Géométrie d'images multiples

William Triggs

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Géométrie d'images multiples

Thèse

pour obtenir le grade de

Docteur en sciences

Spécialité : Informatique (Imagerie, Vision et Robotique)

présentée et soutenue publiquement par

William James Triggs

le 26 novembre 1999

à GRAVIR, INRIA Rhône-Alpes, Montbonnot

Directeur de thèse : Roger Mohr

Membres du jury

M. Jean-Pierre Verjus	président
M. Richard Hartley	rapporteur
M. Andrew Zisserman	rapporteur
M. Roger Mohr	directeur de thèse
M. Olivier Faugeras	

Géométrie d'images multiples

On étudie les relations géométriques entre une scène 3D et ses images perspectives. Les liens entre les images, et la reconstruction 3D de la scène à partir de ces images, sont particulièrement élucidés. L'outil central est un formalisme tensoriel de la géométrie projective des images multiples. La forme et la structure algébrique des contraintes géométriques qui lient les différentes images d'une primitive 3D sont établies. À partir de là, plusieurs nouvelles méthodes de reconstruction 3D projective d'une scène à partir d'images non-calibrées sont développées. Pour rehausser cette structure projective à une structure euclidienne, on introduit un nouveau formalisme d'auto-calibrage d'une caméra en mouvement.

Geometry of Multiple Images

We study the geometric relations that link a 3D scene to its perspective images. The focus is on the connections between the images, and the 3D reconstruction of the scene from these images. Our central tool is a tensorial formulation of the projective multi-image geometry. This is used to determine the form and structure of the geometric constraints between the different images of a 3D primitive. Several new methods for 3D projective reconstruction of scenes from uncalibrated images are derived from this. We then introduce a new formalism for the autocalibration of a moving camera, that converts these projective reconstructions into Euclidean ones.

À tous ceux qu'on aime.
Et à ceux qu'on aimait.

Table des matières

Remerciements

> He had been great already, as he knew, at postponements; but he had only to get afresh into the rhythm of one to feel its fine attraction.
>
> Henry JAMES
> *The Ambassadors*

Autant que d'un milieu scientifique, une thèse issue d'un milieu humain. Je tiens d'abord à remercier mon directeur de thèse Roger MOHR, de sa chaleur et de son enthousiasme inépuisable, de sa patience face à mes très nombreuses réticences inexplicables et esquives d'étudiant confirmé, et surtout de sa force de caractère en m'obligeant à enfin terminer cette petite histoire sans fin. Roger a également – et ceci depuis un lit d'hôpital – porté son aide à toute étape de la production du texte, sans quoi il (le texte!) aurait été infiniment moins compréhensible.

Je remercie également les autres membres de mon jury, Olivier FAUGERAS et Jean-Pierre VERJUS, et surtout mes deux rapporteurs Richard HARTLEY et Andrew ZISSERMAN, qui ont sans broncher lu et rapporté à délai très bref ces quelques centaines de pages de baragouin technique, et qui m'ont en plus fait l'honneur de venir de loin le réentendre une deuxième fois.

Toute l'équipe MOVI m'a donné un accueil superbe pendant mon séjour. Je remercie Long, Peter, Roger et Radu de nos nombreuses et très riches heures de discussion, Danièle et Bart d'avoir débloqué quoi qu'il en soit d'administratif et de matériel, et Cordelia et encore Bart de m'avoir systématiquement nargué afin que je finisse la thèse. En ville, je remercie Bernard de ses multiples contributions soignées aux mêmes buts, et en Angleterre Dhanesh et Carol, Nic, et Al et Gil.

Avant MOVI, j'étais dans l'équipe de robotique SHARP, et avant ça dans le groupe de recherche en robotique à Oxford. Même avant ça, j'étais dans l'équipe de physique mathématique à l'Institut de Mathématique d'Oxford, où j'ai beaucoup appris sur la géométrie projective et les tenseurs, ce qui m'a bien servi pendant cette thèse. Je remercie toutes ces équipes de leurs accueils. Pendant la thèse j'ai eu le support d'une bourse INRIA et du projet ESPRIT LTR 21914 CUMULI, et avant, le support des projets ESPRIT HCM et SECOND, du SERC (maintenant l'ESPRC) et du Commonwealth Scholarships Commission.

Finalement, je remercie mes parents et ma soeur de leur amour et de leur patience pendant toutes ces longues années.

Chapitre 1

Introduction

Cette thèse étudie les relations géométriques entre une scène 3D et ses images perspectives. Les liens entre les différentes images, et la reconstruction 3D de la scène à partir de ces images, sont particulièrement élucidés. L'outil central est un formalisme tensoriel de la géométrie projective des images multiples. Quoique l'orientation du travail soit parfois assez théorique, ce formalisme représente un véhicule de expression très puissant, autant pour le calcul numérique que pour le calcul formel. Tout au long de ce travail, nous avons tenté de ne jamais perdre de vue les aspects algorithmiques et numériques du sujet.

Pourquoi étudier la géométrie multi-images ? – Nous vivons un temps sans précédent historique. L'accroissement explosive de tout qui relève de l'ordinateur – intelligence artificielle, les réseaux, le web, multi-média, réalité virtuelle et augmenté, vidéo et cinéma digitale – risque de changer non seulement nos façons de travailler, mais aussi nos façons de voir notre propre monde. Soit il pour le bien ou non, le bureau et la foyer sont désormais instrumentés et vont certainement devenir de plus en plus réactifs, sinon plus « intelligents ». Il ne s'agira plus de « habiter dans nos espaces », mais plutôt d'« interagir avec eux ». La caméra et l'image seront au centre de cette révolution, car de tous nos sens, la vision est le plus riche et le plus informatif. Les moyens de calcul seront bientôt à ce rendez-vous [1], mais nous manquons cruellement d'algorithmes efficaces, en particulier en tout ce qui concerne l'interprétation et la compréhension de scènes et de structures 3D et dynamiques.

Les révolutions techniques se fait par spécialité, et ici on se focalise sur la théorie d'extraction de la structure 3D à partir de plusieurs images ou séquences d'images. À titre indicative et non-exhaustive, les résultats obtenus porte sur les compétences pratiques suivantes : (*i*) mesurer ou modéliser une scène pour mieux gérer nos interventions sur lui (métrologie photogrammétrique, contrôle de qualité, planification, surveillance, applications médicales) ; (*ii*) resynthétiser d'autres images de la même scène (visualisation, réduction du débit de réseau véhiculant des scènes) ; (*iii*) modifier ou interagir avec la scène (réalité augmenté, studio virtuel).

L'automatisation quasi-complète sera souvent indispensable pour rendre ces applications viable. Pour la plupart d'entre elles, les utilisateurs ne voudraient pas réaliser ou maintenir un calibrage précis des caméras – il leur faut des systèmes qui s'auto-calibraient eux mêmes. Pour toutes ces raisons, il y a un besoin de méthodes améliorées de correspondance entre images, de reconstruction 3D à partir des correspondances trouvées, et d'(auto-)calibrage.

Sous-jacent à tout cela, il y a un besoin de comprendre la structure théorique du domaine. Nous partageons le point de vue qu'« il n'y a rien de plus pratique qu'une bonne théorie » – elle peut aider aux dérivations et aux implantations, indiquer les limites d'application, expliquer comment

1. Pour l'instant, un ordinateur personnel ne peut faire qu'une traitement simpliste d'une séquence d'images de taille raisonnable à temps réel. Mais si on croît la loi de Moore (augmentation des puissances de calcul par un facteur de deux chaque 18 mois), il est à (seulement!) 20-30 ans près de la puissance de calcul du cortex visuel humain, estimé à 10^{13} à 10^{14} opérations par second.

contourner les échecs, suggérer d'autres directions fructueuses ...

La forme de la thèse

Ceci est une thèse sur travaux. C'est un genre que je n'aime guère, mais que les limites de temps et mes autres préoccupations multiples m'obligent à adopter. La plupart du texte consiste en des articles déjà publiés ou soumis, reproduits ici tels quels à une simple mise en page près. J'ai parfois pris une version longue et/ou corrigée s'il en existe, mais ces modifications datent de la même époque de la publication initiale.

J'ai résisté à toute tentation de récrire ces travaux, même légèrement. Je n'ai même pas cédé au désir d'harmoniser les notations qui varient librement de papier en papier. Ceci pour la simple raison que si je commençais à récrire ces textes – et en particulière les plus vieux – je changerais souvent quasiment toute l'exposition ... et parfois même (mais plus rarement) la substance.

Organisation

Le prochain chapitre évoque très brièvement, et sans entrer dans aucun détail, le cadre technique de la thèse. Chacun des trois chapitres suivants introduit, et puis reproduit, plusieurs papiers sur un thème commun : chapitre 3 – les contraintes d'appariement et l'approche tensorielle ; chapitre 4 – la reconstruction projective ; et chapitre 5 – l'auto-calibrage. Un appendice donne un quatrième ensemble de papiers qui n'ont pas trouvé place dans le corps du texte. L'introduction de chaque papier est susceptible de contenir des notes historiques, un bref sommaire technique, et éventuellement des perspectives et commentaires. Ces introductions n'ont pas pour intention de donner une compréhension technique détaillé du travail : pour cela il faut sans exception lire l'article.

Chaque papier a sa bibliographie propre à lui. La bibliographie à la fin de la thèse ne contient que les références citées dans les textes introducteurs.

Chapitre 2

Cadre technique

Géométrie projective

On peut maintenant raisonnablement supposer que la géométrie projective soit familière au lecteur (voir, *ex.* [SK52, HP47]). On adopte toujours les coordonnées homogènes pour décrire l'espace 3D et les images projectives. Chaque point 3D $(X\ Y\ Z)^\top$ se représente par son vecteur homogène $(X\ Y\ Z\ 1)^\top$ ... ou par tout autre vecteur de dimension 4 égal à celui-ci à un facteur d'échelle près. (Cette relation d'équivalence est notée « $\simeq$ »). Il en est de même pour les points images $(x\ y)^\top$ qui deviennent $(x\ y\ 1)^\top$. Quoique redondante, cette représentation homogène à un avantage capital : toute transformation projective prend une apparence *linéaire* quand on l'exprime dans les coordonnées homogènes. C'est le cas pour les transformations projectives (« **homographies** ») 3D–3D et 2D–2D, et plus particulièrement pour les projections perspectives 3D–2D, qui sont au coeur de la formation des images.

Projection centrale et calibrage interne d'une caméra

On n'exposera pas ici systématiquement la théorie de formation d'images. Voir par exemple Faugeras [Fau93] ou Horaud & Monga [HM93] pour le cas perspectif, et [JW76, Sla80] pour les détails optiques. Brièvement, idéalisons par une « **caméra à projection centrale** » tout dispositif de projection d'images avec la propriété que pour chaque demi-droite (« **rayon optique** ») origine à un point 3D particulier (le « **centre optique** » de la caméra), les images de tous les points 3D le long de ce rayon soient confondus. Le modèle sténopé standard en est un exemple. Une caméra centrale peut en principe enregistrer les rayons qui viennent de n'importe quelles directions 3D – une lentille « oeil de poisson » est une approximation – donc une image centrale complète est topologiquement un sphère (le sphère « panoramique » de toutes les directions de vue au centre optique). On autorise des déformations arbitraires dans l'image ... pourvu qu'on puisse les défaire plus tard pour retrouver le « **modèle calibré** » de la caméra, où chaque point image correspond à une direction (rayon 3D au centre) connue.

Supposons qu'on prend comme origine 3D le centre d'une caméra qui est calibrée. L'image du point homogène $\mathbf{X} \simeq (X\ Y\ Z\ 1)^\top$ est évidemment (à un facteur d'échelle près) le point image homogène $\boldsymbol{x} \simeq (X\ Y\ Z)$, car tous les points $(\lambda X\ \lambda Y\ \lambda Z\ 1), \lambda > 0$ se trouvent sur le même rayon issu du centre. Cette projection image s'exprime de façon homogène linéaire comme $\boldsymbol{x} \simeq \mathbf{P}\,\mathbf{X}$, où $\mathbf{P} \equiv (\boldsymbol{I}_{3\times 3} \,|\, \boldsymbol{0})$ est la « **matrice de projection** » 3×4 de la caméra. On peut aussi écrire cela sous la forme $\lambda\,\boldsymbol{x} = \mathbf{P}\,\mathbf{X}$, où on introduit une facteur d'échelle λ pour compenser l'échelle inconnue relative des deux côtés de l'équation. λ s'appelle un « **profondeur projective** » car – moyennant une normalisation convenable de $\boldsymbol{x}$, $\mathbf{X}$ et $\mathbf{P}$ – elle devient la profondeur (distance du centre optique) du point.

Sous un changement de repère euclidien (exprimé en coordonnées homogènes par une matrice $4 \times 4 \left(\begin{smallmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{smallmatrix}\right)$, où $\boldsymbol{R}$ est une rotation 3×3 et $\boldsymbol{t}$ un 3-vecteur de translation) on arrive à une matrice de projection $\mathbf{P} \simeq (\boldsymbol{I}_{3\times 3} \,|\, \boldsymbol{0}) \left(\begin{smallmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{smallmatrix}\right) = (\boldsymbol{R} \,|\, \boldsymbol{t})$. Si on autorise aussi une déformation projective (ou même affine) arbitraire de l'image, on arrive à une matrice générale 3×4 (rang 3) de projection. Au moyen de la décomposition matricielle QR (ou plus exactement RQ), on peut obtenir d'une telle matrice une redécomposition dans la forme :

$$\mathbf{P} \simeq \boldsymbol{K}(\boldsymbol{R} \,|\, \boldsymbol{t}) \qquad\qquad \boldsymbol{K} \equiv \begin{pmatrix} 1 & s & u_0 \\ 0 & a & v_0 \\ 0 & 0 & 1/f \end{pmatrix}$$

où $\boldsymbol{R}$,$\boldsymbol{t}$ sont une rotation et une translation qui donnent la « pose » (position et orientation) de la caméra, et la déformation 2D affine triangulaire $\boldsymbol{K}$ est sa « **matrice de calibrage interne** » . f,a,s,$(u_0$,$v_0)$ s'appellent respectivement la (distance) focale ; le rapport des échelles ; et le *skew* et le point principal géométrique de la caméra. (On paramètre $\boldsymbol{K}$ aussi parfois par les deux focales principales $(f$,$fa)$, et le skew et le point principal *pixéliques* fs et $(fu_0$,$fv_0)$. La focale peut s'exprimer en pixéls ou – si les pixéls sont donnés en millimètres – en millimètres).

On appelle ce modèle le modèle « **projectif sphérique** » d'une caméra. Il prend comme base la géométrie projective sphérique des rayons 3D à un point, notion qui date de la préhistoire de la géométrie projective [1]. Le modèle de projection centrale est en effet très précis pour la plupart des caméras conventionnelles (mis à part les caméras « à balayage » (pushbroom cameras) et les modèles avoisinants). Le modèle projectif de calibrage interne (déformation projective de l'image) est moins précis : il est à la fois trop faible – pour la plupart des caméras, le skew est entièrement négligeable – et trop fort – les distorsions optiques de lentille ne sont *pas* en général négligeables, en particulier avec les lentilles bon marché, de courte focale, ou de zoom. Néanmoins, dans cette thèse on adoptera toujours ce modèle de caméra projectif, car il est très maniable par comparaison avec les modèles internes non-linéaires. En pratique, la distorsion optique ne peut généralement être incluse que : (*i*) par pré-correction ; (*ii*) dans une étape d'estimation non-linéaire – étape quasiment inévitable pour tout système pratique qui prétend à la précision, mais qu'on n'abordera guère ici.

Reconstruction projective et euclidienne

Conséquence inéluctable du fait que le processus de formation d'images soit projectif : tout tentative de « reconstruction » d'une scène à partir des images seules est aussi, de sa nature même, *projective* [2]. Une déformation projective 3D à la fois des caméras et de la scène ne change pas les images, donc la structure déformée ne peut pas être distinguée de la bonne structure au seul moyen de ses images [Fau92, HGC92]. Pour remonter à la structure métrique, il faut des contraintes non-projectives, ou sur la scène, ou sur les mouvements, ou sur les calibrages des caméras [MF92,

1. La projection sur (*i.e.* section par) un plan de la sphère de rayons optiques était déjà courante chez les géomètres grecs, pour résoudre leurs problèmes de trigonométrie sphérique céleste ... Le modèle projectif sphérique s'appelle aussi parfois le modèle « **projectif orienté** » [Sto91].

2. Comme dans les images, la topologie naturelle de l'espace de reconstructions visuelles est toujours celle d'une sphère – ici d'une 3-sphère en 4 dimensions. Par exemple, l'image sphérique d'une droite infinie s'arrêt abruptement aux images de ses deux points de fuite opposés – coupure qui dépend de la structure affine 3D, et qu'on ne peut pas en général localiser dans les images si on ne voit qu'un segment fini de la droite. Dans chaque image sphérique on peut prolonger le demi-cercle image de la droite à une cercle complète. La géométrie de ces points supplémentaires reste cohérente – sauf visibilité c'est identique à celle des points visibles – et on peut les mettre en correspondance comme s'ils étaient les images des points 3D « au delà de l'infini », de la même manière qu'on traite les vecteurs de direction comme les « points à l'infini ». Grâce à ces points 3D virtuels, la reconstruction de la droite devient un cercle topologique (mais elle est droite, de rayon infini), et l'espace 3D devient une sphère topologique. Il est dans la nature même de toute reconstruction visuelle centrale de recréer un tel espace. Mais la reconstruction projective rend la situation plus difficile, car sans le plan à l'infini on ne sait plus quels sont les points virtuels à jeter.

FLM92, Har93a, Har94, MBB93]. C'est pour cela que l'étude de reconstruction visuelle se sépare naturellement en deux parties : la reconstruction 3D projective (*i.e.* à une projectivité 3D près) à partir des données images, et puis la reconstruction 3D euclidienne (*i.e.* jusqu'à une transformation 3D euclidienne rigide près) à partir de la reconstruction projective.

Il faut dire que même une structure projective est déjà très informative. Elle nous donne toute la géométrie 3D métrique de la scène et des caméras – en principe un nombre presque illimité de paramètres – à seulement 9 paramètres près :

- 3 déformations essentiellement projectives (déplacement du plan à l'infini) ;
- 5 étirements affines ;
- un facteur d'échelle global qu'on ne peut jamais obtenir sans connaissances externes, car toute l'optique (au moins dans sa limite géométrique) est invariante à un rééchelonnement global des caméras et de la scène.

La structure projective suffit elle-même pour certaines applications, en particulier celles de la re-synthèse des images quand elles peuvent se limiter aux caméras (réels et virtuelles) projectives non-calibrées. Mais la plupart des applications exigent une structure métrique, donc il faudra se demander comment estimer ces derniers 8–9 paramètres. Dans cette thèse on étudiera plusieurs méthodes pour chacune de ces deux étapes de reconstruction.

Contraintes d'appariement multi-images

Considérons plusieurs images d'une scène, images prises depuis plusieurs points de vue par une ou plusieurs caméras projectives. Les images d'une primitive 3D (qu'elle soit point, droite, courbe, surface ...) ne sont pas entièrement indépendantes entre elles : elles doivent vérifier certaines contraintes de cohérence géométriques, qui exigent qu'elles soient toutes les projections d'une même primitive 3D quelconque. On étudiera la forme algébrique de ces « **contraintes d'appariement multi-images** » en détail plus bas. En effet, elles sont toujours multi-linéaires en les primitives projetées qui apparaissent, avec pour coefficients des tenseurs (tableaux multi-indices) inter-images, fabriqués de matrices de projection de plusieurs caméras. Ces « **tenseurs d'appariement** » sont évidemment fonction de la géométrie (poses relatives et calibrages internes) des caméras. En effet, il se trouve qu'en général l'ensemble des tenseurs *caractérisent* et même *paramétrisent* la partie projective – et moyennant une légère connaissance supplémentaire, souvent aussi la partie euclidienne – de la géométrie caméras, *sans aucune référence explicite aux quantités 3D*. En particulier, les tenseurs peuvent être estimés à partir de un nombre suffisant de correspondances inter-images des primitives, sans connaissances de quantités 3D. D'où les intérêts principaux des contraintes d'appariement :

1° **Correspondances des primitives :** Une fois estimées, elles sont une aide très puissante à ce problème pérenne de la vision, la mise en correspondance des primitives entre images. Elles réduisent la recherche des points correspondants entre deux images aux « droites épipôlaires », et la recherche des points ou droites correspondants dans la troisième et subséquentes images à la simple prédiction et vérification de la présence de la primitive à une position qui peut se précalculer.

2° **Synthèse des nouvelles vues :** La prédiction ci-dessus peut servir plus activement comme « transfert » des primitives correspondantes entre images, pour synthétiser à partir de quelques images d'une scène, de nouvelles vues qui semblent avoir été prises de points de vue différents de ceux des images d'entrée. Ceci constitue une application très à la mode pour la réalité virtuelle.

3° **Reconstruction 3D :** Vu que les contraintes d'appariement dépendent de la géométrie multi-caméras, on peut songer à recouvrir celles-ci des contraintes, et aussi à reconstruire

les primitives appariées. Ce genre de reconstruction géométrique a maintes applications en métrologie, conception, planification, visualisation, réalité virtuelle...

Une fois qu'on a estimé les contraintes d'appariement, toutes ces applications sont à considérer. En plus, les contraintes ne représentent que le début d'une grande toile de relations géométriques, qui relient primitives 3D, primitives projetées, profondeurs projectives, matrices de projection, tenseurs d'appariement et contraintes euclidiennes dans une structure globale, complexe mais cohérente. La plus haute revendication de cette thèse, c'est d'avoir contribué à élucider une partie de cette structure.

Chapitre 3

Contraintes d'appariement, et l'approche tensorielle à la géométrie des images multiples

Ce chapitre, et en particulier son premier papier, pose les fondations de toute cette thèse. Il traite spécifiquement des contraintes d'appariement – contraintes algébriques inter-images, qui exige que les différentes images d'une primitive 3D soient toutes consistantes entre elles. Mais ces contraintes ne sont qu'un aspect de la riche géométrie multi-images, et les techniques tensorielles qu'on développe ici pour ce cas s'étendent et se ramifient à bien d'autres problèmes.

3.1 Resumé de « The Geometry of Projective Reconstruction : Matching Constraints and the Joint Image »

Historique

Ce papier représente mon travail de base sur les contraintes d'appariement multi-images. Il donne un aperçu résolument projective-tensorielle de ces contraintes, approche qui restera sans doute difficile pour les « non-initiés », mais qui représente à mon avis le moyen le plus puissant d'aborder toute la géométrie projective multi-images. Il fut écrit et diffusé en manuscrit vers la fin de 1994, et publié en version courte à ICCV'95[1] [Tri95] (voir appendice). Il fut aussi soumis a IJCV à l'époque, mais n'a jamais à ce jour atteint sa version finale, suite à mes réticences sur sa forme, et surtout à mes préoccupations avec bien d'autres travaux.

Méthode

Avec toutes les contraintes d'appariement, l'essentiel consiste à prendre les équations de projection d'une primitive 3D, « parent » hypothétique de tous les primitives images qu'on voudrait apparier, et d'éliminer algébriquement les coordonnées 3D du parent – et éventuellement aussi ses profondeurs projectives inconnues – afin d'arriver aux équations liant les primitives images entre elles. Pour les classes principales de primitives, on peut choisir une paramétrisation où l'équation de

1. Conférence qui fut un véritable tournant sur notre compréhension de la géométrie multi-images, avec l'apparition (entre autres!) d'importantes papiers par : (*i*) Faugeras & Mourrain [FM95b, FM95a] et Heyden & Åström [Hey95, HÅ95] sur les contraintes d'appariement multi-images – tous les deux traitent à leurs façons à peu près le même domaine que cet article, avec des conclusions cohérentes; (*ii*) Carlsson [Car95] sur la dualité entre points et centres des caméras; (*iii*) Shashua & Werman [SW95] et Hartley [Har95b] sur le tenseur trifocal, et Hartley sur l'estimation stable de la matrice fondamentale [Har95a].

projection est *linéaire* dans les coordonnées 3D inconnues de la primitive, et aussi dans sa profondeur projective (facteur d'échelle inconnu dans l'image). Dans ce cas, les inconnues peuvent être éliminées avec les déterminants, et *en principe* c'est relativement facile de dériver les contraintes d'appariement pour la primitive par cette paramétrisation[2].

Considérons le cas des points. On a plusieurs images $\boldsymbol{x}_i$ d'un point 3D inconnu $\mathbf{X}$, par des caméras projectives $\mathbf{P}_i$, $i = 1 \dots m$. L'équation de projection est $\boldsymbol{x}_i \simeq \mathbf{P}_i \mathbf{X}$, ou, si on introduit une profondeur projective / facteur d'échelle inconnu λ_i, $\lambda_i \boldsymbol{x}_i = \mathbf{P}_i \mathbf{X}$. On peut rassembler toutes ces équations de projection dans un grand système matriciel $(3m) \times (4+m)$:

$$\begin{pmatrix} \mathbf{P}_1 & \boldsymbol{x}_1 & 0 & \dots & 0 \\ \mathbf{P}_2 & 0 & \boldsymbol{x}_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_m & 0 & 0 & \dots & \boldsymbol{x}_m \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ -\lambda_1 \\ -\lambda_2 \\ \vdots \\ -\lambda_m \end{pmatrix} = \mathbf{0}$$

Les points images $\boldsymbol{x}_i$ et les matrices de projection $\mathbf{P}_i$ sont cohérents avec quelque point 3D si et seulement si ce système homogène a une solution. Et bien entendu, la solution donne le point 3D $\mathbf{X}$ correspondant avec ses profondeurs projectives λ_i. Algébriquement, il y a une solution si et seulement si tous les mineurs (déterminants des sous-matrices) $(4+m) \times (4+m)$ de la matrice du système sont nuls. Chaque mineur se forme d'un sous-ensemble spécifique des lignes des matrices de projection et des points images correspondants. La nullité du mineur donne une contrainte algébrique entre les projections et les points, contrainte qui doit être vérifiée si ils sont consistants avec quelque point 3D. Une étude détaille révèle 3 classes de ces contraintes d'appariement de points, qui sont bilinéaire, trilinéaire, et quadrilinéaire, dans les points correspondant dans 2,3,4 images.

Les coefficients des contraintes sont des déterminants 4×4 de 4 rangs des matrices de projection. Ils peuvent être rangés en « **tenseurs inter-images** » – tableaux multi-indices, avec des indices (dimensions) qui appartient aux plusieurs images. Pour les images 2D d'une scène 3D, il y a précisément 4 types de tenseurs d'appariement : épipôle, matrice fondamentale, tenseur trifocal et tenseur quadrifocal. L'épipôle ne figure pas directement dans les contraintes d'appariement, mais d'ailleurs joue un rôle central dans le formalisme.

En effet, les contraintes d'appariement ne sont qu'un premier pas dans les aspects de la vision multi-images. Toute la théorie de la géométrie multi-images s'exprime très naturellement sous forme tensorielle, ce qui nous donne un moyen de calcul puissant pour la géométrie de la vision. Les tenseurs d'appariement ne sont que l'expression la plus courante de cet aspect, et ils apparaissent partout dans le formalisme.

Le lien entre les tenseurs et la géométrie est très naturel. Selon le célèbre « programme d'Erlangen » de Felix KLEIN (*ex.* [Kle39]), une géométrie se caractérise par son groupe de transformations, et par les quantités qui sont invariantes ou covariantes par ce groupe. « **Covariant** » signifie « qui transforme selon une loi cohérente et régulière » – une telle loi s'appelle une « **représentation** » du groupe. Pour les groupes linéaires (euclidien, affine, projectif ...), il se trouve que (quasiment) toutes les représentations sont tensorielles, car étant construites par produit tensoriel d'une ou plusieurs « **représentations de base** » – les « **vecteurs** » du système. Par exemple, dans l'espace projectif il y a deux types de vecteurs – ceux « **contravariants** » qui représentent les points projectifs, et ceux « **covariants** » qui représentent les hyperplans projectifs duaux des points. Les deux lois de transformation sont aussi duales. Quand on construit un tenseur multi-indices, chaque indice correspond

2. « En principe » parce qu'en pratique, mis a part les points, cette approche peut être très lourde. Elle ne peut que difficilement être implantée pour les droites, et je n'ai jamais abouti pour les quadriques, en dépit de plusieurs tentatives. Les équations de projection ne sont plus linéaires dans les matrices de projection associées aux caméras, et en plus la dimension des déterminants monte. Donc la complexité algébrique augmente très significativement.

à un vecteur (ou plutôt à une dimension vectorielle) d'un de ces deux types, et transforme selon la loi appropriée. Mais dans l'espace euclidien, le groupe de transformations autorisées est plus restreint et les deux lois de transformation se confondent, donc il n'y a qu'un seul type de vecteur et d'indice.

En vision multi-images, il faut souvent travailler à la fois dans plusieurs espaces différents – par exemple dans l'espace 3D et dans plusieurs images. En ce cas, les tenseurs peuvent posséder des indices de chacun des types disponibles en chaque espace. La notation devient plus complexe et un peu lourde (si on évite d'être ambiguë ...), mais le calcul tensoriel reste valable.

À titre indicatif, on peut identifier plusieurs facettes du formalisme tensoriel multi-images. Un thème central dans nos approches est de représenter chaque point 3D non par ses coordonnées 3D, mais par l'ensemble de ses coordonnées dans toutes les images. Cette représentation « **par images réunies** » est fortement redondante, mais ses liens aux quantités visibles dans les images sont évidemment beaucoup plus directes. Elle s'est montrée une approche très fructueuse pour notre problématique.

- **La connexion projective / Plücker-Grassmann :** Pour l'essentiel, la géométrie projective est celle de l'alignement, de l'extension, de l'intersection linéaire. Dans un langage tensoriel, ces opérations s'expriment par des déterminants / sommes alternées de composantes. Les sous-espaces projectifs sont coordonnés par leurs « **coordonnées Plücker-Grassmann** » – l'ensemble de leurs déterminants. Cette représentation a l'avantage d'être *linéaire* (et donc relativement maniable) dans ces coordonnées, mais elle devient rapidement très redondante quand la dimension de l'espace augmente. Les coordonnées Plücker-Grassmann sont sujettes aux « **contraintes de consistance de Plücker-Grassmann** », contraintes qui ont une structure quadratique, régulière mais extrêmement lourde en haute dimension. Tout reste maniable en 2 et 3 dimensions, mais représenter une primitive 3D par ces images multiples peut largement augmenter la dimension effective de l'espace ...
- « **Projection inverse** » **des primitives images :** Les primitives 3D principales (points, droites dans la représentation Plücker, quadriques dans la représentation duale) ont toutes une représentation où leurs équations de projection sont *linéaires* dans leurs coordonnées 3D. Réciproquement, on peut (si on connaît la matrice de projection de la caméra) « remonter » d'une primitive image quelconque à sa « **primitive de support 3D** » – la primitive 3D qui contient tous les rayons optiques des points de la primitive image. (Si la primitive image est une projection, les rayons optiques – et donc forcément la projection inverse – contiennent les points de la primitive 3D d'origine). Par exemple : (*i*) d'un point image, on remonte à son rayon optique ; (*ii*) d'une droite image, on remonte à son « **(demi-)plan optique** » – le (demi-)plan qui contient la droite 3D et le centre optique de la caméra ; (*iii*) d'une conique image, on remonte à son « **cône optique** ».

  On pourrait considérer qu'avec les équations de projection et les opérations d'intersection et d'allongement linéaire, les équations de projection inverse sont les unités de base de tout le formalisme projectif-tensoriel.
- **Reconstruction minimale :** Si on connaît les matrices de projection des caméras, on peut reconstruire une primitive 3D à partir d'un nombre suffisant de ses images. Si les équations de projection sont linéaires en la primitive 3D, on peut réduire la reconstruction à la résolution d'un système linéaire, ou – ce qui est équivalent – à l'« **intersection** » des primitives reconstruite par projection inverse depuis les images (rayons optiques d'un point 3D, plans optiques d'une droite 3D ...).

  Si on ne prend que le nombre minimal des contraintes images pour faire la reconstruction, on arrive à un « **système de reconstruction minimal** ». Par exemple, il faut trois contraintes linéaires pour reconstruire un point 3D, donc les deux cas minimaux sont : (*i*) fixer une coor-

donnée du point dans chacune de 3 images ; (*ii*) fixer les deux coordonnées dans une image, et une dans une autre. Toute autre combinaison est ou redondante, ou insuffisante. En général, la reconstruction serait mieux conditionnée si on prenait des contraintes redondantes, mais la reconstruction minimale fournit un lien important aux contraintes de transfert et d'appariement discutées ci-dessous.

- **Équations de transfert :** Une fois obtenue une reconstruction (soit minimale soit redondante) d'une primitive 3D, on peut la reprojeter dans une autre image. Entre les primitives d'entrée et la primitive de sortie, il n'y a aucune référence explicite à l'espace 3D. Donc on peut court-circuiter l'espace 3D et travailler directement entre images. La géométrie 3D des caméras est représentée par ses auxiliaires dans les images, les tenseurs d'appariement. On peut utiliser le transfert par exemple pour la synthèse des nouvelles images depuis des points de vue artificiels, ou pour générer les contraintes d'appariement (voir ci-dessous).
- **Contraintes d'appariement :** On a déjà évoqué ces contraintes. Elles peuvent être interprétées dans les deux façons suivantes : (*i*) une primitive transférée vers une autres images doit être identique à la projection de la primitive d'origine 3D ; ou (*ii*) les primitives 3D reprojetées depuis toutes les images doivent s'intersecter d'une façon cohérente en une primitive 3D bien définie. Ces contraintes sont fortes utiles pour établir les correspondances de primitives entre images. Inversement, elles fournissent une méthode pour estimer les tenseurs d'appariement à partir d'un ensemble de correspondances initiales dans les images.
- **Contraintes de clôture :** Vue la dérivation des matrices de projection, les tenseurs d'appariement doivent satisfaire certaines contraintes de consistance avec ces matrices. De façon tensorielle, ces « **contraintes de clôture** » expriment le fait que l'espace réel est seulement de dimension 3 et « se renferme sur lui même ». La représentation d'une primitive 3D par ses images multiples a beaucoup de dégrées de liberté et aurait pu représenter les images 2D d'un espace de dimension plus grande que trois ... mais puisque ce n'est pas le cas, il doit y avoir une « clôture » à la dimension trois. Les contraintes de clôture sont à la base de la méthode de reconstruction par clôture [Tri96b, Tri97a], qui est décrite dans le chapitre prochain. Elles engendrent aussi les contraintes d'appariement aux profondeurs des contraintes de Grassmann, qui vont être discutées tout de suite.
- **Contraintes d'appariement aux profondeurs projectives :** Les contraintes de clôture sont linéaires dans les matrices de projection qui y apparaissent. Si on applique ces matrices à un point 3D, on génère une série analogue de contraintes qui lient les tenseurs d'appariement aux images du point *avec leurs profondeurs projectives correctes*. Si on connaît les tenseurs et les points images, on peut récupérer de façon linéaire les profondeurs correspondantes. Ces contraintes sont à l'origine de la méthode de reconstruction par factorisation projective [ST96, Tri96a], qui est décrit dans le chapitre prochain. En éliminant les profondeurs (facteurs d'échelle) inconnues, on récupère les contraintes d'appariement traditionnelles dont on a déjà parlé.
- **Identités Plücker-Grassmann :** La dérivation depuis les matrices de projection des tenseurs d'appariement est essentiellement basée sur les déterminants. En effet, les tenseurs peuvent être identifiés aux coordonnées de Plücker-Grassmann de l'espace 3D dans l'espace réuni de toutes les coordonnées images. Ceci implique que les tenseurs doivent vérifier entre eux des relations de consistance qui sont exactement équivalentes aux contraintes Plücker-Grassmann. Il y a un grand nombre de ces relations. Certaines sont très familières, mais pour la plupart elles sont mal connues, bien que parfois utiles. On peut aussi générer les contraintes sur les tenseurs à partir des contraintes de clôture qui sont l'expression la plus primitive de la clôture par déterminants. Les contraintes de Plücker-Grassmann servent à estimer certains tenseurs d'appariement à partir d'autres, par exemple les épipôles s'expriment à partir d'une

matrice fondamentale.

Perspectives

On peut maintenir que ce papier et ses pairs [FM95b, Hey95] ont constitué un tournant de l'étude systématique des contraintes d'appariement multi-images. Au niveau théorique, on a désormais une maîtrise des aspects projectifs et des primitives linéaires (points, droites et plans 3D), qui semble pour l'instant plus ou moins « complète » et « finale ». Mais au niveau pratique le cas est moins clair. Certes la communauté a déjà pu capitaliser sur cette maîtrise pour créer des algorithmes de reconstruction et de transfert qui semblent très efficaces, au moins au niveau des primitives géométriques isolées. Mais à mon avis – comme c'est souvent le cas dans la recherche, et bien qu'on a beaucoup appris dans le processus – c'était une victoire un peu à la Pyrrhus. Mis à part les cas les plus simples de deux et éventuellement de trois images, on a appris définitivement que les contraintes d'appariement – et en particulier leurs contraintes de consistance entre elles – sont algébriquement si complexes et redondantes, qu'il semble plus prudent s'enfuir au plus tôt vers la simplicité relative d'une représentation 3D traditionnelle. J'estime que le tenseur quadrifocal n'a jamais été utilisé de façon convaincante « en vraie grandeur », et que même pour le tenseur trifocal, il est dans la plupart des cas plus facile de basculer dès que possible sur une représentation par matrices de projection (ou ce qui revient en effet à la même chose, sur une représentation homographie + épipôle). Même si on se limite aux représentations hyper-redondantes à la base d'images (plénoptique, mosaïques...), on ne peut pas se passer très longtemps de la consistance géométrique globale, qui semble exiger une représentation plus ou moins explicite du monde 3D.

Il faut également souligner qu'il y a des cas que l'on n'a pas encore pu résoudre, le plus important étant les contraintes d'appariement entre quadriques 3D dans 3 images (ce qui est lié au problème de l'obtention de la structure euclidienne à partir de 3 images – voir plus bas). J'ai abordé ce problème plusieurs fois par plusieurs méthodes différentes, avec des succès parfois partiels mais jamais complets. En principe il est « facile » – l'expansion de certains déterminants 10×10 dont les coefficients sont quadratique aux matrices de projection, et leur regroupement en terme de (termes qui sont un produite de 5) tenseurs d'appariement. Mais en pratique c'est trop lourd, même avec les astuces diverses que j'ai su mettre en oeuvre. Il est bien possible qu'il n'y ait aucune solution simple. Et même s'il n'y en a, il est probable qu'elle aurait un nombre très importante de formes alternatives, grâce aux équivalences Grassmann-Cayley.

3.2 Resumé de « Optimal Estimation of Matching Constraints » – SMILE'98

La version ci dessous de ce papier fut publié au workshop SMILE'98 de ECCV'98 [Tri98]. Il décrit une approche à l'estimation optimale statistique adaptée aux « petits problèmes géométriques tordus » qu'on retrouve si souvent en vision, et plus particulièrement aux contraintes d'appariement multi-images. Puis il résume mes travaux sur une bibliothèque numérique spécialisée pour ce genre de problème. Une version préliminaire du papier contient plus de détail technique sur la façon de formuler l'optimisation [Tri97b].

Le texte repose sur quatre axes principaux : (*i*) une reformulation du problème général d'ajustement d'un modèle géométrique sur les données incertaines, basée sur l'optimisation sous contraintes ; (*ii*) une discussion de la modélisation des erreurs statistiques robustes ; (*iii*) une discussion de la paramétrisation des problèmes géométriques complexes, face aux libertés de choix de jauge (système de coordonnées), contraintes de consistance, *etc*; (*iv*) une brève discussion de comment caractériser la performance d'une telle méthode.

Considérons un problème d'ajustement géométrique simple, par exemple l'ajustement d'une surface implicite sur un ensemble de points 3D incertains. On suppose qu'il y a une « vraie » surface sous-jacente qui est inconnue, et des « vrais » points 3D sous-jacents qui sont également inconnus. Les points tombent précisément sur la surface, donc ils vérifient sans aucun résidu ses équations implicites. Mais on ne connaît ni la surface ni les points – on observe seulement une version bruitée des points, et on voudrait estimer au mieux la surface et éventuellement les points 3D sous-jacents. L'approche classique consiste en : (*i*) minimiser en les paramètres de la surface, la somme des distances (Mahalanobis-) orthogonales des observations à la surface ; (*ii*) estimer le point dans la surface la plus proche à chaque observation. La nouvelle approche consiste en : introduire les positions des points sous-jacents inconnus comme des paramètres supplémentaires dans le problème, et optimiser sur *tous* les paramètres, et de la surface, et des points. Cette deuxième approche est logiquement plus simple et théoriquement plus précise, mais le nombre de paramètres à optimiser est nettement plus grand. Néanmoins, la matrice Jacobienne de ce nouveau système est très creuse et une formulation appropriée de l'algèbre numérique nous donne une algorithme efficace.

3.3 Resumé de « Differential Matching Constraints » – ICCV'99

Cet article fut publié à ICCV'99 [Tri99]. Il reprend les éléments de base du papier « Matching Constraints and the Joint Image » ci dessus, et les redéveloppe au cas (fréquent en pratique) où plusieurs des caméras sont très proches les unes aux autres. Il y avait déjà de nombreuse études sur ce problème dans le cas de deux images calibrées (« flot optique »), mais très peu dans les cas multi-images et/ou non-calibrées [HO93, VF95, VF96, ÅH96, ÅH98, SS97]. Le travail de Åström & Heyden [ÅH96, ÅH98] basé sur les séries de Taylor, était le seul à aborder systématiquement dans cette limite les contraintes multi-images. Mais à mon avis cette approche n'était pas satisfaisante : elle menait à des contraintes et à des tenseurs différentiels très complexes et sans fin, là où la théorie discrète avait des contraintes et tenseurs relativement simples en 4 images au maximum. La source de cette difficulté est en effet les série de Taylor : approche hors pair quand les déplacements sont vraiment infinitésimaux, mais qui requiert un nombre infini de termes pour exprimer tout déplacement fini (et tous les déplacements qu'on voit en pratique *sont* finis !).

On a donc développé une expansion à la base de différences *finies*, qui est mieux adaptée au problème. En plus, pour être capable de traiter les séquences multiples, on généralise au cas où les images tombent en plusieurs groupes, celles de chaque groupe étant proche les unes des autres, et les groupes étant autorisés d'être mieux séparées. On considère aussi brièvement le cas de « **suite d'un tenseur d'appariement** » le long d'une séquence d'images, qui peut être une aide à la suite des cibles et à la recouvrement de la géométrie caméras-scène. Ce cas à des liens forts avec l'estimation optimale itérative des tenseurs, car les mises à jour du tenseur – ou le long de la séquence, ou dans une boucle itérative – se basent sur les mêmes équations.

Perspectives

Ce travail a réussi dans le sens où on a créé un formalisme efficace et facile à mettre en oeuvre pour les petits déplacements. Néanmoins, certaines de mes conclusions restent négatives : dans les cas où *tous* les images sont proches les unes aux autres, bien que les expansions en différences finies soient possibles, elles ne me semblent pas apporter grand chose par rapport aux résultats correspondants non-différentiels. Leur forme est plus complexe, leur précision en pratique semble la même ou légèrement pire à cause des erreurs de troncature, et leur degré de non-linéarité est à peu près le même : il n'y a pas de linéarisation de contraintes de consistance comme dans le cas de la suite d'un tenseur, car le « **point d'expansion** » (le tenseur de bas quand toutes les images coïncident) est singulier.

En plus, pour tous les résultats basés sur le tenseur trifocal, il me semble plus direct de convertir dès que possible dans une représentation basée sur les matrices de projection (ou – ce qui revient à la même chose – sur les homographies et les épipôles).

The Geometry of Projective Reconstruction: Matching Constraints and the Joint Image

Bill Triggs
LIFIA, INRIA Rhône-Alpes,
46 avenue Félix Viallet, 38031 Grenoble, France.
Bill.Triggs@imag.fr

Abstract

This paper studies the geometry of perspective projection into multiple images and the matching constraints that this induces between the images. The combined projections produce a 3D subspace of the space of combined image coordinates called the **joint image**. This is a complete projective replica of the 3D world defined entirely in terms of image coordinates, up to an arbitrary choice of certain scale factors. Projective reconstruction is a canonical process in the joint image requiring only the rescaling of image coordinates. The matching constraints tell whether a set of image points is the projection of a single world point. In 3D there are only three types of matching constraint: the fundamental matrix, Shashua's trilinear tensor, and a new quadrilinear 4 image tensor. All of these fit into a single geometric object, the **joint image Grassmannian** tensor. This encodes exactly the information needed for reconstruction: the location of the joint image in the space of combined image coordinates.

Keywords: Computer Vision, Visual Reconstruction, Projective Geometry, Tensor Calculus, Grassmann Geometry.

1 Introduction

This is the first of two papers that examine the geometry underlying the recovery of 3D projective structure from multiple images. This paper focuses on the geometry of multi-image projection and the **matching constraints** that this induces on image measurements. The second paper will deal with projective reconstruction techniques and error models.

Matching constraints like the fundamental matrix and Shashua's trilinear tensor [19] are currently a topic of lively interest in the vision community. This paper uncovers some of the beautiful and useful structure that lies behind them and should be of interest to anyone working on the geometry of vision. We will show that in three dimensions there are only three types of constraint: the fundamental matrix, Shashua's trilinear tensor, and a new quadrilinear four image tensor. All other matching constraints reduce trivially to one of these three types. Moreover, all of the constraint tensors fit very naturally into a single underlying geometric object, the **joint image Grassmannian**. Structural constraints on the Grassmannian tensor lead to quadratic relations between the matching tensors.

The joint image Grassmannian encodes precisely the portion of the imaging geometry that can be recovered from image measurements. It specifies the location of the **joint image**, a three dimensional submanifold of the space of combined image coordinates containing the matching m-tuples of image points. The topology of the joint image is complicated, but with an arbitrary choice of certain scale factors it becomes a 3D projective space containing a projective 'replica' of the 3D world. This replica is all that can be inferred about the world from image measurements. 3D reconstruction is an intrinsic, canonical geometric process only in the joint image, however an appropriate choice of basis there allows the results to be transferred to the original 3D world up to a projectivity.

This is a paper on the geometry of vision so there will be 'too many equations, no algorithms and no real images'. However it also represents a powerful new way to *think* about projective vision and that *does* have practical consequences. To understand this paper you will need to be comfortable

This unpublished paper dates from 1995. The work was supported by the European Community through Esprit programs HCM and SECOND.

with the tensorial approach to projective geometry: appendix A sketches the necessary background. This approach will be unfamiliar to many vision researchers, although a mathematician should have no problems with it. The change of notation is unfortunate but essential: the traditional matrix-vector notation is simply not powerful enough to express many of the concepts discussed here and becomes a real barrier to clear expression above a certain complexity. However in my experience effort spent learning the tensorial notation is amply repaid by increased clarity of thought.

In origin this work dates from the initial projective reconstruction papers of Faugeras & Maybank [3, 15, 5]. The underlying geometry of the situation was immediately evoked by those papers, although the details took several years to gel. In that time there has been a substantial amount of work on projective reconstruction. Faugeras' book [4] is an excellent general introduction and Maybank [14] provides a more mathematically oriented synthesis. Alternative approaches to projective reconstruction appear in Hartley *et.al.* [8] and Mohr *et.al.* [17]. Luong & Viéville [13] have studied 'canonic decompositions' of projection matrices for multiple views. Shashua [19] has developed the theory of the trilinear matching constraints, with input from Hartley [7]. A brief summary of the present paper appears in [21]. In parallel with the current work, both Werman & Shashua [22] and Faugeras & Mourrain [6] independently discovered the quadrilinear constraint and some of the related structure (but not the 'big picture' — the full joint image geometry). However the deepest debt of the current paper is to time spent in the Oxford mathematical physics research group lead by Roger Penrose [18], whose notation I have 'borrowed' and whose penetrating synthesis of the geometric and algebraic points of view has been a powerful tool and a constant source of inspiration.

2 Conventions and Notation

The world and images will be treated as projective spaces and expressed in homogeneous coordinates. Many equations will apply only up to scale, denoted $a \sim b$. The imaging process will be approximated by a perspective projection. Optical effects such as radial distortion and all the difficult problems of early vision will be ignored: we will basically assume that the images have already been reduced to a smoldering heap of geometry. When token matching between images is required, divine intervention will be invoked (or more likely a graduate student with a mouse).

Our main interest is in sequences of 2D images of ordinary 3D Euclidean space, but when it is straightforward to generalize to D_i dimensional images of d dimensional space we will do so. 1D 'linear' cameras and projection within a 2D plane are also practically important, and for clarity it is often easier to see the general case first.

Our notation is fully tensorial with all indices written out explicitly (*c.f.* appendix A). It is modelled on notation developed for mathematical physics and projective geometry by Roger Penrose [18]. Explicit indices are tedious for simple expressions but make complex tensor calculations *much* easier. Superscripts denote contravariant (*i.e.* point or vector) indices, while subscripts denote covariant (*i.e.* hyperplane, linear form or covector) ones. Contravariant and covariant indices transform inversely under changes of coordinates so that the **contraction** (*i.e.* 'dot product' or sum over all values) of a covariant-contravariant pair is invariant. The 'Einstein summation convention' applies: when the same index symbol appears in covariant and contravariant positions it denotes a contraction (implicit sum) over that index pair. For example $\mathbf{T}^a_b\mathbf{x}^b$ and $\mathbf{x}^b\mathbf{T}^a_b$ both stand for standard matrix-vector multiplication $\sum_b \mathbf{T}^a_b\mathbf{x}^b$. The repeated indices give the contraction, not the order of terms. Non-tensorial labels like image number are never implicitly summed over.

Different types of index denote different space or label types. This makes the notation a little baroque but it helps to keep things clear, especially when there are tensors with indices in several distinct spaces as will be common here. $\mathcal{H}^x$ denotes the *h*omogeneous vector space of objects (*i.e.* tensors) with index type x, while $\mathcal{P}^x$ denotes the associated *p*rojective space of such objects defined only up to nonzero scale: tensors $\mathbf{T}^x$ and $\lambda\mathbf{T}^x$ in $\mathcal{H}^x$ represent the same element of $\mathcal{P}^x$ for all $\lambda \neq 0$. We will not always distinguish points of $\mathcal{P}^x$ from their homogeneous representatives in $\mathcal{H}^x$. Indices $a, b, \ldots$ denote ordinary (projectivized homogenized d-dimensional) Euclidean space $\mathcal{P}^a$

($a = 0, \ldots, d$), while $A_i, B_i, \ldots$ denote homogeneous coordinates in the D_i-dimensional i^{th} image $\mathcal{P}^{A_i}$ ($A_i = 0, \ldots, D_i$). When there are only two images A and A' are used in place of A_1 and A_2. Indices $i, j, \ldots = 1, \ldots, m$ are *i*mage labels, while $p, q, \ldots = 1, \ldots, n$ are *p*oint labels. Greek indices $\alpha, \beta, \ldots$ denote the combined homogeneous coordinates of all the images, thought of as a single big $(D + m)$-dimensional **joint image vector** ($D = \sum_{i=1}^{m} D_i$). This is discussed in section 4.

The same base symbol will be used for 'the same thing' in different spaces, for example the equations $\mathbf{x}^{A_i} \sim \mathbf{P}_a^{A_i}\mathbf{x}^a$ ($i = 1, \ldots, m$) denote the projection of a world point $\mathbf{x}^a \in \mathcal{P}^a$ to m distinct image points $\mathbf{x}^{A_i} \in \mathcal{P}^{A_i}$ via m distinct perspective projection matrices $\mathbf{P}_a^{A_i}$. These equations apply only up to scale and there is an implicit summation over all values of $a = 0, \ldots, d$.

We will follow the mathematicians' convention and use index 0 for homogenization, *i.e.* a Euclidean vector $(x^1 \cdots x^d)^\top$ is represented projectively as $(1\ x^1 \cdots x^d)^\top$ rather than $(x^1 \cdots x^d\ 1)^\top$. This seems more natural and makes notation and coding easier.

$\mathbf{T}^{[ab\ldots c]}$ denotes the result of antisymmetrizing the tensor $\mathbf{T}^{ab\ldots c}$ over all permutations of the indices $ab\ldots c$. For example $\mathbf{T}^{[ab]} \equiv \frac{1}{2}(\mathbf{T}^{ab} - \mathbf{T}^{ba})$. In any $d+1$ dimensional linear space there is a unique-up-to-scale $d+1$ index alternating tensor $\boldsymbol{\varepsilon}^{a_0 a_1 \cdots a_n}$ and its dual $\boldsymbol{\varepsilon}_{a_0 a_1 \cdots a_n}$. Up to scale, these have components ± 1 and 0 as $a_0 a_1 \ldots a_n$ is respectively an even or odd permutation of $01\ldots n$, or not a permutation at all. Any antisymmetric $k+1$ index contravariant tensor $\mathbf{T}^{[a_0 \ldots a_k]}$ can be 'dualized' to an antisymmetric $d-k$ index covariant one $(*\mathbf{T})_{a_{k+1}\cdots a_d} \equiv \frac{1}{(k+1)!}\ \boldsymbol{\varepsilon}_{a_{k+1}\cdots a_d b_0 \cdots b_k}\mathbf{T}^{b_0 \ldots b_k}$, and vice versa $\mathbf{T}^{a_0 \ldots a_k} = \frac{1}{(d-k)!}\ (*\mathbf{T})_{b_{k+1}\cdots b_d}\ \boldsymbol{\varepsilon}^{b_{k+1}\cdots b_d a_0 \cdots a_k}$, without losing information.

A k dimensional projective subspace of the d dimensional projective space $\mathcal{P}^a$ can be denoted by either the span of any $k+1$ independent points $\{\mathbf{x}_i^a |\ i = 0, \ldots, k\}$ in it or the intersection of any $d-k$ independent linear forms (hyperplanes) $\{\mathbf{l}_a^i | i = k+1, \ldots, d\}$ orthogonal to it. The antisymmetric tensors $\mathbf{x}_0^{[a_0} \ldots \mathbf{x}_k^{a_k]}$ and $\mathbf{l}_{[a_{k+1}}^{k+1} \cdots \mathbf{l}_{a_d]}^{d}$ uniquely define the subspace and are (up to scale) independent of the choice of points and forms and dual to each other. They are called respectively **Grassmann coordinates** and **dual Grassmann coordinates** for the subspace. Read appendix A for more details on this.

3 Prelude in F

As a prelude to the arduous general case, we will briefly consider the important sub-case of a single pair of 2D images of 3D space. The low dimensionality of this situation allows a slightly simpler (but ultimately equivalent) method of attack. We will work rapidly in homogeneous coordinates, viewing the 2D projective image spaces $\mathcal{P}^A$ and $\mathcal{P}^{A'}$ as 3D homogeneous vector spaces $\mathcal{H}^A$ and $\mathcal{H}^{A'}$ ($A = 0, 1, 2$; $A' = 0', 1', 2'$) and the 3D projective world space $\mathcal{P}^a$ as a 4D vector space $\mathcal{H}^a$ ($a = 0, \ldots, 3$). The perspective image projections are then 3×4 matrices $\mathbf{P}_a^A$ and $\mathbf{P}_a^{A'}$ defined only up to scale. Assuming that the projection matrices have rank 3, each has a 1D kernel that corresponds to a unique world point killed by the projection: $\mathbf{P}_a^A\ \mathbf{e}^A = \mathbf{0}$ and $\mathbf{P}_a^{A'}\ \mathbf{e}'^a = \mathbf{0}$. These points are called the **centres of projection** and each projects to the **epipole** in the opposite image: $\mathbf{e}^A \equiv \mathbf{P}_a^A \mathbf{e}'^a$ and $\mathbf{e}^{A'} \equiv \mathbf{P}_a^{A'}\mathbf{e}^a$. If the centres of projection are distinct, the two projections define a 3×3 rank 2 tensor called the **fundamental matrix** $\mathbf{F}_{AA'}$ [4]. This maps any given image point $\mathbf{x}^A$ ($\mathbf{x}^{A'}$) to a corresponding **epipolar line** $\mathbf{l}_{A'} \sim \mathbf{F}_{AA'}\mathbf{x}^A$ ($\mathbf{l}_A \sim \mathbf{F}_{AA'}\mathbf{x}^{A'}$) in the other image. Two image points correspond in the sense that they could be the projections of a single world point if and only if each lies on the epipolar line of the other: $\mathbf{F}_{AA'}\ \mathbf{x}^A\mathbf{x}^{A'} = 0$. The null directions of the fundamental matrix are the epipoles: $\mathbf{F}_{AA'}\ \mathbf{e}^A = \mathbf{0}$ and $\mathbf{F}_{AA'}\ \mathbf{e}^{A'} = \mathbf{0}$, so every epipolar line must pass through the corresponding epipole. The fundamental matrix $\mathbf{F}_{AA'}$ can be estimated from image correspondences even when the image projections are unknown.

Two image vectors $\mathbf{x}^A$ and $\mathbf{x}^{A'}$ can be packed into a single 6 component vector $\mathbf{x}^\alpha = (\mathbf{x}^A\ \ \mathbf{x}^{A'})^\top$ where $\alpha = 0, 1, 2, 0', 1', 2'$. The space of such vectors will be called **homogeneous joint image space** $\mathcal{H}^\alpha$. Quotienting out the overall scale factor in $\mathcal{H}^\alpha$ produces a 5 dimensional projective space called **projective joint image space** $\mathcal{P}^\alpha$. The two 3×4 image projection matrices can be stacked into a single 6×4 **joint projection matrix** $\mathbf{P}_a^\alpha \equiv (\mathbf{P}_a^A\ \ \mathbf{P}_a^{A'})^\top$. If the centres of projection are distinct, no point in

$\mathcal{P}^a$ is simultaneously killed by both projections, so the joint projection matrix has a vanishing kernel and hence rank 4. This implies that the joint projection is a nonsingular linear bijection from $\mathcal{H}^a$ onto its image space in $\mathcal{H}^\alpha$. This 4 dimensional image space will be called the **homogeneous joint image** $\mathcal{I}^\alpha$. Descending to $\mathcal{P}^\alpha$, the joint projection becomes a bijective projective equivalence between $\mathcal{P}^a$ and the **projective joint image** $\mathcal{PI}^\alpha$ (the projection of $\mathcal{I}^\alpha$ into $\mathcal{P}^\alpha$). The projection of $\mathcal{PI}^\alpha$ to each image is just a trivial deletion of coordinates, so *the projective joint image is a complete projective replica of the world space in image coordinates*. Unfortunately, $\mathcal{PI}^\alpha$ is not quite unique. Any rescaling $\{\mathbf{P}_a^A, \mathbf{P}_a^{A'}\} \rightarrow \{\lambda\mathbf{P}_a^A, \lambda'\mathbf{P}_a^{A'}\}$ of the underlying projection matrices produces a different but equivalent space $\mathcal{PI}^\alpha$. However modulo this arbitrary choice of scaling the projective joint image is canonically defined by the physical situation.

Now suppose that the projection matrices are unknown but the fundamental matrix has been estimated from image measurements. Since $\mathbf{F}$ has rank 2, it can be decomposed (non-uniquely!) as

$$\mathbf{F}_{AA'} \;=\; \mathbf{u}_A\,\mathbf{v}_{A'} - \mathbf{v}_A\,\mathbf{u}_{A'} \;=\; \mathrm{Det}\begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix}$$

where $\mathbf{u}_A \not\sim \mathbf{v}_A$ and $\mathbf{u}_{A'} \not\sim \mathbf{v}_{A'}$ are two pairs of independent image covectors. It is easy to see that $\mathbf{u}_A \leftrightarrow \mathbf{u}_{A'}$ and $\mathbf{v}_A \leftrightarrow \mathbf{v}_{A'}$ are actually pairs of corresponding epipolar lines[1]. In terms of joint image space, the $\mathbf{u}$'s and $\mathbf{v}$'s can be viewed as a pair of 6 component covectors defining a 4 dimensional linear subspace $\mathcal{I}^\alpha$ of $\mathcal{H}^\alpha$ via the equations:

$$\mathcal{I}^\alpha \;\equiv\; \left\{ \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{A'} \end{pmatrix} \mid \begin{pmatrix} \mathbf{u}_A\,\mathbf{x}^A + \mathbf{u}_{A'}\,\mathbf{x}^{A'} \\ \mathbf{v}_A\,\mathbf{x}^A + \mathbf{v}_{A'}\,\mathbf{x}^{A'} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix} \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{A'} \end{pmatrix} \;=\; \mathbf{0} \right\}$$

Trivial use of the constraint equations shows that any point $(\mathbf{x}^A \;\; \mathbf{x}^{A'})^\top$ of $\mathcal{I}^\alpha$ automatically satisfies the epipolar constraint $\mathbf{F}_{AA'}\,\mathbf{x}^A\mathbf{x}^{A'} = 0$. In fact, given any $(\mathbf{x}^A \;\; \mathbf{x}^{A'})^\top \in \mathcal{H}^\alpha$, the equations

$$\begin{aligned} \mathbf{0} \;&=\; \begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix} \begin{pmatrix} \lambda\,\mathbf{x}^A \\ \lambda'\,\mathbf{x}^{A'} \end{pmatrix} \\ &=\; \begin{pmatrix} \mathbf{u}_A\mathbf{x}^A & \mathbf{u}_{A'}\mathbf{x}^{A'} \\ \mathbf{v}_A\mathbf{x}^A & \mathbf{v}_{A'}\mathbf{x}^{A'} \end{pmatrix} \begin{pmatrix} \lambda \\ \lambda' \end{pmatrix} \end{aligned}$$

have a nontrivial solution if and only if

$$\mathbf{F}_{AA'}\,\mathbf{x}^A\mathbf{x}^{A'} \;=\; \mathrm{Det}\begin{pmatrix} \mathbf{u}_A\mathbf{x}^A & \mathbf{u}_{A'}\mathbf{x}^{A'} \\ \mathbf{v}_A\mathbf{x}^A & \mathbf{v}_{A'}\mathbf{x}^{A'} \end{pmatrix} \;=\; 0$$

In other words, the set of matching point pairs in the two images is exactly the set of pairs that can be rescaled to lie in $\mathcal{I}^\alpha$. *Up to a rescaling, the joint image is the set of matching points in the two images.*

A priori, $\mathcal{I}^\alpha$ depends on the choice of the decomposition $\mathbf{F}_{AA'} = \mathbf{u}_A\,\mathbf{v}_{A'} - \mathbf{v}_A\,\mathbf{u}_{A'}$. In fact appendix B shows that the most general redefinition of the $\mathbf{u}$'s and $\mathbf{v}$'s that leaves $\mathbf{F}$ unchanged up to scale is

$$\begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix} \longrightarrow \Lambda \begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix} \begin{pmatrix} 1/\lambda & 0 \\ 0 & 1/\lambda' \end{pmatrix}$$

where Λ is an arbitrary nonsingular 2×2 matrix and $\{\lambda, \lambda'\}$ are arbitrary nonzero relative scale factors. Λ is a linear mixing of the constraint vectors and has no effect on the location of $\mathcal{I}^\alpha$, but λ and λ' represent rescalings of the image coordinates that move $\mathcal{I}^\alpha$ bodily according to

$$\begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{A'} \end{pmatrix} \longrightarrow \begin{pmatrix} \lambda\,\mathbf{x}^A \\ \lambda'\,\mathbf{x}^{A'} \end{pmatrix}$$

Hence, given $\mathbf{F}$ and an arbitrary choice of the relative image scaling the joint image $\mathcal{I}^\alpha$ is defined uniquely.

Appendix B also shows that given any pair of nonsingular projection matrices $\mathbf{P}_a^A$ and $\mathbf{P}_a^{A'}$ compatible with $\mathbf{F}_{AA'}$ in the sense that the projection of every point of $\mathcal{P}^a$ satisfies the epipolar constraint $\mathbf{F}_{AA'}\,\mathbf{P}_a^A\mathbf{P}_b^{A'}\,\mathbf{x}^a\mathbf{x}^b = 0$, the $\mathcal{I}^\alpha$ arising from factorization of $\mathbf{F}$ is projectively equivalent to the $\mathcal{I}^\alpha$ arising from the projection matrices. (Here, nonsingular means that each matrix has rank 3 and the joint matrix has rank 4, *i.e.* the centres of projection are unique and distinct). In fact there is a constant rescaling $\{\mathbf{P}_a^A, \mathbf{P}_a^{A'}\} \rightarrow \{\lambda\,\mathbf{P}_a^A, \lambda'\,\mathbf{P}_a^{A'}\}$ that makes the two coincide.

In summary, the fundamental matrix can be factorized to define a three dimensional projective subspace $\mathcal{PI}^\alpha$ of the space of combined image coordinates. $\mathcal{PI}^\alpha$ is projectively equivalent to the 3D

[1] *Epipolarity:* $\mathbf{u}_A\,\mathbf{e}^A = 0 = \mathbf{v}_A\,\mathbf{e}^A$ follows from $\mathbf{0} = \mathbf{F}_{AA'}\,\mathbf{e}^A = (\mathbf{u}_A\mathbf{e}^A)\,\mathbf{v}_{A'} - (\mathbf{v}_A\mathbf{e}^A)\,\mathbf{u}_{A'}$, given the independence of $\mathbf{u}_{A'}$ and $\mathbf{v}_{A'}$ for rank 2 $\mathbf{F}$. *Correspondence:* For any $\mathbf{x}^A$ on $\mathbf{u}_A$, $\mathbf{u}_A\,\mathbf{x}^A = 0$ implies that $\mathbf{F}_{AA'}\,\mathbf{x}^A = -(\mathbf{v}_A\mathbf{x}^A)\,\mathbf{u}_{A'} \sim \mathbf{u}_{A'}$.

world and uniquely defined by the images up to an arbitrary choice of a single relative scale factor. Projective reconstruction in $\mathcal{PI}^\alpha$ is simply a matter of rescaling the homogeneous image measurements. This paper investigates the geometry of $\mathcal{PI}^\alpha$ and its multi-image counterparts and argues that up to the choice of scale factor, they provide *the* natural canonical projective reconstruction of the information in the images: all other reconstructions are merely different ways of looking at the information contained in $\mathcal{PI}^\alpha$.

4 Too Many Joint Images

Now consider the general case of projection into $m \geq 1$ images. We will model the world and images respectively as d and D_i dimensional projective spaces $\mathcal{P}^a$ $(a = 0, \ldots, d)$ and $\mathcal{P}^{A_i}$ $(A_i = 0, \ldots, D_i, i = 1, \ldots, m)$ and use homogeneous coordinates everywhere. It may appear more natural to use Euclidean or affine spaces, but when it comes to discussing perspective projection it is simpler to view things as (fragments of) projective space. The usual Cartesian and pixel coordinates are still inhomogeneous local coordinate systems covering almost all of the projective world and image manifolds, so projectivization does not change the essential situation too much.

In homogeneous coordinates the perspective image projections are represented by homogeneous $(D_i+1)\times(d+1)$ matrices $\{\mathbf{P}_a^{A_i} | i = 1, \ldots, m\}$ that take homogeneous representatives of world points $\mathbf{x}^a \in \mathcal{P}^a$ to homogeneous representatives of image points $\mathbf{x}^{A_i} \sim \mathbf{P}_a^{A_i}\mathbf{x}^a \in \mathcal{P}^{A_i}$. The homogeneous vectors and matrices representing world points $\mathbf{x}^a$, image points $\mathbf{x}^{A_i}$ and projections $\mathbf{P}_a^{A_i}$ are each defined only up to scale. Arbitrary nonzero rescalings of them do not change the physical situation because the rescaled world and image vectors still represent the same points of the underlying projective spaces $\mathcal{P}^a$ and $\mathcal{P}^{A_i}$, and the projection equations $\mathbf{x}^{A_i} \sim \mathbf{P}_a^{A_i}$ still hold up to scale.

Any collection of m image points $\{\mathbf{x}^{A_i} | i = 1, \ldots, m\}$ can be viewed as a single point in the Cartesian product $\mathcal{P}^{A_1} \times \mathcal{P}^{A_2} \times \cdots \times \mathcal{P}^{A_m}$ of the individual projective image spaces. This is a $D = \sum_{i=1}^m D_i$ dimensional differentiable manifold whose local inhomogeneous coordinates are just the combined pixel coordinates of all the image

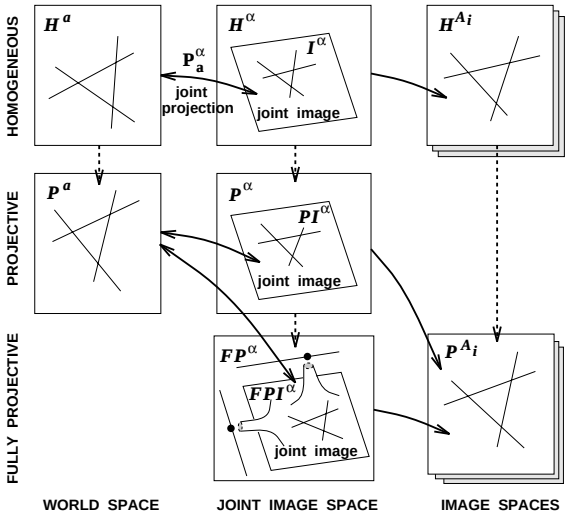


Figure 1: The various joint images and projections.

points. Since any m-tuple of matching points is an element of $\mathcal{P}^{A_1} \times \cdots \times \mathcal{P}^{A_m}$, it may seem that this space is the natural arena for multi-image projective reconstruction. This is almost true but we need to be a little more careful. Although most world points can be represented by their projections in $\mathcal{P}^{A_1} \times \cdots \times \mathcal{P}^{A_m}$, the centres of projection are missing because they fail to project to anything at all in their own images. To represent these, extra points must be glued on to $\mathcal{P}^{A_1} \times \cdots \times \mathcal{P}^{A_m}$.

When discussing perspective projections it is convenient to introduce homogeneous coordinates. A separate homogenizer is required for each image, so the result is just the Cartesian product $\mathcal{H}^{A_1} \times \mathcal{H}^{A_2} \times \cdots \times \mathcal{H}^{A_m}$ of the individual homogeneous image spaces $\mathcal{H}^{A_i}$. We will call this $D + m$ dimensional vector space **homogeneous joint image space** $\mathcal{H}^\alpha$. By quotienting out the overall scale factor in $\mathcal{H}^\alpha$ in the usual way, we can view it as a $D + m - 1$ dimensional projective space $\mathcal{P}^\alpha$ called **projective joint image space**. This is a *bona fide* projective space but it still contains the arbitrary relative scale factors of the component images. A point of $\mathcal{H}^\alpha$ can be represented as a $D + m$ component column vector $\mathbf{x}^\alpha = (\mathbf{x}^{A_1} \cdots \mathbf{x}^{A_m})^\top$ where the $\mathbf{x}^{A_i}$ are homogeneous coordinate vectors in each image. We will think of the index α as taking values $0_1, 1_1, \ldots, D_i, 0_{i+1}, \ldots, D_m$, where the subscripts indicate the image the coordinate came from. An individual image vector $\mathbf{x}^{A_i}$ can be thought of as a vector in $\mathcal{H}^\alpha$ whose non-image-i components vanish.

Since the coordinates of each image are only defined up to scale, the natural definition of the equivalence relation '$\sim$' on $\mathcal{H}^\alpha$ is 'equality up to individual rescalings of the component images': $(\mathbf{x}^{A_1} \ \cdots \ \mathbf{x}^{A_m})^\top \sim (\lambda_1 \mathbf{x}^{A_1} \ \cdots \ \lambda_m \mathbf{x}^{A_m})^\top$ for all $\{\lambda_i \neq 0\}$. So long as none of the $\mathbf{x}^{A_i}$ vectors vanish, the equivalence classes of '$\sim$' are m-dimensional subspaces of $\mathcal{H}^\alpha$ that correspond exactly to the points of $\mathcal{P}^{A_1} \times \cdots \times \mathcal{P}^{A_m}$. However when some of the $\mathbf{x}^{A_i}$ vanish the equivalence classes are lower dimensional subspaces that have no corresponding point in $\mathcal{P}^{A_1} \times \cdots \times \mathcal{P}^{A_m}$. We will call the entire stratified set of equivalence classes **fully projective joint image space** $\mathcal{FP}^\alpha$. This is basically $\mathcal{P}^{A_1} \times \cdots \times \mathcal{P}^{A_m}$ augmented with the lower dimensional product spaces $\mathcal{P}^{A_i} \times \cdots \times \mathcal{P}^{A_j}$ for each proper subset of images $i, \ldots, j$. Most world points project to 'regular' points of $\mathcal{FP}^\alpha$ in $\mathcal{P}^{A_1} \times \cdots \times \mathcal{P}^{A_m}$, but the centres of projection project into lower dimensional fragments of $\mathcal{FP}^\alpha$.

A set of perspective projections into m projective images $\mathcal{P}^{A_i}$ defines a unique **joint projection** into the fully projective joint projective image space $\mathcal{FP}^\alpha$. Given an arbitrary choice of scaling for the homogeneous representatives $\{\mathbf{P}_a^{A_i} \mid i = 1, \ldots, m\}$ of the individual image projections, the joint projection can be represented as a single $(D+m)\times(d+1)$ **joint projection matrix**

$$\mathbf{P}_a^\alpha \equiv \begin{pmatrix} \mathbf{P}_a^{A_1} \\ \vdots \\ \mathbf{P}_a^{A_m} \end{pmatrix} : \mathcal{H}^a \longrightarrow \mathcal{H}^\alpha$$

which defines a projective mapping between the underlying projective spaces $\mathcal{P}^a$ and $\mathcal{P}^\alpha$. A rescaling $\{\mathbf{P}_a^{A_i}\} \rightarrow \{\lambda_i \mathbf{P}_a^{A_i}\}$ of the individual image projection matrices does not change the physical situation or the fully projective joint projection on $\mathcal{FP}^\alpha$, but it *does* change the joint projection matrix $\mathbf{P}_a^\alpha$ and the resulting projections from $\mathcal{H}^a$ to $\mathcal{H}^\alpha$ and from $\mathcal{P}^a$ to $\mathcal{P}^\alpha$. An arbitrary choice of the individual projection scalings is always necessary to make things concrete.

Given a choice of scaling for the components of $\mathbf{P}_a^\alpha$, the image of $\mathcal{H}^a$ in $\mathcal{H}^\alpha$ under the joint projection $\mathbf{P}_a^\alpha$ will be called the **homogeneous joint image** $\mathcal{I}^\alpha$. This is the set of joint image space points that are the projection of some point in world space: $\{\mathbf{P}_a^\alpha \mathbf{x}^a \in \mathcal{H}^\alpha | \, \mathbf{x}^a \in \mathcal{H}^a\}$. In $\mathcal{I}^\alpha$, each world point is represented by its homogeneous vector of image coordinates. Similarly we can define the projective and fully projective joint images $\mathcal{PI}^\alpha$ and $\mathcal{FPI}^\alpha$ as the images of the projective world space $\mathcal{P}^a$ in the projective and fully projective joint image spaces $\mathcal{P}^\alpha$ and $\mathcal{FP}^\alpha$ under the projective and fully projective joint projections. (Equivalently, $\mathcal{PI}^\alpha$ and $\mathcal{FPI}^\alpha$ are the projections of $\mathcal{I}^\alpha$ to $\mathcal{P}^\alpha$ and $\mathcal{FP}^\alpha$).

If the $(D+m)\times(d+1)$ joint projection matrix $\mathbf{P}_a^\alpha$ has rank less than $d+1$ it will have a non-trivial kernel and many world points will project to the same set of image points, so unique reconstruction will be impossible. On the other hand if $\mathbf{P}_a^\alpha$ has rank $d+1$, the homogeneous joint image $\mathcal{I}^\alpha$ will be a $d+1$ dimensional linear subspace of $\mathcal{H}^\alpha$ and $\mathbf{P}_a^\alpha$ will be a nonsingular linear bijection from $\mathcal{H}^a$ onto $\mathcal{I}^\alpha$. Similarly, the projective joint projection will define a nonsingular projective bijection from $\mathcal{P}^a$ onto the d dimensional projective space $\mathcal{PI}^\alpha$ and the fully projective joint projection will be a bijection (and at most points a diffeomorphism) from $\mathcal{P}^a$ onto $\mathcal{FPI}^\alpha$ in $\mathcal{FP}^\alpha$. Structure in $\mathcal{P}^a$ will be mapped bijectively to projectively equivalent structure in $\mathcal{PI}^\alpha$, so $\mathcal{PI}^\alpha$ will be 'as good as' $\mathcal{P}^a$ as far as projective reconstruction is concerned. Moreover, projection from $\mathcal{PI}^\alpha$ to the individual images is a trivial throwing away of coordinates and scale factors, so structure in $\mathcal{PI}^\alpha$ has a very direct relationship with image measurements.

Unfortunately, although $\mathcal{PI}^\alpha$ is closely related to the images it is not quite canonically defined by the physical situation because it moves when the individual image projection matrices are rescaled. However, the truly canonical structure — the fully projective joint image $\mathcal{FPI}^\alpha$ — has a complex stratified structure that is not so easy to handle. When restricted to the product space $\mathcal{P}^{A_1} \times \cdots \times \mathcal{P}^{A_m}$, $\mathcal{FPI}^\alpha$ is equivalent to the projective space $\mathcal{P}^a$ with each centre of projection 'blown up' to the corresponding image space $\mathcal{P}^{A_i}$. The missing centres of projection lie in lower strata of $\mathcal{FP}^\alpha$. Given this complication, it seems easier to work with the simple projective space $\mathcal{PI}^\alpha$ or its homogeneous representative $\mathcal{I}^\alpha$ and to accept that an arbitrary choice of scale factors will be required. We will do this from now on, but it is important to verify that this arbitrary choice does not affect the final results, particularly as far as numerical methods and error models are concerned. It is also essential to realize that although *for any one point* the projection scale fac-

tors can be chosen arbitrarily, once they are chosen they apply uniformly to all other points: *no matter which scaling is chosen, there is a strong coherence between the scalings of different points.* A central theme of this paper is that the essence of projective reconstruction is the recovery of this scale coherence from image measurements.

5 The Joint Image Grassmannian Tensor

We can view the joint projection matrix $\mathbf{P}_a^\alpha$ (with some choice of the internal scalings) in two ways: *(i)* as a collection of m projection matrices from $\mathcal{P}^a$ to the m images $\mathcal{P}^{A_i}$; *(ii)* as a set of $d+1$ $(D+m)$-component column vectors $\{\mathbf{P}_a^\alpha | a = 0, \dots, d\}$ that span the joint image subspace $\mathcal{I}^\alpha$ in $\mathcal{H}^\alpha$. From the second point of view the images of the standard basis $\{(10\cdots0)^\top, (01\cdots0)^\top, \dots, (00\cdots1)^\top\}$ for $\mathcal{H}^a$ (*i.e.* the columns of $\mathbf{P}_a^\alpha$) form a basis for $\mathcal{I}^\alpha$ and a set of homogeneous coordinates $\{x^a | a = 0, \dots, d\}$ can be viewed either as the coordinates of a point $\mathbf{x}^a$ in $\mathcal{P}^a$ or as the coordinates of a point $\mathbf{P}_a^\alpha \mathbf{x}^a$ in $\mathcal{I}^\alpha$ with respect to the basis $\{\mathbf{P}_a^\alpha | a = 0, \dots, d\}$. Similarly, the columns of $\mathbf{P}_a^\alpha$ and the $(d+2)^{nd}$ column $\sum_{a=0}^{d} \mathbf{P}_a^\alpha$ form a projective basis for $\mathcal{PI}^\alpha$ that is the image of the standard projective basis $\{(10\cdots0)^\top, \dots, (00\cdots1)^\top, (11\cdots1)^\top\}$ for $\mathcal{P}^a$.

This means that *any reconstruction in $\mathcal{P}^a$ can be viewed as reconstruction in $\mathcal{PI}^\alpha$ with respect to a particular choice of basis there.* This is important because we will see that (up to a choice of scale factors) *$\mathcal{PI}^\alpha$ is canonically defined by the imaging situation and can be recovered directly from image measurements.* In fact we will show that the information in the combined matching constraints is exactly the location of the subspace $\mathcal{PI}^\alpha$ in $\mathcal{P}^\alpha$, and this is exactly the information we need to make a *canonical geometric reconstruction of $\mathcal{P}^a$ in $\mathcal{PI}^\alpha$ from image measurements.*

By contrast we can not hope to recover the basis in $\mathcal{P}^a$ or the individual columns of $\mathbf{P}_a^\alpha$ by image measurements. In fact any two worlds that project to the same joint image are indistinguishable so far as image measurements are concerned. Under an arbitrary nonsingular projective transformation $\mathbf{x}^a \to \tilde{\mathbf{x}}^{a'} = (\Lambda^{-1})^{a'}{}_b \, \mathbf{x}^b$ between $\mathcal{P}^a$ and some other world space $\mathcal{P}^{a'}$, the projection matrices (and hence the basis vectors for $\mathcal{PI}^\alpha$) must change according to $\mathbf{P}_a^\alpha \to \tilde{\mathbf{P}}_{a'}^\alpha = \mathbf{P}_b^\alpha \, \Lambda^b{}_{a'}$ to compensate. The new basis vectors are a linear combination of the old ones so the space $\mathcal{PI}^\alpha$ they span is not changed, but the individual vectors *are* changed: all we can hope to recover from the images is the geometric location of $\mathcal{PI}^\alpha$, not its particular basis.

But how can we specify the location of $\mathcal{PI}^\alpha$ geometrically? We originally defined it as the span of the columns of the joint projection $\mathbf{P}_a^\alpha$, but that is rather inconvenient. For one thing $\mathcal{PI}^\alpha$ depends only on the span and not on the individual vectors, so it is redundant to specify every component of $\mathbf{P}_a^\alpha$. What is worse, the redundant components are exactly the things that can not be recovered from image measurements. It is not even clear how we would use a 'span' even if we did manage to obtain it.

Algebraic geometers encountered this sort of problem long ago and developed a useful partial solution called **Grassmann coordinates** (see appendix A). Recall that $[a \cdots c]$ denotes antisymmetrization over all permutations of the indices $a \cdots c$. Given $k+1$ independent vectors $\{\mathbf{x}_i^a | \, i = 0, \dots, k\}$ in a $d+1$ dimensional vector space $\mathcal{H}^a$, it turns out that the antisymmetric $k+1$ index **Grassmann tensor** $\mathbf{x}^{a_0 \cdots a_k} \equiv \mathbf{x}_0^{[a_0} \cdots \mathbf{x}_k^{a_k]}$ uniquely characterizes the $k+1$ dimensional subspace spanned by the vectors and (up to scale) does not depend on the particular vectors of the subspace chosen to define it. In fact a point $\mathbf{y}^a$ lies in the span if and only if it satisfies $\mathbf{x}^{[a_0 \cdots a_k} \mathbf{y}^{a_{k+1}]} = \mathbf{0}$, and under a $(k+1) \times (k+1)$ linear redefinition Λ_j^i of the basis elements $\{\mathbf{x}_i^a\}$, $\mathbf{x}^{a_0 \cdots a_k}$ is simply rescaled by $\mathrm{Det}(\Lambda)$. Up to scale, the components of the Grassmann tensor are the $(k+1) \times (k+1)$ minors of the $(d+1) \times (k+1)$ matrix of components of the $\mathbf{x}_i^a$.

The antisymmetric tensors are global coordinates for the k dimensional subspaces in the sense that each subspace is represented by a unique (up to scale) Grassmann tensor. However the parameterization is highly redundant: for $1 \le k \le d-2$ the $k+1$ index antisymmetric tensors have many more independent components than there are degrees of freedom. In fact only the very special antisymmetric tensors that can be written in the above 'simple' form $\mathbf{x}_0^{[a_0} \cdots \mathbf{x}_k^{a_k]}$ specify subspaces. Those that can are characterized by the quadratic **Grassmann simplicity relations** $\mathbf{x}^{a_0 \cdots [a_k} \, \mathbf{x}^{b_0 \cdots b_k]} = \mathbf{0}$.

In the present case the $d+1$ columns of $\mathbf{P}_a^\alpha$ specify the d dimensional joint image subspace $\mathcal{PI}^\alpha$. Instead of antisymmetrizing over the image space indices α we can get the same effect by contracting the world space indices a with the $d+1$ dimensional alternating tensor. This gives the $d+1$ index antisymmetric **joint image Grassmannian** tensor

$$\begin{aligned}\mathbf{I}^{\alpha_0\alpha_1\cdots\alpha_d} &\equiv \tfrac{1}{(d+1)!}\,\mathbf{P}_{a_0}^{\alpha_0}\,\mathbf{P}_{a_1}^{\alpha_1}\cdots\mathbf{P}_{a_d}^{\alpha_d}\;\varepsilon^{a_0a_1\cdots a_d}\\ &\sim \mathbf{P}_0^{[\alpha_0}\,\mathbf{P}_1^{\alpha_1}\cdots\mathbf{P}_d^{\alpha_d]}\end{aligned}$$

Although we have defined the Grassmann tensor in terms of the columns of the projection matrix basis for $\mathcal{PI}^\alpha$, it is actually an intrinsic property of $\mathcal{PI}^\alpha$ that defines and is defined by it in a manner completely independent of the choice of basis (up to scale). In fact we will see that the Grassmann tensor contains exactly the same information as the complete set of matching constraint tensors. Since the matching constraints can be recovered from image measurements, the Grassmann tensor can be too.

As a simple test of plausibility, let us verify that the Grassmann tensor has the correct number of degrees of freedom to encode the imaging geometry required for projective reconstruction. The geometry of an m camera imaging system can be specified by giving each of the m projection mappings modulo an arbitrary overall choice of projective basis in $\mathcal{P}^a$. Up to an arbitrary scale factor, a $(D_i+1)\times(d+1)$ projection matrix is defined by $(D_i+1)(d+1)-1$ parameters while a projective basis in $\mathcal{P}^a$ has $(d+1)(d+1)-1$ degrees of freedom. The m camera projective geometry therefore has

$$\begin{aligned}\sum_{i=1}^{m}\Big((D_i+1)\,(d+1)-1\Big) \;&-\; \big((d+1)^2-1\big)\\ &=\; (D+m-d-1)\,(d+1)-m+1\end{aligned}$$

independent degrees of freedom. For example $11m-15$ parameters are required to specify the geometry of m 2D cameras viewing 3D projective space [13].

The antisymmetric Grassmann tensor $\mathbf{I}^{\alpha_0\cdots\alpha_d}$ has $\binom{D+m}{d+1}$ *linearly* independent components. However the quadratic Grassmann relations reduce the number of *algebraically* independent components to the dimension $(D+m-d-1)(d+1)$ of the space of possible locations of the joint image $\mathcal{I}^\alpha$ in $\mathcal{P}^\alpha$. (Joint image locations are locally parameterized by the $((D+m)-(d+1))\times(d+1)$ matrices, or equivalently by giving $d+1$ $(D+m)$-component spanning basis vectors in $\mathcal{P}^\alpha$ modulo $(d+1)\times(d+1)$ linear redefinitions). The overall scale factor of $\mathbf{I}^{\alpha_0\cdots\alpha_d}$ has already been subtracted from this count, but it still contains the $m-1$ arbitrary *relative* scale factors of the m images. Subtracting these leaves the Grassmann tensor (or the equivalent matching constraint tensors) with $(D+m-d-1)\,(d+1)-m+1$ physically meaningful degrees of freedom. This agrees with the above degree-of-freedom count based on projection matrices.

6 Reconstruction Equations

Suppose we are given a set of m image points $\{\mathbf{x}^{A_i}|\,i=1,\ldots,m\}$ that may correspond to an unknown world point $\mathbf{x}^a$ via some known projection matrices $\mathbf{P}_a^{A_i}$. Can the world point $\mathbf{x}^a$ be recovered, and if so, how?

As usual we will work projectively in homogeneous coordinates and suppose that arbitrary nonzero scalings have been chosen for the $\mathbf{x}^{A_i}$ and $\mathbf{P}_a^{A_i}$. The image vectors can be stacked into a $D+m$ component joint homogeneous image vector $\mathbf{x}^\alpha$ and the projection matrices can be stacked into a $(D+m)\times(d+1)$ component joint homogeneous projection matrix, where d is the world dimension and $D=\sum_{i=1}^m D_i$ is the sum of the image dimensions.

Any candidate reconstruction $\mathbf{x}^a$ must project to the correct point in each image: $\mathbf{x}^{A_i}\sim\mathbf{P}_a^{A_i}\,\mathbf{x}^a$. Inserting variables $\{\lambda_i|\,i=1,\ldots,m\}$ to represent the unknown scale factors gives m homogeneous equations $\mathbf{P}_a^{A_i}\,\mathbf{x}^a-\lambda_i\,\mathbf{x}^{A_i}=\mathbf{0}$. These can be written as a single $(D+m)\times(d+1+m)$ homogeneous linear system, the **basic reconstruction equations**:

$$\left(\begin{array}{c|cccc}\mathbf{P}_a^\alpha & \mathbf{x}^{A_1} & \mathbf{0} & \cdots & \mathbf{0}\\ & \mathbf{0} & \mathbf{x}^{A_2} & \cdots & \mathbf{0}\\ & \vdots & \vdots & \ddots & \vdots\\ & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{x}^{A_m}\end{array}\right)\left(\begin{array}{c}\mathbf{x}^a\\ \hline -\lambda_1\\ -\lambda_2\\ \vdots\\ -\lambda_m\end{array}\right)=\mathbf{0}$$

Any nonzero solution of these equations gives a reconstructed world point $\mathbf{x}^a$ consistent with the image measurements $\mathbf{x}^{A_i}$, and also provides the unknown scale factors $\{\lambda_i\}$.

These equations will be studied in detail in the next section. However we can immediately remark that if there are less image measurements than world dimensions ($D < d$) there will be at least two more free variables than equations and the solution (if it exists) can not be unique. So from now on we require $D \geq d$.

On the other hand, if there are more measurements than world dimensions ($D > d$) the system will usually be overspecified and a solution will exist only when certain constraints between the projection matrices $\mathbf{P}_a^{A_i}$ and the image measurements $\mathbf{x}^{A_i}$ are satisfied. We will call these constraints **matching constraints** and the inter-image tensors they generate **matching tensors**. The simplest example is the epipolar constraint.

It is also clear that there is no hope of a unique solution if the rank of the joint projection matrix $\mathbf{P}_a^\alpha$ is less than $d+1$, because any vector in the kernel of $\mathbf{P}_a^\alpha$ can be added to a solution without changing the projection at all. So we will also require the joint projection matrix to have maximal rank (*i.e.* $d+1$). Recall that this implies that the joint projection $\mathbf{P}_a^\alpha$ is a bijection from $\mathcal{P}^a$ onto its image the joint image $\mathcal{PI}^\alpha$ in $\mathcal{P}^\alpha$. (This is necessary but not always sufficient for a unique reconstruction).

In the usual 3D→2D case the individual projections are 3×4 rank 3 matrices and each has a one dimensional kernel: the centre of projection. Provided there are at least two distinct centres of projection among the image projections, no point will project to zero in every image and the joint projection will have a vanishing kernel and hence maximal rank. (It turns out that in this case $\mathrm{Rank}(\mathbf{P}_a^\alpha) = 4$ is also *sufficient* for a unique reconstruction).

Recalling that the joint projection columns $\{\mathbf{P}_a^\alpha |\, a = 0, \ldots, d\}$ form a basis for the homogeneous joint image $\mathcal{I}^\alpha$ and treating the $\mathbf{x}^{A_i}$ as vectors in $\mathcal{H}^\alpha$ whose other components vanish, we can interpret the reconstruction equations as the geometrical statement that the space spanned by the image vectors $\{\mathbf{x}^{A_i} |\, i = 1, \ldots, m\}$ in $\mathcal{H}^\alpha$ must intersect $\mathcal{I}^\alpha$. At the intersection there is a point of $\mathcal{H}^\alpha$ that can be expressed: *(i)* as a rescaling of the image measurements $\sum_i \lambda_i \mathbf{x}^{A_i}$; *(ii)* as a point of $\mathcal{I}^\alpha$ with coordinates $\mathbf{x}^a$ in the basis $\{\mathbf{P}_a^\alpha |\, a = 0, \ldots, d\}$; *(iii)* as the projection into $\mathcal{I}^\alpha$ of a world point $\mathbf{x}^a$ under $\mathbf{P}_a^\alpha$. (Since $\mathcal{H}^a$ is isomorphic to $\mathcal{I}^\alpha$ under $\mathbf{P}_a^\alpha$, the last two points of view are equivalent).

This construction is important because although neither the coordinate system in $\mathcal{H}^a$ nor the columns of $\mathbf{P}_a^\alpha$ can be recovered from image measurements, the joint image $\mathcal{I}^\alpha$ *can* be recovered (up to an arbitrary choice of relative scaling). In fact the content of the matching constraints is *precisely* the location of $\mathcal{I}^\alpha$ in $\mathcal{H}^\alpha$. This gives a completely geometric and almost canonical projective reconstruction technique in $\mathcal{I}^\alpha$ that requires only the scaling of joint image coordinates. A choice of basis in $\mathcal{I}^\alpha$ is necessary only to map the construction back into world coordinates.

Recalling that the joint image can be located by giving its Grassmann coordinate tensor $\mathbf{I}^{\alpha\beta\cdots\gamma}$ and that in terms of this a point lies in the joint image if and only if $\mathbf{I}^{[\alpha\beta\cdots\gamma}\,\mathbf{x}^{\delta]} = \mathbf{0}$, the basic reconstruction system is equivalent to the following **joint image reconstruction equations**

$$\mathbf{I}^{\,[\alpha\beta\cdots\gamma}\,.\,\left(\sum_{i=1}^{m}\lambda_i\,\mathbf{x}^{A_i]}\right) \;=\; \mathbf{0}$$

This is a redundant system of homogeneous linear equations for the λ_i given the $\mathbf{I}^{\alpha\beta\cdots\gamma}$ and the $\mathbf{x}^{A_i}$. It will be used in section 10 to derive implicit 'reconstruction' methods that are independent of any choice of world or joint image basis.

There is yet another form of the reconstruction equations that is more familiar and compact but slightly less symmetrical. For notational convenience suppose that $\mathbf{x}^{0_i} \neq 0$. (We use component 0 for normalization. Each image vector has at least one nonzero component so the coordinates can be relabelled if necessary so that $\mathbf{x}^{0_i} \neq 0$). The projection equations $\mathbf{P}_a^{A_i}\,\mathbf{x}^a = \lambda_i\,\mathbf{x}^{A_i}$ can be solved for the 0^{th} component to give $\lambda_i = (\mathbf{P}_a^{0_i}\,\mathbf{x}^a)/\mathbf{x}^{0_i}$. Substituting back into the projection equations for the other components yields the following constraint equations for $\mathbf{x}^a$ in terms of $\mathbf{x}^{A_i}$ and $\mathbf{P}_a^{A_i}$:

$$\left(\mathbf{x}^{0_i}\,\mathbf{P}_a^{A_i} - \mathbf{x}^{A_i}\,\mathbf{P}_a^{0_i}\right)\mathbf{x}^a = 0 \qquad A_i = 1, \ldots, D_i$$

(Equivalently, $\mathbf{x}^{A_i} \sim \mathbf{P}_a^{A_i}\mathbf{x}^a$ implies $\mathbf{x}^{[A_i}\mathbf{P}_a^{B_i]}\mathbf{x}^a = \mathbf{0}$, and the constraint follows by setting $B_i = 0_i$). Each of these equations constrains $\mathbf{x}^a$ to lie in a hyperplane in the d-dimensional world space. Combining the constraints from all the images gives the following $D \times (d+1)$ system of **reduced recon-**

struction equations:

$$\begin{pmatrix} \mathbf{x}^{0_1}\,\mathbf{P}_a^{A_1} - \mathbf{x}^{A_1}\,\mathbf{P}_a^{0_1} \\ \vdots \\ \mathbf{x}^{0_m}\,\mathbf{P}_a^{A_m} - \mathbf{x}^{A_m}\,\mathbf{P}_a^{0_m} \end{pmatrix} \mathbf{x}^a \;=\; \mathbf{0} \qquad (A_i{=}1,\dots,D_i)$$

Again a solution of these equations provides the reconstructed homogeneous coordinates of a world point in terms of image measurements, and again the equations are usually overspecified when $D > d$. Provided $\mathbf{x}^{0_i} \neq 0$ the reduced equations are equivalent to the basic ones. Their compactness makes them attractive for numerical work, but their lack of symmetry makes them less suitable for symbolic derivations such as the extraction of the matching constraints. In practice both representations are useful.

7 Matching Constraints

Now we are finally ready to derive the constraints that a set of image points must satisfy in order to be the projections of some world point. We will assume that there are more image than space dimensions ($D > d$) (if not there are no matching constraints) and that the joint projection matrix $\mathbf{P}_a^\alpha$ has rank $d+1$ (if not there are no unique reconstructions). We will work from the basic reconstruction equations, with odd remarks on the equivalent reduced case.

In either case there are $D-d-1$ more equations than variables and the reconstruction systems are overspecified. The image points must satisfy $D-d$ additional independent constraints for there to be a solution, since one degree of freedom is lost in the overall scale factor. For example in the usual 3D→2D case there are $2m-3$ additional scalar constraints: one for the first pair of images and two more for each additional image.

An overspecified homogeneous linear system has nontrivial solutions exactly when its coefficient matrix is rank deficient, which occurs exactly when all of its maximal-size minors vanish. For generic sets of image points the reconstruction systems typically have full rank: solutions exist only for the special sets of image points for which all of the $(d+m+1)\times(d+m+1)$ minors of the basic (or $(d+1)\times(d+1)$ minors of the reduced) reconstruction matrix vanish. These minors are exactly the matching constraints.

In either case each of the minors involves all $d+1$ (world-space) columns and some selection of $d+1$ (image-space) rows of the combined projection matrices, multiplied by image coordinates. This means that the constraints will be polynomials (*i.e.* tensors) in the image coordinates with coefficients that are $(d+1)\times(d+1)$ minors of the $(D+m)\times(d+1)$ joint projection matrix $\mathbf{P}_a^\alpha$. We have already seen in section 5 that these minors are precisely the Grassmann coordinates of the *joint image* $\mathcal{I}^\alpha$, the subspace of homogeneous joint image space spanned by the $d+1$ columns of $\mathbf{P}_a^\alpha$. The complete set of these defines $\mathcal{I}^\alpha$ in a manner entirely independent (up to a scale factor) of the choice of basis in $\mathcal{I}^\alpha$: they are the only quantities that *could* have appeared if the equations were to be invariant to this choice of basis (or equivalently, to arbitrary projective transformations of the world space).

Each of the $(d+m+1)\times(d+m+1)$ minors of the basic reconstruction system contains one column from each image, and hence is linear in the coordinates of each image separately and homogeneous of degree m in the combined image coordinates. The final constraint equations will be linear in the coordinates of each image that appears in them. Any choice of $d+m+1$ of the $D+m$ rows of the matrix specifies a minor, so naively there are $\binom{D+m}{d+m+1}$ distinct constraint polynomials, although the simple degree of freedom count given above shows that even in this naive case only $D-d$ of these can be algebraically independent. However the reconstruction matrix has many zero entries and we need to count more carefully.

Each row comes from (contains components from) exactly one image. The only nonzero entries in the image i column are those from image i itself, so any minor that does not include at least one row from each image will vanish. This leaves only $d+1$ of the $m+d+1$ rows free to apportion. On the other hand, if a minor contains only one row from some image — say the $\mathbf{x}^{A_i}$ row for some particular values of i and A_i — it will simply be the product of $\pm\mathbf{x}^{A_i}$ and an $m-1$ image minor because $\mathbf{x}^{A_i}$ is the *only* nonzero entry in its image i column. But exactly the same $(m-1)$-image minor will appear in several other m-image minors, one for each other choice of the coordinate $A_i = 0,\dots,D_i$. At least

one of these coordinates is nonzero, so the vanishing of the $D_i + 1$ m-image minors is equivalent to the vanishing of the single $(m-1)$-image one.

This allows the full set of m-image matching polynomials to be reduced to terms involving at most $d+1$ images. ($d+1$ because there are only $d+1$ spare rows to share out). In the standard 3D→2D case this leaves the following possibilities ($i \neq j \neq k \neq l = 1, \ldots, m$): (*i*) 3 rows each in images i and j; (*ii*) 3 rows in image i, and 2 rows each in images j and k; and (*iii*) 2 rows each in images i, j, k and l. We will show below that these possibilities correspond respectively to fundamental matrices (*i.e.* bilinear two image constraints), Shashua's trilinear three-image constraints [19], and a new quadrilinear four-image constraint. For 3 dimensional space this is the complete list of possibilities: there are *no* irreducible k-image matching constraints for $k > 4$.

We can look at all this in another way. Consider the $d+m+1$ $(D+m)$-component columns of the reconstruction system matrix. Temporarily writing $\mathbf{x}_i^\alpha$ for the image i column whose only nonzero entries are $\mathbf{x}^{A_i}$, the columns are $\{\mathbf{P}_a^\alpha |\, a = 0, \ldots, d\}$ and $\{\mathbf{x}_i^\alpha |\, i = 1, \ldots, m\}$ and we can form them into a $d+m+1$ index antisymmetric tensor $\mathbf{P}_0^{[\alpha_0} \cdots \mathbf{P}_d^{\alpha_d} \mathbf{x}_1^{\beta_1} \cdots \mathbf{x}_m^{\beta_m]}$. Up to scale, the components of this tensor are exactly the possible $(d+m+1) \times (d+m+1)$ minors of the system matrix. The term $\mathbf{x}_i^\alpha$ vanishes unless α is one of the components A_i, so we need at least one index from each image in the index set $\alpha_0, \ldots, \alpha_d, \beta_1, \ldots, \beta_m$. If only one component from image i is present in the set (B_i say, for some fixed value of B_i), we can extract an overall factor of $\mathbf{x}^{B_i}$ as above. Proceeding in this way the tensor can be reduced to irreducible terms of the form $\mathbf{P}_0^{[\alpha_0} \cdots \mathbf{P}_d^{\alpha_d} \mathbf{x}_i^{B_i} \mathbf{x}_j^{B_j} \cdots \mathbf{x}_k^{B_k]}$. These contain anything from 2 to $d+1$ distinct images $i, j, \ldots, k$. The indices $\alpha_0, \ldots, \alpha_d$ are an arbitrary choice of indices from images $i, j, \ldots, k$ in which each image appears at least once. Recalling that up to scale the components of the joint image Grassmannian $\mathbf{I}^{\alpha_0 \cdots \alpha_d}$ are just $\mathbf{P}_0^{[\alpha_0} \cdots \mathbf{P}_d^{\alpha_d]}$, and dropping the redundant subscripts on the $\mathbf{x}_i^{A_i}$, we can write the final constraint equations in the compact form

$$\mathbf{I}^{[A_i A_j \cdots A_k \alpha \cdots \beta}\, \mathbf{x}^{B_i} \mathbf{x}^{B_j} \cdots \mathbf{x}^{B_k]} = \mathbf{0}$$

where $i, j, \ldots, k$ contains between 2 and $d+1$ distinct images. The remaining indices $\alpha \cdots \beta$ can be chosen arbitrarily from any of the images $i, j, \ldots, k$, up to the maximum of $D_i + 1$ indices from each image. (NB: the $\mathbf{x}^{B_i}$ stand for m distinct vectors whose non-i components vanish, not for the single vector $\mathbf{x}^\alpha$ containing all the image measurements. Since $\mathbf{I}^{\alpha_0 \cdots \alpha_d}$ is already antisymmetric and permutations that place a non-i index on $\mathbf{x}^{B_i}$ vanish, it is enough to antisymmetrize separately over the components from each image).

This is all rather intricate, but in three dimensions the possibilities are as follows ($i \neq j \neq k \neq l = 1, \ldots, m$):

$$\begin{aligned} \mathbf{I}^{[A_i B_i A_j B_j}\, \mathbf{x}^{C_i} \mathbf{x}^{C_j]} &= \mathbf{0} \\ \mathbf{I}^{[A_i B_i A_j A_k}\, \mathbf{x}^{C_i} \mathbf{x}^{B_j} \mathbf{x}^{B_k]} &= \mathbf{0} \\ \mathbf{I}^{[A_i A_j A_k A_l}\, \mathbf{x}^{B_i} \mathbf{x}^{B_j} \mathbf{x}^{B_k} \mathbf{x}^{B_l]} &= \mathbf{0} \end{aligned}$$

These represent respectively the epipolar constraint, Shashua's trilinear constraint and the new quadrilinear four image constraint.

We will discuss each of these possibilities in detail below, but first we take a brief look at the constraints that arise from the *reduced* reconstruction system. Each row of this system is linear in the coordinates of one image and in the corresponding rows of the joint projection matrix, so each $(d+1) \times (d+1)$ minor can be expanded into a sum of degree $d+1$ polynomial terms in the image coordinates, with $(d+1) \times (d+1)$ minors of the joint projection matrix (Grassmann coordinates of $\mathcal{PI}^\alpha$) as coefficients. Moreover, any term that contains two non-zeroth coordinates from the same image (say $A_i \neq 0$ and $B_i \neq 0$) vanishes because the row $\mathbf{P}_a^{0_i}$ appears twice in the corresponding coefficient minor. So each term is at most linear in the non-zeroth coordinates of each image. If k_i is the total number of rows from the i^{th} image in the minor, this implies that the zeroth coordinate $\mathbf{x}^{0_i}$ appears either k_i or $k_i - 1$ times in each term to make up the total homogeneity of k_i in the coordinates of the i^{th} image. Throwing away the nonzero overall factors of $(\mathbf{x}^{0_i})^{k_i - 1}$ leaves a constraint polynomial linear in the coordinates of each image and of total degree at most $d+1$, with $(d+1) \times (d+1)$ minors of the joint projection matrix as coefficients. Closer inspection shows that these are the same as the constraint polynomials found above.

7.1 Bilinear Constraints

Now we restrict attention to 2D images of a 3D world and examine each of the three constraint types in turn. First consider the bilinear joint image Grassmannian constraint $\mathbf{I}^{[B_1C_1B_2C_2}\mathbf{x}^{A_1}\mathbf{x}^{A_2]} = \mathbf{0}$, where as usual $\mathbf{I}^{\alpha\beta\gamma\delta} \equiv \frac{1}{4!}\,\mathbf{P}_a^\alpha\mathbf{P}_b^\beta\mathbf{P}_c^\gamma\mathbf{P}_d^\delta\,\varepsilon^{abcd}$. Recalling that it is enough to antisymmetrize over the components from each image separately, the epipolar constraint becomes

$$\mathbf{x}^{[A_1}\,\mathbf{I}^{B_1C_1][B_2C_2}\,\mathbf{x}^{A_2]} \;=\; \mathbf{0}$$

Dualizing both sets of antisymmetric indices by contracting with $\varepsilon_{A_1B_1C_1}\,\varepsilon_{A_2B_2C_2}$ gives the epipolar constraint the equivalent but more familiar form:

$$\begin{aligned} 0 \;&=\; \mathbf{F}_{A_1A_2}\,\mathbf{x}^{A_1}\mathbf{x}^{A_2} \\ &=\; \tfrac{1}{4\cdot 4!}\,\Big(\varepsilon_{A_1B_1C_1}\mathbf{x}^{A_1}\mathbf{P}_a^{B_1}\mathbf{P}_b^{C_1}\Big)\cdot \\ &\qquad \cdot\Big(\varepsilon_{A_2B_2C_2}\mathbf{x}^{A_2}\mathbf{P}_c^{B_2}\mathbf{P}_d^{C_2}\Big)\;\varepsilon^{abcd} \end{aligned}$$

where the $3\times 3 = 9$ component bilinear constraint tensor or **fundamental matrix** $\mathbf{F}_{A_1A_2}$ is defined by

$$\begin{aligned} \mathbf{F}_{A_1A_2} \;&\equiv\; \tfrac{1}{4}\,\varepsilon_{A_1B_1C_1}\,\varepsilon_{A_2B_2C_2}\,\mathbf{I}^{B_1C_1B_2C_2} \\ &=\; \tfrac{1}{4\cdot 4!}\,\Big(\varepsilon_{A_1B_1C_1}\mathbf{P}_a^{B_1}\mathbf{P}_b^{C_1}\Big)\cdot \\ &\qquad \cdot\Big(\varepsilon_{A_2B_2C_2}\mathbf{P}_c^{B_2}\mathbf{P}_d^{C_2}\Big)\;\varepsilon^{abcd} \\ \mathbf{I}^{B_1C_1B_2C_2} \;&=\; \mathbf{F}_{A_1A_2}\,\varepsilon^{A_1B_1C_1}\varepsilon^{A_2B_2C_2} \end{aligned}$$

Equivalently, the epipolar constraint can be derived by direct expansion of the 6×6 basic reconstruction system minor

$$\mathrm{Det}\begin{pmatrix} \mathbf{P}_a^{A_1} & \mathbf{x}^{A_1} & \mathbf{0} \\ \mathbf{P}_a^{A_2} & \mathbf{0} & \mathbf{x}^{A_2} \end{pmatrix} = 0$$

Choosing the image 1 rows and column and any two columns a and b of $\mathbf{P}$ gives a 3×3 subdeterminant $\varepsilon_{A_1B_1C_1}\mathbf{x}^{A_1}\mathbf{P}_a^{B_1}\mathbf{P}_b^{C_1}$. The remaining rows and columns (for image 2 and the remaining two columns c and d of $\mathbf{P}$, say) give the factor $\varepsilon_{A_2B_2C_2}\mathbf{x}^{A_2}\mathbf{P}_c^{B_2}\mathbf{P}_d^{C_2}$ multiplying this subdeterminant in the determinantal sum. Antisymmetrizing over the possible choices of a through d gives the above bilinear constraint equation. When there are only two images, $\mathbf{F}$ can also be written as the inter-image part of the $\mathcal{P}^\alpha$ (six dimensional) dual $\mathbf{F}_{A_1A_2} = \frac{1}{4}\varepsilon_{A_1B_1C_1A_2B_2C_2}\mathbf{I}^{B_1C_1B_2C_2}$. This is why it was generated by the $6-4=2$ six dimensional constraint covectors $\mathbf{u}_\alpha$ and $\mathbf{v}_\alpha$ for $\mathcal{I}^\alpha$ in section 3.

The bilinear constraint equation

$$\begin{aligned} 0 \;=\; &\Big(\varepsilon_{A_1B_1C_1}\mathbf{x}^{A_1}\mathbf{P}_a^{B_1}\mathbf{P}_b^{C_1}\Big)\cdot \\ &\cdot\Big(\varepsilon_{A_2B_2C_2}\mathbf{x}^{A_2}\mathbf{P}_c^{B_2}\mathbf{P}_d^{C_2}\Big)\;\varepsilon^{abcd} \end{aligned}$$

can be interpreted geometrically as follows. The dualization $\varepsilon_{ABC}\,\mathbf{x}^A$ converts an image point $\mathbf{x}^A$ into covariant coordinates in the image plane. Roughly speaking, this represents the point as the pencil of lines through it: for any two lines $\mathbf{l}_A$ and $\mathbf{m}_A$ through $\mathbf{x}^A$, the tensor $\mathbf{l}_{[B}\mathbf{m}_{C]}$ is proportional to $\varepsilon_{ABC}\mathbf{x}^A$. Any covariant image tensor can be 'pulled back' through the linear projection $\mathbf{P}_a^A$ to a covariant tensor in 3D space. An image line $\mathbf{l}_A$ pulls back to the 3D plane $\mathbf{l}_a = \mathbf{l}_A\mathbf{P}_a^A$ through the projection centre that projects to the line. The tensor $\varepsilon_{ABC}\,\mathbf{x}^A$ pulls back to the 2 index covariant tensor $\mathbf{x}_{[bc]} \equiv \varepsilon_{ABC}\mathbf{x}^A\mathbf{P}_b^B\mathbf{P}_c^C$. This is the covariant representation of a line in 3D: the optical ray through $\mathbf{x}^A$. Given any two lines $\mathbf{x}_{[ab]}$ and $\mathbf{y}_{[ab]}$ in 3D space, the requirement that they intersect is $\mathbf{x}_{ab}\,\mathbf{y}_{cd}\,\varepsilon^{abcd} = 0$. So the above bilinear constraint equation really *is* the standard epipolar constraint, *i.e.* the requirement that the optical rays of the two image points must intersect. Similarly, the $\mathbf{F}_{A_1A_2}$ tensor really is the usual fundamental matrix. Of course this can also be illustrated by explicitly writing out terms.

7.2 Trilinear Constraints

Now consider the trilinear, three image Grassmannian constraint $\mathbf{I}^{[B_1C_1B_2B_3}\,\mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3]} = \mathbf{0}$. This corresponds to a 7×7 basic reconstruction minor formed by selecting all three rows from the first image and two each from the remaining two. Restricting the antisymmetrization to each image and contracting with $\varepsilon_{A_1B_1C_1}$ gives the trilinear constraint

$$\mathbf{x}^{A_1}\mathbf{x}^{[A_2}\,\mathbf{G}_{A_1}{}^{B_2][B_3}\,\mathbf{x}^{A_3]} = \mathbf{0}$$

where the $3\times 3\times 3 = 27$ component trilinear constraint tensor $\mathbf{G}_{A_1}{}^{A_2A_3}$ is defined by

$$\begin{aligned} \mathbf{G}_{A_1}{}^{A_2A_3} \;&\equiv\; \tfrac{1}{2}\,\varepsilon_{A_1B_1C_1}\,\mathbf{I}^{B_1C_1A_2A_3} \\ &=\; \tfrac{1}{2\cdot 4!}\,\Big(\varepsilon_{A_1B_1C_1}\mathbf{P}_a^{B_1}\mathbf{P}_b^{C_1}\Big)\;\mathbf{P}_c^{A_2}\,\mathbf{P}_d^{A_3}\,\varepsilon^{abcd} \\ \mathbf{I}^{A_1B_1A_2A_3} \;&=\; \mathbf{G}_{C_1}{}^{A_2A_3}\,\varepsilon^{C_1A_1B_1} \end{aligned}$$

Dualizing the image 2 and 3 indices by contracting with $\varepsilon_{A_2B_2C_2}\,\varepsilon_{A_3B_3C_3}$ gives the constraint the alternative form

$$\begin{aligned}\mathbf{0} &= \varepsilon_{A_2B_2C_2}\,\varepsilon_{A_3B_3C_3}\cdot \mathbf{G}_{A_1}{}^{B_2B_3}\cdot \mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3}\\ &= \tfrac{1}{2.4!}\left(\varepsilon_{A_1B_1C_1}\mathbf{x}^{A_1}\mathbf{P}_a^{B_1}\mathbf{P}_b^{C_1}\right)\cdot\\ &\quad\cdot\left(\varepsilon_{A_2B_2C_2}\mathbf{x}^{A_2}\mathbf{P}_c^{B_2}\right)\left(\varepsilon_{A_3B_3C_3}\mathbf{x}^{A_3}\mathbf{P}_d^{B_3}\right)\,\varepsilon^{abcd}\end{aligned}$$

These equations must hold for all $3\times 3 = 9$ values of the free indices C_2 and C_3. However when C_2 is projected along the $\mathbf{x}^{C_2}$ direction or C_3 is projected along the $\mathbf{x}^{C_3}$ direction the equations are tautological because, for example, $\varepsilon_{A_2B_2C_2}\,\mathbf{x}^{A_2}\mathbf{x}^{C_2} \equiv 0$. So there are actually only $2\times 2 = 4$ linearly independent scalar constraints among the $3\times 3 = 9$ equations, corresponding to the two image 2 directions 'orthogonal' to $\mathbf{x}^{A_2}$ and the two image 3 directions 'orthogonal' to $\mathbf{x}^{A_3}$. However, each of the $3\times 3 = 9$ constraint equations and $3^3 = 27$ components of the constraint tensor are 'activated' for *some* $\mathbf{x}^{A_i}$, so none can be discarded outright.

The constraint can also be written in matrix notation as follows (*c.f.* [19]). The contraction $\mathbf{x}^{A_1}\mathbf{G}_{A_1}{}^{A_2A_3}$ has free indices A_2A_3 and can be viewed as a 3×3 matrix $[\mathbf{G}\,\mathbf{x}_1]$, and the fragments $\varepsilon_{A_2B_2C_2}\,\mathbf{x}^{A_2}$ and $\varepsilon_{A_3B_3C_3}\,\mathbf{x}^{A_3}$ can be viewed as 3×3 antisymmetric 'cross product' matrices $[\mathbf{x}_2]_\times$ and $[\mathbf{x}_3]_\times$ (where $\mathbf{x}\times\mathbf{y} = [\mathbf{x}]_\times\,\mathbf{y}$ for any 3-vector $\mathbf{y}$). The constraint is then given by the 3×3 matrix equation

$$[\mathbf{x}_2]_\times\,[\mathbf{G}\,\mathbf{x}_1]\,[\mathbf{x}_3]_\times = \mathbf{0}_{\{3\times 3\}}$$

The projections along $\mathbf{x}_2^\top$ (on the left) and $\mathbf{x}_3$ (on the right) vanish identically, so again there are only 4 linearly independent equations.

The trilinear constraint formula

$$\mathbf{x}^{A_1}\mathbf{x}^{[A_2}\,\mathbf{G}_{A_1}{}^{B_2][B_3}\,\mathbf{x}^{A_3]} \;=\; \mathbf{0}$$

also implies that for all values of the free indices $[A_2B_2]$ (or dually C_2)

$$\begin{aligned}\mathbf{x}^{A_3} &\sim \mathbf{x}^{A_1}\mathbf{x}^{[A_2}\,\mathbf{G}_{A_1}{}^{B_2]A_3}\\ &\sim \varepsilon_{C_2A_2B_2}\,\mathbf{x}^{A_1}\mathbf{x}^{A_2}\,\mathbf{G}_{A_1}{}^{B_2A_3}\end{aligned}$$

More precisely, for *matching* $\mathbf{x}^{A_1}$ and $\mathbf{x}^{A_2}$ the quantity $\mathbf{x}^{A_1}\mathbf{x}^{[A_2}\,\mathbf{G}_{A_1}{}^{B_2]A_3}$ can always be factorized as $\mathbf{T}^{[A_2B_2]}\mathbf{x}^{A_3}$ for some $\mathbf{x}^{A_i}$-dependent tensor $\mathbf{T}^{[A_2B_2]}$ (and similarly with $\mathbf{T}_{C_2}$ for the dual form). By fixing suitable values of $[A_2B_2]$ or C_2, these equations can be used to **transfer** points from images 1 and 2 to image 3, *i.e.* to directly predict the projection in image 3 of a 3D point whose projections in images 1 and 2 are known, without any intermediate 3D reconstruction step[2].

The trilinear constraints can be interpreted geometrically as follows. As above the quantity $\varepsilon_{ABC}\,\mathbf{x}^A\,\mathbf{P}_b^B\mathbf{P}_c^C$ represents the optical ray through $\mathbf{x}^A$ in covariant 3D coordinates. For any $\mathbf{y}^A \in \mathcal{P}^A$ the quantity $\varepsilon_{ABC}\,\mathbf{x}^A\mathbf{y}^B\mathbf{P}_c^C$ defines the 3D plane through the optical centre that projects to the image line through $\mathbf{x}^A$ and $\mathbf{y}^A$. All such planes contain the optical ray of $\mathbf{x}^A$, and as $\mathbf{y}^A$ varies the entire pencil of planes through this line is traced out. The constraint then says that for any plane through the optical ray of $\mathbf{x}^{A_2}$ and any other plane through the optical ray of $\mathbf{x}^{A_3}$, the 3D line of intersection of these planes meets the optical ray of $\mathbf{x}^{A_1}$.

The line of intersection always meets the optical rays of both $\mathbf{x}^{A_2}$ and $\mathbf{x}^{A_3}$ because it lies in planes containing those rays. If the rays are skew *every* line through the two rays is generated as the planes vary. The optical ray through $\mathbf{x}^{A_1}$ can not meet every such line, so the constraint implies that the optical rays of $\mathbf{x}^{A_2}$ and $\mathbf{x}^{A_3}$ can not be skew. In other words the image 1 trilinear constraint implies the epipolar constraint between images 2 and 3.

Given that the rays of $\mathbf{x}^{A_2}$ and $\mathbf{x}^{A_3}$ meet (say, at

[2] If $\mathbf{x}^{A_1}$ and $\mathbf{x}^{A_2}$ are *not* matching points, the transfer equations trace out an entire line of mutually inconsistent 'solutions' as $[A_2B_2]$ or C_2 vary. For fixed $\mathbf{x}^{A_1}$ and *any* line $\mathbf{l}_{A_2}$ there is a 'solution' $\mathbf{x}^{A_3}(\mathbf{x}^{A_1},\mathbf{l}_{A_2}) \sim \mathbf{l}_{A_2}\,\mathbf{G}_{A_1}{}^{A_2A_3}\,\mathbf{x}^{A_1}$. This is just the intersection of the image 3 epipolar line of $\mathbf{x}^{A_1}$ with the image 3 epipolar line of the intersection of $\mathbf{l}_{A_2}$ and the image 2 epipolar line of $\mathbf{x}^{A_1}$, *i.e.* the transfer of the only point on $\mathbf{l}_{A_2}$ that *could* be a correct match. In general, as $\mathbf{l}_{A_2}$ traces out the the pencil of lines through $\mathbf{x}^{A_2}$ the corresponding 'solutions' $\mathbf{x}^{A_3}$ trace out the entire epipolar line of $\mathbf{x}^{A_1}$ in image 3. The line of 'solutions' collapses to a point only when $\mathbf{x}^{A_2}$ lies on the epipolar line of $\mathbf{x}^{A_1}$. For reliable transfer the line $\mathbf{l}_{A_2}$ should meet the epipolar line of $\mathbf{x}^{A_1}$ reasonably transversally and if possible should pass close to the image 3 epipole. This can be arranged by projecting the free index C_2 along (an approximation to) the image 3 epipole $\mathbf{e}_3^{A_2}$.

Similarly, $\mathbf{x}^{A_3}$ could be predicted as the intersection of the epipolar lines of $\mathbf{x}^{A_1}$ and $\mathbf{x}^{A_2}$ in $\mathcal{P}^{A_3}$. This intersection always exists, but it is not structurally meaningful if $\mathbf{x}^{A_1}$ and $\mathbf{x}^{A_2}$ do not correspond. The moral is that it is dangerous to use only *some* of the available equations for transfer.

some point $\mathbf{x}^a$), as the two planes through these rays vary their intersection traces out every line through $\mathbf{x}^a$ not in the plane of the rays. The only way that the optical ray of $\mathbf{x}^{A_1}$ can arrange to meet each of these lines is for it to pass through $\mathbf{x}^a$ as well. In other words the trilinear constraint for each image implies that all three optical rays pass through the same point. Thus, the epipolar constraints between images 1 and 2 and images 1 and 3 also follow from the image 1 trilinear constraint.

The constraint tensor $\mathbf{G}_{A_1}{}^{A_2A_3} \equiv \boldsymbol{\varepsilon}_{A_1B_1C_1}\,\mathbf{I}^{B_1C_1A_2A_3}$ treats image 1 specially. The analogous image 2 and image 3 tensors $\mathbf{G}_{A_2}{}^{A_3A_1} \equiv \boldsymbol{\varepsilon}_{A_2B_2C_2}\,\mathbf{I}^{B_2C_2A_3A_1}$ and $\mathbf{G}_{A_3}{}^{A_1A_2} \equiv \boldsymbol{\varepsilon}_{A_3B_3C_3}\,\mathbf{I}^{B_3C_3A_1A_2}$ are linearly independent of $\mathbf{G}_{A_1}{}^{A_2A_3}$ and give further linearly independent trilinear constraints on $\mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3}$. Together, the 3 homogeneous constraint tensors contain $3 \times 27 = 81$ linearly independent components (including 3 arbitrary scale factors) and naïvely give $3 \times 9 = 27$ trilinear scalar constraint equations, of which $3 \times 4 = 12$ are linearly independent for any given triple $\mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3}$.

However, although there are no *linear* relations between the $3 \times 27 = 81$ trilinear and $3 \times 9 = 27$ bilinear matching tensor components for the three images, the matching tensors are certainly not *algebraically* independent of each other: there are many *quadratic* relations between them inherited from the quadratic simplicity constraints on the joint image Grassmannian tensor. In fact, we saw in section 5 that the simplicity constraints reduce the number of algebraically independent degrees of freedom of $\mathbf{I}^{\alpha_0\cdots\alpha_3}$ (and therefore the complete set of bilinear and trilinear matching tensor components) to only $11m - 15 = 18$ for $m = 3$ images. Similarly, there are only $2m - 3 = 3$ *algebraically* independent scalar constraint equations among the *linearly* independent $3 \times 4 = 12$ trilinear and $3 \times 1 = 3$ bilinear constraints on each matching triple of points. One of the main advantages of the Grassmann formalism is the extent to which it clarifies the rich algebraic structure of this matching constraint system. The components of the constraint tensors are essentially just Grassmann coordinates of the joint image, and Grassmann coordinates are *always* linearly independent and quadratically redundant.

Since all three of the epipolar constraints follow from a single trilinear tensor it may seem that the trilinear constraint is more powerful than the epipolar ones, but this is not really so. Given a triple of image points $\{\mathbf{x}^{A_i} | \, i = 1, \ldots, 3\}$, the three pairwise epipolar constraints say that the three optical rays must meet pairwise. If they do not meet at a single point, this implies that each ray must lie in the plane of the other two. Since the rays pass through their respective optical centres, the plane also contains the three optical centres, and is therefore the **trifocal plane**. But this is impossible in general: most image points simply do not lie on the trifocal lines (the projections of the trifocal planes). So for general matching image points the three epipolar constraints together imply that the three optical rays meet at a unique 3D point. This is enough to imply the trilinear constraints. Since we know that only $2m-3 = 3$ of the constraints are algebraically independent, this is as expected.

Similarly, the information contained in just one of the trilinear constraint tensors is generically $4 > 2m - 3 = 3$ linearly independent constraints, which is enough to imply the other two trilinear tensors as well as the three bilinear ones. This explains why most of the early work on trilinear constraints successfully ignores two of the three available tensors [19, 7]. However in the context of purely *linear* reconstruction all three of the tensors would be necessary.

7.3 Quadrilinear Constraints

Finally, the quadrilinear, four image Grassmannian constraint $\mathbf{I}^{[B_1B_2B_3B_4}\,\mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3}\mathbf{x}^{A_4]} = \mathbf{0}$ corresponds to an 8×8 basic reconstruction minor that selects two rows from each of four images. As usual the antisymmetrization applies to each image separately, but in this case the simplest form of the constraint tensor is just a direct selection of $3^4 = 81$ components of the Grassmannian itself

$$\begin{aligned}\mathbf{H}^{A_1A_2A_3A_4} &\equiv \mathbf{I}^{A_1A_2A_3A_4} \\ &= \tfrac{1}{4!}\,\mathbf{P}_a^{A_1}\mathbf{P}_b^{A_2}\mathbf{P}_c^{A_3}\mathbf{P}_d^{A_4}\,\boldsymbol{\varepsilon}^{abcd}\end{aligned}$$

Dualizing the antisymmetric index pairs $[A_iB_i]$ by contracting with $\boldsymbol{\varepsilon}_{A_iB_iC_i}$ for $i = 1, \ldots, 4$ gives the

quadrilinear constraint

$$\begin{aligned}\mathbf{0} \;=\; & \varepsilon_{A_1B_1C_1}\,\varepsilon_{A_2B_2C_2}\,\varepsilon_{A_3B_3C_3}\,\varepsilon_{A_4B_4C_4}\cdot \\ & \qquad \cdot \mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3}\mathbf{x}^{A_4}\;\mathbf{H}^{B_1B_2B_3B_4} \\ =\; & \tfrac{1}{4!}\left(\varepsilon_{A_1B_1C_1}\mathbf{x}^{A_1}\mathbf{P}_a^{B_1}\right)\left(\varepsilon_{A_2B_2C_2}\mathbf{x}^{A_2}\mathbf{P}_b^{B_2}\right)\cdot \\ & \cdot\left(\varepsilon_{A_3B_3C_3}\mathbf{x}^{A_3}\mathbf{P}_c^{B_3}\right)\left(\varepsilon_{A_4B_4C_4}\mathbf{x}^{A_4}\mathbf{P}_d^{B_4}\right)\varepsilon^{abcd}\end{aligned}$$

This must hold for each of the $3^4 = 81$ values of $C_1C_2C_3C_4$. But again the constraints with C_i along the direction $\mathbf{x}^{C_i}$ for any $i = 1, \ldots, 4$ vanish identically, so for any given quadruple of points there are only $2^4 = 16$ linearly independent constraints among the $3^4 = 81$ equations.

Together, these constraints say that for every possible choice of four planes, one through the optical ray defined by $\mathbf{x}^{A_i}$ for each $i = 1, \ldots, 4$, the planes meet in a point. By fixing three of the planes and varying the fourth we immediately find that each of the optical rays passes through the point, and hence that they all meet. This brings us back to the two and three image sub-cases.

Again, there is nothing algebraically new here. The $3^4 = 81$ homogeneous components of the quadrilinear constraint tensor are *linearly* independent of each other and of the $4 \times 3 \times 27 = 324$ homogeneous trilinear and $6 \times 9 = 54$ homogeneous bilinear tensor components; and the $2^4 = 16$ linearly independent quadrilinear scalar constraints are *linearly* independent of each other and of the linearly independent $4 \times 3 \times 4 = 48$ trilinear and $6 \times 1 = 6$ bilinear constraints. However there are only $11m - 15 = 29$ *algebraically* independent tensor components in total, which give $2m - 3 = 5$ *algebraically* independent constraints on each 4-tuple of points. The quadrilinear constraint is algebraically equivalent to various different combinations of two and three image constraints. For example five scalar epipolar constraints will do: take the three pairwise constraints for the first three images, then add two of the three involving the fourth image to force the optical rays from the fourth image to pass through the intersection of the corresponding optical rays from the other three images.

7.4 Matching Constraints for Lines

It is well known that there is no matching constraint for lines in two images. Any two non-epipolar image lines $\mathbf{l}_{A_1}$ and $\mathbf{l}_{A_2}$ are the projection of some unique 3D line: simply pull back the image lines to two 3D planes $\mathbf{l}_{A_1}\mathbf{P}_a^{A_1}$ and $\mathbf{l}_{A_2}\mathbf{P}_a^{A_2}$ through the centres of projection and intersect the planes to find the 3D line $\mathbf{l}_{ab} = \mathbf{l}_{A_1}\mathbf{l}_{A_2}\,\mathbf{P}_{[a}^{A_1}\mathbf{P}_{b]}^{A_2}$.

However for three or more images of a line there are trilinear matching constraints as follows [7]. An image line is the projection of a 3D line if and only if each point on the 3D line projects to a point on the image line. Writing this out, we immediately see that the lines $\{\mathbf{l}_{A_i} \,|\, i = 1, \ldots, m\}$ correspond to a 3D line if and only if the $m \times 4$ reconstruction equations

$$\begin{pmatrix} \mathbf{l}_{A_1}\mathbf{P}_a^{A_1} \\ \vdots \\ \mathbf{l}_{A_m}\mathbf{P}_a^{A_m} \end{pmatrix}\mathbf{x}^a \;=\; \mathbf{0}$$

have a line (*i.e.* a 2D linear space) of solutions $\lambda\mathbf{x}^a + \mu\mathbf{y}^a$ for some solutions $\mathbf{x}^a \not\sim \mathbf{y}^a$.

There is a 2D solution space if and only if the coefficient matrix has rank $4-2 = 2$, which means that every 3×3 minor has to vanish. Obviously each minor is a trilinear function in three $\mathbf{l}_{A_i}$'s and misses out one of the columns of $\mathbf{P}_a^\alpha$. Labelling the missing column as a and expanding produces constraint equations like

$$\mathbf{l}_{A_1}\,\mathbf{l}_{A_2}\,\mathbf{l}_{A_3}\;\left(\mathbf{P}_b^{A_1}\mathbf{P}_c^{A_2}\mathbf{P}_d^{A_3}\;\varepsilon^{abcd}\right) \;=\; \mathbf{0}$$

These simply require that the three pulled back planes $\mathbf{l}_{A_1}\mathbf{P}_a^{A_1}$, $\mathbf{l}_{A_2}\mathbf{P}_a^{A_2}$ and $\mathbf{l}_{A_3}\mathbf{P}_a^{A_3}$ meet in some common 3D line, rather than just a single point. Note the geometry here: each line $\mathbf{l}_{A_i}$ pulls back to a hyperplane in $\mathcal{P}^\alpha$ under the trivial projection. This restricts to a hyperplane in $\mathcal{PI}^\alpha$, which can be expressed as $\mathbf{l}_{A_i}\mathbf{P}_a^{A_i}$ in the basis $\mathbf{P}_a^\alpha$ for $\mathcal{PI}^\alpha$. There are $2m-4$ algebraically independent constraints for m images: two for each image except the first two. There are *no* irreducible higher order constraints for lines in more than 3 images, *e.g.* there is no analogue of the quadrilinear constraint for lines.

By contracting with a final $\mathbf{P}_a^\alpha$, the constraints can also be written in terms of the Grassmannian tensor as

$$\mathbf{l}_{A_1}\,\mathbf{l}_{A_2}\,\mathbf{l}_{A_3}\,\mathbf{I}^{\alpha A_1A_2A_3} \;=\; \mathbf{0}$$

for all α. Choosing α from images 1, 2 or 3 and contracting with an image 1, 2 or 3 epsilon to produce

a trivalent tensor $\mathbf{G}_{A_i}{}^{A_j A_k}$, or choosing α from a fourth image and substituting the quadrivalent tensor $\mathbf{H}^{A_i A_j A_k A_l}$ reduces the line constraints to the form

$$\begin{aligned} \mathbf{l}_{A_2}\, \mathbf{l}_{A_3}\, \mathbf{l}_{[A_1}\, \mathbf{G}_{B_1]}{}^{A_2 A_3} &= \mathbf{0} \\ \mathbf{l}_{A_1}\, \mathbf{l}_{A_2}\, \mathbf{l}_{A_3}\, \mathbf{H}^{A_1 A_2 A_3 A_4} &= \mathbf{0} \end{aligned}$$

These formulae illustrate and extend Hartley's observation that the coefficient tensors of the three-image line constraints are equivalent to those of the trilinear point constraints [7]. Note that although all of these line constraints are *tri*linear, some of them do involve *quadri*valent point constraint tensors.

Since α can take any of $3m$ values A_i, for each triple of lines and $m \geq 3$ images there are very naïvely $3m$ trilinear constraints of the above two forms. However all of these constraints are derived by linearly contracting 4 underlying world constraints with $\mathbf{P}_a^\alpha$'s, so at most 4 of them can be linearly independent. For m matching images of lines this leaves $4\binom{m}{3}$ *linearly* independent constraints of which only $2m-4$ are algebraically independent.

The skew symmetrization in the trivalent tensor based constraint immediately implies the **line transfer** equation

$$\mathbf{l}_{A_1} \sim \mathbf{l}_{A_2}\, \mathbf{l}_{A_3}\, \mathbf{G}_{A_1}{}^{A_2 A_3}$$

This can be used to predict the projection of a 3D line in image 1 given its projections in images 2 and 3, without intermediate 3D reconstruction. Note that line transfer from images 1 and 2 to image 3 is most simply expressed in terms of the image 3 trilinear tensor $\mathbf{G}_{A_3}{}^{A_1 A_2}$, whereas the image 1 or image 2 tensors $\mathbf{G}_{A_1}{}^{A_2 A_3}$ or $\mathbf{G}_{A_2}{}^{A_1 A_3}$ are the preferred form for point transfer.

It is also possible to match *(i)* points against lines that contain them and *(ii)* distinct image lines that are known to intersect in 3D. Such constraints might be useful if a polyhedron vertex is obscured or poorly localized. They are most easily derived by noting that both the line reconstruction equations and the reduced point reconstruction equations are homogeneous in $\mathbf{x}^a$, the coordinates of the intersection point. So line and point rows from several images can be stacked into a single 4 column matrix. As usual there is a solution exactly when all 4×4 minors vanish. This yields two particularly simple irreducible constraints — and correspondingly simple interpretations of the matching tensors' content — for an image point against two lines containing it and four non-corresponding image lines that intersect in 3D:

$$\begin{aligned} \mathbf{x}^{A_1}\, \mathbf{G}_{A_1}{}^{A_2 A_3}\, \mathbf{l}_{A_2} \mathbf{l}'_{A_3} &= 0 \\ \mathbf{H}^{A_1 A_2 A_3 A_4}\, \mathbf{l}_{A_1} \mathbf{l}'_{A_2} \mathbf{l}''_{A_3} \mathbf{l}'''_{A_4} &= 0 \end{aligned}$$

7.5 Matching Constraints for k-Subspaces

More generally, the projections of a k dimensional subspace in d dimensions are (generically) k dimensional image subspaces that can be written as antisymmetric $D_i - k$ index Grassmann tensors $\mathbf{x}_{A_i \cdots B_i \cdots C_i}$. The matching constraints can be built by selecting any $d+1-k$ of these covariant indices from any set $i, j, \ldots, k$ of image tensors and contracting with the Grassmannian to leave k free indices:

$$\mathbf{0} = \mathbf{x}_{A_i \cdots B_i C_i \cdots E_i} \cdots \mathbf{x}_{A_k \cdots B_k C_k \cdots E_k} \cdot \cdot \mathbf{I}^{\alpha_1 \cdots \alpha_k A_i \cdots B_i \cdots A_k \cdots B_k}$$

Dualizing each covariant Grassmann tensor gives an equivalent contravariant form of the constraint, for image subspaces $\mathbf{x}^{A_j \cdots E_j}$ defined by the span of a set of image points

$$\mathbf{0} = \mathbf{I}^{\alpha_1 \cdots \alpha_k [A_i \cdots B_i \cdots A_k \cdots B_k}\, \mathbf{x}^{C_i \cdots E_i} \cdots \mathbf{x}^{C_k \cdots E_k]}$$

As usual it is enough to antisymmetrize over the indices from each image separately. Each set $A_j \cdots B_j C_j \cdots E_j$ is any choice of up to $D_j + 1$ indices from image j, $j = i, \ldots, k$.

7.6 2D Matching Constraints & Homographies

Our formalism also works for 2D projective images of a 2D space. This case is practically important because it applies to 2D images of a planar surface in 3D and there are many useful plane-based vision algorithms. The joint image of a 2D source space is two dimensional, so the corresponding Grassmannian tensor has only three indices and there are only two distinct types of matching constraint: bilinear and trilinear. Let indices a and A_i represent 3D space and the i^{th} image as usual, and indices

$A = 0, 1, 2$ represent homogeneous coordinates on the source plane. If the plane is given by $\mathbf{p}_a \mathbf{x}^a = 0$, the three index epsilon tensor on it is proportional to $\mathbf{p}_a \boldsymbol{\varepsilon}^{abcd}$ when expressed in world coordinates, so the Grassmann tensor becomes

$$\begin{aligned} \mathbf{I}^{\alpha\beta\gamma} &\equiv \tfrac{1}{3!}\, \mathbf{P}_A^\alpha\, \mathbf{P}_B^\beta\, \mathbf{P}_C^\gamma\, \boldsymbol{\varepsilon}^{ABC} \\ &\sim \tfrac{1}{4!}\, \mathbf{p}_a\, \mathbf{P}_b^\alpha\, \mathbf{P}_c^\beta\, \mathbf{P}_d^\gamma\, \boldsymbol{\varepsilon}^{abcd} \end{aligned}$$

This yields the following bilinear and trilinear matching constraints with free indices respectively C_2 and $C_1 C_2 C_3$

$$\begin{aligned} \mathbf{0} &= \mathbf{p}_a \left(\boldsymbol{\varepsilon}_{A_1 B_1 C_1}\, \mathbf{x}^{A_1}\, \mathbf{P}_b^{B_1} \mathbf{P}_c^{C_1}\right) \cdot \\ &\qquad \cdot \left(\boldsymbol{\varepsilon}_{A_2 B_2 C_2}\, \mathbf{x}^{A_2}\, \mathbf{P}_d^{B_2}\right)\, \boldsymbol{\varepsilon}^{abcd} \\ \mathbf{0} &= \mathbf{p}_a \left(\boldsymbol{\varepsilon}_{A_1 B_1 C_1}\, \mathbf{x}^{A_1} \mathbf{P}_b^{B_1}\right)\left(\boldsymbol{\varepsilon}_{A_2 B_2 C_2}\, \mathbf{x}^{A_2} \mathbf{P}_c^{B_2}\right) \cdot \\ &\qquad \cdot \left(\boldsymbol{\varepsilon}_{A_3 B_3 C_3}\, \mathbf{x}^{A_3}\, \mathbf{P}_d^{B_3}\right)\, \boldsymbol{\varepsilon}^{abcd} \end{aligned}$$

The bilinear equation says that $\mathbf{x}^{A_2}$ is the image of the intersection of optical ray of $\mathbf{x}^{A_1}$ with the plane $\mathbf{p}_a$: $\mathbf{x}^{A_2} \sim \left(\mathbf{p}_a \cdot \boldsymbol{\varepsilon}_{A_1 B_1 C_1}\, \mathbf{P}_b^{B_1} \mathbf{P}_c^{C_1} \cdot \mathbf{P}_d^{A_2} \cdot \boldsymbol{\varepsilon}^{abcd}\right) \mathbf{x}^{A_1}$. In fact it is well known that any two images of a plane are projectively equivalent under a transformation (homography) $\mathbf{x}^{A_2} \sim \mathbf{H}_{A_1}^{A_2}\, \mathbf{x}^{A_1}$. In our notation the homography is just

$$\mathbf{H}_{A_1}^{A_2} \equiv \mathbf{p}_a \cdot \boldsymbol{\varepsilon}_{A_1 B_1 C_1}\, \mathbf{P}_b^{B_1} \mathbf{P}_c^{C_1} \cdot \mathbf{P}_d^{A_2} \cdot \boldsymbol{\varepsilon}^{abcd}$$

The trilinear constraint says that any three image lines through the three image points $\mathbf{x}^{A_1}$, $\mathbf{x}^{A_2}$ and $\mathbf{x}^{A_3}$ always meet in a point when pulled back to the plane $\mathbf{p}_a$. This implies that the optical rays of the three points intersect at a common point on the plane, and hence gives the obvious cyclic consistency condition $\mathbf{H}_{A_2}^{A_1}\, \mathbf{H}_{A_3}^{A_2} \sim \mathbf{H}_{A_3}^{A_1}$ (or equivalently $\mathbf{H}_{A_2}^{A_1}\, \mathbf{H}_{A_3}^{A_2}\, \mathbf{H}_{B_1}^{A_3} \sim \boldsymbol{\delta}_{B_1}^{A_1}$) between the three homographies.

7.7 Matching Constraints for 1D Cameras

If some of the images are taken with one dimensional 'linear' cameras, a similar analysis applies but the corresponding entries in the reconstruction equations have only two rows instead of three. Constraints that would require three rows from a 1D image no longer exist, and the remaining constraints lose their free indices. In particular, when all of the cameras are 1D there are no bilinear or trilinear tensors and the only irreducible matching constraint is the quadrilinear scalar:

$$\begin{aligned} 0 &= \mathbf{H}_{A_1 A_2 A_3 A_4}\, \mathbf{x}^{A_1} \mathbf{x}^{A_2} \mathbf{x}^{A_3} \mathbf{x}^{A_4} \\ &= \left(\boldsymbol{\varepsilon}_{A_1 B_1}\, \mathbf{x}^{A_1}\, \mathbf{P}_a^{B_1}\right)\left(\boldsymbol{\varepsilon}_{A_2 B_2}\, \mathbf{x}^{A_2}\, \mathbf{P}_b^{B_2}\right) \cdot \\ &\qquad \cdot \left(\boldsymbol{\varepsilon}_{A_3 B_3}\, \mathbf{x}^{A_3}\, \mathbf{P}_c^{B_3}\right)\left(\boldsymbol{\varepsilon}_{A_4 B_4}\, \mathbf{x}^{A_4}\, \mathbf{P}_d^{B_4}\right)\, \boldsymbol{\varepsilon}^{abcd} \end{aligned}$$

This says that the four planes pulled back from the four image points must meet in a 3D point. If one of the cameras is 2D and the other two are 1D a scalar trilinear constraint also exists.

7.8 3D to 2D Matching

It is also useful to be able to match known 3D structure to 2D image structure, for example when building a reconstruction incrementally from a sequence of images. This case is rather trivial as the 'constraint tensor' is just the projection matrix, but for comparison it is perhaps worth writing down the equations. For an image point $\mathbf{x}^A$ projected from a world point $\mathbf{x}^a$ we have $\mathbf{x}^A \sim \mathbf{P}_a^A\, \mathbf{x}^a$ and hence the equivalent constraints

$$\mathbf{x}^{[A} \mathbf{P}_a^{B]} \mathbf{x}^a = \mathbf{0} \quad \Longleftrightarrow \quad \boldsymbol{\varepsilon}_{ABC} \mathbf{x}^A \mathbf{P}_a^B \mathbf{x}^a = \mathbf{0}$$

There are three bilinear equations, only two of which are independent for any given image point. Similarly, a world line $\mathbf{l}_{[ab]}$ (or dually, $\mathbf{l}^{[ab]}$) and a corresponding image line $\mathbf{l}_A$ satisfy the equivalent bilinear constraints

$$\mathbf{l}_A\, \mathbf{P}_{[a}^A\, \mathbf{l}_{bc]} = \mathbf{0} \quad \Longleftrightarrow \quad \mathbf{l}_A\, \mathbf{P}_a^A\, \mathbf{l}_{bc}\, \boldsymbol{\varepsilon}^{abcd} = \mathbf{0}$$

or dually

$$\mathbf{l}_A\, \mathbf{P}_a^A\, \mathbf{l}^{ab} = \mathbf{0}$$

Each form contains four bilinear equations, only two of which are linearly independent for any given image line. For example, if the line is specified by giving two points on it $\mathbf{l}^{ab} \sim \mathbf{x}^{[a} \mathbf{y}^{b]}$, we have the two scalar equations $\mathbf{l}_A\, \mathbf{P}_a^A\, \mathbf{x}^a = 0$ and $\mathbf{l}_A\, \mathbf{P}_a^A\, \mathbf{y}^a = 0$.

7.9 Epipoles

There is still one aspect of $\mathbf{I}^{\alpha_0 \cdots \alpha_d}$ that we have not yet seen: the Grassmannian tensor also directly contains the *epipoles*. In fact, the epipoles are most

naturally viewed as the first order term in the sequence of matching tensors, although they do not themselves induce any matching constraints.

Assuming that it has rank d, the $d\times(d+1)$ projection matrix of a $d-1$ dimensional image of d dimensional space defines a unique **centre of projection** ${\mathbf{e}_i}^a$ by $\mathbf{P}_a^{A_i}\,{\mathbf{e}_i}^a = \mathbf{0}$. The solution of this equation is given (*c.f.* section 8) by the vector of $d \times d$ minors of $\mathbf{P}_a^{A_i}$, *i.e.*

$$ {\mathbf{e}_i}^a \;\sim\; \varepsilon_{A_i\cdots C_i}\,\mathbf{P}^{A_i}_{a_1}\cdots\mathbf{P}^{C_i}_{a_d}\,\varepsilon^{aa_1\cdots a_d} $$

The projection of a centre of projection in another image is an **epipole**

$$ {\mathbf{e}_i}^{A_j} \;\sim\; \varepsilon_{A_i\cdots C_i}\,\mathbf{P}^{A_j}_{a_0}\mathbf{P}^{A_i}_{a_1}\cdots\mathbf{P}^{C_i}_{a_d}\,\varepsilon^{a_0a_1\cdots a_d} $$

Recognizing the factor of $\mathbf{I}^{A_jA_iB_i\cdots C_i}$, we can fix the scale factors for the epipoles so that

$$ \begin{aligned} {\mathbf{e}_i}^{A_j} &\equiv \tfrac{1}{d!}\,\varepsilon_{A_iB_i\cdots C_i}\;\mathbf{I}^{A_jA_iB_i\cdots C_i} \\ \mathbf{I}^{A_jA_iB_i\cdots C_i} &= {\mathbf{e}_i}^{A_j}\,\varepsilon^{A_iB_i\cdots C_i} \end{aligned} $$

The d-dimensional joint image subspace $\mathcal{PI}^\alpha$ of $\mathcal{P}^\alpha$ passes through the d-*co*dimensional projective subspace $\mathbf{x}^{A_i} = \mathbf{0}$ at the **joint image epipole**

$$ {\mathbf{e}_i}^\alpha \equiv \left({\mathbf{e}_i}^{A_1},\ldots,{\mathbf{e}_i}^{A_{i-1}},\mathbf{0},{\mathbf{e}_i}^{A_{i+1}},\ldots,{\mathbf{e}_i}^{A_m}\right)^\top $$

As usual, an arbitrary choice of the relative scale factors is required.

Counting up the components of the $\binom{m}{4}$ quadrilinear, $3\binom{m}{3}$ trilinear, $\binom{m}{2}$ bilinear and $m(m-1)$ monolinear (epipole) tensors for m images of a 3D world, we find a total of

$$ \begin{aligned} \binom{3m}{4} \;=\; 81\cdot\binom{m}{4} \;+\; 27\cdot 3\binom{m}{3} \\ +\; 9\cdot\binom{m}{2} \;+\; 3\cdot m(m-1) \end{aligned} $$

linearly independent components. These are linearly equivalent to the complete set of $\binom{3m}{4}$ linearly independent components of $\mathbf{I}^{\alpha_0\cdots\alpha_d}$, so the joint image Grassmannian tensor can be reconstructed *linearly* given the entire set of (appropriately scaled) matching tensors.

8 Minimal Reconstructions and Uniqueness

The matching constraints found above are closely associated with a set of **minimal reconstruction techniques** that produce candidate solutions $\mathbf{x}^a$ from minimal sets of d image measurements (three in the 3D case). Geometrically, measuring an image coordinate restricts the corresponding world point to a hyperplane in $\mathcal{P}^a$. The intersection of any d independent hyperplanes gives a unique solution candidate $\mathbf{x}^a$, so there is a minimal reconstruction technique based on any set of d independent image measurements. Matching is equivalent to the requirement that this candidate lies in the hyperplane of each of the remaining measurements. If d measurements are not independent the corresponding minimal reconstruction technique will fail to give a unique candidate, but so long as the images contain *some* set of d independent measurements at least one of the minimal reconstructions will succeed and the overall reconstruction solution will be unique (or fail to exist altogether if the matching constraints are violated).

Algebraically, we can restate this as follows. Consider a general $k \times (k+1)$ system of homogeneous linear equations with rank k. Up to scale the system has a unique solution given by the $(k+1)$-component vector of $k\times k$ minors of the system matrix[3]. Adding an extra row to the system destroys the solution unless the new row is orthogonal to the existing minor vector: this is exactly the requirement that the determinant of the $(k+1)\times(k+1)$ matrix vanish so that the system still has rank k. With an overspecified rank k system: any choice of k rows gives a minor vector; at least one minor vector is nonzero by rank-k-ness; every minor vector is orthogonal to every row of the system matrix by non-rank-$(k+1)$-ness; and all of the minor vectors are equal up to scale because there is only one direction orthogonal to any given k independent rows. In other words the existence of a solution can be ex-

[3] *Proof*: By the rank k condition the vector of minors does not vanish. Adding any $(k+1)^{st}$ row vector $\mathbf{v}$ to the system gives a $(k+1)\times(k+1)$ matrix. By the usual cofactor expansion, the determinant of this matrix is exactly the dot product of $\mathbf{v}$ with the vector of minors. The determinant vanishes when $\mathbf{v}$ is chosen to be any of the existing rows of the matrix, so the minor vector is orthogonal to each row.

pressed as a set of simple orthogonality relations on a candidate solution (minor vector) produced from any set of k independent rows.

We can apply this to the $(d+m)\times(d+m)$ minors of the $(D+m)\times(d+m+1)$ basic reconstruction system, or equivalently to the $d\times d$ minors of the $D\times(d+1)$ reduced reconstruction system. The situation is very similar to that for matching constraints and a similar analysis applies. The result is that if $i,j,\ldots,k$ is a set of $2\leq m'\leq d$ distinct images and $\gamma,\ldots,\delta$ is any selection of $d-m'$ indices from images $i,j,\ldots,k$ (at most D_i-1 from any one image), there is a pair of equivalent minimal reconstruction techniques for $\mathbf{x}^a\in\mathcal{P}^a$ and $\mathbf{x}^\alpha\in\mathcal{P}^\alpha$:

$$\begin{aligned}\mathbf{x}^a &\sim \mathbf{P}^{a[B_iB_j\cdots B_k\gamma\cdots\delta}\,\mathbf{x}^{A_i}\mathbf{x}^{A_j}\cdots\mathbf{x}^{A_k]}\\ \mathbf{x}^\alpha &\sim \mathbf{I}^{\alpha[B_iB_j\cdots B_k\gamma\cdots\delta}\,\mathbf{x}^{A_i}\mathbf{x}^{A_j}\cdots\mathbf{x}^{A_k]}\end{aligned}$$

where

$$\mathbf{P}^{a[\alpha_1\cdots\alpha_d]} \equiv \tfrac{1}{d!}\,\mathbf{P}^{\alpha_1}_{a_1}\cdots\mathbf{P}^{\alpha_d}_{a_d}\,\boldsymbol{\varepsilon}^{aa_1\cdots a_d}$$

In these equations, the right hand side has tensorial indices $[B_i\cdots B_k\gamma\cdots\delta A_i\cdots A_k]$ in addition to a or α, but so long as the matching constraints hold any value of these indices gives a vector parallel to $\mathbf{x}^a$ or $\mathbf{x}^\alpha$ (*i.e.* for *matching* image points the tensor $\mathbf{P}^{a[B_i\cdots B_k\gamma\cdots\delta}\,\mathbf{x}^{A_i}\cdots\mathbf{x}^{A_k]}$ can be factorized as $\mathbf{x}^a\,\mathbf{T}^{[B_i\cdots B_k\gamma\cdots\delta A_i\cdots A_k]}$ for some tensors $\mathbf{x}^a$ and $\mathbf{T}$). Again it is enough to antisymmetrize over the indices of each image separately. For 2D images of 3D space the possible minimal reconstruction techniques are $\mathbf{P}^{a[B_1C_1B_2}\,\mathbf{x}^{A_1}\mathbf{x}^{A_2]}$ and $\mathbf{P}^{a[B_1B_2B_3}\,\mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3]}$:

$$\begin{aligned}\mathbf{x}^a &\sim \left(\boldsymbol{\varepsilon}_{A_1B_1C_1}\,\mathbf{x}^{A_1}\,\mathbf{P}^{B_1}_b\mathbf{P}^{C_1}_c\right)\cdot\\ &\qquad\cdot\left(\boldsymbol{\varepsilon}_{A_2B_2C_2}\,\mathbf{x}^{A_2}\,\mathbf{P}^{C_2}_d\right)\,\boldsymbol{\varepsilon}^{abcd}\\ \mathbf{x}^a &\sim \left(\boldsymbol{\varepsilon}_{A_1B_1C_1}\,\mathbf{x}^{A_1}\,\mathbf{P}^{C_1}_b\right)\left(\boldsymbol{\varepsilon}_{A_2B_2C_2}\,\mathbf{x}^{A_2}\,\mathbf{P}^{C_2}_c\right)\cdot\\ &\qquad\cdot\left(\boldsymbol{\varepsilon}_{A_3B_3C_3}\,\mathbf{x}^{A_3}\,\mathbf{P}^{C_2}_d\right)\,\boldsymbol{\varepsilon}^{abcd}\end{aligned}$$

These correspond respectively to finding the intersection of the optical ray from one image and the constraint plane from one coordinate of the second one, and to finding the intersection of three constraint planes from one coordinate in each of three images.

To recover the additional matching constraints that apply to the minimal reconstruction solution with indices $[B_i\cdots B_k\gamma\cdots\delta A_i\cdots A_k]$, project the solution to some image l to get

$$\mathbf{P}^{C_l}_a\,\mathbf{x}^a \;=\; \mathbf{I}^{C_l[B_i\cdots B_k\gamma\cdots\delta}\,\mathbf{x}^{A_i}\cdots\mathbf{x}^{A_k]}$$

If the constraint is to hold, this must be proportional to $\mathbf{x}^{C_l}$. If l is one of the existing images (i, say) $\mathbf{x}^{A_l}$ is already in the antisymmetrization, so if we extend the antisymmetrization to C_l the result must vanish: $\mathbf{I}^{[C_lB_i\cdots B_k\gamma\cdots\delta}\,\mathbf{x}^{A_i}\cdots\mathbf{x}^{A_k]} = \mathbf{0}$. If l is distinct from the existing images we can explicitly add $\mathbf{x}^{A_l}$ to the antisymmetrization list, to get $\mathbf{I}^{[C_lB_i\cdots B_k\gamma\cdots\delta}\,\mathbf{x}^{A_i}\cdots\mathbf{x}^{A_k}\mathbf{x}^{A_l]} = \mathbf{0}$.

Similarly, the minimal reconstruction solution for 3D lines from two images is just the pull-back

$$\mathbf{l}_{ab} \;\sim\; \mathbf{l}_{A_1}\mathbf{l}_{A_2}\,\mathbf{P}^{A_1}_{[a}\mathbf{P}^{A_2}_{b]}$$

or in contravariant form

$$\mathbf{l}^{ab} \;\sim\; \mathbf{l}_{A_1}\mathbf{l}_{A_2}\,\mathbf{P}^{A_1}_c\mathbf{P}^{A_2}_d\,\boldsymbol{\varepsilon}^{abcd}$$

This can be projected into a third image and dualized to give the previously stated line transfer equation

$$\begin{aligned}\mathbf{l}_{A_3} &\sim \mathbf{l}_{A_1}\mathbf{l}_{A_2}\cdot\boldsymbol{\varepsilon}_{A_3B_3C_3}\,\mathbf{P}^{B_3}_a\mathbf{P}^{C_3}_b\mathbf{P}^{A_1}_c\mathbf{P}^{A_2}_d\,\boldsymbol{\varepsilon}^{abcd}\\ &\sim \mathbf{l}_{A_1}\mathbf{l}_{A_2}\,\mathbf{G}_{A_3}{}^{A_1A_2}\end{aligned}$$

More generally, the covariant form of the k-subspace constraint equations given in section 7.5 generates basic reconstruction equations for k dimensional subspaces of the j^{th} image or the world space by dropping one index A_j from the contraction and using it as the α_0 of a set of $k+1$ free indices $\alpha_0\cdots\alpha_k$ designating the reconstructed k-subspace in $\mathcal{P}^{A_j}$. To reconstruct the k-subspace in world coordinates, the projection tensors $\mathbf{P}^{\alpha_i}_{a_i}$ corresponding to the free indices must also be dropped, leaving free world indices $a_0\cdots a_k$.

9 Grassmann Relations between Matching Tensors

The components of any Grassmann tensor must satisfy a set of quadratic 'simplicity' constraints called the **Grassmann relations**. In our case the joint image Grassmannian satisfies

$$\begin{aligned}\mathbf{0} &= \mathbf{I}^{\alpha_0\cdots\alpha_{d-1}[\beta_0}\,\mathbf{I}^{\beta_0\cdots\beta_{d+1}]}\\ &= \tfrac{1}{d+2}\sum_{a=0}^{d+1}(-1)^a\,\mathbf{I}^{\alpha_0\cdots\alpha_{d-1}\beta_a}\,\mathbf{I}^{\beta_0\cdots\beta_{a-1}\beta_{a+1}\cdots\beta_{d+1}}\end{aligned}$$

Table 1: The Grassmann identities between the matching tensors of two and three images.

$\mathbf{0}_{A_1}$	$= \mathbf{F}_{A_1A_2}\,\mathbf{e}_1{}^{A_2}$	$[111, 11122]$
$\mathbf{0}^{A_1A_2}$	$= \mathbf{F}_{B_1B_2}\,\mathbf{F}_{C_1C_2}\,\boldsymbol{\varepsilon}^{B_1C_1A_1}\,\boldsymbol{\varepsilon}^{B_2C_2A_2} + 2\,\mathbf{e}_2{}^{A_1}\,\mathbf{e}_1{}^{A_2}$	$[112, 11222]$
$\mathbf{0}_{A_3}$	$= \mathbf{F}_{A_2A_3}\,\mathbf{e}_1{}^{A_2} - \boldsymbol{\varepsilon}_{A_3B_3C_3}\,\mathbf{e}_1{}^{B_3}\,\mathbf{e}_2{}^{C_3}$	$[111, 22233]$
$\mathbf{0}^{A_3}_{A_1A_2}$	$= \boldsymbol{\varepsilon}_{A_2B_2C_2}\,\mathbf{e}_1{}^{B_2}\,\mathbf{G}_{A_1}{}^{C_2A_3} + \mathbf{e}_1{}^{A_3}\,\mathbf{F}_{A_1A_2}$	$[111, 11223]$
$\mathbf{0}^{A_1}_{A_2A_3}$	$= \boldsymbol{\varepsilon}_{A_2B_2C_2}\,\mathbf{e}_1{}^{B_2}\,\mathbf{G}_{A_3}{}^{A_1C_2} + \boldsymbol{\varepsilon}_{A_3B_3C_3}\,\mathbf{e}_1{}^{B_3}\,\mathbf{G}_{A_2}{}^{A_1C_3}$	$[111, 12233]$
$\mathbf{0}^{A_1A_2A_3}_{B_2}$	$= \mathbf{F}_{B_1B_2}\,\mathbf{G}_{C_1}{}^{A_2A_3}\,\boldsymbol{\varepsilon}^{B_1C_1A_1} - \mathbf{e}_1{}^{A_2}\,\mathbf{G}_{B_2}{}^{A_1A_3} + \boldsymbol{\delta}_{B_2}{}^{A_2}\,\mathbf{e}_1{}^{C_2}\,\mathbf{G}_{C_2}{}^{A_1A_3}$	$[112, 11223]$
$\mathbf{0}^{A_2B_2}_{A_1B_1A_3}$	$= \boldsymbol{\varepsilon}_{A_3B_3C_3}\,\mathbf{G}_{A_1}{}^{A_2B_3}\,\mathbf{G}_{B_1}{}^{B_2C_3} - \mathbf{e}_1{}^{A_2}\,\boldsymbol{\varepsilon}_{A_1B_1C_1}\,\mathbf{G}_{A_3}{}^{C_1B_2}$	$[112, 11233]$
	$-\mathbf{F}_{A_1C_2}\,\boldsymbol{\varepsilon}^{C_2A_2B_2}\,\mathbf{F}_{B_1A_3}$	
$\mathbf{0}^{A_2}_{A_1B_1}$	$= \mathbf{F}_{A_1A_3}\,\mathbf{G}_{B_1}{}^{A_2A_3} + \boldsymbol{\varepsilon}_{A_1B_1C_1}\,\mathbf{e}_3{}^{C_1}\,\mathbf{e}_1{}^{A_2}$	$[112, 11333]$
$\mathbf{0}^{B_1B_2}_{A_1A_2A_3}$	$= \boldsymbol{\varepsilon}_{A_3B_3C_3}\,\mathbf{G}_{A_1}{}^{B_2B_3}\mathbf{G}_{A_2}{}^{B_1C_3} - \mathbf{F}_{A_1A_2}\,\mathbf{G}_{A_3}{}^{B_1B_2}$	$[112, 12233]$
	$+\boldsymbol{\delta}_{A_2}{}^{B_2}\,\mathbf{F}_{A_1C_2}\,\mathbf{G}_{A_3}{}^{B_1C_2} + \boldsymbol{\delta}_{A_1}{}^{B_1}\,\mathbf{e}_1{}^{B_2}\,\mathbf{F}_{A_2A_3}$	
$\mathbf{0}^{B_1A_2B_2}_{A_1}$	$= \mathbf{G}_{C_3}{}^{B_1B_2}\,\mathbf{G}_{A_1}{}^{A_2C_3} + \mathbf{e}_3{}^{B_1}\,\mathbf{F}_{A_1C_2}\,\boldsymbol{\varepsilon}^{C_2A_2B_2} + \boldsymbol{\delta}_{A_1}{}^{B_1}\,\mathbf{e}_1{}^{A_2}\,\mathbf{e}_3{}^{B_2}$	$[112, 12333]$
$\mathbf{0}^{A_2}_{A_1A_3}$	$= \boldsymbol{\varepsilon}_{A_3B_3C_3}\,\mathbf{e}_2{}^{B_3}\,\mathbf{G}_{A_1}{}^{A_2C_3} - \mathbf{F}_{A_1B_2}\mathbf{F}_{C_2A_3}\,\boldsymbol{\varepsilon}^{B_2C_2A_2}$	$[112, 22233]$
$\mathbf{0}^{B_2}_{A_1A_2}$	$= \mathbf{F}_{A_2A_3}\,\mathbf{G}_{A_1}{}^{B_2A_3} + \mathbf{F}_{A_1A_2}\,\mathbf{e}_3{}^{B_2} - \boldsymbol{\delta}_{A_2}{}^{B_2}\,\mathbf{F}_{A_1C_2}\,\mathbf{e}_3{}^{C_2}$	$[112, 22333]$
$\mathbf{0}^{A_1A_2B_2A_3B_3}$	$= \mathbf{G}_{B_1}{}^{A_2A_3}\,\mathbf{G}_{C_1}{}^{B_2B_3}\,\boldsymbol{\varepsilon}^{B_1C_1A_1} - \mathbf{G}_{C_2}{}^{A_1A_3}\,\boldsymbol{\varepsilon}^{C_2A_2B_2}\,\mathbf{e}_1{}^{B_3}$	$[123, 11123]$
	$-\mathbf{G}_{C_3}{}^{A_1A_2}\,\boldsymbol{\varepsilon}^{C_3A_3B_3}\,\mathbf{e}_1{}^{B_2}$	
$\mathbf{0}^{B_1B_2A_3B_3}_{A_1A_2}$	$= \mathbf{G}_{A_2}{}^{B_1A_3}\,\mathbf{G}_{A_1}{}^{B_2A_3} - \mathbf{G}_{A_2}{}^{B_1B_3}\,\mathbf{G}_{A_1}{}^{B_2A_3} - \mathbf{F}_{A_1A_2}\,\mathbf{G}_{C_3}{}^{B_1B_2}\,\boldsymbol{\varepsilon}^{C_3A_3B_3}$	$[123, 11223]$
	$-\boldsymbol{\delta}_{A_2}{}^{B_2}\,\mathbf{G}_{C_2}{}^{B_1A_3}\,\mathbf{G}_{A_1}{}^{C_2B_3} + \boldsymbol{\delta}_{A_1}{}^{B_1}\,\mathbf{G}_{A_2}{}^{C_1B_3}\,\mathbf{G}_{C_1}{}^{B_2A_3}$	

Mechanically substituting expressions for the various components of $\mathbf{I}^{\alpha_0\cdots\alpha_d}$ in terms of the matching tensors produces a long list of quadratic relations between the matching tensors. For reference, table 1 gives a (hopefully complete) list of the identities that can be generated between the matching tensors of two and three images in $d = 3$ dimensions, modulo image permutation, traces of identities with covariant and contravariant indices from the same image, and (anti-)symmetrization operations on identities with several covariant or contravariant indices from the same image. (For example, $\mathbf{F}_{A_2A_3}\,\mathbf{G}_{A_1}{}^{A_2A_3} = 2\,\mathbf{F}_{A_1A_2}\,\mathbf{e}_3{}^{A_2}$ and $\mathbf{F}_{A_3(A_1}\,\mathbf{G}_{B_1)}{}^{A_2A_3} = \mathbf{0}$ follow respectively from tracing $[112, 22233]$ and symmetrizing $[112, 11333]$). The constraint tensors are assumed to be normalized as in their above definitions, in terms of an arbitrary choice of scale for the underlying image projections. In practice, these scale factors must often be recovered from the Grassmann relations themselves. Note that with these conventions, $\mathbf{F}_{A_1A_2} = \mathbf{F}_{A_2A_1}$ and $\mathbf{G}_{A_1}{}^{A_2A_3} = -\mathbf{G}_{A_1}{}^{A_3A_2}$. For clarity the free indices have been displayed on the (zero) left-hand side tensors. The labels indicate one choice of image numbers for the indices of the Grassmann simplicity relation that will generate the identity (there may be others).

As an example of the use of these identities, $\mathbf{G}_{A_1}{}^{A_2A_3}$ follows from linearly from $\mathbf{F}_{A_1A_2}$, $\mathbf{F}_{A_1A_3}$ and the corresponding epipoles $\mathbf{e}_1{}^{A_2}$, $\mathbf{e}_3{}^{A_1}$ and $\mathbf{e}_3{}^{A_2}$ by applying $[112, 11333]$ and $[112, 22333]$.

10 Reconstruction in Joint Image Space

We have argued that multi-image projective reconstruction is essentially a matter of recovering a coherent set of projective scale factors for the measured image points, that it canonically takes place in the joint image space $\mathcal{P}^\alpha$, and that reconstruction in world coordinates is best seen as a choice of basis in the resulting joint image subspace $\mathcal{PI}^\alpha$. To emphasize these points it is interesting to develop 'reconstruction' techniques that work directly in joint image space using measured image coordinates, without reference to *any* 3D world or basis.

First suppose that the complete set of matching tensors between the images has been recovered. It is still necessary to fix an arbitrary overall scale factor for each image. This can be done by choosing any coherent set of relative scalings for the matching tensors, so that they verify the Grassmann simplicity relations as given above. Then, since the components of the joint image Grassmann tensor $\mathbf{I}^{\alpha\beta\cdots\gamma}$ can be recovered directly from the matching tensors, the location of the joint image $\mathcal{PI}^\alpha$ has been fixed.

Now consider a matching set of image points $\{\mathbf{x}^{A_1}, \ldots, \mathbf{x}^{A_m}\}$ with arbitrary relative scalings. As discussed in section 6, the matching constraints are equivalent to the requirement that there be a rescaling of the image points that places the joint image space vector $\sum_{i=1}^m \lambda_i \, \mathbf{x}^{A_i}$ in the joint image $\mathcal{PI}^\alpha$. Expressed in terms of the Grassmannian, this becomes the **joint image reconstruction system**

$$\mathbf{I}^{\,[\alpha\beta\cdots\gamma} \cdot \left(\sum_{i=1}^{m} \lambda_i \, \mathbf{x}^{A_i]} \right) \;=\; \mathbf{0}$$

This is a redundant set of homogeneous multilinear equations in the Grassmannian $\mathbf{I}^{\alpha\beta\cdots\gamma}$, the image points $\mathbf{x}^{A_i}$, and the scale factors λ_i, that can be used to 'reconstruct' the scale factors given the Grassmannian and the image measurements.

These equations can be reexpressed in terms of the matching tensors, in much the same way as the Grassmann simplicity relations can. The types of constraint that can arise for 2D images of 3D space are shown in table 2. The left hand sides are zero tensors and the labels give index image numbers that will generate the equation. The numerical coefficients are valid only for correctly scaled matching tensors. Permuting the images generates further equations. Note that since the equations are algebraically redundant it is only necessary to apply a subset of at least $m-1$ of them to solve for the m scale factors. The optimal choice of equations probably depends on the ease and accuracy with which the various matching tensor components can be estimated.

Recovery of the scale factors locates the reconstructed joint image point $\mathbf{x}^\alpha$ unambiguously in the subspace $\mathcal{PI}^\alpha$. Its coordinates in any chosen basis (*i.e.* with respect to any given choice of the basis-vector columns of the joint projection matrix $\mathbf{P}_a^\alpha$) can easily be obtained, if required. Although this process is arguably too abstract to be called 'reconstruction', all of the relevant structure is certainly present in the joint image representation and can easily be extracted from it.

Given an efficient numerical technique for the resolution of sets of bilinear equations and a sufficient number of matching points, it would also be possible to solve the above equations simultaneously for the vector of matching tensor components and the vector of scale factors, given the measured image coordinates as coefficients. Algebraic elimination of the scale factors from these equations should ultimately lead back to the various matching constraints (modulo probably heavy use of the Grassmann relations). Elimination of the matching tensors (modulo the matching constraints viewed as constraints on the matching tensor components) for sufficiently many matching points would lead to (high degree!) basic reconstruction methods for the recovery of the scale factors directly from measured image coordinates.

Geometrically, the reconstruction process can be pictured as follows. Each image point is a D_i-codimensional subset of its D_i-dimensional image, so under the trivial projection it can be pulled back to a D_i-codimensional subspace of the joint image space $\mathcal{P}^\alpha$. Intersecting the subspaces pulled back from the different images results in an $(m-1)$-dimensional projective subspace of $\mathcal{P}^\alpha$. This is precisely the set of all possible rescalings of the $\mathbf{x}^{A_i}$. The joint image $\mathcal{PI}^\alpha$ intersects this subspace if and only if the matching constraints are satisfied, and the intersection is of course the desired reconstruction. So the problem of multi-image projective reconstruction from points can be viewed as the search for the $(d+m-1)$-dimensional subspace of $\mathcal{P}^\alpha$

Table 2: The five basic types of reconstruction equation for a point in the joint image.

$$\mathbf{0}_{A_2} = (\mathbf{F}_{A_1A_2}\,\mathbf{x}^{A_1})\lambda_1 + (\varepsilon_{A_2B_2C_2}\,\mathbf{e}_1{}^{B_2}\,\mathbf{x}^{C_2})\lambda_2 \quad [11122]$$

$$\mathbf{0}^{A_2A_3} = (\mathbf{G}_{A_1}{}^{A_2A_3}\,\mathbf{x}^{A_1})\lambda_1 - (\mathbf{e}_1{}^{A_3}\,\mathbf{x}^{A_2})\lambda_2 + (\mathbf{e}_1{}^{A_2}\,\mathbf{x}^{A_3})\lambda_3 \quad [11123]$$

$$\mathbf{0}^{A_3}_{A_1A_2} = (\varepsilon_{A_1B_1C_1}\,\mathbf{G}_{A_2}{}^{B_1A_3}\,\mathbf{x}^{C_1})\lambda_1 + (\varepsilon_{A_2B_2C_2}\,\mathbf{G}_{A_1}{}^{B_2A_3}\,\mathbf{x}^{C_2})\lambda_2 - (\mathbf{F}_{A_1A_2}\,\mathbf{x}^{A_3})\lambda_3 \quad [11223]$$

$$\mathbf{0}^{A_2A_3A_4}_{A_1} = (\varepsilon_{A_1B_1C_1}\,\mathbf{H}^{B_1A_2A_3A_4}\,\mathbf{x}^{C_1})\lambda_1 + (\mathbf{G}_{A_1}{}^{A_4A_3}\,\mathbf{x}^{A_2})\lambda_2 - (\mathbf{G}_{A_1}{}^{A_2A_4}\,\mathbf{x}^{A_3})\lambda_3 + (\mathbf{G}_{A_1}{}^{A_2A_3}\,\mathbf{x}^{A_4})\lambda_4 \quad [11234]$$

$$\mathbf{0}^{A_1A_2A_3A_4A_5} = (\mathbf{H}^{A_2A_3A_4A_5}\,\mathbf{x}^{A_1})\lambda_1 - (\mathbf{H}^{A_1A_3A_4A_5}\,\mathbf{x}^{A_2})\lambda_2 + (\mathbf{H}^{A_1A_2A_4A_5}\,\mathbf{x}^{A_3})\lambda_3 - (\mathbf{H}^{A_1A_2A_3A_5}\,\mathbf{x}^{A_4})\lambda_4 + (\mathbf{H}^{A_1A_2A_3A_4}\,\mathbf{x}^{A_5})\lambda_5 \quad [12345]$$

that contains (or comes closest to containing) a given set of $(m-1)$-dimensional joint-image-point subspaces, followed by an arbitrary choice (the scale factors) of a d-dimensional subspace (the joint image) of the $(d+m-1)$-dimensional space that meets each joint-image-point subspace transversally. The reconstruction of lines and higher dimensional subspaces can be viewed in similarly geometric terms.

11 Perspectives

The theoretical part of the paper is now finished, but before closing it may be worthwhile to reflect a little on our two principal themes: projective reconstruction and the tensor calculus. We will take it for granted that the projective and algebraic-geometric approaches to vision are here to stay: the 'unreasonable efficacy of mathematics in the physical sciences' can only lead to an increasing mathematization of the field.

11.1 Matching & Reconstruction

Clearly visual scene reconstruction is a large and complex problem that is not going to be 'solved' by any one contribution, so we will restrict ourselves to a few technical remarks. To the extent that the problem can be decomposed at all, the most difficult parts of it will probably always be the low level feature extraction and token matching. 3D reconstruction seems relatively straightforward once image tokens have been put into correspondence, although much remains to be done on the practical aspects, particularly on error models [17, 4, 20] and the recovery of Euclidean structure [17].

Given the complexity and algebraic redundancy of the trilinear and quadrilinear constraints it is certainly legitimate to ask whether they are actually likely to be *useful* in practice. I think that the answer is a clear 'yes' for the trilinear constraints and the overall joint image/Grassmannian picture, but the case for the quadrilinear constraints is still open.

The principal application of the matching tensors must be for token matching and verification. The trilinear constraints can be used directly to verify the correspondence of a triple of points or lines, or indirectly to transfer a hypothesized feature location to a third image given its location in two others, in a hypothesize-and-test framework. Image synthesis (*e.g.* image sequence compression and interpolation) is likely to be another important application of transfer [10].

Fundamental matrices can also be used for these applications, but because the higher order constraints 'holistically' combine data from several images and there is built-in redundancy in the constraint equations, it is likely that they will prove less prone to mismatches and numerically more stable than a sequence of applications of the epipolar constraint. For example Shashua [19] has reported that a single trilinear constraint gives more reliable transfer results than two epipolar ones, and Faugeras and Mourrain [6] have pointed out that bilinear constraint based transfer breaks down when the 3D point lies in the trifocal plane or the three op-

tical centres are aligned, whereas trilinear transfer continues to be reasonably well conditioned.

When there are four images the quadrilinear constraint can also be used for point matching and transfer, but the equations are highly redundant and it seems likely that bilinear and trilinear methods will prove adequate for the majority of applications. The trilinear constraint is nonsingular for almost all situations involving points, provided the optical centres do not coincide and the points avoid the lines passing between them.

The most important failure for lines is probably that for lines lying in an epipolar plane of two of the images. In this case the constraints mediated by trivalent tensors are vacuous (although there is still enough information to reconstruct the corresponding 3D line unless it lies in the trifocal plane or the optical centres are aligned) and those mediated by quadrivalent tensors are rank deficient. But given the linear dependence of the various line constraints it is not clear that the quadrivalent ones have any advantage over an equivalent choice of trivalent ones.

A closely related issue is that of linear versus higher order methods. Where possible, linear formulations are usually preferred. They tend to be simpler, faster, better understood and numerically more stable than their nonlinear counterparts, and they are usually much easier to adapt to redundant data, which is common in vision and provides increased accuracy and robustness. On the other hand, nonlinear constraints can not be represented accurately within a linear framework.

This is especially relevant to the estimation of the matching tensors. We have emphasized that the matching tensor components and constraint equations are *linearly* independent but *quadratically* highly dependent. It is straightforward to provide linear minimum-eigenvector methods to estimate: the 9-component fundamental matrix from at least 8 pairs of corresponding points in two images [11, 12]; each of the three linearly independent 27-component trilinear tensors from at least 7 triples of points in three images; and the 81-component quadrilinear tensor from at least 6 quadruples of corresponding points in four images [20]. For complex applications several of these tensors might be needed, for example a fundamental constraint might provide initial feature pairings that can be used to check for corresponding features in a third image using further fundamental or trilinear constraints. Also, different trilinear tensors are required for point transfer and line transfer.

Unfortunately, it turns out that the above linear estimation techniques (particularly that for the fundamental matrix) are numerically rather poorly conditioned, so that the final estimates are very sensitive to measurement errors and outliers. Moreover, even in the case of a single fundamental matrix there is a nonlinear constraint that can not be expressed within the linear framework. The quadratic epipolar relation $\mathbf{F}_{A_1A_2}\,\mathbf{e}_1{}^{A_2} = \mathbf{0}$ implies the cubic constraint $\mathrm{Det}(\mathbf{F}) = 0$. If this constraint is ignored, one finds that the resulting estimates of $\mathbf{F}$ and the epipoles tend to be rather inaccurate [12]. In fact, the linear method is often used only to initialize nonlinear optimization routines that take account of the nonlinearity and the estimated measurement errors in the input data.

This leads to the following open question: *When several matching tensors are being estimated, to what extent is it possible or necessary to take account of the quadratic constraints between them?* The full set of quadratic relations is very complex and it is probably not practical to account for all of them individually: it would be much simpler just to work directly in terms of the 3D joint image geometry. Moreover, many of the relations depend on the relative scaling of the constraint tensors and the recovery of these further complicates the issue (it is a question of exactly which combinations of components need to be fixed to ensure consistency and numerical stability). On the other hand, experience with the fundamental matrix suggests that it is dangerous to ignore the constraints entirely. Some at least of them are likely to be important in any given situation. Our current understanding of these matters is very sketchy: essentially all we have is a few *ad hoc* comparisons of particular techniques.

As a final point, a few people seem to have been hoping for some 'magic' reconstruction technique that completely avoids the difficulties of image-to-image matching, perhaps by holistically combining data from a large number of images (or a single dense image sequence). The fact that the matching constraints stop at four images (or equivalently three time derivatives) does not preclude this, but perhaps makes it seem a little less likely. On the other hand,

the simplicity of the joint image picture makes incremental recursive reconstruction techniques that correctly handle the measurement errors and constraint geometry seem more likely (*c.f.* [16]).

11.2 Tensors vs. the Rest

This paper is as much about the use of tensors as a vehicle for mathematical vision as it is about image projection geometry. Tensors have seldom been used in vision and many people appear to be rather tensor-phobic, so it seems appropriate to say a few words in their favour: "*Don't panic!*" [1].

First of all, what *is* a tensor? — It is a collection (a multidimensional array) of components that represent a single geometric object with respect to some system of coordinates, and that are intermixed when the coordinate system is changed. This immediately evokes the two principal concerns of tensor calculus: *(i)* to perform manipulations *abstractly* at the object level rather than explicitly at the component level; and *(ii)* to ensure that all expressions are properly *covariant* (*i.e.* have the correct transformation laws) under changes of basis. The advantages are rather obvious: the higher level of abstraction brings greater compactness, clarity and insight, and the guaranteed covariance of well-formed tensorial expressions ensures that no hidden assumptions are made and that the correct algebraic symmetries and relationships between the components are automatically preserved.

Vectors are the simplest type of tensor and the familiar 3D vector calculus is a good example of the above points: it is much simpler and less error prone to write a single vector $\mathbf{x}$ instead of three components $(x^1,\ x^2,\ x^3)$ and a symbolic cross product $\mathbf{z} = \mathbf{x} \times \mathbf{y}$ instead of three equations $z^1 = x^2y^3 - x^3y^2$, $z^2 = x^3y^1 - x^1y^3$ and $z^3 = x^1y^2 - x^2y^1$. Unfortunately, the simple index-free matrix-vector notation seems to be difficult to extend to higher-order tensors with the required degree of flexibility. (Mathematicians sometimes define tensors as multilinear functions $\mathbf{T}(\mathbf{x}, \ldots, \mathbf{z})$ where $\mathbf{x}, \ldots, \mathbf{z}$ are vectors of some type and the result is a scalar, but this notation becomes hopelessly clumsy when it comes to inter-tensor contractions, antisymmetrization and so forth). In fact, the index-free notation becomes as much a dangerous weapon as a useful tool as soon as one steps outside the realm of simple vector calculations in a single Euclidean space. It is only too easy to write $\mathbf{x}^\top\mathbf{x} = 1$ in a projective space where no transpose (metric tensor) exists, or a meaningless 'epipolar equation' $\mathbf{l}^\top\mathbf{F}\,\mathbf{x} = 0$ where $\mathbf{l}$ is actually the 3-component vector of an image line (rather than an image *point*) and $\mathbf{x}$ belongs to the wrong image for the fundamental matrix $\mathbf{F}$ (which should have been transposed in any case).

To avoid this sort of confusion, it is essential to use a notation that clearly distinguishes the space and covariant/contravariant type of each index. Although it can not be denied that this sometimes leads to rather baroque-looking formulae — especially when there are many indices from many different spaces as in this paper — it is much preferable to the alternatives of using either no indices at all or i, j, and k for everything, so that one can never quite see what is supposed to be happening. It is important not to be fooled into thinking that tensor equations are intrinsically difficult just because they have indices. For simple calculations the indexed notation is not significantly more difficult to use than the traditional index-free one, and it becomes *much* clearer and more powerful in complex situations. For a visually appealing (but typographically inconvenient) pictorial notation, see the appendix of [18].

Simultaneously with the work presented in this paper, at least two other groups independently converged on parts of the constraint geometry from component-based points of view: Faugeras & Mourrain [6] using the Grassmann-Cayley algebra of skew linear forms, and Werman & Shashua [22] using Gröbner bases and algebraic elimination theory. Both approaches make very heavy use of computer algebra whereas all of the calculations in the present paper were done by hand, and neither (notwithstanding the considerable value of their results) succeeded in obtaining anything like a complete picture of the constraint geometry. My feeling is that it is perhaps no accident that in each of the three categories: level of geometric abstraction, efficiency of calculation, and insight gained, the relative ordering is the same: tensor calculus $>$ Grassmann-Cayley algebra $>$ elimination theory.

Elimination-theoretic approaches using resultants and Gröbner bases seem to be intrinsically component-based. They take no account of the tensorial structure of the equations and therefore make no use of the many symmetries between them,

so even when the coordinate systems are carefully adapted to the problem they tend to carry a significant amount of computational redundancy. Werman & Shashua [22] suggest that an advantage of such approaches is the fact that very little geometric insight is required. Unfortunately, one might also suggest that very little geometric insight is *gained*: the output is a complex set of equations with no very clearly articulated structure.

The Grassmann-Cayley algebra [6, 2] is spiritually much closer to the tensorial point of view. Indeed, it can be viewed as a specialized index-free notation for manipulating completely antisymmetric covariant and contravariant tensors. It supports operations such as antisymmetrization over indices from several tensors (wedge product), contractions over corresponding sets of covariant and contravariant antisymmetric indices (hook product), and contravariant-covariant dualization (sometimes used to identify the covariant and contravariant algebras and then viewed as the identity, in which case the hook product is replaced by the join product). Given the connection with Grassmann coordinates, the Grassmann-Cayley algebra can be viewed as a calculus of intersection and union (span) for projective subspaces: clearly a powerful and highly relevant concept. It is likely that this approach would have lead fairly rapidly to the full Grassmannian matching constraint geometry, notwithstanding the relative opacity of the initial component-oriented formulations.

Despite its elegance, there are two problems with the Grassmann-Cayley algebra as a general formalism. The first is that it is not actually very general: it is good for calculations with linear or projective subspaces, but it does not extend gracefully to more complex situations or higher-degree objects. For example quadric surfaces are represented by *symmetric* tensors which do not fit at all well into the antisymmetric algebra. Tensors are much more flexible in this regard. The second problem with the Grassmann-Cayley algebra is that it is often infuriatingly vague about geometric (covariance) issues. Forms of different degree with indices from different spaces can be added formally within the algebra, but this makes no sense at all tensorially: such objects do not transform reasonably under changes of coordinates, and consequently do not have any clear *geometric* meaning, whatever the status of the algebra. The fact that the algebra has a stratified tensorial structure is usually hidden in the definitions of the basic product operations, but it becomes a central issue as soon as geometric invariance is called into question.

In summary, my feeling is that the tensorial approach is ultimately the most promising. The indexed notation is an extraordinarily powerful, general and flexible tool for the algebraic manipulation of geometric objects. It displays the underlying the structure and covariance of the equations very clearly, and it naturally seems to work at about the right level of abstraction for practical calculations: neither so abstract nor so detailed as to hide the essential structure of the problem. Component-based approaches are undoubtedly useful, but they are probably best reserved until *after* a general tensorial derivation has been made, to specialize and simplify a set of abstract tensorial equations to the particular application in hand.

As an example of this, a $k+1$ index antisymmetric tensor representing a k dimensional subspace of a d dimensional projective space has (very naïvely) $(d+1)^{k+1}$ components, but only $\binom{d+1}{k+1}$ of these are linearly independent owing to antisymmetry. The independent components can easily be enumerated (the indices $i_0 i_1 \cdots i_k$ for $0 \leq i_0 < i_1 < \ldots < i_k \leq d$ form a spanning set) and gathered into an explicit $\binom{d+1}{k+1}$ component vector for further numerical or symbolic manipulation. In fact, these components span exactly one tensorial stratum of the Grassmann-Cayley algebra.

It is perhaps unfortunate that current computer algebra systems seem to have very few tools for manipulating general tensorial expressions, as these would greatly streamline the derivation and specialization processes. However, there does not appear to be any serious obstacle to the development of such tools and it is likely that they will become available in the near future.

12 Summary

Given a set of perspective projections into m projective image spaces, there is a 3D subspace of the space of combined image coordinates called the **joint image**. This is a complete projective replica of the 3D world expressed directly in terms of scaled image coordinates. It is defined intrinsically by the

physical situation up to an arbitrary choice of some internal scalings. Projective reconstruction in the joint image is a canonical process requiring only a rescaling of the image coordinates. A choice of basis in the joint image allows the reconstruction to be transferred to world space.

There are multilinear **matching constraints** between the images that determine whether a set of image points could be the projection of a single world point. For 3D worlds only three types of constraint exist: the epipolar constraint generated by the fundamental matrix between pairs of images, Shashua's trilinear constraints between triples of images and a new quadrilinear constraint on sets of corresponding points from four images.

Moreover, the entire set of constraint tensors for all the images can be combined into a single compact geometric object, the antisymmetric 4 index **joint image Grassmannian** tensor. This can be recovered from image measurements whenever the individual constraint tensors can. It encodes precisely the information needed for reconstruction: the location of the joint image in the space of combined image coordinates. It also generates the matching constraints for images of lines and a set of **minimal reconstruction techniques** closely associated with the matching constraints. Structural constraints on the Grassmannian tensor produce quadratic identities between the various constraint tensors.

Acknowledgements

This work was supported by the European Community through Esprit programs HCM and SECOND and benefited from discussions with many colleagues. The notation and the tensorial treatment of projective space are based on Roger Penrose's approach to relativistic spinors and twistors [18], which partly derives (I believe) from Hodge & Pedoe's excellent treatise on algebraic geometry [9]. Without these powerful tools this work would not have been possible.

A Mathematical Background

This appendix provides a very brief overview of the linear algebra and projective geometry need to understand this paper, and a little background information on our notation. For more details on using tensor calculus for projective space see [9, 18].

A.1 Vectors and Tensors

A **vector space** $\mathcal{H}^a$ is a space on which addition and scaling of elements are defined: $\lambda\mathbf{x}^a + \mu\mathbf{y}^a$ is in $\mathcal{H}^a$ for all scalars λ and μ and elements $\mathbf{x}^a$ and $\mathbf{y}^a$ of $\mathcal{H}^a$. The **span** of a set $\{\mathbf{e}^a_1, \ldots, \mathbf{e}^a_k\}$ of elements of $\mathcal{H}^a$ is the vector space of linear combinations $x^1\mathbf{e}^a_1 + \cdots + x^k\mathbf{e}^a_k$ of elements of the set. A minimal set that spans the entire space is called a **basis** and the number of elements in the set is the **dimension** of the space. Given a basis $\{\mathbf{e}^a_1, \ldots, \mathbf{e}^a_d\}$ for a d dimensional vector space $\mathcal{H}^a$, any element $\mathbf{x}^a$ of the space can be expressed as $x^1\mathbf{e}^a_1 + \cdots + x^d\mathbf{e}^a_d$ and associated with the coordinate vector $(x^1, \ldots, x^d)$.

It is helpful to view the superscript a as an **abstract index** [18], *i.e.* an abstract label or placeholder denoting the space the object belongs to. However given a choice of basis it can also be thought of as a variable indexing the coordinate vector that represents the object in that basis.

For every vector space $\mathcal{H}^a$ there is a dual vector space of linear mappings on $\mathcal{H}^a$, denoted $\mathcal{H}_a$. An element $\mathbf{l}_a$ of $\mathcal{H}_a$ acts linearly on an element $\mathbf{x}^a$ of $\mathcal{H}^a$ to produce a scalar. This action is denoted symbolically by $\mathbf{l}_a\mathbf{x}^a$ and called **contraction**. Any basis $\{\mathbf{e}^a_1, \ldots, \mathbf{e}^a_d\}$ for $\mathcal{H}^a$ defines a unique **dual basis** $\{\mathbf{e}^1_a, \ldots, \mathbf{e}^d_a\}$ for $\mathcal{H}_a$ with $\mathbf{e}^i_a\mathbf{e}^a_j = \delta^i_j$, where δ^i_j is 1 when $i = j$ and 0 otherwise. The i^{th} coordinate of $\mathbf{x}^a$ in the basis $\{\mathbf{e}^a_j\}$ is just $x^i \equiv \mathbf{e}^i_a\,\mathbf{x}^a$. If elements of $\mathcal{H}^a$ are represented in the basis $\{\mathbf{e}^a_i\}$ as d index column vectors, elements of $\mathcal{H}_a$ in the dual basis $\{\mathbf{e}^i_a\}$ behave like d index row vectors. Contraction is then just the dot product of the coordinate vectors: $(u_1\,\mathbf{e}^1_a + \cdots + u_d\,\mathbf{e}^d_a)(x^1\,\mathbf{e}^a_1 + \cdots + x^d\,\mathbf{e}^a_d) = u_1\,x^1 + \cdots + u_d\,x^d$. Contraction involves a sum over coordinates but we do not explicitly write the summation signs: whenever a superscript label also appears as a subscript a summation is implied. This is called the **Einstein summation convention**. The order of terms is unimportant: $\mathbf{u}_a\,\mathbf{x}^a$ and $\mathbf{x}^a\,\mathbf{u}_a$ both denote the contraction of the dual vector $\mathbf{u}_a$ with the vector $\mathbf{x}^a$.

Suppose we change the basis in $\mathcal{H}^a$ according to $\mathbf{e}^a_i \rightarrow \tilde{\mathbf{e}}^a_i = \sum_j \mathbf{e}^a_j\,\Lambda^j{}_i$ for some matrix $\Lambda^j{}_i$. To keep the resulting abstract element of $\mathcal{H}^a$ the same, coordinate vectors must transform inversely

according to $x^i \rightarrow \tilde{x}^i = \sum_j (\Lambda^{-1})^i{}_j \, x^j$. To preserve the relations $\tilde{\mathbf{e}}^i_a \, \tilde{\mathbf{e}}^a_j = \delta^i_j$, the dual basis must also transform as $\mathbf{e}^i_a \rightarrow \tilde{\mathbf{e}}^i_a = \sum_j (\Lambda^{-1})^i{}_j \, \mathbf{e}^j_a$. Finally, to leave the abstract element of the dual space the same, dual coordinate vectors must transform as $u_i \rightarrow \tilde{u}_i = \sum_j u_j \, \Lambda^j{}_i$. Because of the transformations of their coordinates under changes of basis, vectors $\mathbf{x}^a$ are called **contravariant** and dual vectors $\mathbf{u}_a$ are called **covariant**.

An element $\mathbf{x}^a$ of $\mathcal{H}^a$ can also be viewed as a linear mapping on elements of $\mathcal{H}_a$ defined by $\mathbf{u}_a \mathbf{x}^a$, in other words as an element of the dual of the dual of $\mathcal{H}^a$. For finite dimensional spaces every linear mapping on $\mathcal{H}_a$ can be written this way, so there is a complete symmetry between $\mathcal{H}^a$ and $\mathcal{H}_a$: neither is 'more primitive'.

Any nonzero element of $\mathcal{H}_a$ defines a $d-1$ dimensional subspace of $\mathcal{H}^a$ by the equations $\mathbf{u}_a \mathbf{x}^a = 0$, and conversely any $d-1$ dimensional subspace defines a unique element of $\mathcal{H}_a$ up to scale.

It is possible to take formal ('tensor' or 'outer') products of n-tuples of elements of vector spaces, for example a formal element $\mathbf{T}^{aA}{}_\alpha \equiv \mathbf{x}^a \, \mathbf{y}^A \, \mathbf{z}_\alpha$ can be made from elements $\mathbf{x}^a$, $\mathbf{y}^A$, $\mathbf{z}_\alpha$ of vector spaces $\mathcal{H}^a$, $\mathcal{H}^A$ and $\mathcal{H}_\alpha$. The vector space of linear combinations of such objects (for different choices of $\mathbf{x}^a$, $\mathbf{y}^A$ and $\mathbf{z}_\alpha$) is called the tensor product space $\mathcal{H}^{aA}{}_\alpha = \mathcal{H}^a \otimes \mathcal{H}^A \otimes \mathcal{H}_\alpha$. When there are several distinct copies of $\mathcal{H}^a$ we use distinct letters to denote them, *e.g.* $\mathcal{H}^{ab}{}_c = \mathcal{H}^a \otimes \mathcal{H}^b \otimes \mathcal{H}_c$ contains two copies of $\mathcal{H}^a$. Elements of a tensor product space are called **tensors** and can be thought of as multidimensional arrays of components in some chosen set of bases. Under changes of basis each of the indices must be transformed individually.

There are a number of important generic operations on tensors. A set of tensors can be contracted together over any appropriate subset of their indices, for example $\mathbf{u}_{ab} \, \mathbf{x}^a \in \mathcal{H}_b$, $\mathbf{u}_a \, \mathbf{T}^{aB}{}_c \, \mathbf{x}^c \in \mathcal{H}^B$. Self contractions $\mathbf{T}^{ab\cdots}{}_{ac\cdots} \in \mathcal{H}^{b\cdots}{}_{c\cdots}$ are called **traces**. A group of indices can be **(anti-)symmetrized** by averaging over all possible permutations of their positions, with an additional minus sign for odd permutations during antisymmetrization. On indices, $(\cdots)$ denotes symmetrization and $[\cdots]$ antisymmetrization. For example $\mathbf{T}^{(ab)} = \frac{1}{2}(\mathbf{T}^{ab} + \mathbf{T}^{ba})$ and $\mathbf{T}^{[ab]} = \frac{1}{2}(\mathbf{T}^{ab} - \mathbf{T}^{ba})$ can be viewed as symmetric and antisymmetric matrices, and $\mathbf{T}^{[abc]} = \frac{1}{3!}(\mathbf{T}^{abc} - \mathbf{T}^{bac} + \mathbf{T}^{bca} - \mathbf{T}^{cba} + \mathbf{T}^{cab} - \mathbf{T}^{acb})$ is an antisymmetric 3 index tensor. A group of indices is **(anti-)symmetric** if (anti-)symmetrization over them does not change the tensor: $(\cdots)$ and $[\cdots]$ are also used to denote this, for example $\mathbf{T}^{[ab]}_{(cd)} \in \mathcal{H}^{[ab]}{}_{(cd)}$ is antisymmetric in ab and symmetric in cd. Permutation of (anti-)symmetric indices changes at most the sign of the tensor.

In d dimensions antisymmetrizations over more than d indices vanish: in any basis each index must take a distinct value between 1 and d. Up to scale there is a unique antisymmetric d index tensor $\boldsymbol{\varepsilon}^{a_1 a_2 \cdots a_d} \in \mathcal{H}^{[a_1 a_2 \cdots a_d]}$: choosing $\boldsymbol{\varepsilon}^{12\cdots d} = +1$ in some basis, all other components are ± 1 or 0. Under a change of basis the components of $\boldsymbol{\varepsilon}^{a_1 \cdots a_d}$ are rescaled by the determinant of the transformation matrix. There is a corresponding dual tensor $\boldsymbol{\varepsilon}_{a_1 a_2 \cdots a_d} \in \mathcal{H}_{[a_1 a_2 \cdots a_d]}$ with components ± 1 or 0 in the dual basis. $\boldsymbol{\varepsilon}_{a_1 a_2 \cdots a_d}$ defines a volume element on $\mathcal{H}^a$, giving the volume of the hyper-parallelepiped formed by d vectors $\mathbf{x}^a_1, \ldots, \mathbf{x}^a_d$ as $\boldsymbol{\varepsilon}_{a_1 a_2 \cdots a_d} \, \mathbf{x}^{a_1}_1 \cdots \mathbf{x}^{a_d}_d$. The determinant of a linear transformation $\mathbf{T}^a_b$ on $\mathcal{H}^a$ can be defined as $\frac{1}{d!} \boldsymbol{\varepsilon}_{a_1 a_2 \cdots a_d} \, \mathbf{T}^{a_1}_{b_1} \cdots \mathbf{T}^{a_d}_{b_d} \, \boldsymbol{\varepsilon}^{b_1 b_2 \cdots b_d}$, and this agrees with the determinant of the matrix of $\mathbf{T}^a_b$ in any coordinate basis. A contravariant antisymmetric k index tensor $\mathbf{T}^{[a_1 \cdots a_k]}$ has a covariant antisymmetric $d-k$ index **dual** $(*\mathbf{T})_{a_{k+1} \cdots a_d} \equiv \frac{1}{k!} \, \boldsymbol{\varepsilon}_{a_{k+1} \cdots a_d b_1 \cdots b_k} \, \mathbf{T}^{b_1 \cdots b_k}$. Conversely $\mathbf{T}^{a_1 \cdots a_k} = \frac{1}{(d-k)!} \, (*\mathbf{T})_{b_{k+1} \cdots b_d} \, \boldsymbol{\varepsilon}^{b_{k+1} \cdots b_d a_1 \cdots a_k}$. A tensor and its dual contain the same information and both have $\binom{d}{k}$ independent components.

A.2 Grassmann Coordinates

Antisymmetrization and duality are important in the theory of linear subspaces. Consider a set $\{\mathbf{v}^a_1, \ldots, \mathbf{v}^a_k\}$ of k independent vectors spanning a k dimensional subspace Σ of $\mathcal{H}^a$. Given some choice of basis the vectors can be viewed as column vectors and combined into a single $d \times k$ matrix. Any set $\{a_1, \ldots, a_k\}$ of k distinct rows of this matrix defines a $k \times k$ submatrix whose determinant is a $k \times k$ **minor** of the original matrix. Up to a constant scale factor these minors are exactly the components of the tensor $\Sigma^{a_1 \cdots a_k} \equiv \mathbf{v}^{[a_1}_1 \cdots \mathbf{v}^{a_k]}_k$. If the original vectors are independent the $d \times k$ matrix has rank k and at least one of the $k \times k$ minors (and hence the tensor $\Sigma^{a_1 \cdots a_k}$) will not vanish. Conversely, if the

tensor vanishes the vectors are linearly dependent.

A vector $\mathbf{x}^a$ lies in the subspace Σ if and only if all of the $(k+1)\times(k+1)$ minors of the $d\times(k+1)$ matrix whose columns are $\mathbf{x}^a$ and the $\mathbf{v}_i^a$ vanish. In tensorial terms: $\mathbf{x}^a$ is an element of Σ if and only if $\Sigma^{[a_1\cdots a_k}\,\mathbf{x}^{a]} = \mathbf{0}$. So no two distinct subspaces have the same $\Sigma^{a_1\cdots a_k}$. Under a $k\times k$ linear redefinition $\mathbf{v}_i^a \to \tilde{\mathbf{v}}_i^a = \sum_j \Lambda_i{}^j \mathbf{v}_j^a$ of the spanning vectors, the $k\times k$ minors are simply a constant factor of $\mathrm{Det}(\Lambda_i{}^j)$ different from the old ones by the usual determinant of a product rule. So up to scale $\Sigma^{a_1\cdots a_k}$ is independent of the set of vectors in Σ chosen to span it.

A subspace Σ can also be defined as the null space of a set of $d-k$ independent linear forms $\{\mathbf{u}_a^{k+1},\ldots,\mathbf{u}_a^d\}$, *i.e.* as the set of $\mathbf{x}^a$ on which all of the $\mathbf{u}_a^i$ vanish: $\mathbf{u}_a^i\,\mathbf{x}^a = 0$. The $\mathbf{u}_a^i$ can be viewed as a $(d-k)\times d$ matrix of row vectors. Arguments analogous to those above show that the covariant antisymmetric $d-k$ index tensor $\Sigma_{a_{k+1}\cdots a_d} \equiv \mathbf{u}^{k+1}_{[a_{k+1}}\cdots\mathbf{u}^d_{a_d]}$ is independent (up to scale) of the $\{\mathbf{u}_a^i\}$ chosen to characterize Σ and defines Σ as the set of points for which $\Sigma_{a_{k+1}\cdots a_d}\,\mathbf{x}^{a_d} = \mathbf{0}$. We use the same symbol for $\Sigma_{a_{k+1}\cdots a_d}$ and $\Sigma^{a_1\cdots a_k}$ because up to scale they turn out to be mutually dual: $\Sigma_{a_{k+1}\cdots a_d} \sim \frac{1}{k!}\,\boldsymbol{\varepsilon}_{a_{k+1}\cdots a_d b_1\cdots b_k}\,\Sigma^{b_1\cdots b_k}$. In particular a hypersurface can be denoted either by $\mathbf{u}_a$ or by $\mathbf{u}^{[a_1\cdots a_{d-1}]}$.

Hence, up to scale, $\Sigma^{a_1\cdots a_k}$ and its dual $\Sigma_{a_{k+1}\cdots a_d}$ are intrinsic characteristics of the subspace Σ, independent of the bases chosen to span it and uniquely defined by and defining it. In this sense the antisymmetric tensors provide a sort of coordinate system on the space of linear subspaces of $\mathcal{H}^a$, called **Grassmann coordinates**.

Unfortunately, only very special antisymmetric tensors specify subspaces. The space of k dimensional linear subspaces of a d dimensional vector space is only $k\,(d-k)$ dimensional, whereas the antisymmetric k index tensors have $\binom{d}{k}$ independent components, so the Grassmann coordinates are massively redundant. The tensors that do define subspaces are called **simple** because they satisfy the following complex quadratic **Grassmann relations**:

$$\Sigma^{a_1\cdots[a_k}\,\Sigma^{b_1\cdots b_k]} = \mathbf{0}$$

or in terms of the dual

$$\Sigma_{a_{k+1}\cdots a_d}\,\Sigma^{a_d b_2\cdots b_k} = \mathbf{0}$$

These relations obviously hold for any tensor of the form $\mathbf{v}_1^{[a_1}\cdots\mathbf{v}_k^{a_k]}$ because one of the vectors must appear twice in an antisymmetrization. What is less obvious is that they do not hold for any tensor that can not be written in this form.

Although their redundancy and the complexity of the Grassmann relations makes them rather inconvenient for numerical work, Grassmann coordinates are a powerful tool for the algebraization of geometric operations on subspaces. For example the union of two independent subspaces is just $\Sigma^{[a_1\cdots a_k}\Gamma^{b_1\cdots b_l]}$ and dually the intersection of two (minimally) intersecting subspaces is $\Sigma_{[a_1\cdots a_k}\Gamma_{b_1\cdots b_l]}$.

A.3 Projective Geometry

Given a $d+1$ dimensional vector space $\mathcal{H}^a$ with nonzero elements $\mathbf{x}^a$ and $\mathbf{y}^a$ ($a = 0,\ldots,d$), we will write $\mathbf{x}^a \sim \mathbf{y}^a$ and say that $\mathbf{x}^a$ and $\mathbf{y}^a$ are *equivalent up to scale* whenever there is a nonzero scalar λ such that $\mathbf{x}^a = \lambda\,\mathbf{y}^a$. The d dimensional **projective space** $\mathcal{P}^a$ is defined to be the set of nonzero elements of $\mathcal{H}^a$ under equivalence up to scale. When we write $\mathbf{x}^a \in \mathcal{P}^a$ we really mean the equivalence class $\{\lambda\,\mathbf{x}^a |\ \lambda \neq 0\}$ of $\mathbf{x}^a$ under $\sim$.

The span of any $k+1$ independent representatives $\{\mathbf{x}_0^a,\ldots,\mathbf{x}_k^a\}$ of points in $\mathcal{P}^a$ is a $k+1$ dimensional vector subspace of $\mathcal{H}^a$ that projects to a well-defined k dimensional projective subspace of $\mathcal{P}^a$ called the subspace **through** the points. Two independent points define a one dimensional projective subspace called a projective line, three points define a projective plane, and so forth. The vector subspaces of $\mathcal{H}^a$ support notions of subspace dimension, independence, identity, containment, intersection, and union (vector space sum or smallest containing subspace). All of these descend to the projective subspaces of $\mathcal{P}^a$. Similarly, linear mappings between vector spaces, kernels and images, injectivity and surjectivity, and so on all have their counterparts for projective mappings between projective spaces.

Tensors on $\mathcal{H}^a$ also descend to projective tensors defined up to scale on $\mathcal{P}^a$. Elements $\mathbf{u}_a$ of the projective version $\mathcal{P}_a$ of the dual space $\mathcal{H}_a$ define $d-1$ dimensional projective hyperplanes in $\mathcal{P}^a$ via $\mathbf{u}_a\mathbf{x}^a = 0$. The duality of $\mathcal{H}^a$ and $\mathcal{H}_a$ descends to a powerful duality principle between points and hyperplanes on $\mathcal{P}^a$ and $\mathcal{P}_a$.

More generally the antisymmetric $k+1$ index contravariant and $d-k$ index covariant Grassmann tensors on $\mathcal{H}^a$ define k dimensional projective subspaces of $\mathcal{P}^a$. For example given independent points $\mathbf{x}^a$, $\mathbf{y}^a$ and $\mathbf{z}^a$ of $\mathcal{P}^a$ the projective tensor $\mathbf{x}^{[a}\mathbf{y}^{b]}$ defines the line through $\mathbf{x}^a$ and $\mathbf{y}^a$ and $\mathbf{x}^{[a}\mathbf{y}^b\mathbf{z}^{c]}$ defines the plane through $\mathbf{x}^a$, $\mathbf{y}^a$ and $\mathbf{z}^a$. Similarly, in 3D a line can be represented dually as the intersection of two hyperplanes $\mathbf{u}_{[a}\mathbf{v}_{b]}$ while a point requires three $\mathbf{u}_{[a}\mathbf{v}_b\mathbf{w}_{c]}$. In 2D a single hyperplane $\mathbf{u}_a$ suffices for a line, and two are required for a point $\mathbf{u}_{[a}\mathbf{v}_{b]}$. Dualization gives back the contravariant representation, *e.g.* $\mathbf{x}^a = \mathbf{u}_b\mathbf{v}_c\,\varepsilon^{abc}$ are the coordinates of the intersection of the two lines $\mathbf{u}_a$ and $\mathbf{v}_a$ in 2D.

A d dimensional projective space can be thought of as a d dimensional affine space (*i.e.* a Euclidean space with points, lines, planes, and so on, but no origin or notion of absolute distance) with a number of **ideal** points added 'at infinity'. Choosing a basis for $\mathcal{H}^a$, any representative $\mathbf{x}^a$ of an element $\mathcal{P}^a$ with $x^0 \neq 0$ can be rescaled to the form $(1, x^1, \ldots, x^d)^\top$. This defines an inclusion of the affine space $(x^1, \ldots, x^d)$ in $\mathcal{P}^a$, but the $d-1$ dimensional projective subspace 'at infinity' of elements of $\mathcal{P}^a$ with $x^0 = 0$ is not represented. Under this inclusion affine subspaces (lines, planes, etc) become projective ones, and all of affine geometry can be transferred to projective space. However projective geometry is simpler than affine geometry because projective spaces are significantly more uniform than affine ones — there are far fewer special cases to consider. For example two distinct lines always meet exactly once in the projective plane, whereas in the affine plane they always meet *except* when they are parallel. Similarly, there are natural transformations that preserve projective structure (*i.e.* that map lines to lines, preserve intersections and so) that are quite complicated when expressed in affine space but very simple and natural in projective terms. The 3D→2D pinhole camera projection is one of these, hence the importance of projective geometry to computer vision.

B Factorization of the Fundamental Matrix

This appendix proves two claims made in section 3.

(1) Given the factorization $\mathbf{F}_{AA'} = \mathbf{u}_A\,\mathbf{v}_{A'} - \mathbf{v}_A\,\mathbf{u}_{A'}$*, the most general redefinition of the* $\mathbf{u}$*'s and* $\mathbf{v}$*'s that leaves* $\mathbf{F}$ *unchanged up to scale is*

$$\begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix} \longrightarrow \Lambda \begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix} \begin{pmatrix} 1/\lambda & 0 \\ 0 & 1/\lambda' \end{pmatrix}$$

where Λ *is an arbitrary nonsingular* 2×2 *matrix and* $\{\lambda, \lambda'\}$ *are arbitrary nonzero relative scale factors.*

Since $\mathbf{u}_A$ and $\mathbf{v}_A$ are independent epipolar lines and there is only a two parameter family of these, any other choice $\tilde{\mathbf{u}}_A, \tilde{\mathbf{v}}_A$ must be a nonsingular linear combination of these two, and similarly for $\mathbf{u}_{A'}$ and $\mathbf{v}_{A'}$. Hence the only possibilities are:

$$\begin{pmatrix} \mathbf{u}_A \\ \mathbf{v}_A \end{pmatrix} \longrightarrow \Lambda \begin{pmatrix} \mathbf{u}_A \\ \mathbf{v}_A \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{u}_{A'} \\ \mathbf{v}_{A'} \end{pmatrix} \longrightarrow \Lambda' \begin{pmatrix} \mathbf{u}_{A'} \\ \mathbf{v}_{A'} \end{pmatrix}$$

for nonsingular 2×2 matrices Λ and Λ'. Then

$$\begin{aligned} \mathbf{F}_{AA'} &= \begin{pmatrix} \mathbf{u}_A & \mathbf{v}_A \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{A'} \\ \mathbf{v}_{A'} \end{pmatrix} \\ &\rightarrow \begin{pmatrix} \mathbf{u}_A & \mathbf{v}_A \end{pmatrix} \Lambda^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \Lambda' \begin{pmatrix} \mathbf{u}_{A'} \\ \mathbf{v}_{A'} \end{pmatrix} \end{aligned}$$

Since the covectors $\mathbf{u}_A, \mathbf{v}_A$ and $\mathbf{u}_{A'}, \mathbf{v}_{A'}$ are independent, for $\mathbf{F}$ to remain unchanged up to scale we must have

$$\Lambda^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \Lambda' \sim \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Using the 2×2 matrix identity

$$\Lambda = -\mathrm{Det}\,(\Lambda) \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \Lambda^{-\top} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

we find that $\Lambda' \sim \Lambda$ up to scale. Defining λ'/λ to reflect the difference in scale, the result follows.

(2) Given any factorization $\mathbf{F}_{AA'} = \mathbf{u}_A\,\mathbf{v}_{A'} - \mathbf{v}_A\,\mathbf{u}_{A'}$ *defining a 4D subspace* $\mathcal{I}^\alpha$ *of* $\mathcal{H}^\alpha$ *via*

$$\begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix} \begin{pmatrix} \mathbf{x}^A \\ \mathbf{x}^{A'} \end{pmatrix} = \mathbf{0}$$

and any pair $\{\mathbf{P}_a^A, \mathbf{P}_a^{A'}\}$ *of rank 3 projection matrices with distinct centres of projection compatible with* $\mathbf{F}_{AA'}$ *in the sense that* $\mathbf{F}_{AA'}\,\mathbf{P}_a^A\mathbf{P}_b^{A'}\,\mathbf{x}^a\mathbf{x}^b = 0$ *for all* $\mathbf{x}^a \in \mathcal{H}^a$*, there is a fixed rescaling* $\{\lambda, \lambda'\}$ *that makes* $\mathcal{I}^\alpha$ *coincide with the image of* $\mathcal{H}^a$ *under the joint projection* $(\lambda\,\mathbf{P}_a^A \;\; \lambda'\,\mathbf{P}_a^{A'})^\top$.

If the compatibility condition holds for all $\mathbf{x}^a$, the symmetric part of the quadratic form $\mathbf{F}_{AA'}\,\mathbf{P}_a^A\mathbf{P}_b^{A'}$ must vanish. Expanding $\mathbf{F}$ and for clarity defining $\mathbf{u}_a \equiv \mathbf{u}_A\,\mathbf{P}_a^A$, $\mathbf{u}'_a \equiv \mathbf{u}_{A'}\,\mathbf{P}_a^{A'}$, $\mathbf{v}_a \equiv \mathbf{v}_A\,\mathbf{P}_a^A$, and $\mathbf{v}'_a \equiv \mathbf{v}_{A'}\,\mathbf{P}_a^{A'}$ we find:

$$\mathbf{u}_a\,\mathbf{v}'_b + \mathbf{v}'_a\,\mathbf{u}_b - \mathbf{v}_a\,\mathbf{u}'_b - \mathbf{u}'_a\,\mathbf{v}_b \;=\; \mathbf{0}$$

Since both projections have rank 3 none of the pulled back covectors $\mathbf{u}_a, \mathbf{u}'_a, \mathbf{v}_a, \mathbf{v}'_a$ vanish, and since the pairs $\mathbf{u}_A \not\sim \mathbf{v}_A$ and $\mathbf{u}_{A'} \not\sim \mathbf{v}_{A'}$ are independent, $\mathbf{u}_a \not\sim \mathbf{v}_a$ and $\mathbf{u}'_a \not\sim \mathbf{v}'_a$ are independent too. Contracting with any vector $\mathbf{x}^a$ orthogonal to both $\mathbf{u}_a$ and $\mathbf{u}'_a$ we find that

$$(\mathbf{v}'_a\,\mathbf{x}^a)\,\mathbf{u}_b - (\mathbf{v}_a\,\mathbf{x}^a)\,\mathbf{u}'_b \;=\; \mathbf{0}$$

Either there is some $\mathbf{x}^a$ for which one (and hence both) of the coefficients $\mathbf{v}_a\,\mathbf{x}^a$ and $\mathbf{v}'_a\,\mathbf{x}^a$ are nonzero — which implies that $\mathbf{u}_a \sim \mathbf{u}'_a$ — or both coefficients vanish for all such $\mathbf{x}^a$. But in this case we could conclude that $\mathbf{v}_a$ and $\mathbf{v}'_a$ were in $\mathrm{Span}(\mathbf{u}_a, \mathbf{u}'_a)$ and since $\mathbf{v}_a$ is independent of $\mathbf{u}_a$ and $\mathbf{v}'_a$ of $\mathbf{u}'_a$ that $\mathbf{v}_a \sim \mathbf{u}'_a$ and $\mathbf{v}'_a \sim \mathbf{u}_a$. Substituting back into $\mathbf{F}$ immediately shows that $\lambda\,\mathbf{u}_a\mathbf{u}_b - \lambda'\,\mathbf{v}_a\mathbf{v}_b = \mathbf{0}$ with nonzero λ and λ', and hence that $\mathbf{u}_a \sim \mathbf{v}_a$. So this branch is not possible and we can conclude that for some nonzero λ and λ', $\lambda\,\mathbf{u}_a + \lambda'\,\mathbf{u}'_a = \mathbf{0}$. Similarly, $\mu\,\mathbf{v}_a + \mu'\,\mathbf{v}'_a = \mathbf{0}$ for some nonzero μ and μ'. Substituting back into $\mathbf{F}$ gives $(\lambda/\lambda' - \mu/\mu')\,(\mathbf{u}_a\mathbf{v}_b + \mathbf{v}_a\mathbf{u}_b) = \mathbf{0}$, so up to scale $\{\mu, \mu'\} \sim \{\lambda, \lambda'\}$. The rescaling $\{\mathbf{P}_A^A,\ \mathbf{P}_A^{A'}\} \longrightarrow \{\lambda\,\mathbf{P}_A^A,\ \lambda'\,\mathbf{P}_A^{A'}\}$ then takes the projection of any $\mathbf{x}^a$ to a vector lying in $\mathcal{I}^\alpha$:

$$\begin{pmatrix} \mathbf{u}_A & \mathbf{u}_{A'} \\ \mathbf{v}_A & \mathbf{v}_{A'} \end{pmatrix} \begin{pmatrix} \lambda\,\mathbf{P}_a^A \\ \lambda'\,\mathbf{P}_a^{A'} \end{pmatrix} \mathbf{x}^a$$
$$= \begin{pmatrix} \lambda\,\mathbf{u}_a + \lambda'\,\mathbf{u}'_a \\ \lambda\,\mathbf{v}_a + \lambda'\,\mathbf{v}'_a \end{pmatrix} \mathbf{x}^a \;=\; \begin{pmatrix} 0 \\ 0 \end{pmatrix} \mathbf{x}^a \;=\; \mathbf{0}$$

References

[1] D. Adams. *The Hitchhikers Guide to the Galaxy*. Pan Books, 1979.

[2] S. Carlsson. Multiple image invariance using the double algebra. In J. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[3] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[4] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

[5] O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[6] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[7] R. Hartley. Lines and points in three views – an integrated approach. In *Image Understanding Workshop*, Monterey, California, November 1994.

[8] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 761–4, Urbana-Champaign, Illinois, 1992.

[9] W. V. D. Hodge and D. Pedoe. *Methods of Algebraic Geometry*, volume 1. Cambridge University Press, 1947.

[10] S. Laveau and O. Faugeras. 3d scene representation as a collection of images and fundamental matrices. Technical Report RR-2205, INRIA, Sophia Antipolis, France, February 1994.

[11] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–5, 1981.

[12] Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimental results. Technical Report RR-1894, INRIA, Sophia Antipolis, France, 1993.

[13] Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. Technical Report UCB/CSD-93-772, Dept. EECS, Berkeley, California, 1993.

[14] S. J. Maybank. *Theory of Reconstruction from Image Motion*, volume 28 of *Series in Information Sciences*. Springer-Verlag, 1993.

[15] S. J. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. J. Computer Vision*, 8(2):123–151, 1992.

[16] P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 314–20, Cambridge, MA, June 1995.

[17] R. Mohr, B. Boufama, and P. Brand. Accurate relative positioning from multiple images. Technical Report RT-102, LIFIA, INRIA Rhône-Alpes, Grenoble, France, 1993. Submitted to *Journal of Artificial Intelligence*.

[18] R. Penrose and W. Rindler. *Spinors and space-time. Vol. 1, Two-spinor calculus and relativistic fields*. Cambridge University Press, 1984.

[19] A. Shashua. Projective structure from uncalibrated images: Structure from motion and recognition. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 16(8), 1994.

[20] B. Triggs. Least squares estimation in projective spaces. To appear, 1995.

[21] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.

[22] M. Werman and A. Shashua. The study of 3D-from-2D using elimination. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 473–9, Cambridge, MA, June 1995.

Optimal Estimation of Matching Constraints

Bill Triggs

INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot St. Martin, France
Bill.Triggs@inrialpes.fr — http://www.inrialpes.fr/movi/people/Triggs

Abstract

We describe work in progress on a numerical library for estimating multi-image matching constraints, or more precisely the multi-camera geometry underlying them. The library will cover several variants of homographic, epipolar, and trifocal constraints, using various different feature types. It is designed to be modular and open-ended, so that (*i*) new feature types or error models, (*ii*) new constraint types or parametrizations, and (*iii*) new numerical resolution methods, are relatively easy to add. The ultimate goal is to provide practical code for stable, reliable, statistically optimal estimation of matching geometry under a choice of robust error models, taking full account of any nonlinear constraints involved. More immediately, the library will be used to study the relative performance of the various competing problem parametrizations, error models and numerical methods. The paper focuses on the overall design, parametrization and numerical optimization issues. The methods described extend to many other geometric estimation problems in vision, *e.g.* curve and surface fitting.
Keywords: Matching constraints, multi-camera geometry, geometric fitting, statistical estimation, constrained optimization.

1 Introduction and Motivation

This paper describes work in progress on a numerical library for the estimation of multi-image matching constraints. The library will cover several variants of homographic, epipolar, and trifocal constraints, using various common feature types. It is designed to be modular and open-ended, so that new feature types or error models, new constraint types or parametrizations, and new numerical resolution methods are relatively easy to add. The ultimate goal is to provide practical code for stable, reliable, statistically optimal estimation of matching geometry under a choice of robust error models, taking full account of any nonlinear constraints involved. More immediately, the library is being used to study the relative performance of the various competing problem parametrizations, error models and numerical methods. Key questions include: (*i*) how much difference does an *accurate statistical error model* make; (*ii*) which *constraint parametrizations*, *initialization methods* and *numerical optimization schemes* offer the best reliability/speed/simplicity. The answers are most interesting for *near-degenerate* problems, as these are the most difficult to handle reliably. This paper focuses on architectural, parametrization and numerical optimization issues. I have tried to give an overview of the relevant choices and technology, rather than going into too much detail on any one subject. The methods described extend to many other geometric estimation problems, such as curve and surface fitting.

After motivating the library and giving notation in this section, we develop a general statistical framework for geometric fitting in §2 and discuss parametrization issues in §3. §4 summarizes the library architecture and numerical techniques, §5 discusses experimental testing, and §6 concludes.

This paper appeared in SMILE'98, European Workshop on 3D Structure from Multiple Images of Large-scale Environments, Springer LNCS, 1998. The work was supported by Esprit LTR project CUMULI.

Why study matching constraint estimation? — Practically, matching constraints are central to both feature grouping and 3D reconstruction, so better algorithms should immediately benefit many geometric vision applications. But there are many variations to implement, depending on the feature type, number of images, image projection model, camera calibration, and camera and scene geometry. So a systematic approach seems more appropriate than an *ad hoc* case-by-case one. Matching constraints also have a rather delicate algebraic structure which makes them difficult to estimate accurately. Many common camera and scene geometries correspond to degenerate cases whose special properties need to be detected and exploited for stability. Even in stable cases it is not yet clear how best to parametrize the constraints — usually, they belong to fairly complicated algebraic varieties and redundant or constrained parametrizations are required. Some numerical sophistication is needed to implement these efficiently, and the advantages of different models and parametrizations need to be studied experimentally: the library is a vehicle for this.

It is also becoming clear that in many cases no single model suffices. One should rather think in terms of a continuum of nested models linked by specialization/generalization relations. For example, rather than simply assuming a generic fundamental matrix, one should use inter-image homographies for small camera motions or large flat scenes, affine fundamental matrices for small, distant objects, essential matrices for constant intrinsic parameters, fundamental matrices for wide views of large close objects, lens distortion corrections for real images, *etc*. Ideally, the model should be chosen to maximize the statistically expected end-to-end system performance, given the observed input data. Although there are many specific decision criteria (ML, AIC, BIC, ...), the key issue is always the *bias* of over-restrictive models versus the *variability* of over-general ones with superfluous parameters poorly controlled by the data. Any model selection approach requires several models to be fitted so that the best can be chosen. Some of the models must always be inappropriate — either biased or highly variable — so fast, reliable, accurate fitting in difficult cases is indispensable for practical model selection.

Terminology and notation: We use homogeneous coordinates throughout, with upright bold for 3D quantities and italic bold for image ones. Image projections are described by 3×4 perspective ***projection matrices*** $\boldsymbol{P}$, with specialized forms for calibrated or very distant cameras. Given m images of a static scene, our goal is to recover as much information as possible about the camera calibrations and poses, using only image measurements. We will call the recoverable information the ***inter-image geometry*** to emphasize that no explicit 3D structure is involved. The ensemble of projection matrices is defined only up to a 3D coordinate transformation (projectivity or similarity) $\mathbf{T}$: $(\boldsymbol{P}_1, \dots, \boldsymbol{P}_m) \rightarrow (\boldsymbol{P}_1\mathbf{T}, \dots, \boldsymbol{P}_m\mathbf{T})$. We call such coordinate freedoms ***gauge freedoms***. So our first representation of the inter-image geometry is as ***projection matrices modulo a transformation group***. In the uncalibrated case this gives an $11m$ parameter representation with 15 gauge freedoms, leaving $11m - 15$ essential d.o.f. ($= 7, 18, 29$ for $m = 2, 3, 4$). In the calibrated case there are $6m - 7$ essential degrees of freedom.

Any set of four (perhaps not distinct) projection matrices can be combined to form a ***matching tensor*** [14, 5] — a multi-image object independent of the 3D coordinates. The possible types are: ***epipoles*** $\boldsymbol{e}_i^j$; 3×3 ***fundamental matrices*** $\boldsymbol{F}_{ij}$; $3 \times 3 \times 3$ ***trifocal tensors*** $\boldsymbol{G}_i^{jk}$; and $3 \times 3 \times 3 \times 3$ ***quadrifocal tensors*** $\boldsymbol{H}^{ijkl}$. Their key property is that they are the coefficients of inter-image ***matching constraints*** — the consistency relations linking corresponding features in different images. *E.g.*, for images $\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}''$ of a 3D point we have the 2-image ***epipolar constraint*** $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{F}\,\boldsymbol{x}' = 0$; the 3-image ***trinocular constraint*** which can be written symbolically as $[\,\boldsymbol{x}'\,]_\times(\boldsymbol{G} \cdot \boldsymbol{x})\,[\,\boldsymbol{x}''\,]_\times = \boldsymbol{0}$ where $[\,\boldsymbol{x}\,]_\times$ is the matrix generating the cross product $[\,\boldsymbol{x}\,]_\times\boldsymbol{y} \equiv \boldsymbol{x} \wedge \boldsymbol{y}$; and a 4-image ***quadrinocular constraint***. The matching tensors also characterize the inter-image geometry. This is attractive because they are intimately connected to the image measurements — it is much easier to get linearized initial estimates of matching tensors than of projection matrices. Unfortunately, this linearity is deceptive. Matching tensors are not really linear objects: they only

represent a valid, realizable inter-image geometry if they satisfy a set of nonlinear algebraic ***consistency constraints***. These rapidly become intractable beyond 2–3 images, and are still only partially understood [4, 14, 5, 9, 6]. Our second parametrization of the inter-image geometry is as *matching tensors subject to consistency constraints*.

We emphasize that camera matrices or matching tensors are only a means to an end: it is the underlying inter-image geometry that we are really trying to estimate. Unfortunately, this is abstract and somewhat difficult to pin down because it is a ***nontrivial algebraic variety*** — there *are* no simple, minimal, global parametrizations.

2 Optimal Geometric Fitting

2.1 Direct Approach

Matching constraint estimation is an instance of an ***abstract geometric fitting problem*** which also includes curve and surface fitting and many other geometric estimation problems: estimate the parameters of a model $\mathbf{u}$ defining implicit constraints $\mathbf{c}_i(\mathbf{x}_i, \mathbf{u}) = \mathbf{0}$ on underlying features $\mathbf{x}_i$, from noisy measurements of the features. More specifically we assume:

1. There are unknown ***true underlying features*** $\overline{\mathbf{x}}_i$ and an unknown ***true underlying model*** $\overline{\mathbf{u}}$ which exactly satisfy implicit ***model-feature consistency constraints*** $\mathbf{c}_i(\overline{\mathbf{x}}_i, \overline{\mathbf{u}}) = \mathbf{0}$. (For matching constraint estimation, these 'features' are actually ensembles of several corresponding image ones).

2. Each underlying feature $\overline{\mathbf{x}}_i$ is linked to observations $\underline{\mathbf{x}}_i$ or other prior information by an additive ***posterior statistical error measure*** $\rho_i(\mathbf{x}_i) \equiv \rho_i(\mathbf{x}_i \,|\, \underline{\mathbf{x}}_i)$. For example, ρ_i might be (robustified, bias corrected) ***posterior log likelihood***. There may also be a ***model prior*** $\rho_{\text{prior}}(\mathbf{u})$. These distributions are independent.

3. The model parametrization $\mathbf{u}$ may itself be complex, *e.g.* with internal constraints $\mathbf{k}(\mathbf{u}) = \mathbf{0}$, gauge freedoms, *etc*.

4. We want to find ***optimal consistent point estimates*** $(\hat{\mathbf{x}}_i, \hat{\mathbf{u}})$ of the true underlying model $\overline{\mathbf{u}}$ and features $\overline{\mathbf{x}}_i$

$$(\hat{\mathbf{x}}_i, \ldots, \hat{\mathbf{u}}) \equiv \arg\min \left(\rho_{\text{prior}}(\mathbf{u}) + \sum_i \rho_i(\mathbf{x}_i | \underline{\mathbf{x}}_i) \;\middle|\; \mathbf{c}_i(\mathbf{x}_i, \mathbf{u}) = \mathbf{0},\ \mathbf{k}(\mathbf{u}) = \mathbf{0} \right)$$

   Consistent means that $(\hat{\mathbf{x}}_i, \hat{\mathbf{u}})$ exactly satisfy all the constraints. ***Optimal*** means that they minimize the total error over all such estimates. ***Point estimate*** means that we are attempting to "summarize" the joint posterior distribution $\rho(\mathbf{x}_i, \ldots, \mathbf{u} | \underline{\mathbf{x}}_i, \ldots)$ with just the few numbers $(\hat{\mathbf{x}}_i, \ldots, \hat{\mathbf{u}})$.

We call this the ***direct approach*** to geometric fitting because it involves direct numerical optimization over the "natural" variables $(\mathbf{x}_i, \mathbf{u})$. Its most important characteristics are: (*i*) It gives exact, optimal results — no approximations are involved. (*ii*) It produces optimal consistent estimates $\hat{\mathbf{x}}_i$ of the underlying features $\overline{\mathbf{x}}_i$. These are useful whenever the measurements need to be made coherent with the model. For matching constraint estimation such feature estimates are "pre-triangulated" or "implicitly reconstructed" in that they have already been made exactly consistent with exactly one reconstructed 3D feature. (*iii*) Natural variables are used and the error function is relatively simple, typically just a sum of (robustified, covariance weighted) squared deviations $\|\mathbf{x}_i - \underline{\mathbf{x}}_i\|^2$. (*iv*) However, a sparse constrained nonlinear optimization routine is required: the problem is large, constrained and usually nonlinear, but the features couple only to the model, not to each other.

As an example, for the uncalibrated epipolar geometry: the "features" are pairs of corresponding underlying image points $(\boldsymbol{x}_i, \boldsymbol{x}_i')$; the "model" $\mathbf{u}$ is the fundamental matrix $\boldsymbol{F}$ subject to the consistency constraint $\det(\boldsymbol{F}) = 0$; the "model-feature constraints" are the epipolar constraints $\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{F}\, \boldsymbol{x}_i' = 0$; and the "feature error model" $\rho_i(\mathbf{x}_i)$ might be (a robustified, covariance-weighted variant of) the squared feature-observation distance $\|\boldsymbol{x} - \underline{\boldsymbol{x}}\|^2 + \|\boldsymbol{x}' - \underline{\boldsymbol{x}}'\|^2$.

2.2 Reduced Approach

If explicit estimates of the underlying features are not required, one can attempt to replace step 4 above with an optimization over $\mathbf{u}$ alone:

$4'$. Find an ***optimal consistent point estimate*** $\hat{\mathbf{u}}$ of the true underlying model $\overline{\mathbf{u}}$

$$\hat{\mathbf{u}} \;\equiv\; \arg\min \left(\rho_{\text{prior}}(\mathbf{u}) + \sum_i \rho_i(\mathbf{u}|\underline{\mathbf{x}}_i) \;\middle|\; \mathbf{k}(\mathbf{u}) = \mathbf{0} \right)$$

Here, the ***reduced error functions*** $\rho_i(\mathbf{u}|\underline{\mathbf{x}}_i)$ are obtained by freezing $\mathbf{u}$ and eliminating the unknown features from the problem using either: (*i*) ***point estimates***

$$\mathbf{x}_i(\underline{\mathbf{x}}_i, \mathbf{u}) \;\equiv\; \arg\min \left(\rho_i(\mathbf{x}_i|\underline{\mathbf{x}}_i) \mid \mathbf{c}_i(\mathbf{x}_i, \mathbf{u}) = \mathbf{0} \right)$$

of $\mathbf{x}_i$ given $\underline{\mathbf{x}}_i$ and $\mathbf{u}$, with $\rho_i(\mathbf{u}|\underline{\mathbf{x}}_i) \equiv \rho_i(\mathbf{x}_i(\underline{\mathbf{x}}_i, \mathbf{u})|\underline{\mathbf{x}}_i)$; or (*ii*) ***marginalization*** with respect to $\mathbf{x}_i$: $\rho_i(\mathbf{u}|\underline{\mathbf{x}}_i) \equiv \int_{\mathbf{c}_i(\mathbf{x}_i,\mathbf{u})=0} \rho_i(\mathbf{x}_i|\underline{\mathbf{x}}_i)\, \mathbf{d}\mathbf{x}_i$. These two methods are not equivalent in general, although their answers happen to agree in the linear/Gaussian limit. But both represent reasonable estimation techniques.

We call this the ***reduced approach*** to geometric fitting, because the problem is ***reduced*** to one involving only the model parameters $\mathbf{u}$. The main advantage is that the optimization is over relatively few variables $\mathbf{u}$. The constraints $\mathbf{c}_i$ do not appear, so a non-sparse and (perhaps) unconstrained optimization routine can be used. The disadvantage is that the reduced cost $\rho(\mathbf{u})$ is seldom available in closed form. Usually, it can only be evaluated to first order in a linearized + central distribution approximation. In fact, the direct method (with $\mathbf{u}$ frozen, and perhaps limited to a single iteration) is often the easiest way to evaluate the point-estimate-based reduced cost. The only real difference is that the direct method explicitly calculates and applies feature updates $\mathbf{d}\mathbf{x}_i$, while the reduced method restarts each time from $\mathbf{x}_i \equiv \underline{\mathbf{x}}_i$. But the feature updates are relatively easy to calculate given the factorizations needed for cost evaluation, so it seems a pity not to use them.

The first order reduced cost can be estimated in two ways, either (*i*) directly from the definition by projecting $\underline{\mathbf{x}}_i$ Mahalanobis-orthogonally onto the local first-order constraint surface $\mathbf{c}_i + \frac{\mathbf{d}\mathbf{c}_i}{\mathbf{d}\mathbf{x}_i} \cdot \mathbf{d}\mathbf{x}_i = \mathbf{0}$; or (*ii*) by treating $\mathbf{c}_i \equiv \mathbf{c}_i(\underline{\mathbf{x}}_i, \mathbf{u})$ as a random variable, using covariance propagation w.r.t. $\underline{\mathbf{x}}_i$ to find its covariance, and calculating the χ^2-like variable $\mathbf{c}_i^{\mathrm{T}} \mathrm{Cov}(\mathbf{c}_i)^{-1} \mathbf{c}_i$. In either case we obtain the ***gradient weighted least squares*** cost function[1] [13]

$$\rho(\mathbf{u}) \;=\; \sum_i \mathbf{c}_i^{\mathrm{T}} \left(\frac{\mathbf{d}\mathbf{c}_i}{\mathbf{d}\mathbf{x}_i} \left(\frac{\mathbf{d}^2 \rho_i}{\mathbf{d}\mathbf{x}_i^2} \right)^{-1} \frac{\mathbf{d}\mathbf{c}_i}{\mathbf{d}\mathbf{x}_i}^{\mathrm{T}} \right)^{-1} \mathbf{c}_i \;\Bigg|_{(\underline{\mathbf{x}}_i, \mathbf{u})}$$

This is simplest for problems with scalar constraints. *E.g.* for the uncalibrated epipolar constraint we get the well-known form [10]

$$\rho(\mathbf{u}) \;=\; \sum_i \frac{(\underline{\boldsymbol{x}}_i^{\mathrm{T}} \boldsymbol{F}\, \underline{\boldsymbol{x}}_i')^2}{\underline{\boldsymbol{x}}_i^{\mathrm{T}} \boldsymbol{F}\, \mathrm{Cov}(\underline{\boldsymbol{x}}_i')\, \boldsymbol{F}^{\mathrm{T}}\, \underline{\boldsymbol{x}}_i + \underline{\boldsymbol{x}}_i'^{\mathrm{T}} \boldsymbol{F}^{\mathrm{T}}\, \mathrm{Cov}(\underline{\boldsymbol{x}}_i)\, \boldsymbol{F}\, \underline{\boldsymbol{x}}_i'}$$

[1] If any of the covariance matrices is singular (which happens for redundant constraints or homogeneous data $\mathbf{x}_i$), the matrix inverses can be replaced with pseudo-inverses.

2.3 Robustification — Total Distribution Approach

Outliers are omnipresent in vision data and it is essential to protect against them. In general, they are distinguished only by their failure to agree with the consensus established by the inliers, so one should really think in terms of *inlier* or *coherence* detection. The hardest part is establishing a reliable initial estimate, *i.e.* the combinatorial problem of finding enough inliers to estimate the model, without being able to tell in advance that they *are* inliers. Exhaustive enumeration is usually impracticable, so one falls back on either RANSAC-like random sampling or (in low dimensions) Hough-like voting. Initialization from an outlier-polluted linear estimate is seldom completely reliable.

Among the many approaches to robustness, I prefer M-like estimators and particularly the ***total distribution*** approach: hypothesize a parametric form for the ***total observation distribution*** — *i.e.* including *both* inliers *and* outliers — and fit this to the data using some standard criterion, *e.g.* maximum likelihood. No explicit inlier/outlier decision is needed: the correct model is located simply because it provides an explanation more probable than randomness for the coherence of the inliers[2]. The total approach is really just classical parametric statistics with a more realistic or "robust" choice of parametric family. Any required distribution parameters can in principle be estimated during fitting (*e.g.* covariances, outlier densities). For centrally peaked mixtures one can view the total distribution as a kind of M-estimator, although it long predates these and gives a much clearer meaning to the rather arbitrary functional forms usually adopted for them. As with other M-like-estimators, the estimation problem is nonlinear and numerical optimization is required. With this approach, both of the above geometric fitting methods are 'naturally' robust — we just need to use an appropriate total likelihood.

Reasons for preferring M-like estimators over trimmed ones like RANSAC's consensus and rank-based ones like least median squares include: (*i*) to the extent that the total distribution is realistic, the total approach is actually the statistically optimal one; (*ii*) only M-like cost functions are smooth and hence easy to optimize; (*iii*) the 'soft' transitions of M-like estimators allow better use of weak 'near outlier' data, *e.g.* points which are relatively uncertain owing to feature extraction problems, or "false outliers" caused by misestimated covariances or a skewed, biased, or badly initialized model; (*iv*) including an explicit covariance scale makes the results more reliable and increases the *expected* breakdown point — 'scale free' rank based estimators can not tell whether the measurements they are including are "plausible" or not; (*v*) all of these estimators assume an underlying ranking of errors 'by relative size', and none are robust against mismodelling of this — rank based estimators only add a little extra robustness against the likelihood *vs.* error size assignment.

3 Parametrizing the Inter-image Geometry

As discussed above, what we are really trying to estimate is the ***inter-image geometry*** — the part of the multi-camera calibration and pose that is recoverable from image measurements alone. However, this is described by a nontrivial algebraic variety — it has *no* simple, minimal, concrete, global parametrization. For example, the uncalibrated epipolar geometry is "the variety of all homographic mappings between line pencils in the plane", but it is unclear how best to parametrize this. We will consider three general parametrization strategies for algebraic varieties: (*i*) redundant parametrizations with internal gauge freedoms; (*ii*) redundant parametrizations with internal constraints; (*iii*) overlapping local coordinate patches. *Mathematically* these are all equivalent — they only differ in relative convenience and numerical properties. Different methods are convenient for

[2] If the total distribution happens to be an inlier/outlier *mixture* — *e.g.* Gaussian peak + uniform background — posterior inlier/outlier probabilities are easily extracted as a side effect.

different uses, so it is important to be able to convert between them. Even the numerical differences are slight for strong geometries and careful implementations, but for weak geometries there can be significant differences.

3.1 Redundant Parametrizations with Gauge Freedom

In many geometric problems, ***arbitrary choices of coordinates*** are required to reduce the problem to a concrete algebraic form. Such choices are called ***gauge freedoms*** — 'gauge' just means coordinate system. They are associated with an internal ***symmetry*** or ***coordinate transformation*** group and its representations. Formulae expressed in gauged coordinates reflect the symmetry by obeying well-defined transformation rules under changes of coordinates, *i.e.* by belonging to well-defined group representations. 3D Cartesian coordinates are a familiar example: the gauge group is the group of rigid motions, and the representations are (roughly speaking) Cartesian tensors.

Common gauge freedoms include: (*i*) 3D projective or Euclidean coordinate freedoms in reconstruction and projection-matrix-based camera parametrizations; (*ii*) arbitrary homogeneous-projective scale factors; and (*iii*) choice-of-plane freedoms in ***homographic parametrizations*** of the inter-image geometry. These latter represent matching tensors as products of epipoles and inter-image homographies induced by an arbitrary 3D plane. The gauge freedom is the 3 d.o.f. choice of plane. The fundamental matrix can be written as $\boldsymbol{F} \simeq [\,\boldsymbol{e}\,]_\times \boldsymbol{H}$ where $\boldsymbol{e}$ is the epipole and $\boldsymbol{H}$ is any inter-image homography [11, 3]. Redefining the 3D plane changes $\boldsymbol{H}$ to $\boldsymbol{H} + \boldsymbol{e}\,\boldsymbol{a}^{\mathrm{T}}$ for some image line 3-vector $\boldsymbol{a}$. This leaves $\boldsymbol{F}$ unchanged, as do rescalings $\boldsymbol{e} \to \lambda\boldsymbol{e}$, $\boldsymbol{H} \to \mu\boldsymbol{H}$. So there are $3+1+1$ gauge freedoms in the $3+3\times 3 = 12$ variable parametrization $\boldsymbol{F} \simeq \boldsymbol{F}(\boldsymbol{e}, \boldsymbol{H})$, leaving the correct $12-5=7$ degrees of freedom of the uncalibrated epipolar geometry. Similarly [8], the image $(1,2,3)$ trifocal tensor $\boldsymbol{G}$ can be written in terms of the epipoles $(\boldsymbol{e}', \boldsymbol{e}'')$ and inter-image homographies $(\boldsymbol{H}', \boldsymbol{H}'')$ of image 1 in images 2 and 3

$$\boldsymbol{G} \;\simeq\; \boldsymbol{e}' \otimes \boldsymbol{H}'' - \boldsymbol{H}' \otimes \boldsymbol{e}'' \quad \text{with freedom} \quad \begin{pmatrix} \boldsymbol{H}' \\ \boldsymbol{H}'' \end{pmatrix} \;\to\; \begin{pmatrix} \boldsymbol{H}' \\ \boldsymbol{H}'' \end{pmatrix} + \begin{pmatrix} \boldsymbol{e}' \\ \boldsymbol{e}'' \end{pmatrix} \boldsymbol{a}^{\mathrm{T}}$$

The gauge freedom corresponds to the choice of 3D plane and 3 scale d.o.f. — the relative scaling of $(\boldsymbol{e}', \boldsymbol{H}')$ *vs.* $(\boldsymbol{e}'', \boldsymbol{H}'')$ being significant — so the 18 d.o.f. of the uncalibrated trifocal geometry are parametrized by $3+3+9+9 = 24$ parameters modulo $3+1+1+1 = 6$ gauge freedoms. For calibrated cameras it is useful to place the 3D plane at infinity so that the resulting absolute homographies are represented by 3×3 rotation matrices. This gives well-known 6 and 12 parameter representations of the calibrated epipolar and trifocal geometries, each with just one redundant scale d.o.f.: $\boldsymbol{E} \simeq [\,\boldsymbol{e}\,]_\times \boldsymbol{R}$, $\boldsymbol{G} \simeq \boldsymbol{e}' \otimes \boldsymbol{R}'' - \boldsymbol{R}' \otimes \boldsymbol{e}''$. All of these homography + epipole parametrizations can also be viewed as projection matrix based ones, in a 3D frame where the first projection takes the form $(\boldsymbol{I}_{3\times 3} | \boldsymbol{0})$. The plane position freedom $\boldsymbol{a}$ corresponds to the 3 remaining d.o.f. of the 3D projective frame [8]. These methods seem to be a good compromise: compared to 'free' projections, they reduce the number of extraneous d.o.f. from 15 to 3. However their numerical stability does depend on that of the key image.

Gauged parametrizations have the following advantages: (*i*) they are very natural when the inter-image geometry is derived from the 3D one; (*ii*) they are close to the underlying geometry, so it is relatively easy to derive further properties from them (projection matrices, reconstruction methods, matching tensors); (*iii*) a single homogeneous coordinate system covers the whole variety; (*iv*) they are numerically fairly stable. Their main disadvantage is that they include extraneous, strictly irrelevant degrees of freedom which have no effect at all on the residual error. Hence, gauged Jacobians are exactly rank deficient: specially stabilized numerical methods are needed to handle them. The additional variables and stabilization also tend to make gauged parametrizations slow.

3.2 Constrained Parametrizations

Another way to define a variety is in terms of ***consistency constraints*** that "cut the variety out of" a larger, usually linear space. Any coordinate system in the larger space then parametrizes the variety, but this is an over-parametrization subject to nonlinear constraints. Points which fail to satisfy the constraints have no meaning in terms of the variety. ***Matching tensors*** are the most familiar example. In the 2- and 3-image cases a single fundamental matrix or trifocal tensor suffices to characterize the inter-image geometry. But this is a linear over-parametrization, subject to the tensor's nonlinear consistency constraints — only so is a coherent, realizable inter-image geometry represented. Such parametrizations are valuable because they are close to the image data, and (inconsistent!) linear initial estimates of the tensors are easy to obtain. Their main disadvantages are: (*i*) the consistency conditions rapidly become complicated and non-obvious; (*ii*) the representation is only implicit — it is not immediately obvious how to go from the tensor to other properties of the geometry such as projection matrices. The first problem is serious and puts severe limitations on the use of (ensembles of) matching tensors to represent camera geometries, even in transfer-type applications where explicit projection matrices are not required. Three images seems to be about the practical limit if a guaranteed-consistent geometry is required, although — at the peril of a build-up of rounding error — one can chain together a series of such three image solutions [12, 15, 1].

For the fundamental matrix the codimension is 1 and the consistency constraint is $\det(\boldsymbol{F}) = 0$ — this is perhaps the simplest of all representations of the uncalibrated epipolar geometry. For the essential matrix $\boldsymbol{E}$ the codimension is 3, spanned either by the requirement that $\boldsymbol{E}$ should have two equal (which counts for 2) and one zero singular values, or by a local choice of 3 of the 9 Demazure constraints $(\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}} - \frac{1}{2}\mathrm{trace}(\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}))\ \boldsymbol{E} = \boldsymbol{0}$ [4]. For the uncalibrated trifocal tensor $\boldsymbol{G}$ we locally need $26 - 18 = 8$ linearly independent constraints. Locally (only!) these can be spanned by the 10 determinantal constraints $\frac{\mathbf{d}^3}{\mathbf{dx}^3}\det(\boldsymbol{G}\cdot\mathbf{x}) = \boldsymbol{0}$ — see [6] for several global sets. For the quadrifocal tensor $\boldsymbol{H}$ the codimension is $80 - 29 = 51$ which is locally (but almost certainly not globally) spanned by the $3! \cdot 3 \cdot 3 = 54$ determinantal constraints $\det_{ij}(\boldsymbol{H}^{ijkl}) = \boldsymbol{0}$ + permutations.

Note that the redundancy and complexity of the matching tensor representation rises rapidly as more images or calibration constraints are added. Also, ***constraint redundancy*** is common. Many algebraic varieties require a number of generators greater than their codimension. Intersections of the minimal number of polynomials *locally* give the correct variety, but typically have other, unwanted components elsewhere in the space. Extra polynomials must be included to suppress these, and it rapidly becomes difficult to say which sets of polynomials are globally sufficient.

3.3 Local Coordinate Patches / Minimal Parametrizations

Both gauged and constrained parametrizations are redundant and require specialized numerical methods. Why not simplify life by using a ***minimal set of independent parameters***? — The basic problem is that no such parametrization can cover the whole of a topologically nontrivial variety without singularities. Minimal parametrizations are intrinsically *local*: to cover the whole variety we need several such partially overlapping 'local coordinate patches', and also code to select the appropriate patch and manage any inter-patch transitions that occur. This can greatly complicate the optimization loop.

Also, although infinitely many local parametrizations exist, they are not usually very 'natural' and finding one with good properties may not be easy. Basically, starting from some 'natural' redundant representation, we must either come up with some inspired nonlinear change of variables which locally removes the redundancy, or algebraically eliminate variables by brute force using consistency or gauge fixing constraints. For example, Luong *et al* [10] guarantee $\det(\boldsymbol{F}) = 0$ by writing each row of the fundamental matrix as a linear combination of the other two. Each parametrization fails when its two rows are linearly dependent, but the three of them suffice to

cover the whole variety. In more complicated situations, intuition fails and we have to fall back on algebraic elimination, which rapidly leads to intractable results. Elimination-based parametrizations are usually highly anisotropic: they do not respect the symmetries of the underlying geometry. This tends to mean that they are messy to implement, and numerically ill-behaved, particularly near the patch boundaries.

The above comments apply only to *algebraically* derived parametrizations. Many of the numerical techniques for gauged or constrained problems eliminate redundant variables *numerically* to first order, using the constraint Jacobians. Such local parametrizations are much better behaved because they are always used at the centre of their valid region, and because stabilizing techniques like pivoting can be used. *It is usually preferable to eliminate variables locally and numerically rather than algebraically.*

4 Library Architecture and Numerical Methods

The library is designed to be modular so that different problems and approaches are easy to implement and compare. We separate: (*i*) the matching geometry type and parametrization; (*ii*) each contributing feature-group type, parametrization and error model; (*iii*) the numerical optimization method, and its associated linear algebra; (*iv*) the search controller (step acceptance and damping, convergence tests). This decomposition puts some constraints on the types of algorithms that can be implemented, but these do not seem to be too severe in practice. Modularization also greatly simplifies the implementation.

Perhaps the most important assumption is the adoption throughout of a "square root" or normalized residual vector based framework, and the associated use of Gauss-Newton techniques. ***Normalized residual vectors*** are quantities $\mathbf{e}_i$ for which the squared norm $\|\mathbf{e}_i\|^2$ — or more generally a robust, nonlinear function $\rho_i(\|\mathbf{e}_i\|^2)$ — is a meaningful statistical error measure. *E.g.* $\mathbf{e}_i(\mathbf{x}_i) \equiv \text{Cov}(\underline{\mathbf{x}}_i)^{-\frac{1}{2}}(\mathbf{x}_i - \underline{\mathbf{x}}_i)$. This allows a nonlinear-least-squares-like approach. Whenever possible, we work directly with the residual $\mathbf{e}$ and its Jacobian $\frac{\mathbf{de}}{\mathbf{dx}}$ rather than with $\|\mathbf{e}\|^2$, its gradient $\frac{\mathbf{d}(\|\mathbf{e}\|^2)}{\mathbf{dx}} = \mathbf{e}^{\mathrm{T}}\frac{\mathbf{de}}{\mathbf{dx}}$ and its Hessian $\frac{\mathbf{d}^2(\|\mathbf{e}\|^2)}{\mathbf{dx}^2} = \mathbf{e}^{\mathrm{T}}\frac{\mathbf{d}^2\mathbf{e}}{\mathbf{dx}^2} + \frac{\mathbf{de}}{\mathbf{dx}}^{\mathrm{T}}\frac{\mathbf{de}}{\mathbf{dx}}$. We use the ***Gauss-Newton approximation***, *i.e.* we discard the second derivative term $\mathbf{e}^{\mathrm{T}}\frac{\mathbf{d}^2\mathbf{e}}{\mathbf{dx}^2}$ in the Hessian. This buys us simplicity (no second derivatives are needed) and also numerical stability because we can use stable ***linear least squares*** methods for step prediction: by default we use ***QR decomposition with column pivoting*** of $\frac{\mathbf{de}}{\mathbf{dx}}$, rather than Cholesky decomposition of the normal matrix $\frac{\mathbf{de}}{\mathbf{dx}}^{\mathrm{T}}\frac{\mathbf{de}}{\mathbf{dx}}$. This is potentially slightly slower, but for ill-conditioned Jacobians it has much better resistance to rounding error. (The default implementation is intended for use as a reference, so it is deliberately rather conservative). The main disadvantage of Gauss-Newton is that convergence may be slow if the problem has both *large residual* and *strong nonlinearity* — *i.e.* if the ignored Hessian term $\mathbf{e}^{\mathrm{T}}\frac{\mathbf{d}^2\mathbf{e}}{\mathbf{dx}^2}$ is large. However, *geometric vision problems usually have small residuals* — the noise is usually much smaller than the scale of the geometric nonlinearities.

4.1 Numerical Methods for Gauge Freedom

The basic numerical difficulty with gauge freedom is that because gauge motions represent exact redundancies that have no effect at all on the residual error, in a classical optimization framework there is nothing to say what they should be: the error gradient and Hessian in a gauge direction both vanish, so the Newton step is undefined. If left undamped, this leads to ***large gauge fluctuations*** which can destabilize the rest of the system, prevent convergence tests from operating, *etc*. There are two ways around this problem:

1. Gauge fixing conditions break the degeneracy by adding ***artificial constraints***. Unless we are

clever enough to choose constraints that eliminate variables in closed form, this reduces the problem to constrained optimization. The constraints are necessarily non-gauge-invariant, *i.e.* non-tensorial under the gauge group. For example, to fix the 3D projective coordinate freedom, Hartley [8] sets $\boldsymbol{P}_1 \equiv (\boldsymbol{I}_{3\times3}|\boldsymbol{0})$ and $\sum_i \boldsymbol{e}^i \boldsymbol{H}^i_j = 0$ where $\boldsymbol{P}_2 = (\boldsymbol{H}|\boldsymbol{e})$. Neither of these constraints is tensorial — the results depend on the chosen image coordinates.

2. Free gauge methods — like photogrammetric ***free bundle*** ones — leave the gauge free to drift, but ensure that it does not move too far at each step. Typically, it is also monitored and reset "by hand" when necessary to ensure good conditioning. The basic tools are ***rank deficient least squares*** methods (*e.g.* [2]). These embody some form of damping to preclude large fluctuations in near-deficient directions. The popular ***regularization*** method minimizes $\|\text{residual}\|^2 + \lambda^2\|\text{step size}\|^2$ for some small $\lambda > 0$ — an approach that fits very well with Levenberg-Marquardt-like search control schemes. Alternatively, a ***basic solution*** — a solution where certain uncontrolled components are set to zero — can be calculated from a standard pivoted QR or Cholesky decomposition, simply by ignoring the last few (degenerate) columns. One can also find vectors spanning the local gauge directions and treat them as 'virtual constraints' with zero residual, so that the gauge motion is locally zeroed. ***Householder reduction***, which orthogonalizes the rows of $\frac{\mathbf{de}}{\mathbf{dx}}$ w.r.t. the gauge matrix by partial QR decomposition, is a nice example of this.

4.2 Numerical Methods for Constrained Optimization

There are at least three ways to handle linear constraints numerically: (*i*) ***eliminate variables*** using the constraint Jacobian; (*ii*) introduce ***Lagrange multipliers*** and solve for these too; (*iii*) ***weighting methods*** treat the constraints as heavily weighted residual errors. Each method has many variants, depending on the matrix factorization used, the ordering of operations, *etc*. As a rough rule of thumb, for dense problems variable elimination is the fastest and stablest method, but also the most complex. Lagrange multipliers are slower because there are more variables. Weighting is simple, but slow and inexact — stable orthogonal decompositions are needed as weighted problems are ill-conditioned.

For efficiency, direct geometric fitting requires a sparse implementation — the features couple to the model, but not to each other. The above methods all extend to sparse problems, but the implementation complexity increases by about one order of magnitude in each case. My initial implementation [16] used Lagrange multipliers and Cholesky decomposition, but I currently prefer a stabler, faster 'multifrontal QR' elimination method. There is no space for full details here, but it works roughly as follows (NB: the implementation orders the steps differently for efficiency): For each constrained system, the constraint Jacobian $\frac{\mathbf{dc}}{\mathbf{dx}}$ is factorized and the results are propagated to the error Jacobian $\frac{\mathbf{de}}{\mathbf{dx}}$. This eliminates the $\dim(\mathbf{c})$ variables best controlled by the constraints from $\frac{\mathbf{de}}{\mathbf{dx}}$, leaving a 'reduced' $\dim(\mathbf{e}) \times (\dim(\mathbf{x}) - \dim(\mathbf{c}))$ least squares problem. Many factorization methods can be used for the elimination and the reduced problem. I currently use column pivoted QR decomposition for both, which means that the elimination step is essentially Gaussian elimination. All this is done for each feature system. The elimination also carries the $\frac{\mathbf{dc}}{\mathbf{du}}$ columns into the reduced system. The residual error of the reduced system can not be reduced by changing $\mathbf{x}$, but it is affected by changes in $\mathbf{u}$ acting via these reduced $\frac{\mathbf{dc}}{\mathbf{du}}$ columns, which thus give contributions to an effective reduced error Jacobian $\frac{\mathbf{de(u)}}{\mathbf{du}}$ for the model $\mathbf{u}$. (This is the reduced geometric fitting method's error function). The resulting model system is reduced against any model constraints and factorized by pivoted QR. Back-substitution through the various stages then gives the required model update and finally the feature updates.

4.3 Search Control

All of the above techniques are linear. For nonlinear problems they must be used in a loop with appropriate step damping and search control strategies. This has been an unexpectedly troublesome part of the implementation — there seems to be a lack of efficient, reliable search control heuristics for constrained optimization. The basic problem is that the dual goals of reducing the constraint violation and reducing the residual error often conflict, and it is difficult to find a compromise that is good in all circumstances. Traditionally, a ***penalty function*** [7] is used, but all such methods have a 'stiffness' parameter which is difficult to set — too weak and the constraints are violated, too strong and the motion along the constraints towards the cost minimum is slowed. Currently, rather than a strict penalty function, I use a heuristic designed to allow a reasonable amount of 'slop' during motions along the constraints. The residual/constraint conflict also affects ***step damping*** — the control of step length to ensure acceptable progress. The principle of a ***trust region*** — a dynamic local region of the search space where the local function approximations are thought to hold good — applies, but interacts badly with ***quadratic programming*** based step prediction routines which try to satisfy the constraints exactly no matter how far away they are. Existing heuristics for this seemed to be poor, so I have developed a new 'dual control' strategy which damps the towards-constraint and along-constraint parts of the step separately using two Levenberg-Marquardt parameters linked to the same trust region.

Another difficulty is ***constraint redundancy***. Many algebraic varieties require a number of generators greater than their codimension to eliminate spurious components elsewhere in the space. The corresponding constraint Jacobians theoretically have rank $=$ codimension on the variety, but usually rank $>$ codimension away from it. Numerically, a reasonably complete and well-conditioned set of generators is advisable to reduce the possibility of convergence to spurious solutions, but the high degree of rank degeneracy on the variety, and the rank transition as we approach it, are numerically troublesome. Currently, my only effective way to handle this is to assume known codimension r and numerically project out and enforce only the r strongest constraints at each iteration. This is straightforward to do during the constraint factorization step, once r is known. As examples: the trifocal point constraints $[\,\boldsymbol{x}'\,]_\times(\boldsymbol{G}\cdot\boldsymbol{x})[\,\boldsymbol{x}''\,]_\times = \boldsymbol{0}$ have rank 4 in $(\boldsymbol{x},\boldsymbol{x}',\boldsymbol{x}'')$ for most invalid tensors, but only rank 3 for valid ones; and the trifocal consistency constraints $\frac{\mathbf{d}^3}{\mathbf{d}\boldsymbol{x}^3}\det(\boldsymbol{G}\cdot\boldsymbol{x}) = \boldsymbol{0}$ have rank 10 for most invalid tensors, but only rank 8 for valid ones. In both cases, overestimating the rank causes severe ill-conditioning.

4.4 Robustification

We assume that each feature has a ***central*** robust cost function $\rho_i(\mathbf{x}_i) \equiv \rho_i(\|\mathbf{e}_i(\mathbf{x}_i)\|^2)$ defined in terms of a covariance-weighted ***normalized residual error*** $\mathbf{e}_i(\mathbf{x}_i) \equiv \mathbf{e}_i(\mathbf{x}_i|\underline{\mathbf{x}}_i)$. This defines the 'granularity' — entire 'features' (for matching constraints, ensembles of corresponding image features) are robustified, not their individual components. The robust cost ρ_i is usually some M-estimator, often a total log likelihood. For a uniform-outlier-polluted Gaussian it has the form $\rho(z) \equiv -2\log\left(e^{-z/2}+\beta\right)$, where β is related to outlier density. Typically, $\rho(z)$ is linear near 0, monotonic but sublinear for $z > 0$ and tends to a constant at $z \to \infty$ if distant outliers have vanishing influence. Hence, $\rho' \equiv \frac{d\rho}{dz}$ decreases monotonically to 0 and $\rho'' \equiv \frac{d^2\rho}{dz^2}$ is negative.

Robustification can lead to numerical problems, so care is needed. Firstly, since the cost is often nonconvex for outlying points, strong regularization may be required to guarantee a positive Hessian and hence a cost reducing step. This can slow convergence. To partially compensate for this curvature, and to allow us to use a 'naïve' Gauss-Newton step calculation while still accounting for robustness, we define a weighted, rank-one-corrected ***effective residual*** $\tilde{\mathbf{e}} \equiv \frac{\sqrt{\rho'}}{1-\alpha}\mathbf{e}$ and ***effective Jacobian*** $\widetilde{\frac{\mathbf{de}}{\mathbf{dx}}} \equiv \sqrt{\rho'}\,(\mathbf{I} - \frac{\alpha}{\|\mathbf{e}\|^2}\,\mathbf{e}\,\mathbf{e}^{\mathrm{T}})\frac{\mathbf{de}}{\mathbf{dx}}$ where $\alpha \equiv \mathrm{RootOf}(\frac{1}{2}\alpha^2 - \alpha - \frac{\rho''}{\rho'}\|\mathbf{e}\|^2)$. These definitions

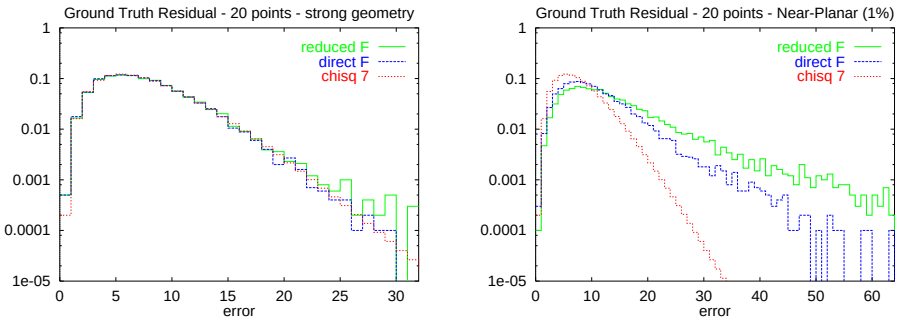


Figure 1: Ground feature residuals for strong and near-coplanar epipolar geometries.

ensure that to second order in ρ and $\mathbf{dx}$ and up to an irrelevant constant, the true robust cost $\rho(\|\mathbf{e} + \frac{\mathbf{de}}{\mathbf{dx}}\mathbf{dx}\|^2)$ is the same as the naïve effective squared error $\|\tilde{\mathbf{e}} + \widetilde{\frac{\mathbf{de}}{\mathbf{dx}}}\mathbf{dx}\|^2$. *I.e.* the same step $\mathbf{dx}$ is generated, so if we use effective quantities, we need think no further about robustness[3]. Here the $\sqrt{\rho'}$ weighting is the first order correction, and the α terms are the second order one. Usually $\rho' \to 0$ for distant outliers. Since the whole feature system is scaled by $\sqrt{\rho'}$, this might cause numerical conditioning or scaling problems in the direct method. To avoid this, we actually apply the $\sqrt{\rho'}$-weighting at the last possible moment — the contribution of the feature to the model error — and leave the feature systems themselves unweighted.

5 Measuring Performance

We currently test mainly on synthetic data, to allow systematic comparisons over a wide range of problems. We are particularly concerned with verifying theoretical statistical performance bounds, as these are the best guarantee that we are doing as well as could reasonably be expected. Any tendency to return occasional outliers is suspect and needs to be investigated. Histograms of the ***ground-truth-feature residual*** (GFR) have proven particularly useful for this. These plot frequency *vs.* size of the total squared deviation of the *ground truth* values of the noisy features used in the estimate, from the estimated matching relations. This measures how *consistent* the estimated geometry is with the underlying noise-free features. For weak feature sets the geometry might still be far from the true one, but consistency is the most we can expect given the data. In the linear approximation the GFR is χ^2_ν distributed for any sufficient model and number of features, where ν is the number of d.o.f. of the underlying inter-image geometry. This makes GFR easy to test and very sensitive to residual biases and oversized errors, as these are typically proportional to the number of features n and hence easily seen against the fixed χ^2_ν background for $n \gg \nu$. For example, fig.1 shows GFR histograms for the 7 d.o.f. uncalibrated epipolar geometry for direct and reduced $\boldsymbol{F}$-matrix estimators and strong and weak (1% non-coplanar) feature sets. For the strong geometry both methods agree perfectly with the theoretical χ^2_7 distribution without any sign of outliers, so both methods do as well as could be hoped. This holds for any number of points from 9 to 1000 — the estimated geometry (error per point) becomes more accurate, but the total GFR error stays constant. For the weak geometry both methods do significantly worse than the theoretical limit — in fact they turn out to have a small but roughly constant residual error *per point* rather than in total — with the direct method being somewhat better than the reduced one. We are currently investigating this: in theory it should be possible to get near the limit, even for exactly singular geometries.

[3] If $\frac{\rho''}{\rho'}\|\mathbf{e}\|^2 < -\frac{1}{2}$ the robust Hessian has negative curvature and there is no real solution for α. In practice we limit $\alpha < 1 - \epsilon$ to prevent too much ill-conditioning. We would have had to regularize this case away anyway, so nothing is lost.

6 Summary

We have described work in progress on a generic, modular library for the optimal nonlinear estimation of matching constraints, discussing especially the overall approach, parametrization and numerical optimization issues. The library will cover many different constraint types & parametrizations and feature types & error models in a uniform framework. It aims to be efficient and stable even in near-degenerate cases, *e.g.* so that it can be used reliably for model selection. Several fairly sophisticated numerical methods are included, including a sparse constrained optimization method designed for ***direct geometric fitting***. Future work will concentrate mainly on (*i*) implementing and comparing different constraint types and parametrizations, feature types, and numerical resolution methods; and (*ii*) improving the reliability of the initialization and optimization stages, especially in near-degenerate cases.

References

[1] S. Avidan and A. Shashua. Threading fundamental matrices. In *European Conf. Computer Vision*, pages 124–140, Freiburg, 1998.

[2] Åke Björk. *Numerical Methods for Least Squares Problems*. SIAM Press, Philadelphia, PA, 1996.

[3] B. Boufama and R. Mohr. Epipole and fundamental matrix estimation using the virtual parallax property. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1030–1036, Cambridge, MA, June 1995.

[4] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

[5] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[6] O. Faugeras and T. Papadopoulo. Grassmann-Cayley algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors. *Transactions of the Royal society A*, 1998.

[7] R. Fletcher. *Practical Methods of Optimization*. John Wiley, 1987.

[8] R.I. Hartley. Lines and points in three views and the trifocal tensor. *Int. J. Computer Vision*, 22(2):125–140, 1997.

[9] Anders Heyden. *Geometry and Algebra of Multiple Projective Transformations*. Ph.D. Thesis, University of Lund, 1995.

[10] Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimental results. Technical Report RR-1894, INRIA, Sophia Antipolis, France, 1993.

[11] Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. In *European Conf. Computer Vision*, pages 589–599, 1994.

[12] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–20, Cambridge, U.K., 1996. Springer-Verlag.

[13] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 13(11):1115–38, 1991.

[14] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.

[15] B. Triggs. Linear projective reconstruction from matching tensors. In *British Machine Vision Conference*, pages 665–74, Edinburgh, September 1996.

[16] B. Triggs. A new approach to geometric fitting. Available from http://www.inrialpes.fr/movi/people/Triggs, 1997.

Differential Matching Constraints

Bill Triggs
INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot St. Martin, France.
Bill.Triggs@inrialpes.fr ◇ *http://www.inrialpes.fr/movi/people/Triggs*

Abstract

We introduce a finite difference expansion for closely spaced cameras in projective vision, and use it to derive differential analogues of the finite-displacement projective matching tensors and constraints. The results are simpler, more general and easier to use than Åström & Heyden's time-derivative based 'continuous time matching constraints'. We suggest how to use the formalism for 'tensor tracking' — propagation of matching relations against a fixed base image along an image sequence. We relate this to nonlinear tensor estimators and show how 'unwrapping the optimization loop' along the sequence allows simple 'linear n point' update estimates to converge rapidly to statistically near-optimal, near-consistent tensor estimates as the sequence proceeds. We also give guidelines as to when difference expansion is likely to be worthwhile as compared to a discrete approach.

Keywords: Matching Constraints, Matching Tensors, Image Sequences, Tensor Tracking, Difference Expansion.

1 Introduction

This paper studies **differential matching constraints** — limiting forms of ordinary multi-image matching constraints [5, 7, 8, 12, 15], when some of the image projections nearly coincide. We introduce a finite difference based formalism that is easy to use and covers most aspects of projective multi-image geometry: matching constraints and tensors, feature transfer, reconstruction. Modulo suitable image rectification (fixation, dominant plane stabilization [9, 10]), the results extend to all *small translation* geometries, *i.e.* whenever some of the camera centres are near-coincident on the scale of the scene. For convenience we will often express results in terms of feature displacements ('flow'). But this is largely cosmetic: feature positions could equally well be used. Our method spans the gap between infinitesimal [17, 2] and discrete approaches: only *some* of the cameras need coincide and our difference expansions are short, finite polynomials not infinite Taylor series.

This section gives motivation and previous work, §2 reviews discrete matching constraints, §3 reviews and critiques Åström & Heyden's differential approach, §4 introduces our difference formalism and differential matching tensors, §5 derives various differential matching constraints, and §6 summarizes and concludes.

Motivation: Theoretically, "nothing is gained" by a differential approach: the same underlying geometric constraints and image error models apply in both differential and discrete approaches. However, small displacements are practically common (*e.g.* video sequences) and have special properties that make purpose-built methods desirable:
(+) ***Feature correspondence*** is much easier so more data is available, especially with region based ('direct', 'least squares', 'intensity based') approaches.
(+) Differential problems are often ***less nonlinear*** than discrete ones, as nonlinear geometry (rotations, calibration, matching tensor consistency) can be locally linearized and included in the initial linear estimation for improved stability. Simpler models can be used, and local minima may be less of a problem.
(−) Small motion linearization is only an ***approximation***. It has limited validity and introduces bias/truncation error.
(−) The additional correspondences are often of ***low quality***: they may add a lot of computation but relatively little precision.
(−) ***Signal-to-noise ratio*** is lower with small motion, so fewer parameters can be estimated accurately (*e.g.* SFM, perspective) and error modelling

This paper appeared in ICCV'99. The work was supported by Esprit LTR project CUMULI. I would like to thank P. Anandan and T. Viéville for useful discussions.

is more critical: bias, outliers, linearization error.

Given that geometric constraints are known to improve robustness and efficiency even for small motion (*c.f.* 'Geometrically Constrained Multiphoto Matching' [3]), it seems worthwhile to develop the matching constraint formalism in this direction. We will also link our differential matching constraints to the local linearization used in nonlinear estimators for the discrete case, so a better understanding of differential case may lead to better estimators for the discrete one. Another motivation was to develop routines for ***matching constraint tracking***, *i.e.* updating the matching geometry along an image sequence from linear change estimates, rather than wastefully recalculating it from scratch each time, or using the image tracks only to get correspondences between the two widely-spaced end images.

Previous Work: There are many papers on all aspects of optical flow — see [4] for references — but here we will focus on differential analogues of the *uncalibrated* discrete matching constraints. The key contributions on this are by Viéville & Faugeras [16, 17] for the two image case and Åström & Heyden [1, 2] for the multi-image one. We will return to the Åström-Heyden approach below. Other related work includes [13, 6, 14].

2 Discrete Matching Constraints

In homogeneous coordinates, image i has 3×4 projection matrix $\mathbf{P}_i$. The image $\boldsymbol{x}_i$ of a 3D point $\mathbf{X}$ is $\lambda_i \boldsymbol{x}_i = \mathbf{P}_i \mathbf{X}$. The scale factors λ_i are called ***projective depths***. Gather m image projections of $\mathbf{X}$ into a big $3m \times (4+m)$ matrix [15]:

$$\begin{pmatrix} \mathbf{P}_1 & \boldsymbol{x}_1 & 0 & \cdots & 0 \\ \mathbf{P}_2 & 0 & \boldsymbol{x}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ \mathbf{P}_m & 0 & 0 & \cdots & \boldsymbol{x}_m \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ -\lambda_1 \\ \vdots \\ -\lambda_m \end{pmatrix} = \mathbf{0} \quad (1)$$

As there is a solution, the matrix has rank $\leq 3+m$, *i.e.* all of its $(4+m) \times (4+m)$ minors must vanish. Expanding and simplifying gives 'epipolar', 'trifocal' and 'quadrifocal' ***multi-image matching constraints*** linking corresponding points $\boldsymbol{x}_i$ in 2,3,4 images. Similar constraints exist for 3 images of a line and for 2 images of a line plus 1 image of a point on it. Each constraint is multilinear in the 2–4 image features involved, with coefficients that are 4×4 determinants built from 4 rows taken from 2–4 projection matrices. The determinants can be arranged into 4 types of ***matching tensor***[1], depending on how many rows are taken from each image. It will be useful to view the tensors as multi-index, multilinear forms in the components of 4 (possibly repeated) projection matrices. Symbolically:

$$\begin{array}{rcll} \boldsymbol{e}_1^2 & \equiv & \boldsymbol{e}(1,1,1,2) & \text{epipole} \\ \boldsymbol{F}_{12} & \equiv & \boldsymbol{F}(1,1,2,2) & \text{fundamental matrix} \\ \boldsymbol{T}_1^{23} & \equiv & \boldsymbol{T}(1,1,2,3) & \text{trifocal tensor} \\ \boldsymbol{Q}^{1234} & \equiv & \boldsymbol{Q}(1,2,3,4) & \text{quadrifocal tensor} \end{array} \quad (2)$$

where, *e.g.* $\boldsymbol{F}(1,1',2,2')$ stands for a 3×3-matrix-valued quadrilinear form $\boldsymbol{F}(\mathbf{P}_1, \mathbf{P}'_1, \mathbf{P}_2, \mathbf{P}'_2)$ in the four projection matrices $\mathbf{P}_1, \mathbf{P}'_1, \mathbf{P}_2, \mathbf{P}'_2$, and the fundamental matrix $\boldsymbol{F}_{12}(\mathbf{P}_1, \mathbf{P}_2)$ is the result of substituting $\mathbf{P}'_1 = \mathbf{P}_1$ and $\mathbf{P}'_2 = \mathbf{P}_2$ into this. As multilinear forms in four projections, the components of $\boldsymbol{e}(\cdot), \boldsymbol{F}(\cdot), \boldsymbol{T}(\cdot)$ are simple, fixed linear combinations[2] of those of $\boldsymbol{Q}(\cdot)$. When their arguments are repeated as shown above, $\boldsymbol{e}(\cdot), \boldsymbol{F}(\cdot), \boldsymbol{T}(\cdot)$ contain exactly the same information as the corresponding version of $\boldsymbol{Q}(\cdot)$, in a more compact, easier-to-use form. Even when the arguments are not repeated, $\boldsymbol{e}(\cdot), \boldsymbol{F}(\cdot), \boldsymbol{T}(\cdot)$ are automatically symmetric in the arguments shown as repeated, *e.g.* $\boldsymbol{e}(1,1',1'',2)$ and $\boldsymbol{F}(1,1',2,2')$ are symmetric under all permutations of the three $\mathbf{P}_1$'s and two $\mathbf{P}_2$'s.

Given the tensors, the matching constraints we will differentialize below can be written symbolically as:

$$\begin{array}{rcll} \boldsymbol{x}_1^\top \boldsymbol{F}_{12}\, \boldsymbol{x}_2 & = & 0 & \text{epipolar constraint} \\ \boldsymbol{x}_2 \wedge \left(\boldsymbol{T}_1^{23} \cdot \boldsymbol{x}_1\right) \wedge \boldsymbol{x}_3 & = & \boldsymbol{0} & \text{trifocal point constraint} \\ \boldsymbol{l}_2^\top \left(\boldsymbol{T}_1^{23} \wedge \boldsymbol{l}_1\right) \boldsymbol{l}_3 & = & \boldsymbol{0} & \text{trifocal line constraint} \\ \boldsymbol{l}_2^\top \left(\boldsymbol{T}_1^{23} \cdot \boldsymbol{x}_1\right) \boldsymbol{l}_3 & = & \boldsymbol{0} & \text{trifocal point-line const.} \end{array}$$

Here, $\boldsymbol{x}_i$ ($\boldsymbol{l}_i$) denote corresponding image points

[1] **Tensors** are just multi-index arrays of components. They are not intrinsically difficult to handle, but lie outside the usual matrix-vector notation. For simplicity I'll display results as matrices whenever possible, and switch into indexed notation [15] when matrix notation is too weak. For calculations I use ***tensor diagrams*** — 'circuit diagrams' that show graphically which indices are connected.

[2] They are contractions of $\boldsymbol{Q}(\cdot)$ against image $\boldsymbol{\epsilon}$ tensors — *e.g.* $\boldsymbol{F}_{AB}(1,1',2,2') \equiv \frac{1}{4}\boldsymbol{\epsilon}_{ACD}\, \boldsymbol{\epsilon}_{BEF}\, \boldsymbol{Q}^{CDEF}(1,1',2,2')$ [15].

(lines) in image i, and $\wedge$ or $[\,\cdot\,]_\times$ denotes vector-vector or matrix-vector cross product.

Geometrically, the matching constraints express 3D incidence relations between the optical rays / planes pulled back from corresponding image points / lines. The matching tensors are a nonlinear encoding of the camera geometry in image coordinates. They can be estimated "linearly" from image data using the matching constraints, but only by: (*i*) using a heuristic error model; (*ii*) ignoring *nonlinear self-consistency constraints* that guarantee that the tensor(s) correspond to some underlying set of projection matrices. Examples of such constraints include $\boldsymbol{F}_{12}\,\boldsymbol{e}_1^2 = 0$, $\det(\boldsymbol{F}_{12}) = 0$, $\det\left(\boldsymbol{T}_1^{23}\cdot\boldsymbol{x}_1\right) = 0$ for all $\boldsymbol{x}_1$, and many more [15]. One advantage of the differential approach is that it often allows the consistency constraints and the true statistical error model to be locally linearized, so that simple linear least squares tensor estimators can take nearly full account of both.

3 The Åström-Heyden Approach

This section summarizes and critiques Åström & Heyden's approach to differential multi-image matching constraints [1, 2]. A moving camera with time varying projection matrix $\mathbf{P}(t)$ viewing a static scene generates image projections $\lambda(t)\,\boldsymbol{x}(t) = \mathbf{P}(t)\,\mathbf{X}$. Taylor expand at t:

$$\mathbf{P}(t+\Delta t) \;=\; \mathbf{P}^{(0)} + \mathbf{P}^{(1)}\,\Delta t + \mathbf{P}^{(2)}\,(\Delta t)^2 + \ldots$$

where $\mathbf{P}^{(k)} \equiv \frac{1}{k!}\frac{d^k}{dt^k}\mathbf{P}$, and similarly for $\boldsymbol{x}(t+\Delta t)$ and $\lambda(t+\Delta t)$. Substitute into the projection equations, truncate at order m, split by powers of Δt, and gather the resulting equations into a $3(m+1)\times(4+(m+1))$ matrix

$$\begin{pmatrix} \mathbf{P}^{(0)} & \boldsymbol{x}^{(0)} & 0 & \cdots & 0 \\ \mathbf{P}^{(1)} & \boldsymbol{x}^{(1)} & \boldsymbol{x}^{(0)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ \mathbf{P}^{(m)} & \boldsymbol{x}^{(m)} & \boldsymbol{x}^{(m-1)} & \cdots & \boldsymbol{x}^{(0)} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ -\lambda^{(0)} \\ \vdots \\ -\lambda^{(m)} \end{pmatrix} = \mathbf{0}$$

As in (1), all maximal minors vanish. Expanding gives multilinear **differential matching constraints** involving all of the point derivatives $\boldsymbol{x}^{(0)},\ldots,\boldsymbol{x}^{(m)}$. The coefficients are **differential matching tensors** formed from 4×4 minors of 4 rows of the projection derivatives $\mathbf{P}^{(0)},\ldots,\mathbf{P}^{(m)}$.

This approach is certainly powerful, but I feel that it is not "the right thing" for most applications: (*i*) The constraints combine infinitely many feature derivatives and differential matching tensors of arbitrarily high orders, even though the discrete case stops at $m = 4$ features and tensors. (*ii*) The constraints are *extremely complicated*, even for $m = 3$. (*iii*) It is very difficult to relate them to the discrete case, even though their derivation is almost identical. (*iv*) They depend on the exact form of the camera motion between t and $t+\Delta t$, whereas we often know or care only about the camera *positions* at the endpoints t and $t+\Delta t$. (*v*) Many things remain to be done: lines, transfer, depth recovery, cases where some images are from other, more widely-spaced cameras, *etc*.

Note that only the geometric path of the camera matters for the constraints, not its time parametrization. So they should really be formulated in terms of some geometric, parametrization-invariant analogue of differential equations such as **exterior differential systems** (*c.f.* also [13]). This was my first intention, but on reflection it does not solve the main problem, which is simply that *differentiation is not the appropriate tool here.*

In applications, images are always *finitely* (though perhaps closely) spaced. What we measure is feature positions at these discrete times, and what we *use* is matching constraints, projection matrices, *etc*, again at these discrete times. Time derivatives never explicitly appear, and if introduced, they are serve only to re-synthesize the finite-time positions that we actually measure or use. Finite differences are a more appropriate tool for such discrete-time problems. Given measurements of some quantity $\boldsymbol{x}(t), \boldsymbol{x}(t+\Delta t)$, their **finite difference** is simply $\Delta\boldsymbol{x} \;\equiv\; \boldsymbol{x}(t+\Delta t) - \boldsymbol{x}(t)$. So we have a finite, one term 'expansion' $\boldsymbol{x}(t+\Delta t) \;=\; \boldsymbol{x}(t) + \Delta\boldsymbol{x}$ rather than an infinite Taylor series $\boldsymbol{x}(t+\Delta t) = \boldsymbol{x}(t) + \dot{\boldsymbol{x}}\,\Delta t + \frac{1}{2}\,\ddot{\boldsymbol{x}}\,\Delta t^2 + \ldots$. If we use $\boldsymbol{x}(t+\Delta t)$ in some polynomial expression (matching constraints, transfer, SFM), difference expansion gives a relatively simple polynomial in $\Delta\boldsymbol{x}$, while Taylor expansion a very complicated infinite series in Δt. The Taylor series is ultimately more powerful in that it implies values of $\boldsymbol{x}$ for *all* Δt. But if we measure and use $\boldsymbol{x}$ only at *one* Δt as here, $\dot{\boldsymbol{x}}\,\Delta t + \frac{1}{2}\,\ddot{\boldsymbol{x}}\,\Delta t^2 + \ldots$ is a very complicated way of parametrizing the simple difference $\Delta\boldsymbol{x}$.

In summary, Åström & Heyden got an infinite series of complicated equations rather than a finite series of simple ones simply because they asked for too much. Their results are like a series solution to a differential equation: they imply the matching constraints for *every* Δt with *any* analytic camera motion, whereas in practice we usually only want them at the endpoints of *one particular* Δt.

4 Projective Difference Expansion

Now we begin to assemble the elements of our finite difference approach to projective vision. First, a clarification. We work with projective quantities expressed in homogeneous coordinates, *e.g.* image points $\boldsymbol{x}$, projections $\mathbf{P}$. We want to expand projective expressions in $\boldsymbol{x}', \mathbf{P}'$ in terms of "nearby" base quantities $\boldsymbol{x}, \mathbf{P}$ and "projective differences" $\Delta\boldsymbol{x} = \boldsymbol{x}' - \boldsymbol{x}$, $\Delta\mathbf{P} = \mathbf{P}' - \mathbf{P}$. Unfortunately, homogeneous quantities like $\boldsymbol{x}, \boldsymbol{x}'$ are only defined up to scale, so differences like $\boldsymbol{x}' - \boldsymbol{x}$ are not well defined: as their relative scale changes, $\boldsymbol{x}' - \boldsymbol{x}$ sweeps out the entire projective line through $\boldsymbol{x}, \boldsymbol{x}'$. Nevertheless, if we are careful about scales, we can still use $\Delta\boldsymbol{x} \equiv \boldsymbol{x}' - \boldsymbol{x}$ to represent the displacement between two projective points. Fix the scale of $\boldsymbol{x}$ once and for all. Under rescaling $\boldsymbol{x}' \to (1+\mu)\boldsymbol{x}'$, $\Delta\boldsymbol{x}$ changes as $\Delta\boldsymbol{x} \to \Delta\boldsymbol{x} + \mu\boldsymbol{x}' \approx \Delta\boldsymbol{x} + \mu\boldsymbol{x} + \mathcal{O}(\mu\Delta\boldsymbol{x})$. So for small rescalings μ and displacements $\Delta\boldsymbol{x}$, $\Delta\boldsymbol{x}$ is only defined modulo the approximate affine freedom $\Delta\boldsymbol{x} \to \Delta\boldsymbol{x} + \mu\boldsymbol{x}$. The expressions we need to expand are always separately homogeneous in $\boldsymbol{x}$ and $\boldsymbol{x}' = \boldsymbol{x} + \Delta\boldsymbol{x}$, so this freedom leads to the following important **invariance principle**: *The term of lowest nonvanishing order in $\Delta\boldsymbol{x}$ is explicitly invariant under shifts $\Delta\boldsymbol{x} \to \Delta\boldsymbol{x} + \mu\boldsymbol{x}$.* We usually work only to this order, so formulae which *use* $\Delta\boldsymbol{x}$ are invariant, and formulae which *calculate* it can do so only up to an unknown multiple of $\boldsymbol{x}$. For example, our formulae for differential matching tensors are defined only up to multiples of the underlying base tensor. In practice, for input data we simply choose similar normalizations for $\boldsymbol{x}, \boldsymbol{x}'$ so that μ is small. But for numerically calculated Δ's we always need to enforce some sort of normalization condition to remove the superfluous rescaling degree of freedom.

A related point which greatly simplifies many of the formulae is that: *Difference expansion in a variable is only worthwhile if the problem is nonlinear in that variable.* One can certainly derive expansions for linearly-appearing variables of the form $(\mathbf{A} + \Delta\mathbf{A} + \dots)\cdot(\boldsymbol{x} + \Delta\boldsymbol{x}) \approx \mathbf{A}\cdot\boldsymbol{x} + \mathbf{A}\cdot\Delta\boldsymbol{x} + \Delta\mathbf{A}\cdot\boldsymbol{x} + \mathcal{O}(\Delta^2)$, where $\mathbf{A}$ stands for other stuff independent of $\boldsymbol{x}' = \boldsymbol{x} + \Delta\boldsymbol{x}$ and hence $\Delta\boldsymbol{x}$. But there's really no point. If you already have $\boldsymbol{x}, \Delta\boldsymbol{x}$ and are trying to calculate $\mathbf{A}, \Delta\mathbf{A}$, you might as well just use $\boldsymbol{x}'$ in the exact expression. This is simpler, has less truncation error, and (at least in vision) is unlikely even to cause problems with numerical loss of precision: Δ's usually scale roughly as measured image differences, which have a minimum relative size of about 10^{-4} as differences much smaller than a pixel or greater than the image width can not be measured. In fact, since we are working to lowest nonvanishing order in Δ and $\mathbf{A}$ is independent of $\boldsymbol{x}'$, invariance under $\Delta\boldsymbol{x} \to \Delta\boldsymbol{x} + \mu\boldsymbol{x}$ implies that $\mathbf{A}\cdot\boldsymbol{x}$ must actually *vanish* (at least in the zero noise case). Conversely, if you are trying to calculate $\Delta\boldsymbol{x}$ given $\mathbf{A}, \Delta\mathbf{A}$, the equation is linear in either $\Delta\boldsymbol{x}$ or $\boldsymbol{x}' = \boldsymbol{x} + \Delta\boldsymbol{x}$, so you might as well just form the update $(\mathbf{A} + \Delta\mathbf{A} + \dots)$ and calculate $\boldsymbol{x}'$ directly. This remains true even if $\mathbf{A}$ depends on $\boldsymbol{x}$, so long as it is independent of $\boldsymbol{x}'$.

For example, matching constraints and transfer relations are usually linear in each of their image features, so there is no real advantage in using image displacements or 'flow' for them — one can just as well use the underlying features $\boldsymbol{x}, \boldsymbol{x}'$. Arguably, this also applies to 'direct' (intensity based, optical flow) approaches — one can use intensity differences to estimate local correlation shifts just as well as image derivatives[3]. Similarly, for epipoles, homographies and trifocal tensors, some of the projection matrices appear linearly and there is no real advantage in making a difference expansion in these. (More precisely, there is none once the coefficients multiplying the projection to form the epipole, *etc*, have been recovered). On the other hand, for linear tensor-based parametrizations, the consistency constraints are always nonlinear and hence *do* benefit from expansion.

We will sometimes need to take differences in several images simultaneously, *e.g.* for each i, if $\mathbf{P}_i'$ is near to $\mathbf{P}_i$ we define $\Delta\mathbf{P}_i \equiv \mathbf{P}_i' - \mathbf{P}_i$. If there

[3] As with the Taylor series above, the derivatives are only an indirect way of synthesizing image displacements, which could have been produced more directly using (sub-pixel/multi-scale/ ...) image interpolation.

are several projections $\mathbf{P}'_i, \mathbf{P}''_i$ near the same base projection $\mathbf{P}_i$, each generates its own independent difference $\Delta\mathbf{P}'_i, \Delta\mathbf{P}''_i$.

By substituting the updates $\mathbf{P}'_i = \mathbf{P}_i + \Delta\mathbf{P}_i = (1+\Delta)\mathbf{P}_i$ into the multilinear matching forms (2) and expanding, we can derive exact finite difference expansions of all the matching tensors. For example, for the $1'$–2 fundamental matrix

$$\begin{aligned}\boldsymbol{F}_{1'2} &\equiv \boldsymbol{F}(1',1',2,2)\\ &= \boldsymbol{F}\left((1+\Delta)\mathbf{P}_1,(1+\Delta)\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_2\right)\\ &= \boldsymbol{F}(1,1,2,2) + 2\,\boldsymbol{F}(\Delta 1,1,2,2) + \boldsymbol{F}(\Delta 1,\Delta 1,2,2)\end{aligned}$$

where $\Delta 1$ stands for $\Delta\mathbf{P}_1$, *etc*. If only one projection varies, the full list of such expansion types is:

$$\begin{aligned}\boldsymbol{e}^2_{1'} &= \boldsymbol{e}^2_1 + \boldsymbol{e}^2_{\Delta 1} + \boldsymbol{e}^2_{\Delta^2 1} + \boldsymbol{e}^2_{\Delta^3 1}\\ \boldsymbol{F}_{1'2} &= \boldsymbol{F}_{12} + \boldsymbol{F}_{\Delta 12} + \boldsymbol{F}_{\Delta^2 12}\\ \boldsymbol{T}^{23}_{1'} &= \boldsymbol{T}^{23}_1 + \boldsymbol{T}^{23}_{\Delta 1} + \boldsymbol{T}^{23}_{\Delta^2 1}\\ \boldsymbol{e}^{2'}_1 &= \boldsymbol{e}^2_1 + \boldsymbol{e}^{\Delta 2}_1\\ \boldsymbol{T}^{2'3}_1 &= \boldsymbol{T}^{23}_1 + \boldsymbol{T}^{\Delta 23}_1\\ \boldsymbol{Q}^{1'234} &= \boldsymbol{Q}^{1234} + \boldsymbol{Q}^{\Delta 1234}\end{aligned} \tag{3}$$

where we define the following **differential matching tensors** by successively replacing projections $\mathbf{P}'$ with projection differences $\Delta\mathbf{P} \equiv \mathbf{P}' - \mathbf{P}$:

$$\begin{aligned}\boldsymbol{e}^2_{\Delta 1} &\equiv 3\,\boldsymbol{e}(\Delta 1,1,1,2)\\ \boldsymbol{F}_{\Delta 12} &\equiv 2\,\boldsymbol{F}(\Delta 1,1,2,2)\\ \boldsymbol{T}^{23}_{\Delta 1} &\equiv 2\,\boldsymbol{T}(\Delta 1,1,2,3)\\ \boldsymbol{Q}^{\Delta 1234} &\equiv \boldsymbol{Q}(\Delta 1,2,3,4)\\ \boldsymbol{e}^{\Delta 2}_1 &\equiv \boldsymbol{e}(1,1,1,\Delta 2)\\ \boldsymbol{T}^{\Delta 23}_1 &\equiv \boldsymbol{T}(1,1,\Delta 2,3)\\ \boldsymbol{e}^2_{\Delta^2 1} &\equiv 3\,\boldsymbol{e}(\Delta 1,\Delta 1,1,2)\\ \boldsymbol{F}_{\Delta^2 12} &\equiv \boldsymbol{F}(\Delta 1,\Delta 1,2,2)\\ \boldsymbol{T}^{23}_{\Delta^2 1} &\equiv \boldsymbol{T}(\Delta 1,\Delta 1,2,3)\\ \boldsymbol{e}^2_{\Delta^3 1} &\equiv \boldsymbol{e}(\Delta 1,\Delta 1,\Delta 1,2)\end{aligned}$$

Very few of these are needed in any one application. If $\Delta\mathbf{P}$ is small, we can truncate the finite difference expansions at any desired order. The scales of the differential tensors were chosen to make the difference expansions simple, as this is essentially the only place they appear. The derivations use the symmetry of the forms $\boldsymbol{e}(\cdot), \boldsymbol{F}(\cdot), \boldsymbol{T}(\cdot)$. There are analogous expansions when several projections vary at once. We attach primes and Δ's to indices rather than whole tensors (*e.g.* $\boldsymbol{F}_{1'2}$, $\boldsymbol{e}^2_{\Delta 1}$), because the latter becomes hopelessly confusing when several projections vary at once.

The differential tensors depend on the normalizations of the $\Delta\mathbf{P}$'s, and are only defined up to admixtures of lower order terms, *e.g.* $\boldsymbol{F}_{1\Delta 2} \rightarrow \boldsymbol{F}_{1\Delta 2} + \mu\,\boldsymbol{F}_{12}$. **Saturated** differential tensors have all $\mathbf{P}$'s of a certain type replaced by $\Delta\mathbf{P}$'s. They behave just like ordinary matching tensors formed with "projections" $\Delta\mathbf{P}$, *e.g.* the "fundamental matrix" $\boldsymbol{F}_{\Delta^2 12} = \boldsymbol{F}(\Delta 1,\Delta 1,2,2)$ satisfies $\det(\boldsymbol{F}_{1\Delta^2 2}) = 0$ and has "epipoles" $\boldsymbol{e}^{\Delta 1}_2$ and $\boldsymbol{e}^2_{\Delta^3 1}$. But unsaturated tensors are more common in low order expansions: these have the same index structure but different properties.

5 Differential Matching Constraints

Given these expansions, it is very straightforward to develop differential forms of the various discrete matching constraints, transfer relations, *etc*. Simply take each discrete formula, choose the type of near-coincidence that should occur between its projection matrices, substitute the corresponding difference expansions (and optionally the difference expansions of the corresponding image features), expand, and truncate at the desired order.

Note that only *some* of the projections need be near coincident, unlike, *e.g.* [2]. In particular, we are investigating methods for *matching constraint tracking*, *i.e.* propagating a matching tensor against a base image along an image sequence by small updates, without having to recalculate it from scratch at each new image. This sort of approach should be useful for providing search constraints in geometrically guided feature trackers, as a tensor is available at each time step. And numerically it should allow linearized approximations to nonlinear error models and tensor consistency relations, so that a linearly-estimated tensor converges to a near-consistent, near-optimal estimate as the sequence continues. *I.e.*, the usual iterative refinement loop for the tensor would be 'unwrapped along the image sequence', tracking the moving tensor by a kind of locally-linearized control law, *c.f.* [13].

Differential Epipolar Constraint: The simplest

case is the epipolar constraint between a fixed camera $\mathbf{P}_1$ and a moving one $\mathbf{P}_2(t)$. We suppose that we have already calculated the fundamental matrix $\boldsymbol{F}_{12} \neq 0$, and want to update it to $\boldsymbol{F}_{12'}$ where $\mathbf{P}'_2 = \mathbf{P}_2 + \Delta\mathbf{P}_2$. Using (3), and optionally $\boldsymbol{x}'_2 = \boldsymbol{x}_2 + \Delta\boldsymbol{x}_2$ and the 1–2 epipolar constraint $\boldsymbol{x}_1^\top \boldsymbol{F}_{12}\, \boldsymbol{x}_2 = 0$, the first order expansion of the 1–2′ epipolar constraint is simply

$$\begin{aligned} 0 \;=\; \boldsymbol{x}_1^\top \boldsymbol{F}_{12'}\, \boldsymbol{x}_{2'} \;&\approx\; \boldsymbol{x}_1^\top \left(\boldsymbol{F}_{12} + \boldsymbol{F}_{1\Delta 2}\right) \boldsymbol{x}'_2 \\ &\approx\; \boldsymbol{x}_1^\top \boldsymbol{F}_{12}\, \Delta\boldsymbol{x}_2 + \boldsymbol{x}_1^\top \boldsymbol{F}_{1\Delta 2}\, \boldsymbol{x}_2 \end{aligned}$$

Using either form, $\boldsymbol{F}_{1\Delta 2}$ can be estimated linearly from $\boldsymbol{F}_{12}$, $\boldsymbol{x}_1$, and $\boldsymbol{x}'_2$ or $\boldsymbol{x}_2, \Delta\boldsymbol{x}_2$. $\boldsymbol{F}_{12'}$ can be recovered from $\boldsymbol{F}_{12'} \approx \boldsymbol{F}_{12} + \boldsymbol{F}_{1\Delta 2}$. The advantages over direct 'linear 8 point' estimation of $\boldsymbol{F}_{12'}$ are: (*i*) we can enforce the consistency constraint $\det(\boldsymbol{F}) = 0$, at least to a 1^{st}-order approximation; (*ii*) because of this, we need only 7 points; (*iii*) we can use $\boldsymbol{F}_{12}$ to pre-calculate approximately statistically optimal error weightings, so the initial linear estimator should have near-optimal accuracy. The linearization of the consistency constraint $\det(\boldsymbol{F}_{12'}) = 0$ is

$$\mathrm{trace}\left(\,\mathrm{cof}(\boldsymbol{F}_{12})\, \boldsymbol{F}_{1\Delta 2}\right) + \det(\boldsymbol{F}_{12}) \;=\; 0 \qquad (4)$$

where $\mathrm{cof}(\boldsymbol{F}_{12}) \;\approx\; \boldsymbol{e}_2^1\, \boldsymbol{e}_1^{2\top}$ is the matrix of cofactors of $\boldsymbol{F}_{12}$. Even if $\boldsymbol{F}_{12}$ is inconsistent, this equation enforces $\det(\boldsymbol{F}_{12'}) = 0$ to first order, and hence converges rapidly towards consistency.

As expected, $\boldsymbol{F}_{1\Delta 2}$ is only defined up to multiples of $\boldsymbol{F}_{12}$. For example, the error term $\boldsymbol{x}_1^\top \boldsymbol{F}_{1\Delta 2}\, \boldsymbol{x}_2$ and the linearized consistency constraint (4) have such invariances if $\boldsymbol{x}_1^\top \boldsymbol{F}_{12}\, \boldsymbol{x}_2$ and $\det(\boldsymbol{F}_{12})$ are exactly 0. The exact multiple we choose is irrelevant so long as it is small, but some choice is needed to avoid numerical ill-conditioning. In practice, we constrain $\boldsymbol{F}_{1\Delta 2}$ to be orthogonal to $\boldsymbol{F}_{12}$ as a 9-vector, *i.e.* $\mathrm{trace}(\boldsymbol{F}_{12}^\top \boldsymbol{F}_{1\Delta 2}) = 0$. Given the above and $\boldsymbol{F}_{12}$, near optimal '7 point' estimation of $\boldsymbol{F}_{1\Delta 2}$ reduces to a 9 variable linear least squares problem with 2 linear constraints. Any standard numerical method can be used, *e.g.* Gauss (LU) or Householder (LQ) based constraint elimination followed by QR decomposition to solve the reduced least squares problem. (For 7 point RANSAC, the problem becomes a simple 9×9 linear system).

Only the 1–2 and 1–2′ epipolar constraints were used here: the 1–2–2′ trifocal one will be considered below.

The Optimization Point-of-View: The above discussion should sound very familiar to anyone who has implemented a nonlinear fundamental matrix estimator. In fact, the above $\boldsymbol{F}_{12} \rightarrow \boldsymbol{F}_{12'}$ update rule is exactly one step of a Sequential Quadratic Programming (SQP) style refinement routine for $\boldsymbol{F}_{12'}$, started from the estimate $\boldsymbol{F}_{12}$. Further iterations could be used to improve the accuracy, if desired. The moral is that: *Tensor tracking and nonlinear tensor refinement are basically the same problem*. So the same numerical methods can be used for both. We also emphasize that there is really no advantage to using 'flow' $\Delta\boldsymbol{x}$ rather than position $\boldsymbol{x}'$, and the differential tensor $\boldsymbol{F}_{1\Delta 2}$ plays exactly the same role as a conventional first order model update $\Delta\boldsymbol{F}$. The difference expansion merely serves as a systematic way to derive such update equations.

Differential Trifocal Constraints: First order expansion of the 1–2′–3 and 1′–2–3 trifocal point, line and point-line matching constraints modulo the 1–2–3 ones gives:

$$\begin{aligned} \left(\, \boldsymbol{x}_2 \wedge \left(\boldsymbol{T}_1^{\Delta 23} \cdot \boldsymbol{x}_1\right) + \Delta\boldsymbol{x}_2 \wedge \left(\boldsymbol{T}_1^{23} \cdot \boldsymbol{x}_1\right) \,\right) \wedge \boldsymbol{x}_3 \;&\approx\; 0 \\ \left(\boldsymbol{l}_2^\top \left(\boldsymbol{T}_1^{\Delta 23} \wedge \boldsymbol{l}_1\right) + \Delta\boldsymbol{l}_2^\top \left(\boldsymbol{T}_1^{23} \wedge \boldsymbol{l}_1\right) \right) \boldsymbol{l}_3 \;&\approx\; 0 \\ \left(\boldsymbol{l}_2^\top \left(\boldsymbol{T}_1^{\Delta 23} \cdot \boldsymbol{x}_1\right) + \Delta\boldsymbol{l}_2^\top \left(\boldsymbol{T}_1^{23} \cdot \boldsymbol{x}_1\right) \right) \boldsymbol{l}_3 \;&\approx\; 0 \\ \boldsymbol{x}_2 \wedge \left(\, \boldsymbol{T}_{\Delta 1}^{23} \cdot \boldsymbol{x}_1 + \boldsymbol{T}_1^{23} \cdot \Delta\boldsymbol{x}_1 \,\right) \wedge \boldsymbol{x}_3 \;&\approx\; 0 \\ \boldsymbol{l}_2^\top \left(\, \boldsymbol{T}_{\Delta 1}^{23} \wedge \boldsymbol{l}_1 + \boldsymbol{T}_1^{23} \wedge \Delta\boldsymbol{l}_1 \,\right) \boldsymbol{l}_3 \;&\approx\; 0 \\ \boldsymbol{l}_2^\top \left(\, \boldsymbol{T}_{\Delta 1}^{23} \cdot \boldsymbol{x}_1 + \boldsymbol{T}_1^{23} \cdot \Delta\boldsymbol{x}_1 \,\right) \boldsymbol{l}_3 \;&\approx\; 0 \end{aligned}$$

As in the two image case, the 27 components of $\boldsymbol{T}_1^{\Delta 23}$ or $\boldsymbol{T}_{\Delta 1}^{23}$ can be estimated linearly from the constraints, modulo a multiple of $\boldsymbol{T}_1^{23}$. However this is a gross overparametrization as the unknown projections $\Delta\mathbf{P}_{2'}, \Delta\mathbf{P}_{1'}$ have only 12 d.o.f. apiece. We need to constrain the $\Delta\boldsymbol{T}$'s to respect the constancy of the constant $\mathbf{P}$'s involved. This is possible using inter-tensor consistency constraints, *e.g.* for $\boldsymbol{T}_1^{2'3}$ use either of

$$\begin{aligned} \boldsymbol{T}_{A1}^{B2'\,C3}\, \boldsymbol{F}_{B1\,C3} \;+\; (A1 \leftrightarrow B1) \;&=\; 0 \\ \boldsymbol{\epsilon}_{A3\,B3\,C3}\, \boldsymbol{T}_{A1}^{A2'\,A3}\, \boldsymbol{T}_{B1}^{B2\,B3}\, \boldsymbol{e}_1^{C3} \;+\; (A1 \leftrightarrow B1) \;&=\; 0 \end{aligned}$$

where as usual $\boldsymbol{T}_1^{2'3} \approx \boldsymbol{T}_1^{23} + \boldsymbol{T}_1^{\Delta 23}$. But this whole approach seems over-complicated. Given that $\boldsymbol{T}_1^{23}$ is actually linear in $\mathbf{P}_2$, we might as well just find a

homography-epipole decomposition [7, 11]

$$\begin{aligned}\boldsymbol{T}_1^{23} &= \boldsymbol{H}_1^2 \otimes \boldsymbol{e}_1^3 - \boldsymbol{e}_1^2 \otimes \boldsymbol{H}_1^3 \\ \left(\boldsymbol{T}_1^{23} \cdot \boldsymbol{x}_1\right) &= \left(\boldsymbol{H}_1^2 \,|\, \boldsymbol{e}_1^2\right) \begin{pmatrix} \boldsymbol{0} & \boldsymbol{x}_1 \\ -\boldsymbol{x}_1^\top & 0 \end{pmatrix} \left(\boldsymbol{H}_1^3 \,|\, \boldsymbol{e}_1^3\right)^\top\end{aligned}$$

and work directly in terms of $\mathbf{P}_i = \left(\boldsymbol{H}_1^i \,|\, \boldsymbol{e}_1^i\right)$ for $i = 1, 2, 2', 3$. As always, $\boldsymbol{H} - \boldsymbol{e}$ parametrization of $\boldsymbol{T}$ (or $\boldsymbol{F}$) is just a closet form of projective camera reconstruction, so we might as well do things properly with a clean reconstruction method, followed by conventional tracking of the moving projection using the 'linear 6 point' DLT estimator (or better). My experiments suggest that this is not only the easiest, but also the stablest and most accurate way to work — the tensor is *only* useful for the initial reconstruction. *I.e.*, tracking of the trifocal tensor is certainly possible, but I have not found any advantage over conventional projection matrix tracking.

5.1 Coincident Images & Degeneracy

Now we study what happens to the differential matching constraints when more of their images are near-coincident. When some of the cameras (or modulo image rectification, some of their centres) coincide, the discrete matching tensors either vanish or degenerate to lower degree ones

$$\begin{aligned}\boldsymbol{e}_1^1 &= \boldsymbol{0} \\ \boldsymbol{F}_{11} &= \boldsymbol{0} \\ \boldsymbol{T}_1^{12} &= \boldsymbol{\delta}_1^1 \otimes \boldsymbol{e}_1^2 \\ \boldsymbol{T}_1^{21} &= -\boldsymbol{e}_1^2 \otimes \boldsymbol{\delta}_1^1 \\ \boldsymbol{T}_1^{22} &= \boldsymbol{F}_{A1\,A2}\, \boldsymbol{\epsilon}^{A2\,B2\,C2} \\ \boldsymbol{Q}^{1123} &= \boldsymbol{T}_{A1}^{A2\,A3}\, \boldsymbol{\epsilon}^{A1\,B1\,C1}\end{aligned}$$

The corresponding matching constraints also degenerate, *e.g.* the trifocal point constraint $\boldsymbol{x}_2 \wedge \left(\boldsymbol{T}_1^{23} \cdot \boldsymbol{x}_1\right) \wedge \boldsymbol{x}_3 = 0$ becomes $\left(\boldsymbol{x}_1^\top \boldsymbol{F}_{12}\, \boldsymbol{x}_2\right)\, [\,\boldsymbol{x}_2\,]_\times = 0$ for $\mathbf{P}_3 \to \mathbf{P}_2$ and vanishes for $\mathbf{P}_3 \to \mathbf{P}_1$. Similarly, some the differential matching tensors degenerate to lower degree ones when their base images coincide

$$\begin{aligned}-\boldsymbol{e}_{\Delta 1}^1 &= \boldsymbol{e}_1^{\Delta 1} = \boldsymbol{e}_1^{1'} \\ \boldsymbol{F}_{1\Delta 1} &= \left[\,\boldsymbol{e}_1^{\Delta 1}\,\right]_\times = \left[\,\boldsymbol{e}_1^{1'}\,\right]_\times \\ \boldsymbol{T}_1^{1\Delta 2} &= \boldsymbol{\delta}_1^1 \otimes \boldsymbol{e}_1^{\Delta 2} \\ \boldsymbol{T}_{\Delta 1}^{12} &= \boldsymbol{\delta}_1^1 \otimes \boldsymbol{e}_{\Delta 1}^2 - \boldsymbol{T}_1^{\Delta 12}\end{aligned}$$

Coincidence also produces redundancies between various differential tensors, *e.g.* $\boldsymbol{F}_{A1\,\Delta A2} = \frac{1}{2!}\, \boldsymbol{\epsilon}_{A2\,B2\,C2}\, \boldsymbol{T}_{A1}^{B2\,\Delta C2}$. We will silently adopt whichever form is the most convenient.

Differential Epipolar Constraint: If $\mathbf{P}_1$ and $\mathbf{P}_2$ coincide, $\boldsymbol{F}_{12}$ vanishes and $\boldsymbol{F}_{1\Delta 2}$ reduces to $\left[\,\boldsymbol{e}_1^{1'}\,\right]_\times$. We relabel $1' \to 2$ for clarity, *i.e.* $\Delta\mathbf{P}_1 = \mathbf{P}_2 - \mathbf{P}_1$. The *exact* expansion of $\boldsymbol{F}_{12}$ is

$$\begin{aligned}\boldsymbol{F}_{12} &= \boldsymbol{F}_{11} + \boldsymbol{F}_{1\Delta 1} + \boldsymbol{F}_{1\Delta^2 1} \\ &= 0 + \left[\,\boldsymbol{e}_1^2\,\right]_\times + \boldsymbol{F}_{1\Delta^2 1}\end{aligned}$$

The leading term is skew so the epipolar constraint vanishes to first order. The second order term is Viéville & Faugeras' **'first order' motion equation** [16, 17] :

$$\boldsymbol{x}_1^\top\, \boldsymbol{F}_{12}^{(s)}\, \boldsymbol{x}_1 \;+\; \boldsymbol{x}_1^\top \left[\,\boldsymbol{e}_1^2\,\right]_\times \Delta\boldsymbol{x}_1 \;\approx\; 0 \qquad (5)$$

where $\boldsymbol{F}_{12}^{(s)} \equiv \frac{1}{2}\left(\boldsymbol{F}_{12} + \boldsymbol{F}_{12}^\top\right)$ is the symmetric part of $\boldsymbol{F}_{12}$ or $\boldsymbol{F}_{1\Delta^2 1}$. The constraint uses only $\boldsymbol{e}_1^2$ and $\boldsymbol{F}_{12}^{(s)}$ so it has $3 + 6 = 9$ linearly independent components, modulo joint overall rescaling and the consistency constraint $\det(\boldsymbol{F}) = 0$ which becomes ${\boldsymbol{e}_1^2}^\top\, \boldsymbol{F}_{12}^{(s)}\, \boldsymbol{e}_1^2 = 0$. Like $\det(\boldsymbol{F}_{12}) = 0$, this is cubic in the unknowns. The linearization base point $\boldsymbol{F}_{11}$ vanishes, so we can no longer linearize the consistency constraint and error model. Hence, the differential method has about the same degree of complexity and nonlinearity as direct estimation of $\boldsymbol{F}_{12}$. Normalizing $\left(\boldsymbol{F}_{12}^{(s)}, \boldsymbol{e}_1^2\right)$ so that $\|\boldsymbol{e}_1^2\| = 1$, we can recover $\boldsymbol{F}_{12}$ from

$$\begin{aligned}\boldsymbol{F}_{12} &= [\,\boldsymbol{e}\,]_\times + \boldsymbol{F}^{(s)} + \boldsymbol{e}\, \left(\boldsymbol{F}^{(s)}\, \boldsymbol{e}\right)^\top - \left(\boldsymbol{F}^{(s)}\, \boldsymbol{e}\right)\, \boldsymbol{e}^\top \\ &= [\,\boldsymbol{e}\,]_\times + \left(\boldsymbol{I} + \boldsymbol{e}\,\boldsymbol{e}^\top\right) \boldsymbol{F}^{(s)} \left(\boldsymbol{I} - \boldsymbol{e}\,\boldsymbol{e}^\top\right)\end{aligned}$$

(The second form is preferred as it automatically projects onto $\boldsymbol{e}^\top \boldsymbol{F}^{(s)}\, \boldsymbol{e} = 0$). In general $\det(\boldsymbol{F}^{(s)}) \neq 0$: it vanishes iff the motion is planar or a parallel twist.

I have investigated matching and depth recovery using this differential approach, but found no practical advantage over direct '8 point' estimation of $\boldsymbol{F}_{12}$. The accuracy and stability are at best the same, and become worse whenever truncation error in (5) is above the noise level.

Trifocal Constraints: The differential trifocal constraints remain nondegenerate when two of their images coincide, but their coefficient tensors simplify. This case is especially interesting because it allows us to propagate matches from a base image

plus the current one to the next image in the sequence. To first order in Δ, both the 1–1′–2 and 1′–1–2 trifocal point, line and point-line matching constraints reduce to

$$\begin{aligned}
\boldsymbol{x}_1 \wedge \left(\boldsymbol{T}_1^{\Delta 12} \cdot \boldsymbol{x}_1\right) \wedge \boldsymbol{x}_2 \\
- \left(\boldsymbol{x}_1 \wedge \Delta\boldsymbol{x}_1\right) \left(\boldsymbol{e}_1^2 \wedge \boldsymbol{x}_2\right)^\top &\approx \boldsymbol{0} \\
\boldsymbol{l}_1^\top \left(\boldsymbol{T}_1^{\Delta 12} \wedge \boldsymbol{l}_1\right) \boldsymbol{l}_2 - \left(\boldsymbol{l}_1 \wedge \Delta\boldsymbol{l}_1\right) \left(\boldsymbol{l}_2^\top \boldsymbol{e}_1^2\right) &\approx \boldsymbol{0} \\
\boldsymbol{l}_1^\top \left(\boldsymbol{T}_1^{\Delta 12} \cdot \boldsymbol{x}_1\right) \boldsymbol{l}_2 - \left(\boldsymbol{l}_1^\top \Delta\boldsymbol{x}_1\right) \left(\boldsymbol{l}_2^\top \boldsymbol{e}_1^2\right) &\approx \boldsymbol{0}
\end{aligned}$$

Similarly, the 2–1–1′ constraints become

$$\begin{aligned}
\boldsymbol{x}_1 \wedge \left(\boldsymbol{T}_2^{\Delta 11} \cdot \boldsymbol{x}_2\right) \wedge \boldsymbol{x}_1 + \left(\boldsymbol{F}_{12}\, \boldsymbol{x}_2\right) \left(\boldsymbol{x}_1 \wedge \Delta\boldsymbol{x}_1\right)^\top \\
+ \left(\Delta\boldsymbol{x}_1^\top \boldsymbol{F}_{12}\, \boldsymbol{x}_2\right) \cdot \left[\, \boldsymbol{x}_1 \,\right]_\times &\approx \boldsymbol{0} \\
\boldsymbol{l}_1^\top \left(\boldsymbol{T}_2^{\Delta 11} \wedge \boldsymbol{l}_2\right) \boldsymbol{l}_1 - \left(\left(\boldsymbol{l}_1 \wedge \Delta\boldsymbol{l}_1\right)^\top \boldsymbol{F}_{12}\right) \wedge \boldsymbol{l}_2 &\approx \boldsymbol{0} \\
\boldsymbol{l}_1^\top \left(\boldsymbol{T}_2^{\Delta 11} \cdot \boldsymbol{x}_2\right) \boldsymbol{l}_1 - \left(\boldsymbol{l}_1 \wedge \Delta\boldsymbol{l}_1\right)^\top \boldsymbol{F}_{12}\, \boldsymbol{x}_2 &\approx \boldsymbol{0}
\end{aligned}$$

All of these are modulo the ordinary 1–2 epipolar constraint and maintenance of point-line incidence $\Delta\left(\boldsymbol{l}_1^\top \boldsymbol{x}_1\right) = \boldsymbol{l}_1^\top \Delta\boldsymbol{x}_1 + \Delta\boldsymbol{l}_1^\top \boldsymbol{x}_1 = 0$.

Once again, the tensor-based parameterization is feasible but seems overly complex. A homography-epipole one is preferable, but reduces the problem to classical reconstruction-reprojection. The parametrization can be initialized using any homography obtained from $\boldsymbol{F}_{12}$ (*e.g.* $\boldsymbol{H}_1^2 = \left[\, \boldsymbol{l}_2 \,\right]_\times \boldsymbol{F}_{21} + \boldsymbol{e}_1^2\, \boldsymbol{l}_1^\top$ for any non-epipolar $\boldsymbol{l}_1, \boldsymbol{l}_2$, or $\boldsymbol{H}_1^2 = \left[\, \boldsymbol{e}_1^2 \,\right]_\times \boldsymbol{F}_{21} + \lambda\, \boldsymbol{e}_1^2\, \boldsymbol{e}_2^{1\top}$ in a well-normalized image frame). The initial $\boldsymbol{H} - \boldsymbol{e}$ decompositions are then $\boldsymbol{T}_1^{12} = \boldsymbol{\delta}_1^1 \otimes \boldsymbol{e}_1^2 - \boldsymbol{0} \otimes \boldsymbol{H}_1^2$ and $\boldsymbol{T}_2^{11} = \boldsymbol{H}_2^1 \otimes \boldsymbol{e}_2^1 - \boldsymbol{e}_2^1 \otimes \boldsymbol{H}_2^1$.

If all three images nearly coincide, the trifocal constraints degenerate further and a 2^{nd}-order 1–1′–1″ expansion is needed. For clarity, we rename 1′,1″ to 2,3 and use our normalization freedom to replace $\boldsymbol{T}_1^{\Delta 2\Delta 3}$ with $\boldsymbol{T}_1^{23} \approx \boldsymbol{\delta}_1^1 \otimes \boldsymbol{e}_1^3 - \boldsymbol{e}_1^2 \otimes \boldsymbol{\delta}_1^1 + \boldsymbol{T}_1^{\Delta 2\Delta 3}$, giving matching constraints:

$$\begin{aligned}
\boldsymbol{x}_1 \wedge \left(\left(\boldsymbol{T}_1^{23} \cdot \boldsymbol{x}_1\right) + \Delta\boldsymbol{x}_2\, \boldsymbol{e}_1^{3\top} - \boldsymbol{e}_1^2\, \Delta\boldsymbol{x}_3^\top \right) \wedge \boldsymbol{x}_1 &\approx \boldsymbol{0} \\
\boldsymbol{l}_1^\top \left(\boldsymbol{T}_1^{(23)} \wedge \boldsymbol{l}_1\right) \boldsymbol{l}_1 + \left(\boldsymbol{l}_1 \wedge \Delta\boldsymbol{l}_2\right) \left(\boldsymbol{l}_1^\top \boldsymbol{e}_1^3\right) \\
- \left(\boldsymbol{l}_1^\top \boldsymbol{e}_1^2\right) \left(\boldsymbol{l}_1 \wedge \Delta\boldsymbol{l}_3\right) &\approx \boldsymbol{0} \\
\boldsymbol{l}_1^\top \left(\boldsymbol{T}_1^{(23)} \cdot \boldsymbol{x}_1\right) \boldsymbol{l}_1 - \left(\Delta\boldsymbol{l}_2^\top \boldsymbol{x}_1\right) \left(\boldsymbol{l}_1^\top \boldsymbol{e}_1^3\right) \\
+ \left(\boldsymbol{l}_1^\top \boldsymbol{e}_1^2\right) \left(\Delta\boldsymbol{l}_3^\top \boldsymbol{x}_1\right) &\approx \boldsymbol{0}
\end{aligned}$$

Here, $\boldsymbol{T}_1^{(23)}$ is the 18 d.o.f. symmetric part of $\boldsymbol{T}_1^{23}$ on its two upper indices. The point equation uses 24 d.o.f. of $\boldsymbol{T}_1^{23}$ plus two epipoles, so it does not seem competitive with standard finite $\boldsymbol{T}_1^{23}$ estimation. The line and point-line equations use only $\boldsymbol{T}_1^{(23)}, \boldsymbol{e}_1^2, \boldsymbol{e}_1^3$ and hence have $18 + 3 + 3 = 24$ linear parameters to estimate. The point-line equation is the basis of Stein & Shashua's 'tensor brightness constraint' [14], where the lines are local tangents to the iso-intensity contour at $\boldsymbol{x}_1$, displaced by normal flow into nearby images 2 and 3. But in this case the line-based constraints are quite ill-conditioned and they require special motion assumptions which reduce the problem to one considered by [6].

6 Conclusions

We have introduced a finite difference expansion for projective vision problems with near-coincident cameras. In contrast to Åström & Heyden's time-derivative based approach, it gives fairly manageable expansions for geometric vision problems like matching tensors and constraints, transfer and reconstruction. Here, we used it to systematically derive various differential matching constraints. Basically, three cases occur when difference expansion is used:

- For problems linear in the expanded variables, expansion is possible but redundant. This happens for most feature-based calculations once the matching tensors or homographies are known — *e.g.* feature transfer or reconstruction.
- For nonlinear, non-degenerate problems, first order difference expansion gives a useful local linearization. Consistency-constraint-satisfying, statistically-near-optimal tensor update becomes a simple constrained linear least squares problem. This is always equivalent to one step of an iterative nonlinear estimator started from the base tensor.
- For nonlinear problems where the expansion base case is degenerate, second (or higher) order expansion gives a valid but nonlinear local parametrization. This may be simpler or less nonlinear than the original one, but it is not clear that much is really gained. So far none of my experiments have shown any clear advantage for the differential approach in this case.

Future work will include experimental studies of constraint tracking in the 1′–2 and 1–1′–2 cases, and

development of analogous expansions for more constrained problems like calibrated cameras and autocalibration.

References

[1] K. Åström and A. Heyden. Multilinear constraints in the infinitessimal-time case. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 833–8, San Francisco, 1996.

[2] K. Åström and A. Heyden. Continuous time matching constraints for image streams. *Int. J. Computer Vision*, 28(1):85–96, 1998.

[3] E.P. Baltsavias. *Multiphoto Geometrically Constrained Matching*. PhD thesis, ETH-Zurich, 1992.

[4] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. Computer Vision*, 12(1):43–77, 1994.

[5] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[6] K. Hanna and N. Okamoto. Combining stereo and motion analysis for direct estimation of scene structure. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 357–65, 1993.

[7] R.I. Hartley. Lines and points in three views and the trifocal tensor. *Int. J. Computer Vision*, 22(2):125–140, 1997.

[8] A. Heyden and K. Åström. A canonical framework for sequences of images. In *IEEE Workshop on Representations of Visual Scenes*, Cambridge, MA, June 1995.

[9] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *Int. J. Computer Vision*, 12(1):5–16, 1994.

[10] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: a parallax based approach. In *Int. Conf. Pattern Recognition*, pages 685–688, 1994.

[11] A. Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 17(8):779–89, 1995.

[12] A. Shashua and M. Werman. On the trilinear tensor of three perspective views and its underlying geometry. In *IEEE Int. Conf. Computer Vision*, Boston, MA, June 1995.

[13] S. Soatto and P. Perona. Motion estimation using subspace constraints. *Int. J. Computer Vision*, 22(3):235–59, 1997.

[14] G. Stein and A. Shashua. Model-based brightness constraints: On direct estimation of structure and motion. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 400–406, 1997.

[15] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.

[16] T. Viéville and O. Faugeras. Motion analysis with a camera with unknown and possibly varying intrinsic parameters. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 750–6, Cambridge, MA, June 1995.

[17] T. Viéville and O. Faugeras. The first order expansion of motion equations in the uncalibrated case. *Computer Vision and Image Understanding*, 64(1):128–46, 1996.

Chapitre 4

Reconstruction projective

Ce chapitre décrit trois papiers sur le recouvrement à partir de plusieurs images projectives non-calibrées, de la géométrie 3D projective d'une scène statique et des caméras. On suppose que l'approche tensorielle décrite ci-dessus pour la géométrie des images multiples est familière au lecteur. Sur le plan pratique, on suppose que les primitives géométriques 2D (pour la plupart des points, mais aussi parfois des droites) ont déjà été extraites des images, et mis en correspondance entre images.

4.1 Resumé de « A Factorization-based Algorithm for Multi-image Projective Structure and Motion » – ECCV'96

Historique

Ce papier avec Peter STURM fut publié à ECCV'96 [ST96]. Il donne une méthode de reconstruction projective multi-images qui se montre très stable en pratique, et qui reste de loin ma méthode générale préférée pour ce problème. Historiquement, elle est une façon de coller ensemble des reconstructions partielles 3D obtenues par les équations « estimation des profondeurs projectives » décrites ci-dessus [Tri95]. Mais elle a été vulgarisée comme une généralisation projective de la méthode de factorisation affine de Tomasi & Kanade [TK92].

Méthode

Supposons qu'on a n points 3D $\mathbf{X}_1, \dots \mathbf{X}_n$ visibles dans m images projectives avec des matrices de projection $\mathbf{P}_1, \dots, \mathbf{P}_m$. Pour chaque paire, d'image $\mathbf{P}_i$ et de point 3D $\mathbf{X}_p$, on a un point image $\boldsymbol{x}_{ip}$ avec l'équation de projection $\lambda_{ip}\,\boldsymbol{x}_{ip} = \mathbf{P}_i\,\mathbf{X}_p$, où λ_{ip} est la profondeur / facteur d'échelle projective correspondant. On peut réunir toutes ces mn équations dans une grande système matricielle :

$$\begin{pmatrix} \lambda_{11}\,\boldsymbol{x}_{11} & \lambda_{12}\,\boldsymbol{x}_{12} & \dots & \lambda_{1n}\,\boldsymbol{x}_{1n} \\ \lambda_{21}\,\boldsymbol{x}_{21} & \lambda_{22}\,\boldsymbol{x}_{22} & \dots & \lambda_{2n}\,\boldsymbol{x}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1}\,\boldsymbol{x}_{m1} & \lambda_{m2}\,\boldsymbol{x}_{m2} & \dots & \lambda_{mn}\,\boldsymbol{x}_{mn} \end{pmatrix}_{(3m)\times n} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_m \end{pmatrix}_{(3m)\times 4} \begin{pmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_n \end{pmatrix}_{4\times n}$$

L'essentiel de la méthode est que si on peut retrouver les profondeurs projectives λ_{ip}, la matrice des $\lambda_{ip}\,\boldsymbol{x}_{ip}$ serait forcément – comme le côté droit – de rang 4. On peut toujours décomposer numériquement une telle matrice en forme du côté droit, par exemple par moyenne de la SVD (Décomposition par Valeurs Singulières). Il y a l'ambiguïté d'une transformation linéaire 4×4 non-singulière dans

cette décomposition, mais cette ambiguïté ne fait que représenter l'homographie 4×4 libre associé au choix d'un repère projectif arbitraire : toute factorisation à rang 4 donne une reconstruction 3D valable des caméras et des points, et ce dans un repère projectif.

Pour retrouver les λ_{ip}, on applique les « équations d'estimation des profondeurs projectives » [Tri], qui lie les primitives images, leurs profondeurs / facteurs d'échelle projectives, et les tenseurs d'appariement. Il existe des contraintes pour tout type de tenseur, mais ici on ne se servira que de celles de la matrice fondamentale, qui prennent la forme :

$$\boldsymbol{F}_{ij}\,(\lambda_{jp}\,\boldsymbol{x}_{jp}) \;=\; \boldsymbol{e}_{ji} \wedge (\lambda_{ip}\,\boldsymbol{x}_{ip})$$

Cette équation vectorielle impose que les droites épipôlaires des deux points correspondants coïncident, et en plus elle relie les positions relatives de ces points le long ces droites à leurs profondeurs projectives relatives. Ici c'est seulement les profondeurs projectives qu'on veut, donc on peut résoudre ces équations au moindre carrées :

$$\lambda_{ip} \;=\; \frac{(\boldsymbol{e}_{ji} \wedge \boldsymbol{x}_{ip}) \cdot (\boldsymbol{F}_{ij}\,\boldsymbol{x}_{jp})}{\|\boldsymbol{e}_{ji} \wedge \boldsymbol{x}_{ip}\|^2}\,\lambda_{jp}$$

Les matrices fondamentales sont estimées à partir des données images. On peut fixer l'échelle λ_{jp} de chaque point arbitrairement en une image, puis on enchaîne ces équations pour retrouver ses échelles correspondants dans tous les autres images. Une fois ceci fait, on construit la matrice des $\lambda_{ip}\,\boldsymbol{x}_{ip}$, et on la factorise pour extraire la reconstruction. En pratique, c'est aussi important d'appliquer une étape de renormalisation numérique qui est décrite dans le papier, afin de mieux conditionner le modèle du bruit qui est implicite au système.

Perspective

Il se trouve qu'en pratique cette méthode fonctionne très bien. Elle est certainement parmi les méthodes les plus stables et précises pour la reconstruction projective, grâce sans doute au fait qu'elle intègre d'une façon équilibrée toutes les données images à la fois. (La plupart des autres méthodes ne font qu'intégrer les données d'un nombre fixe d'images, ou se basent sur le choix d'une « image de référence » qui n'est pas intégré symétriquement aux autres). Mais cette méthode a aussi une faiblesse significative qui limite son application pratique : elle exige la visibilité et l'extraction de *tous* les points à reconstruire dans *toutes* les images à utiliser, ce qui n'est guère réaliste pour les séquences longues. Il existe plusieurs façons de contourner cette limitation fondamentale, mais aucune solution nette ne se dégage pour l'instant. Le problème de factorisation d'une matrice dont certaines éléments sont inconnus est important aussi en statistique et en traitement du signal. Il existe des algorithmes type optimisation non-linéaire [Wib76, SIR95], mais ils ont besoin d'une initialisation approximative de la structure, ce qui n'est pas le cas pour SVD.

Un aspect surprenant de la méthode de factorisation projective, c'est sa stabilité face aux incertitudes des points et des tenseurs d'entrée. Avec la méthode basée sur la matrice fondamentale, on peut enchaîner une bonne vingtaine ou trentaine d'équations de profondeur avant que cela nuise à la précision des sorties 3D. Je n'ai pas de très bonne explication, mais on peut noter que quand il y a une séquence d'images avec des géométries épipôlaires similaires entre chaque paire, les erreurs dans les profondeurs ont une forte tendance de s'annuler entre une image et la prochaine. Par exemple, si un point estimé se trouve un peu trop proche à l'épipôle, il donne une profondeur relative un peu trop petite dans cette image, mais du même fait une profondeur relative un peu trop grande dans la prochaine, et les différences ont tendance à s'annuler.

4.2 Resumé de « Factorization Methods for Projective Structure & Motion » – CVPR'96

Ce papier fut publié à CVPR'96 [Tri96a]. Il donne plusieurs raffinements au papier précédent, il y inclut une discussion préliminaire des méthodes de factorisation accélérées (spécialisées au cas de bas rang) et une comparaison expérimentale avec plusieurs autres méthodes de reconstruction projectives.

Mais sa contribution la plus importante est l'extension de la méthode de factorisation aux droites. Si on pouvait représenter chaque droite par deux points 3D le long de la droite, la reprojection de ces points donnerait deux points images sur chaque droite image, les points étant en correspondance entre les images. Quand on ne voit pas de tels points spéciaux, on peut les synthétiser : faire une choix arbitraire de deux points sur la droite dans la première image, et on coupe les droites épipôlaires de chacun de ces points dans les autres images par les images des droites d'origine. Ceci donne les points correspondants requis, qui peuvent être reconstruits comme des points normaux pour reconstituer la droite mère 3D. En plus, un tel transfert des points donne automatiquement les bons facteurs d'échelle pour la reconstruction projective, sans qu'on ait à les recouvrir explicitement. On peut aussi utiliser le tenseur trifocal comme moteur de transfert, pour le même effet. La méthode intègre des points et des droites dans la même factorisation. Elle marche bien tant que les droites 3D ne passent pas trop près des centres de projection, et donc forcément, les droites images sont éloignées des droites épipôlaires. (Dans le cas inverse, l'image d'une droite est très sensible aux perturbations 3D).

4.3 Resumé de « Linear Projective Reconstruction from Matching Tensors » – IVC'97

Ce papier fut publié en « Image & Vision Computing » [Tri97a], après la publication d'une version préliminaire à BMVC'96 [Tri96b]. Le talon d'Achille des méthodes basées sur la factorisation matricielle est qu'elles ne peuvent pas tolérer des données manquantes. Dans notre cas, *tous* les points 3D à reconstruire doivent être visibles dans *toutes* les images à utiliser ... ce qui n'est guère réaliste en pratique pour les séquences longues. Alors qu'il existe plusieurs moyens d'esquiver ce problème en pratique [TK92, SIR95, Jac97], on peut souhaiter des méthodes de reconstruction projectives qui fonctionnent même avec des données manquantes.

Cet article décrit une telle famille de méthodes, qui extraient des matrices de caméra projectives consistantes directement des tenseurs d'appariement. Les primitives image sont utilisées seulement pour estimer les tenseurs, donc les données manquantes ne présentent aucune difficulté. Une fois les matrices de projection des caméras obtenues, les primitives 3D peuvent être estimées linéairement à partir de leurs projections images respectives. Au coeur de la méthode sont les « **contraintes de clôture** » liant les tenseurs d'appariement et leurs matrices de projection génératrices. En empilant ces contraintes (tenseurs) – et sur condition d'avoir choisi de façon compatible leurs échelles relatives – on crée une grande matrice dont l'espace nul est de dimension 4 et contient les 4 colonnes de toutes les matrices de projection. Les projections elles mêmes peuvent être obtenues par la décomposition SVD ou tout autre algorithme permettant de déterminer le noyau d'une application linéaire.

Les résultats de la méthode sont en pratique plus ou moins bons, mais ils ne sont pas aussi stables que ceux de la reconstruction par factorisation. En particulier, elle échoue quand on inclut seulement les matrices fondamentales dans les contraintes *et* tous les centres optiques sont alignés. Ceci représente un échec fondamental de la représentation de la géométrie multi-caméras par matrices fondamentales, déjà bien connu dans d'autres circonstances (*ex.* [LF94]). Par contre, la reconstruction par factorisation des matrices fondamentales n'est *pas* mise en défaut par l'alignement

des centres, car elle n'élimine pas les coordonnées des points images.

A Factorization Based Algorithm for Multi-Image Projective Structure and Motion

Peter Sturm and Bill Triggs

GRAVIR-IMAG & INRIA Rhône-Alpes*
46, Avenue Félix Viallet, 38031 Grenoble, France
email: Peter.Sturm@imag.fr, Bill.Triggs@imag.fr

Abstract

We propose a method for the recovery of projective shape and motion from multiple images of a scene by the factorization of a matrix containing the images of all points in all views. This factorization is only possible when the image points are correctly scaled. The major technical contribution of this paper is a practical method for the recovery of these scalings, using only fundamental matrices and epipoles estimated from the image data. The resulting projective reconstruction algorithm runs quickly and provides accurate reconstructions. Results are presented for simulated and real images.

1 Introduction

In the last few years, the geometric and algebraic relations between uncalibrated views have found lively interest in the computer vision community. A first key result states that, from two uncalibrated views, one can recover the 3D structure of a scene up to an unknown projective transformation [Fau92, HGC92]. The information one needs to do so is entirely contained in the fundamental matrix, which represents the epipolar geometry of the 2 views.

Up to now, projective reconstruction has been investigated mainly for the case of 2 views. Faugeras [Fau92] studied projective reconstruction using 5 reference points. Hartley [HGC92] derives from the fundamental matrix 2 projection matrices, equal to the true ones up to an unknown projective transformation. These are then used to perform reconstruction by triangulation[HS94]. As for multiple images, most of the current methods [MVQ93, Har93, MM95] initially privilege a few views or points and thus do not treat all data uniformly.

Recently, multi-linear matching constraints have been discovered that extend the epipolar geometry of 2 views to 3 and 4 views. Shashua [Sha95] described the trilinear relationships between 3 views. Faugeras and Mourrain [FM95], and independently Triggs [Tri95a] have systematically studied the relationships between N images. Triggs introduced a new way of thinking about projective reconstruction. The image coordinates of the projections of a 3D point are combined into a single "joint image vector". Then, projective reconstruction consists essentially of rescaling the image coordinates in order to place the joint image vector in a certain 4-dimensional subspace of the joint image space called the *joint image*. This subspace is characterized by the multi-linear matching constraints between the views.

The projective reconstruction method we propose in this paper is based on the joint image formalism, but it is not necessary to understand this formalism to read the paper. We show that by

*This work was performed within a joint research programme between CNRS, INPG, INRIA, UJF.

rescaling the image coordinates we can obtain a *measurement matrix* (the combined image coordinates of all the points in all the images), which is of rank 4. Projective structure and motion can then be obtained by a singular value factorization of this matrix. So, in a sense this work can be considered as an extension of Tomasi-Kanade's and Poelman-Kanade's factorization methods [TK92, PK94] from affine to perspective projections.

The paper is organized as follows. (1) We motivate the idea of reconstruction through the rescaling of image coordinates. Throughout this paper we will restrict attention to the case of bilinear matching constraints (fundamental matrix), although the full theory [Tri95b] also allows tri- and quadrilinear matching constraints to be used. (2) We discuss some numerical considerations and describe the proposed projective reconstruction algorithm. (3) We show results that we have obtained with real and simulated data. (4) We conclude and discuss several open issues, which will be part of our future work.

2 Projective Reconstruction from Multiple Views

2.1 The Projective Reconstruction Problem

Suppose we have a set of n 3D points visible in m perspective images. Our goal is to recover 3D structure (point locations) and motion (camera locations) from the image measurements. We will assume no camera calibration or additional 3D information, so we will only be able to reconstruct the scene up to an overall projective transformation of the 3D space [Fau92, HGC92].

We will work in homogeneous coordinates with respect to arbitrary projective coordinate frames. Let $\mathbf{Q}_p$ be the unknown homogeneous coordinate vectors of the 3D points, $\mathbf{P}_i$ the unknown 3×4 image projection matrices, and $\mathbf{q}_{ip}$ the measured homogeneous coordinate vectors of the image points, where $p = 1, \ldots, n$ labels points and $i = 1, \ldots, m$ labels images. Each object is defined only up to an arbitrary nonzero rescaling, *e.g.* $\mathbf{Q}_p \sim \mu_p \mathbf{Q}_p$. The basic image projection equations say that — up to a set of unknown scale factors — the $\mathbf{q}_{ip}$ are the projections of the $\mathbf{Q}_p$:

$$\lambda_{ip}\mathbf{q}_{ip} \;=\; \mathbf{P}_i\mathbf{Q}_p$$

We will call the unknown scale factors λ_{ip} **projective depths**[1]. If the $\mathbf{Q}_p$ and the $\mathbf{q}_{ip}$ are chosen to have affine normalization ('weight' components equal to 1) and the $\mathbf{P}_i$ are normalized so that the vectorial part of the 'weight' component row has norm 1, the projective depths become true optical depths, *i.e.* true orthogonal distances from the focal plane of the camera.

The complete set of image projections can be gathered into a single $3m \times n$ matrix equation:

$$\mathbf{W} \;\equiv\; \begin{pmatrix} \lambda_{11}\mathbf{q}_{11} & \lambda_{12}\mathbf{q}_{12} & \cdots & \lambda_{1n}\mathbf{q}_{1n} \\ \lambda_{21}\mathbf{q}_{21} & \lambda_{22}\mathbf{q}_{22} & \cdots & \lambda_{2n}\mathbf{q}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{q}_{m1} & \lambda_{m2}\mathbf{q}_{m2} & \cdots & \lambda_{mn}\mathbf{q}_{mn} \end{pmatrix} \;=\; \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_m \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_n \end{pmatrix}$$

Notice that *with the correct projective depths* λ_{ip}, the $3m \times n$ **rescaled measurement matrix** $\mathbf{W}$ has rank at most 4. If we could recover the depths, we could apply an SVD based factorization technique similar to that used by Tomasi and Kanade [TK92] to $\mathbf{W}$, and thereby recover both 3D structure and camera motion for the scene. The main technical advance of this paper is a practical method for the recovery of the unknown projective depths, using fundamental matrices and epipoles estimated from the image data.

[1] This is not the same notion as the "projective depth" of Shashua, which is a cross ratio of distances along epipolar lines [Sha94]

Taken individually, the projective depths are arbitrary because they depend on the arbitrary scale factors chosen for the $\mathbf{P}_i$, the $\mathbf{Q}_p$ and the $\mathbf{q}_{ip}$. However taken as a whole the rescaled measurements $\mathbf{W}$ have a strong internal coherence. The overall scale of each triple of rows and each column of $\mathbf{W}$ can be chosen arbitrarily (*c.f.* the arbitrary scales of the projections $\mathbf{P}_i$ and the 3D points $\mathbf{Q}_p$), but once these $m+n$ overall scales have been fixed there is no further freedom of choice for the remaining $mn-m-n$ scale factors in λ_{ip}. Hence, the projective depths really do contain useful information.

2.2 Recovery of Projective Depths

Now we will show how the projective depths can be recovered from fundamental matrices and epipoles, modulo overall row and column rescalings. The point projection equation $\lambda_{ip}\mathbf{q}_{ip} = \mathbf{P}_i\mathbf{Q}_p$ implies that the 6×5 matrix

$$\left(\begin{array}{c|c} \mathbf{P}_i & \lambda_{ip}\mathbf{q}_{ip} \\ \mathbf{P}_j & \lambda_{jp}\mathbf{q}_{jp} \end{array}\right) = \left(\begin{array}{c|c} \mathbf{P}_i & \mathbf{P}_i\mathbf{Q}_p \\ \mathbf{P}_j & \mathbf{P}_j\mathbf{Q}_p \end{array}\right) = \left(\begin{array}{c} \mathbf{P}_i \\ \mathbf{P}_j \end{array}\right)\left(\begin{array}{c|c} \mathbf{I}_{4\times 4} & \mathbf{Q}_p \end{array}\right)$$

has rank at most 4. Hence, all of its 5×5 minors vanish. We can expand these by cofactors in the last column to get homogeneous linear equations in the components of $\lambda_{ip}\mathbf{q}_{ip}$ and $\lambda_{jp}\mathbf{q}_{jp}$. The coefficients are 4×4 determinants of projection matrix rows. These turn out to be just fundamental matrix and epipole components [Tri95a, FM95]. In particular, if abc and $a'b'c'$ are even permutations of 123 and $\mathbf{P}_i^a$ denotes row a of $\mathbf{P}_i$, we have:

$$[\mathbf{F}_{ij}]_{aa'} = \left|\begin{array}{c} \mathbf{P}_i^b \\ \mathbf{P}_i^c \\ \mathbf{P}_j^{b'} \\ \mathbf{P}_j^{c'} \end{array}\right| \qquad [\mathbf{e}_{ij}]^a = \left|\begin{array}{c} \mathbf{P}_i^a \\ \mathbf{P}_j^1 \\ \mathbf{P}_j^2 \\ \mathbf{P}_j^3 \end{array}\right| \tag{1}$$

Applying these relations to the three 5×5 determinants built from two rows of image i and three rows of image j gives the following fundamental relation between epipolar lines:

$$(\mathbf{F}_{ij}\mathbf{q}_{jp})\,\lambda_{jp} = (\mathbf{e}_{ij} \wedge \mathbf{q}_{ip})\,\lambda_{ip} \tag{2}$$

This relation says two things:

- **Equality up to scale:** The epipolar line of $\mathbf{q}_{jp}$ in image i is the line through the corresponding point $\mathbf{q}_{ip}$ and the epipole $\mathbf{e}_{ij}$. This is just a direct re-statement of the standard epipolar constraint.
- **Equality of scale factors:** If the correct projective depths are used in (2), the two terms have *exactly the same size* — the equality is exact, not just up to scale. This is the new result that allows us to recover projective depths using fundamental matrices and epipoles. Analogous results based on higher order matching tensors can be found in [Tri95b], but in this paper we will use only equation (2).

Our strategy for the recovery of projective depths is quite straightforward. Equation (2) relates the projective depths of a single 3D point in two images. By estimating a sufficient number of fundamental matrices and epipoles, we can amass a system of homogeneous linear equations that allows the complete set of projective depths of a given point to be found, up to an arbitrary overall scale factor. At a minimum, this can be done with any set of $m-1$ fundamental matrices that link the m images into a single connected graph. If additional fundamental matrices are available, the equations become redundant and (hopefully) more robust. In the limit, all $m(m-1)/2$ fundamental matrices and all $m(m-1)$ equations could be used to find the m unknown depths for each point, but this would be computationally very expensive. We are currently investigating policies for choosing

economical but robust sets of equations, but in this paper we will restrict ourselves to the simplest possible choice: the images are taken pairwise in sequence, $\mathbf{F}_{12}, \mathbf{F}_{23}, \ldots, \mathbf{F}_{m-1\,m}$.

This is almost certainly not the most robust choice, but it (or any other minimal selection) has the advantage that it makes the depth recovery equations trivial to solve. Solving the vector equation (2) in least squares for λ_{ip} in terms of λ_{jp} gives:

$$\lambda_{ip} = \frac{(\mathbf{e}_{ij} \wedge \mathbf{q}_{ip}) \cdot (\mathbf{F}_{ij}\mathbf{q}_{jp})}{\|\mathbf{e}_{ij} \wedge \mathbf{q}_{ip}\|^2}\ \lambda_{jp} \tag{3}$$

Such equations can be recursively chained together to give estimates for the complete set of depths for point p, starting from some arbitrary initial value such as $\lambda_{1p} = 1$.

However there is a flaw in the above argument: fundamental matrices and epipoles can only be recovered up to an unknown scale factor, so we do not actually know the scale factors in equations (1) or (2) after all! In fact this does not turn out to be a major problem. It is a non-issue if a minimal set of depth-recovery equations is used, because the arbitrary overall scale factor for each image can absorb the arbitrary relative scale of the $\mathbf{F}$ and $\mathbf{e}$ used to recover the projective depths for that image. However if redundant depth-recovery equations are used it is essential to choose a self-consistent scaling for the estimated fundamental matrices and epipoles. We will not describe this process here, except to mention that it is based on the quadratic identities between matching tensors described in [Tri95b].

Note that with unbalanced choices of scale for the fundamental matrices and epipoles, the average scale of the recovered depths might tend to increase or decrease exponentially during the recursive chaining process. Theoretically this is not a problem because the overall scales are arbitrary, but it could well make the factorization phase of the reconstruction algorithm numerically ill-conditioned. To counter this we re-balance the recovered matrix of projective depths after it has been built, by judicious overall row and column scalings.

2.3 Projective Shape and Motion by Factorization

Once we have obtained the projective depths, we can extract projective shape and motion from the rescaled measurement matrix $\mathbf{W}$.

Let

$$\mathbf{W} = \mathbf{U}\ \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)\ \mathbf{V}$$

be a Singular Value Decomposition (SVD) of $\mathbf{W}$, with $s = \min\{3m, n\}$ and singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s \geq 0$. Since $\mathbf{W}$ is of rank 4, the σ_i for $i > 4$ vanish. Thus, only the first 4 columns (rows) of $\mathbf{U}$ ($\mathbf{V}$) contribute to this matrix product. Let $\mathbf{U}'$ ($\mathbf{V}'$) the matrix of the first 4 columns (rows) of $\mathbf{U}$ ($\mathbf{V}$). Then,

$$\mathbf{W} = \mathbf{U}'_{3m\times 4}\ \underbrace{\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)}_{\mathbf{\Sigma}}\ \mathbf{V}'_{4\times n} = \mathbf{U}'\ \mathbf{\Sigma}\ \mathbf{V}'\ .$$

Any factorization of $\mathbf{\Sigma}$ into two 4×4 matrices $\mathbf{\Sigma}'$ and $\mathbf{\Sigma}''$, $\mathbf{\Sigma} = \mathbf{\Sigma}'\mathbf{\Sigma}''$, leads to

$$\mathbf{W} = \underbrace{\mathbf{U}'\mathbf{\Sigma}'}_{\hat{\mathbf{U}}}\underbrace{\mathbf{\Sigma}''\mathbf{V}'}_{\hat{\mathbf{V}}} = \hat{\mathbf{U}}_{3m\times 4}\hat{\mathbf{V}}_{4\times n}\ .$$

We can interpret the matrix $\hat{\mathbf{U}}$ as a collection of m (3×4) projection matrices $\hat{\mathbf{P}}_i$ and $\hat{\mathbf{V}}$ as collection of n 4-vectors $\hat{\mathbf{Q}}_p$, representing 3D shape :

$$\mathbf{W} = \hat{\mathbf{U}}\hat{\mathbf{V}} = \begin{pmatrix} \hat{\mathbf{P}}_1 \\ \hat{\mathbf{P}}_2 \\ \vdots \\ \hat{\mathbf{P}}_m \end{pmatrix}_{3m\times 4} \begin{pmatrix} \hat{\mathbf{Q}}_1 & \hat{\mathbf{Q}}_2 & \cdots & \hat{\mathbf{Q}}_n \end{pmatrix}_{4\times n} \tag{4}$$

Equation (4) shows that the $\hat{\mathbf{P}}_i$ and $\hat{\mathbf{Q}}_p$ represent at least projective motion and shape, since

$$\hat{\mathbf{P}}_i\hat{\mathbf{Q}}_p = \lambda_{ip}\mathbf{q}_{ip} \sim \mathbf{q}_{ip} \quad .$$

Unlike the case of orthographic projections [TK92], there are no further constraints on the $\hat{\mathbf{P}}_i$ or $\hat{\mathbf{Q}}_p$: we can *only* recover projective shape and motion. For any non singular projective transformation $\mathbf{T}_{4\times4}$, $\hat{\mathbf{P}}_i\mathbf{T}$ and $\mathbf{T}^{-1}\hat{\mathbf{Q}}_p$ is an equally valid factorization of the data into projective motion and shape :

$$(\hat{\mathbf{P}}_i\mathbf{T})(\mathbf{T}^{-1}\hat{\mathbf{Q}}_p) = \hat{\mathbf{P}}_i\hat{\mathbf{Q}}_p \sim \mathbf{q}_{ip} \quad .$$

A consequence of this is that the factorization of $\mathbf{\Sigma}$ is arbitrary. For the implementation, we chose $\mathbf{\Sigma}' = \mathbf{\Sigma}'' = \mathbf{\Sigma}^{1/2} = \mathrm{diag}(\sigma_1^{1/2}, \sigma_2^{1/2}, \sigma_3^{1/2}, \sigma_4^{1/2})$.

3 The Algorithm

Based on the observations made above, we have developed a practical algorithm for projective reconstruction from multiple views. Besides the major two steps, determination of the scale factors λ_{ip} and factorization of the rescaled measurement matrix, the outline of our algorithm is based on some numerical considerations.

3.1 Normalization of Image Coordinates

To ensure good numerical conditioning of the method, we work with normalized image coordinates, as described in [Har95]. This normalization consists of applying a similarity transformation (translation and uniform scaling) $\mathbf{T}_i$ to each image, so that the transformed points are centered at the origin and the mean distance from the origin is $\sqrt{2}$.

All of the remaining steps of the algorithm are done in normalized coordinates. Since we actually compute projective motion and shape for the transformed image points $\mathbf{T}_i\mathbf{q}_{ip}$, $\hat{\mathbf{P}}_i\hat{\mathbf{Q}}_p = \lambda_{ip}\mathbf{T}_i\mathbf{q}_{ip} \sim \mathbf{T}_i\mathbf{q}_{ip}$, the resulting projection estimates $\hat{\mathbf{P}}_i$ must be corrected : $\hat{\mathbf{P}}_i{}' = \mathbf{T}_i^{-1}\hat{\mathbf{P}}_i$. The $\hat{\mathbf{P}}_i{}'$ and $\hat{\mathbf{Q}}_p$ then represent projective motion and shape corresponding to the measured image points $\mathbf{q}_{ip}$.

Our results show that this simple normalization drastically improves the results of the projective reconstruction.

3.2 Balancing the Rescaled Measurement Matrix

Consider the factorization of the rescaled measurement matrix W in projective motion and shape :

$$\mathbf{W} = \begin{pmatrix} \lambda_{11}\mathbf{q}_{11} & \lambda_{12}\mathbf{q}_{12} & \cdots & \lambda_{1n}\mathbf{q}_{1n} \\ \lambda_{21}\mathbf{q}_{21} & \lambda_{22}\mathbf{q}_{22} & \cdots & \lambda_{2n}\mathbf{q}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{q}_{m1} & \lambda_{m2}\mathbf{q}_{m2} & \cdots & \lambda_{mn}\mathbf{q}_{mn} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{P}}_1 \\ \hat{\mathbf{P}}_2 \\ \vdots \\ \hat{\mathbf{P}}_m \end{pmatrix} \begin{pmatrix} \hat{\mathbf{Q}}_1 & \hat{\mathbf{Q}}_2 & \cdots & \hat{\mathbf{Q}}_n \end{pmatrix}$$

Multiplying column l of $\mathbf{W}$ by a non zero scalar ν_l corresponds to multiplying $\hat{\mathbf{Q}}_l$ by ν_l. Analogously, multiplying the image k rows $(3k-2, 3k-1, 3k)$ by a non zero scalar μ_k corresponds to multiplying the projection matrix $\hat{\mathbf{P}}_k$ by μ_k. Hence, point-wise and image-wise rescalings of $\mathbf{W}$ do not affect the recovered projective motion and shape.

However, these considerations are only valid in the absence of noise. In presence of noise, $\mathbf{W}$ will only be approximately of rank 4, and scalar multiplications of $\mathbf{W}$ as described above *will* affect

the results. We therefore aim to improve the results of the factorization by applying appropriate point- and image-wise rescalings to $\mathbf{W}$. The goal is to ensure good numerical conditioning by rescaling so that all rows and columns of $\mathbf{W}$ have on average the same order of magnitude. To do this we use the following iterative scheme :

1. Rescale each column l so that $\sum_{r=1}^{3m}(w_{rl})^2 = 1$.
2. Rescale each triplet of rows $(3k-2, 3k-1, 3k)$ so that $\sum_{l=1}^{n}\sum_{i=3k-2}^{3k} w_{il}^2 = 1$.
3. If the entries of $\mathbf{W}$ changed significantly, repeat 1 and 2.

Note that, since we work with normalized image coordinates $\mathbf{q}_{ip}$, it would be sufficient to balance only the $m \times n$ matrix (λ_{ip}) instead of $\mathbf{W}$.

3.3 Outline of the Algorithm

The complete algorithm is composed of the following steps.

1. Normalize the image coordinates, by applying transformations $\mathbf{T}_i$.
2. Estimate the fundamental matrices and epipoles with the method of [Har95].
3. Determine the scale factors λ_{ip} using equation (3).
4. Build the rescaled measurement matrix $\mathbf{W}$.
5. Balance $\mathbf{W}$ by column-wise and "triplet-of-rows"-wise scalar mutliplications.
6. Compute the SVD of the balanced matrix $\mathbf{W}$.
7. From the SVD, recover projective motion and shape.
8. Adapt projective motion, to account for the normalization transformations $\mathbf{T}_i$ of step 1.

4 Experimental Evaluation of the Algorithm

4.1 Experiments with Simulated Images

We conducted a large number of experiments with simulated images to quantify the performance of the algorithm. The simulations used three different configurations : lateral movement of a camera, movement towards the scene, and a circular movement around the scene (see figure 1). In configuration 2, the depths of points lying on the line joining the projection centers can not be recovered. Reconstruction of points lying close to this line is extremely difficult, as was confirmed by the experiments, which resulted in quite inaccurate reconstructions for this configuration.

For the circular movement, the overall trajectory of the camera formed a quarter circle, centered on the scene. For each specific experiment, the trajectory length was the same for all three configurations. The m different viewing positions were equidistantly distributed along the trajectory.

In order to simulate realistic situations, we adopted the following parameters : the camera's calibration matrix was $\mathrm{diag}(1000, 1000, 1)$. The scene was composed of points distributed uniformly in a sphere of radius 100. The distance between the camera and the center of the sphere was 200 (for configuration 2 this was the distance with respect to the view m).

For each configuration, the following experiment was conducted 50 times :

1. Determine at random 50 points in the sphere.

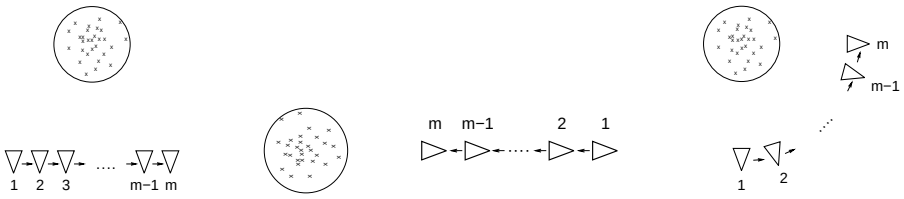


Figure 1: **The 3 configurations for simulation. (1)** *Lateral movement.* **(2)** *Translation towards the scene.* **(3)** *Circular movement.*

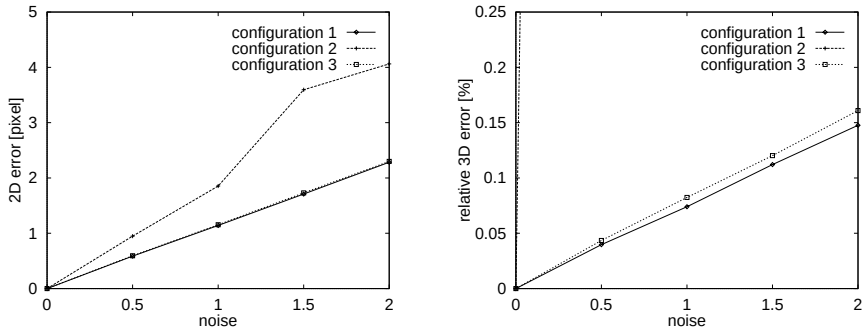
2. Project the points into the m views.
3. Add Gaussian noise of levels $0.0, 0.5, \ldots, 2.0$ to the image coordinates.
4. Carry out projective reconstruction with our algorithm.
5. Compute the image distance error of the backprojected points (2D error) : $\frac{1}{mn}\sum_{i=1}^{m}\sum_{p=1}^{n}\|\hat{\mathbf{P}}_i\hat{\mathbf{Q}}_p - \mathbf{q}_{ip}\|$, where $\|.\|$ means the Euclidean vector norm.
6. Align the projective reconstruction with the Euclidean model and compute the distance error in the Euclidean frame (3D error).

The results of these experiments were analyzed with respect to several variables, as reported in the following subsections. All values represented in the graphs are the mean result over 50 trials. To monitor the effect of outliers on the results, we also computed the median values. These gave graphs similar to those for the means, which we will not show here.

2D errors are given in pixels, whereas 3D errors are given relative to the scene's size, in percent.

4.1.1 Sensitivity to Noise

Graphs 1 and 2 show the behavior of the algorithm with respect to different noise levels for the three configurations. For this experiment, reconstruction was done from 10 views.

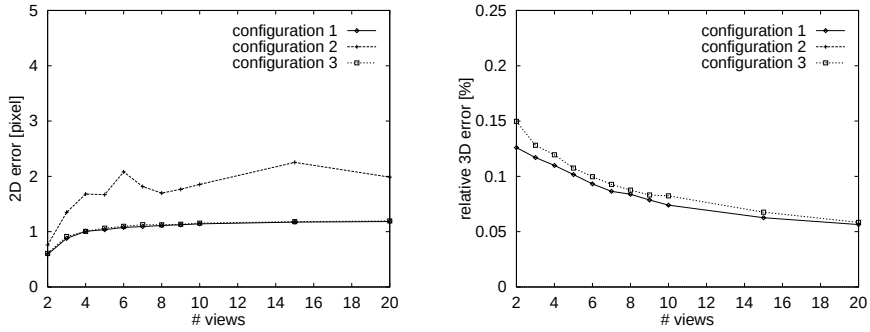


Graphs 1 and 2 : Sensitivity to noise. *The 2D error curves for the configurations 1 and 3 are nearly undistinguishable. 3D error for configuration 2 goes rapidly off scale.*

The algorithm performed almost equally well for configurations 1 and 3, whereas the 3D error for configuration 2 exceeds 100 % for 2.0 pixels noise. Considering the graphs of configuration 2, we also see that 2D and 3D error are not always well correlated. For configurations 1 and 3, the 2D error is of the same order as pixel noise. Note also the linear shape of the graphs.

4.1.2 Number of Views

The image noise for this experiment was 1.0 pixel.

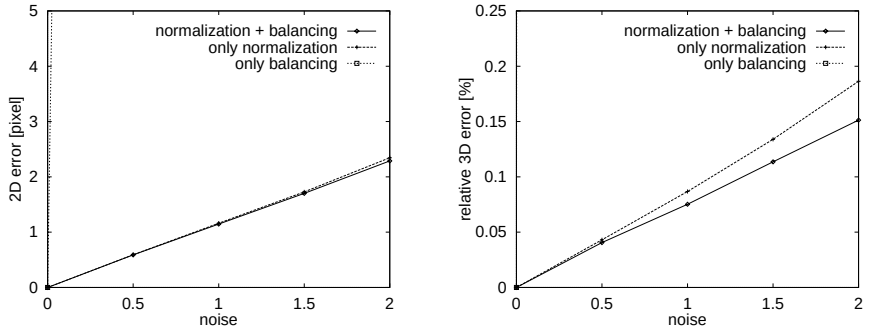


Graphs 3 and 4 : Behavior with respect to number of views. *The 2D error curves for the configurations 1 and 3 are nearly undistinguishable. The 3D error for configuration 2 lies above 5 %. The curve is thus not visible in the graph.*

The graphs show the expected behavior : when more views are used for reconstruction, the structure is recovered more accurately. Secondly, 2D error augments with increasing number of views, but shows a clearly asymptotic behavior. 1. Note that the use of 20 views reduces the 3D error to 50 % of that for 2 views.

4.1.3 Importance of Normalization and Balancing

The error values in the previous graphs were obtained with the algorithm as described in subsection 3.3. To underline the importance of using normalized image coordinates, we also ran the algorithm using unnormalized ones. The effects of not balancing the rescaled measurement matrix before factorization were also examined.



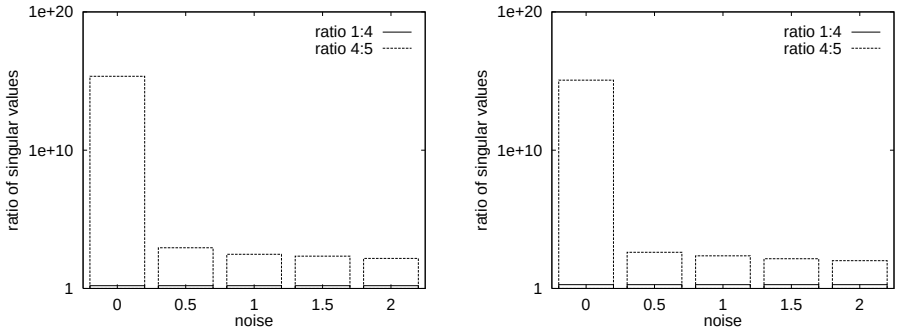
Graphs 5 and 6 : Influence of normalization and balancing. *The results presented here were obtained for configuration 1. The 2D error curve for "only balancing" goes off scale even for 0.5 pixels noise and the 3D curve is so steep that it is not even visible.*

When the image coordinates are not normalized, the error is already off scale for 0.5 pixel noise. An explanation for this is the bad conditioning of the rescaled measurement matrix (see also next paragraph). As for balancing, we see that this improves 3D errors up to 20 %, and hence should always be part of the algorithm.

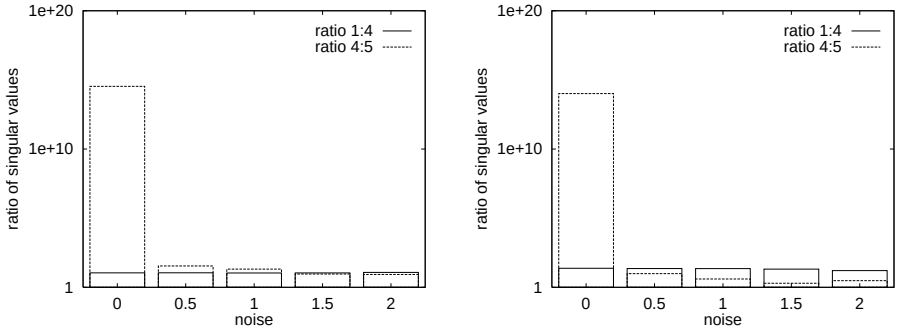
4.1.4 Robustness of the Factorization

The applicability of our factorization method is based on the rank 4-ness of the rescaled measurement matrix $\mathbf{W}$ (in the noiseless case). To test the robustness of this property, we evaluated how

close $\mathbf{W}$ is to rank 4 in practice. To be close to rank 4, the ratio of the 4th and 5th largest singular values , $\sigma_4 : \sigma_5$, should be large with respect to the ratio of the 1st and 4th largest, $\sigma_1 : \sigma_4$. In the following graphs, these two ratios are represented, for configurations 1 and 2 and for 2 and 20 views. Note that the y-axes are scaled logarithmically.



Graphs 7 and 8 : Ratios of singular values for configuration 1. *The graph on the left shows the situation for 2 views, on the right for 20 views.*



Graphs 9 and 10 : Ratios of singular values for configuration 2. *The graph on the left shows the situation for 2 views, on the right for 20 views.*

We see that for configuration 1, the matrix is always very close to rank 4 : $(\sigma_1 : \sigma_4)$ is lower than 2, whereas $(\sigma_4 : \sigma_5)$ lies clearly above 100. As for configuration 2, the graphs reflect the bad performance in 3D reconstruction. $(\sigma_1 : \sigma_4)$ is about 10, while for high noise levels or many views $(\sigma_4 : \sigma_5)$ is close to 1.

4.2 Evaluation with Real Images

The algorithm has also been tested on several sequences of real images. For 2 of them we show results.

4.2.1 The House Sequence

Figure 2 shows the first and last image of a sequence of 6 images of a scene with a wooden house. 38 points were tracked over the whole sequence, but only extracted with ± 1 pixel accuracy.

To estimate the quality of the projective reconstruction, we aligned it with an approximate Euclidean model of the scene obtained from calibrated views (see figure 3). Lines have been drawn between some of the points to aid visualization.

In the side and front views we see that right angles are approximately conserved, and that the windows are coplanar with the wall. The bumpiness on the left side of the roof is due to the fact that the roof stands out slightly from the house's front wall (see figure 2), thus causing occlusion in the last view of the edge point between roof and wall.

Figure 2: **First and last image of the house sequence and one image of the castle sequence.**

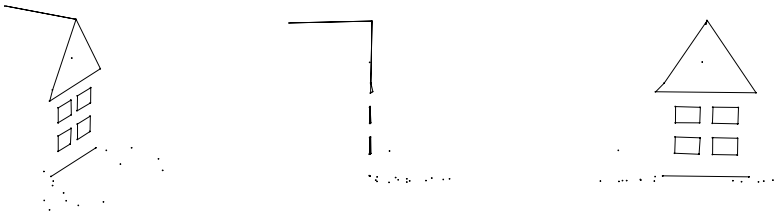


Figure 3: **Three views of the reconstructed house. (1)** *"General view"*. **(2)** *Side view.* **(3)** *Front view.*

4.2.2 The Castle Sequence

28 points have been tracked through the 11 images of the scene shown in the right part of figure 2. 3D ground truth is available, and the reconstruction errors have been evaluated quantitatively. The projective reconstruction was aligned with the Euclidean model and the resulting RMS error was 4.45 mm for an object size of about $220\text{mm} \times 210\text{mm} \times 280\text{mm}$. The RMS error of the reprojected structure with respect to the measured image points was less than 0.02 pixels.

We also applied a Levenberg-Marquardt nonlinear least-squares estimation algorithm, with the results of our method as initialization. This slightly improved the 2D reprojection error, however the 3D reconstruction error was not significantly changed.

5 Discussion and Further Work

In this paper, we have proposed a method of projective reconstruction from multiple uncalibrated images. The method is very elegant, recovering shape and motion by factorization of one matrix, containing all image points of all views. This factorization is only possible when the image points are correctly scaled. We have proposed a very simple way to obtain the individual scale factors, using only fundamental matrices and epipoles estimated from the image data.

The algorithm proves to work well with real images. Quantitative evaluation by numerical simulations shows the robustness of the factorization and the good performance with respect to noise. The results also show that it is essential to work with normalized image coordinates.

Some aspects of the method remain to be examined. In the current implementation, we recover projective depths by chaining equation (2) for pairs of views $(12), (23), \ldots, (m-1, m)$. However, it would be worth investigating whether other kinds of chaining are not more stable. Furthermore, uncertainty estimates on the fundamental matrices should be considered when choosing which of the equations (2) to use. To run the algorithm in practice, it should also be able to treat points which are not visible in all images. Finally the method could be extended to use trilinear and perhaps even quadrilinear matching tensors.

Acknowledgements. This work was partially supported by INRIA France and E.C. projects HCM and SECOND. Data for this research were partially provided by the Calibrated Imaging Laboratory at Carnegie Mellon University, supported by ARPA, NSF, and NASA (the castle sequence can be found at http://www.cs.cmu.edu/ cil/cil-ster.html).

References

[Fau92] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. 2nd ECCV, Santa Margherita Ligure, Italy*, pages 563–578, May 1992.

[FM95] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proc. 5th ICCV, Cambridge, Massachusetts*, pages 951–956, June 1995.

[Har93] R.I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proc.* DARPA–ESPRIT *Workshop on Applications of Invariants in Computer Vision, Azores, Portugal*, pages 187–202, October 1993.

[Har95] R. Hartley. In Defence of the 8-point Algorithm. In *Proc. 5th ICCV, Cambridge, Massachusetts*, pages 1064–1070, June 1995.

[HGC92] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. CVPR, Urbana-Champaign, Illinois*, pages 761–764, 1992.

[HS94] R. Hartley and P. Sturm. Triangulation. In *Proc.* ARPA *IUW, Monterey, California*, pages 957–966, November 1994.

[MM95] P.F. McLauchlan and D.W. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proc. 5th ICCV, Cambridge, Massachusetts*, pages 314–320, June 1995.

[MVQ93] R. Mohr, F. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. In *Proc. CVPR, New York*, pages 543–548, June 1993.

[PK94] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In *Proc. 3rd ECCV, Stockholm, Sweden*, pages 97–108, May 1994.

[Sha94] A. Shashua. Projective structure from uncalibrated images: Structure from motion and recognition. IEEE *Trans. on PAMI*, 16(8):778–790, August 1994.

[Sha95] A. Shashua. Algebraic functions for recognition. IEEE *Trans. on PAMI*, 17(8):779–789, August 1995.

[TK92] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137–154, 1992.

[Tri95a] B. Triggs. Matching constraints and the joint image. In *Proc. 5th ICCV, Cambridge, Massachusetts*, pages 338–343, June 1995.

[Tri95b] B. Triggs. The geometry of projective reconstruction I: matching constraints and the joint image. *IJCV*, 1995. submitted.

Factorization Methods for Projective Structure and Motion

Bill Triggs
INRIA Rhône-Alpes,
655 avenue de l'Europe, 38330 Montbonnot Saint-Martin, France.
Bill.Triggs@inrialpes.fr http://www.inrialpes.fr/MOVI/Triggs

Abstract

This paper describes a family of factorization-based algorithms that recover 3D projective structure and motion from multiple uncalibrated perspective images of 3D points and lines. They can be viewed as generalizations of the Tomasi-Kanade algorithm from affine to fully perspective cameras, and from points to lines. They make no restrictive assumptions about scene or camera geometry, and unlike most existing reconstruction methods they do not rely on 'privileged' points or images. All of the available image data is used, and each feature in each image is treated uniformly. The key to projective factorization is the recovery of a consistent set of *projective depths* (scale factors) for the image points: this is done using fundamental matrices and epipoles estimated from the image data. We compare the performance of the new techniques with several existing ones, and also describe an approximate factorization method that gives similar results to SVD-based factorization, but runs much more quickly for large problems.

Keywords: Multi-image Structure, Projective Reconstruction, Matrix Factorization.

1 Introduction

There has been considerable progress on scene reconstruction from multiple images in the last few years, aimed at applications ranging from very precise industrial measurement systems with several fixed cameras, to approximate structure and motion from real time video for active robot navigation. One can usefully begin by ignoring the issues of camera calibration and metric structure, initially recovering the scene up to an overall projective transformation and only later adding metric information if needed [5, 10, 1]. The key result is that projective reconstruction is the best that can be done without calibration or metric information about the scene, and that it is possible from at least two views of point-scenes or three views of line-scenes [2, 3, 8, 6].

Most current reconstruction methods either work *only* for the minimal number of views (typically two), or single out a few 'privileged' views for initialization before bootstrapping themselves to the multi-view case [5, 10, 9]. For robustness and accuracy, there is a need for methods that uniformly take account of all the data in all the images, without making restrictive special assumptions or relying on privileged features or images for initialization. The orthographic and paraperspective structure/motion factorization methods of Tomasi, Kanade and Poelman [17, 11] partially fulfill these requirements, but they only apply when the camera projections are well approximated by affine mappings. This happens only for cameras viewing small, distant scenes, which is seldom the case in practice. Factorization methods for perspective images are needed, however it has not been clear how to find the unknown projective scale factors of the image measurements that are required for this. (In the affine case the scales are constant and can be eliminated).

As part of the current blossoming of interest in multi-image reconstruction, Shashua [14] recently extended the well-known two-image epipolar constraint to a trilinear constraint between matching points in three images. Hartley [6] showed that this constraint also applies to lines in three images, and Faugeras & Mourrain [4] and I [18, 19] completed that corner of the puzzle by systematically studying the constraints for lines and points in any number of images. A key aspect of the viewpoint presented in [18, 19] is that projective reconstruction is essen-

This paper appeared in CVPR'96. The work was supported by an EC HCM grant and INRIA Rhône-Alpes. I would like to thank Peter Sturm and Richard Hartley for enlightening discussions.

tially a matter of recovering a coherent set of **projective depths** — projective scale factors that represent the depth information lost during image projection. These are exactly the missing factorization scales mentioned above. They satisfy a set of consistency conditions called 'joint image reconstruction equations' [18], that link them together via the corresponding image point coordinates and the various inter-image matching tensors.

In the MOVI group, we have recently been developing projective structure and motion algorithms based on this 'projective depth' picture. Several of these methods use the factorization paradigm, and so can be viewed as generalizations of the Tomasi-Kanade method from affine to fully perspective projections. However they also require a depth recovery phase that is not present in the affine case. The basic reconstruction method for point images was introduced in [15]. The current paper extends this in several directions, and presents a detailed assessment of the performance of the new methods in comparison to existing techniques such as Tomasi-Kanade factorization and Levenberg-Marquardt nonlinear least squares. Perhaps the most significant result in the paper is the extension of the method to work for lines as well as points, but I will also show how the factorization can be iteratively 'polished' (with results similar to nonlinear least squares iteration), and how any factorization-based method can be speeded up significantly for large problems, by using an approximate fixed-rank factorization technique in place of the Singular Value Decomposition.

The factorization paradigm has two key attractions that are only enhanced by moving from the affine to the projective case: *(i)* All of the data in all of the images is treated uniformly — there is no need to single out 'privileged' features or images for special treatment; *(ii)* No initialization is required and convergence is virtually guaranteed by the nature of the numerical methods used. Factorization also has some well known disadvantages: Every primitive must be visible in every image. This is unrealistic in practice given occlusion and extraction and tracking failures. It is not possible to incorporate a full statistical error model for the image data, although some sort of implicit least-squares trade-off *is* made. It is not clear how to incorporate additional points or images incrementally: the whole calculation must be redone. SVD-based factorization is slow for large problems.

Only the speed problem will be considered here. SVD is slow because it was designed for general, full rank matrices. For matrices of fixed low rank r (as here, where the rank is 3 for the affine method or 4 for the projective one), approximate factorizations can be computed in time $\mathcal{O}(mnr)$, *i.e.* directly proportional to the size of the input data.

The Tomasi-Kanade 'hallucination' process can be used to work around missing data [17], as in the affine case. However this greatly complicates the method and dilutes some of its principal benefits. There is no obvious solution to the error modelling problem, beyond using the factorization to initialize a nonlinear least squares routine (as is done in some of the experiments below). It would probably be possible to develop incremental factorization update methods, although there do not seem to be any in the standard numerical algebra literature.

The rest of the paper outlines the theory of projective factorization for points and lines, describes the final algorithms and implementation, reports on experimental results using synthetic and real data, and concludes with a discussion. The full theory of projective depth recovery applies equally to two, three and four image matching tensors, but throughout this paper I will concentrate on the two-image (fundamental matrix) case for simplicity. The underlying theory for the higher valency cases can be found in [18].

2 Point Reconstruction

We need to recover 3D structure (point locations) and motion (camera calibrations and locations) from m uncalibrated perspective images of a scene containing n 3D points. Without further information it is only possible to reconstruct the scene up to an overall projective transformation [2, 8], so we will work in homogeneous coordinates with respect to arbitrary projective coordinate frames. Let $\mathbf{X}_{\mathrm{p}}$ ($p = 1, \ldots, n$) be the unknown homogeneous 3D point vectors, $\mathbf{P}_{\mathrm{i}}$ ($i = 1, \ldots, m$) the unknown 3×4 image projections, and $\mathbf{x}_{\mathrm{ip}}$ the measured homogeneous image point vectors. Modulo some scale factors λ_{ip}, the image points are projected from the world points: $\lambda_{\mathrm{ip}}\,\mathbf{x}_{\mathrm{ip}} = \mathbf{P}_{\mathrm{i}}\,\mathbf{X}_{\mathrm{p}}$. Each object is defined only up to rescaling. The λ's 'cancel out' the arbitrary scales of the image points, but there is

still the freedom to: (*i*) arbitrarily rescale each world point $\mathbf{X}_{\mathrm{p}}$ and each projection $\mathbf{P}_{\mathrm{i}}$; (*ii*) apply an arbitrary nonsingular 4×4 projective deformation $\mathbf{T}$: $\mathbf{X}_{\mathrm{p}} \rightarrow \mathbf{T}\mathbf{X}_{\mathrm{p}}$, $\mathbf{P}_{\mathrm{i}} \rightarrow \mathbf{P}_{\mathrm{i}}\mathbf{T}^{-1}$. Modulo changes of the λ_{ip}, the image projections are invariant under both of these transformations.

The scale factors λ_{ip} will be called **projective depths**. With correctly normalized points and projections they become true optical depths, *i.e.* orthogonal distances from the focal planes of the cameras. (NB: this is not the same as Shashua's 'projective depth' [13]). In general, $m+n-1$ projective depths can be set arbitrarily by choosing appropriate scales for the $\mathbf{X}_{\mathrm{p}}$ and $\mathbf{P}_{\mathrm{i}}$. However, once this is done the remaining $(m-1)(n-1)$ degrees of freedom contain real information that can be used for 3D reconstruction: taken as a whole the projective depths have a strong internal coherence. In fact, [18, 19] argues that just as the key to calibrated stereo reconstruction is the recovery of Euclidean depth, the essence of projective reconstruction is precisely the recovery of a coherent set of projective depths modulo overall projection and world point rescalings. Once this is done, reconstruction reduces to choosing a projective basis for a certain abstract three dimensional 'joint image' subspace, and reading off point coordinates with respect to it.

2.1 Factorization

Gather the point projections into a single $3m \times n$ matrix equation:

$$\mathbf{W} \equiv \begin{pmatrix} \lambda_{11}\,\mathbf{x}_{11} & \lambda_{12}\,\mathbf{x}_{12} & \cdots & \lambda_{1\mathrm{n}}\,\mathbf{x}_{1\mathrm{n}} \\ \lambda_{21}\,\mathbf{x}_{21} & \lambda_{22}\,\mathbf{x}_{22} & \cdots & \lambda_{2\mathrm{n}}\,\mathbf{x}_{2\mathrm{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{\mathrm{m}1}\,\mathbf{x}_{\mathrm{m}1} & \lambda_{\mathrm{m}2}\,\mathbf{x}_{\mathrm{m}2} & \cdots & \lambda_{\mathrm{mn}}\,\mathbf{x}_{\mathrm{mn}} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{\mathrm{m}} \end{pmatrix} \begin{pmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{\mathrm{n}} \end{pmatrix}$$

Hence, with a consistent set of projective depths the **rescaled measurement matrix $\mathbf{W}$** has rank at most 4. Any rank 4 matrix can be factorized into some $3m \times 4$ matrix of 'projections' multiplying a $4 \times n$ matrix of 'points' as shown, and any such factorization corresponds to a valid projective reconstruction: the freedom in factorization is exactly a 4×4 nonsingular linear transformation $\mathbf{P} \rightarrow \mathbf{P}\,\mathbf{T}^{-1}$, $\mathbf{X} \rightarrow \mathbf{T}\,\mathbf{X}$, which can be regarded as a projective transformation of the reconstructed 3D space.

One practical method of factorizing $\mathbf{W}$ is the Singular Value Decomposition [12]. This decomposes an arbitrary $k \times l$ matrix $\mathbf{W}_{\mathrm{k}\times\mathrm{l}}$ of rank r into a product $\mathbf{W}_{\mathrm{k}\times\mathrm{l}} = \mathbf{U}_{\mathrm{k}\times\mathrm{r}}\,\mathbf{D}_{\mathrm{r}\times\mathrm{r}}\,\mathbf{V}_{\mathrm{l}\times\mathrm{r}}^{\top}$, where the columns of $\mathbf{V}_{\mathrm{l}\times\mathrm{r}}$ and $\mathbf{U}_{\mathrm{k}\times\mathrm{r}}$ are orthonormal bases for the input (co-kernel) and output (range) spaces of $\mathbf{W}_{\mathrm{k}\times\mathrm{l}}$, and $\mathbf{D}_{\mathrm{r}\times\mathrm{r}}$ is a diagonal matrix of positive decreasing 'singular values'. The decomposition is unique when the singular values are distinct, and can be computed stably and reliably in time $\mathcal{O}(kl\min(k,l))$. The matrix $\mathbf{D}$ of singular values can be absorbed into either $\mathbf{U}$ or $\mathbf{V}$ to give a decomposition of the projection/point form $\mathbf{P}\mathbf{X}$. (I absorb it into $\mathbf{V}$ to form $\mathbf{X}$).

The SVD has been used by Tomasi, Kanade and Poelman [17, 11] for their affine (orthographic and paraperspective) reconstruction techniques. The current application can be viewed as a generalization of these methods to projective reconstruction. The projective case leads to slightly larger matrices ($3m \times n$ rank 4 as opposed to $2m \times n$ rank 3), but is actually simpler than the affine case as there is no need to subtract translation terms or apply nonlinear constraints to guarantee the orthogonality of the projection matrices.

Ideally, one would like to find reconstructions in time $\mathcal{O}(mn)$ (the size of the input data). SVD is a factor of $\mathcal{O}(\min(3m,n))$ slower than this, which can be significant if there are many points and images. Although SVD is probably near-optimal for full-rank matrices, rank r matrices can be factorized in 'output sensitive' time $\mathcal{O}(mnr)$. I have experimented with one such 'fixed rank' method, and find it to be almost as accurate as SVD and significantly faster for large problems. The method repeatedly sweeps the matrix, at each sweep guessing and subtracting a column-vector that 'explains' as much as possible of the residual error in the matrix columns. A rank r matrix is factorized in r sweeps. When the matrix is not exactly of rank r the guesses are not quite optimal and it is useful to include further sweeps (say $2r$ in total) and then SVD the matrix of extracted columns to estimate the best r combinations of them.

2.2 Projective Depth Recovery

The above factorization techniques can *only* be used if a self-consistent set of projective depths λ_{ip} can be found. The key technical advance that makes this work possible is a practical method for estimating these using fundamental matrices and epipoles obtained from the image data. The full theory can be found in [18], which also describes how to use trivalent and quadrivalent matching tensors for depth recovery. Here we briefly sketch the fundamental matrix case. The image projections $\lambda_{\mathrm{ip}}\,\mathbf{x}_{\mathrm{ip}} = \mathbf{P}_{\mathrm{i}}\,\mathbf{X}_{\mathrm{p}}$ imply that the 6×5 matrix

$$\left(\begin{array}{c|c} \mathbf{P}_{\mathrm{i}} & \lambda_{\mathrm{ip}}\,\mathbf{x}_{\mathrm{ip}} \\ \mathbf{P}_{\mathrm{j}} & \lambda_{\mathrm{jp}}\,\mathbf{x}_{\mathrm{jp}} \end{array}\right) = \left(\begin{array}{c} \mathbf{P}_{\mathrm{i}} \\ \mathbf{P}_{\mathrm{j}} \end{array}\right) \left(\begin{array}{c|c} \mathbf{I}_{4\times 4} & \mathbf{X}_{\mathrm{p}} \end{array}\right)$$

has rank at most 4, so all of its 5×5 minors vanish. Expanding by cofactors in the last column gives homogeneous linear equations in the components of $\lambda_{\mathrm{ip}}\,\mathbf{x}_{\mathrm{ip}}$ and $\lambda_{\mathrm{jp}}\,\mathbf{x}_{\mathrm{jp}}$, with coefficients that are 4×4 determinants of projection matrix rows. These turn out to be the expressions for the fundamental matrix $\mathbf{F}_{\mathrm{ij}}$ and epipole $\mathbf{e}_{\mathrm{ji}}$ of camera j in image i in terms of projection matrix components [19, 4]. The result is the **projective depth recovery equation**:

$$(\mathbf{F}_{\mathrm{ij}}\,\mathbf{x}_{\mathrm{jp}})\,\lambda_{\mathrm{jp}} \;=\; (\mathbf{e}_{\mathrm{ji}} \wedge \mathbf{x}_{\mathrm{ip}})\,\lambda_{\mathrm{ip}} \qquad (1)$$

This says two things: (*i*) The epipolar line of $\mathbf{x}_{\mathrm{jp}}$ in image i is the same as the line through the corresponding point $\mathbf{x}_{\mathrm{ip}}$ and epipole $\mathbf{e}_{\mathrm{ji}}$ (as is well known); (*ii*) *With the correct projective depths and scalings for* $\mathbf{F}_{\mathrm{ij}}$ *and* $\mathbf{e}_{\mathrm{ji}}$*, the two terms have exactly the same size*. The equality is exact, not just up to scale. This is the new result that allows us to recover projective depths using fundamental matrices and epipoles. Analogous results based on higher order matching tensors can be found in [18].

It is straightforward to recover projective depths using (1). Each instance of it linearly relates the depths of a single 3D point in two images. By estimating a sufficient number of fundamental matrices and epipoles, we can amass a system of homogeneous linear equations that allows the complete set of depths for a given point to be found, up to an arbitrary overall scale factor. At a minimum, this can be done by selecting any set of $m-1$ equations that link the m images into a single connected graph. With such a non-redundant set of equations the depths for each point p can be found trivially by chaining together the solutions for each image, starting from some arbitrary initial value such as $\lambda_{1\mathrm{p}} = 1$. Solving the depth recovery equation in least squares gives a simple recursion relation for λ_{ip} in terms of λ_{jp}:

$$\lambda_{\mathrm{ip}} \;:=\; \frac{(\mathbf{e}_{\mathrm{ji}} \wedge \mathbf{x}_{\mathrm{ip}}) \cdot (\mathbf{F}_{\mathrm{ij}}\,\mathbf{x}_{\mathrm{jp}})}{\|\mathbf{e}_{\mathrm{ji}} \wedge \mathbf{x}_{\mathrm{ip}}\|^2}\,\lambda_{\mathrm{jp}}$$

If additional depth recovery equations are used, this simple recursion must be replaced by a redundant (and hence potentially more robust) homogeneous linear system. However, care is needed. The depth recovery equations are sensitive to the scale factors chosen for the $\mathbf{F}$'s and $\mathbf{e}$'s, and these can not be recovered directly from the image data. This is irrelevant when a single chain of equations is used, as rescalings of $\mathbf{F}$ and $\mathbf{e}$ affect all points equally and hence amount to rescalings of the corresponding projection matrices. However with redundant equations it is essential to choose a mutually self-consistent set of scales for the $\mathbf{F}$'s and $\mathbf{e}$'s. I will not describe this process here, except to note that the consistency condition is the Grassmann identity $\mathbf{F}_{\mathrm{kj}}\mathbf{e}_{\mathrm{ij}} = \mathbf{e}_{\mathrm{ik}} \wedge \mathbf{e}_{\mathrm{jk}}$ [18].

It is still unclear what the best trade-off between economy and robustness is for depth recovery. This paper considers only two simple non-redundant choices: either the images are taken pairwise in sequence, $\mathbf{F}_{21}, \mathbf{F}_{32}, \ldots, \mathbf{F}_{\mathrm{m\,m-1}}$, or all subsequent images are scaled in parallel from the first, $\mathbf{F}_{21}, \mathbf{F}_{31}, \ldots, \mathbf{F}_{\mathrm{m1}}$. It might seem that long chains of rescalings would prove numerically unstable, but in practice depth recovery is surprisingly well conditioned. Both serial and parallel chains work very well despite their non-redundancy and chain length or reliance on a 'key' image. The two methods give similar results except when there are many ($>$40) images, when the shorter chains of the parallel system become more robust. Both are stable even when epipolar *point* transfer is ill-conditioned (*e.g.* for a camera moving in a straight line, when the epipolar lines of different images coincide): the image observations act as stable 'anchors' for the transfer process.

Balancing: A further point is that with arbitrary choices of scale for the fundamental matrices and epipoles, the average size of the recovered depths might tend to increase or decrease exponentially during the solution-chaining process. Theoretically

this is not a problem as the overall scales are arbitrary, but it could easily make the factorization phase numerically ill-conditioned. To counter this the recovered matrix of projective depths must be balanced after it has been built, by judicious overall row and column rescalings. The process is very simple. The image points are normalized on input, so ideally all of the scale factors λ_{ip} should have roughly the same order of magnitude, $\mathcal{O}(1)$ say. For each point the depths are estimated as above, and then: (*i*) each row (image) of the estimated depth matrix is rescaled to have length $\sqrt{n}$; (*ii*) each column (point) of the resulting matrix is rescaled to length $\sqrt{m}$. This process is repeated until it roughly converges, which happens very quickly (within 2–3 iterations).

3 Line Reconstruction

3D lines can also be reconstructed using the above techniques. A line $\mathbf{L}$ can be represented by any two 3D points lying on it, say $\mathbf{Y}$ and $\mathbf{Z}$. In image i, $\mathbf{L}$ projects to some image line $\mathbf{l}_{\mathrm{i}}$ and $\mathbf{Y}$ and $\mathbf{Z}$ project to image points $\mathbf{y}_{\mathrm{i}}$ and $\mathbf{z}_{\mathrm{i}}$ lying on $\mathbf{l}_{\mathrm{i}}$. The points $\{\mathbf{y}_{\mathrm{i}} | i = 1, \ldots, m\}$ are in epipolar correspondence, so they can be used in the depth recovery equation (1) to reconstruct $\mathbf{Y}$, and similarly for $\mathbf{Z}$. The representatives $\mathbf{Y}$ and $\mathbf{Z}$ can be fixed implicitly by choosing $\mathbf{y}_1$ and $\mathbf{z}_1$ arbitrarily on $\mathbf{l}_1$ in the first image, and using the epipolar constraint to transfer these to the corresponding points in the remaining images: $\mathbf{y}_{\mathrm{i}}$ lies on both $\mathbf{l}_{\mathrm{i}}$ and the epipolar line of $\mathbf{y}_1$, so is located at their intersection.

In fact, epipolar transfer and depth recovery can be done in one step. Let $\mathbf{y}_{\mathrm{i}}$ stand for the *rescaled* via points $\mathbf{P}_{\mathrm{i}}\mathbf{Y}$. Substitute these into equation (1), cross-product with $\mathbf{l}_{\mathrm{i}}$, expand, and simplify using $\mathbf{l}_{\mathrm{i}} \cdot \mathbf{y}_{\mathrm{i}} = 0$:

$$\begin{aligned} \mathbf{l}_{\mathrm{i}} \wedge (\mathbf{F}_{\mathrm{ij}}\, \mathbf{y}_{\mathrm{j}}) &= \mathbf{l}_{\mathrm{i}} \wedge (\mathbf{e}_{\mathrm{ji}} \wedge \mathbf{y}_{\mathrm{i}}) \\ &= -(\mathbf{l}_{\mathrm{i}} \cdot \mathbf{e}_{\mathrm{ji}})\, \mathbf{y}_{\mathrm{i}} + (\mathbf{l}_{\mathrm{i}} \cdot \mathbf{y}_{\mathrm{i}})\, \mathbf{e}_{\mathrm{ji}} \\ &= -(\mathbf{l}_{\mathrm{i}} \cdot \mathbf{e}_{\mathrm{ji}})\, \mathbf{y}_{\mathrm{i}} \end{aligned} \qquad (2)$$

Up to a factor of $\mathbf{l}_{\mathrm{i}} \cdot \mathbf{e}_{\mathrm{ji}}$*, the intersection* $\mathbf{l}_{\mathrm{i}} \wedge (\mathbf{F}_{\mathrm{ij}}\, \mathbf{y}_{\mathrm{j}})$ *of* $\mathbf{l}_{\mathrm{i}}$ *with the epipolar line of* $\mathbf{y}_{\mathrm{j}}$ *automatically gives the correct projective depth for reconstruction.* Hence, factorization-based line reconstruction can be implemented by choosing a suitable (widely spaced) pair of via-points on each line in the first image, and then chaining together instances of equation (2) to find the corresponding, correctly scaled via-points in the other images. The required fundamental matrices can not be found directly from line matches, but they can be estimated from point matches, or from the trilinear line matching constraints (trivalent tensor) [6, 14, 4, 19, 18]. Alternatively, the trivalent tensor can be used directly: in tensorial notation [18], the trivalent via-point transfer equation is $\mathrm{l}_{\mathrm{B_k}}\, \mathbf{G}_{\mathrm{C_j}}{}^{\mathrm{A_i B_k}}\, \mathbf{y}^{\mathrm{C_j}} = (\mathrm{l}_{\mathrm{B_k}}\, \mathbf{e}_{\mathrm{j}}^{\mathrm{B_k}}) \mathbf{y}^{\mathrm{A_i}}$.

As with points, redundant equations may be included if and only if a self-consistent normalization is chosen for the fundamental matrices and epipoles. For numerical stability, it is essential to balance the resulting via-points (*i.e.* depth estimates). This works with the $3m \times 2n_{\mathrm{lines}}$ '$\mathbf{W}$' matrix of via-points, iteratively rescaling all coordinates of each image (triple of rows) and all coordinates of each line (pair of columns) until an approximate equilibrium is reached, where the overall mean square size of each coordinate is $\mathcal{O}(1)$ in each case. To ensure that the via-points representing each line are on average well separated, I also orthonormalize the two $3m$-component column vectors for each line with respect to one another. The via-point equations (2) are linear and hence invariant with respect to this, but it does of course change the 3D representatives $\mathbf{Y}$ and $\mathbf{Z}$ recovered for each line.

4 Implementation

This section summarizes the complete algorithm for factorization-based 3D projective reconstruction from image points and lines, and discusses a few important implementation details and variants. The algorithm goes as follows: Extract and match points and lines across all images.

Standardize all image coordinates (see below).

Estimate a set of fundamental matrices and epipoles sufficient to chain all the images together (*e.g.* using point matches).

For each point, estimate the projective depths using equation (1). Build and balance the depth matrix λ_{ip}, and use it to build the rescaled point measurement matrix $\mathbf{W}$.

For each line choose two via-points and transfer them to the other images using the transfer equations (2). Build and balance the rescaled line via-point matrix.

Combine the line and point measurement matrices into a $3m \times (n_{\text{points}} + 2n_{\text{lines}})$ data matrix and factorize it using either SVD or the fixed-rank method. Recover 3D projective structure (point and via-point coordinates) and motion (projection matrices) from the factorization.

Un-standardize the projection matrices (see below).

Complexity: The algorithm is dominated by the $\mathcal{O}(mn \min(3m, n))$ SVD step if this is used, while if an approximate factorization is used it is proportional to the input data size $\mathcal{O}(mn)$.

Standardization: To get acceptable results from the above algorithm, it is *absolutely essential* to work in a well-adapted image coordinate system. The basic idea is to choose working coordinates that reflect the least squares trade-offs implicit in the factorization algorithm. This is standard practice in numerical analysis, but it does not seem to have been widely known in vision until Hartley [7] pointed out its importance for fundamental matrix estimation. The exact scheme used is not critical, provided that the homogeneous working coordinates are all of the same order of magnitude. I currently prefer to scale the image into the unit square $[-1, 1] \times [-1, 1]$, homogenize, and then normalize the resulting homogeneous 3-vectors to unit length $x^2 + y^2 + z^2 = 1$. This simple scheme works very well in practice. The normalization applies to line vectors as well as point ones, and behaves well even for points (*e.g.* epipoles) near the line at infinity. After reconstruction, the camera projections need to be un-standardized by multiplying by the inverse transformation.

4.1 Generalizations & Variants

I have implemented and experimented with a number of variants of the above algorithm, the more promising of which are featured in the experiments described below.

Iterative Factorization: The projective depths depend on the 3D structure, which in turn derives from the depths. The reconstruction can be iteratively improved by reprojecting to refine the depth estimates and then re-factorizing. For points one finds the component of the reprojected 3D point vector along each image vector, while for lines the reprojected via-point is perturbed orthogonally to lie on the measured image line. With SVD-based factorization and standardized image coordinates the iteration turns out to be extremely stable, and always improves the recovered structure slightly (often significantly for lines). For points, one can even start with arbitrary initial depths (say the affine ones $\lambda_{\text{ip}} = 1$) and iterate to convergence. This requires no fundamental matrices or depth recovery equations and converges reliably in practice, although it can be rather slow if started far from the true solution.

Nonlinear Least Squares: The 'linear' factorization-based projective reconstruction methods described above are a suitable starting point for more refined nonlinear least-squares estimation. This can take account of image point error models, camera calibrations, or Euclidean constraints, as in the work of Szeliski and Kang [16], Hartley [5] and Mohr, Boufama and Brand [10]. The standard workhorse for such problems is Levenberg-Marquardt iteration [12], so for comparison with the linear methods I have implemented simple L-M based projective reconstruction algorithms. These can be initialized from either fixed-rank or SVD-based factorizations. For lines the recovered structure is often improved significantly, while for points the improvement over the linear methods is usually small.

Affine Factorization: To illustrate the advantages of projective factorization over the original Tomasi-Kanade-Poelman work [17, 11], I have also implemented affine SVD-based point reconstruction. This gives rather poor results in the below experiments because the perspective distortions are quite large.

5 Experiments

To quantify the performance of the various algorithms, I have run a large number of simulations using synthetic data, and also tested the algorithms on manually matched primitives derived from real images. There is only space for a very brief summary here, more details can be found in [20].

The simulations are based on trial scenes consisting of random 3D points and lines in the unit cube $[-1, 1] \times [-1, 1] \times [-1, 1]$, perturbed by uniform noise and viewed by identical perspective cameras in various arrangements. In the graphs shown here, the cameras are spaced uniformly along a 90 degree arc of radius 2 in the equatorial plane of the scene,

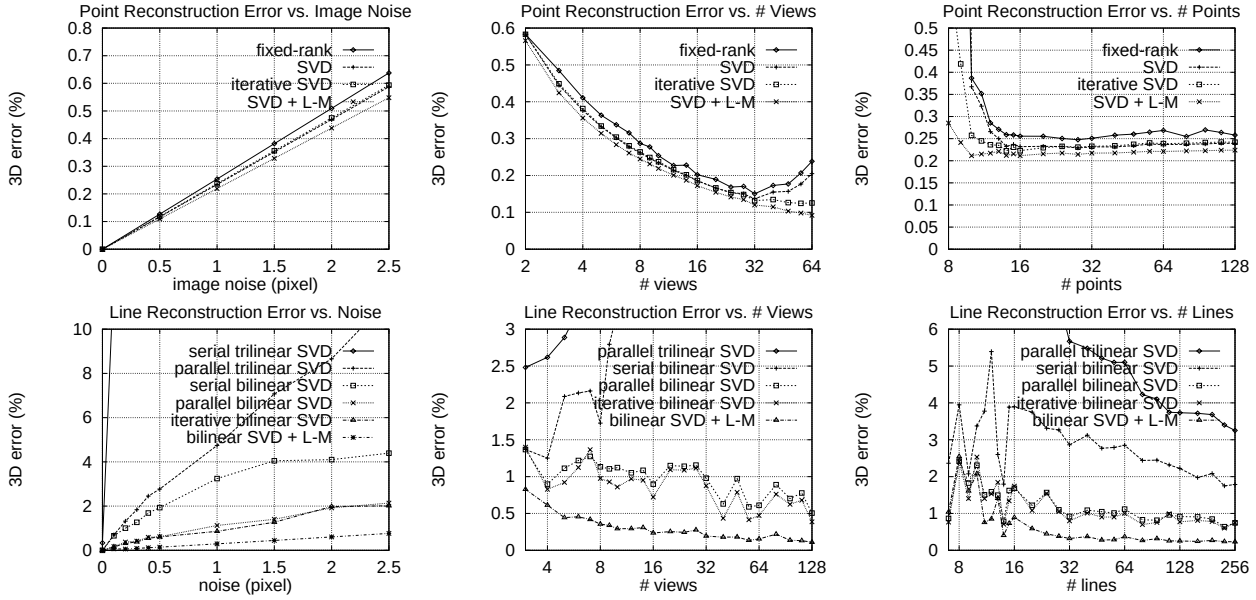


Figure 1: Mean 3D reconstruction error for points and lines, *vs.* noise, number of views and number of primitives. Defaults: ± 1 pixel noise; 10 views; 50 primitives.

and are directed towards the scene centre (*i.e.* there is a large baseline and significant perspective distortion). Reconstruction error is measured over 50 trials, after least-squares projective alignment with the true 3D structure. Mean errors are reported for points, while for lines there are always outliers so median errors are used[1].

Fundamental matrices and epipoles are estimated using the linear least squares method with all the available point matches, followed by a supplementary SVD to project the fundamental matrices to rank 2 and find the epipoles. In standardized coordinates this method performs very well [7], and it has not proved necessary to refine the results with a nonlinear method. Unless otherwise noted, the projective depths of points are recovered by chaining sequentially through the images: $\mathbf{F}_{12}, \mathbf{F}_{23}, \ldots, \mathbf{F}_{\mathrm{m-1\,m}}$. A parallel chain $\mathbf{F}_{12}, \mathbf{F}_{13}, \ldots, \mathbf{F}_{1\,\mathrm{m}}$ usually gives similar results. For lines in more than a few images, the parallel chain is superior and is used by default.

Fig. 1 shows the sensitivity of various point and line reconstruction methods to image noise, number of views, and number of scene primitives (points or lines). The methods shown are: *points:* fundamental matrix depth recovery with SVD and fixed-rank factorization, iterated SVD and nonlinear least-squares initialized from SVD; *lines:* fundamental matrix and trilinear parallel and serial via-point transfer followed by SVD, iterated SVD, and SVD plus nonlinear least-squares.

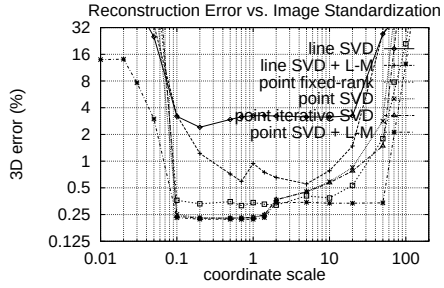


Figure 2: Reconstruction error *vs.* image standardization.

All of the point methods are very stable. Their errors vary linearly with noise and decrease as more points or views are added. There is not much difference in precision, but generally the fixed-rank method is slightly less accurate (but significantly faster) than SVD. Iterating the SVD makes a small improvement, and nonlinear least-squares is slightly more accurate again. Serial depth recovery chains become ill-conditioned when more than 30-40 im-

[1]The image of a line passing near the optical centre of a camera is extremely sensitive to small 3D perturbations. Also, if the camera centres lie in a plane (as here), all lines in that plane have the same image, so such lines can not be uniquely reconstructed (*c.f.* axial points for cameras lying in a line; in this case, only lines skew with the axis can be reconstructed).

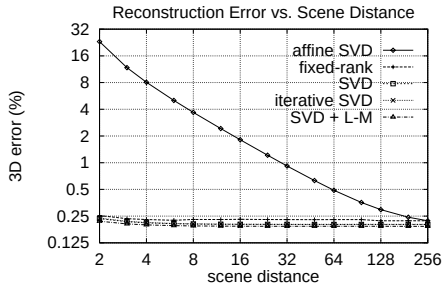


Figure 3: Projective and affine reconstruction *vs.* scene distance.

ages are chained: beyond this parallel chaining is advised.

Line reconstruction is less stable. Only the least-squares methods consistently give reconstruction errors commensurate with the input noise. Parallel F-matrix transfer plus factorization is a factor of 2 or more worse than this, and serial transfer is worse again. Iterative factorization helps a little, but the use of a nonlinear least-squares routine is still advisable. Any of these methods are accurate enough for reliable initialization of the least-squares iteration. If my implementation is correct, trilinear transfer based reconstruction is too sensitive to noise to be useful (this requires confirmation). For all of the above methods, there are outliers corresponding to lines that either can not be reconstructed uniquely, or are very sensitive to small 3D perturbations.

The importance of standardization is illustrated in fig. 2, where the image coordinates are standardized to $\mathcal{O}(\text{scale})$ rather than $\mathcal{O}(1)$ before reconstruction. Pixel coordinates correspond to a scale of 256 and give errors hundreds of times worse than well-standardized coordinates. The rapid increase in error at scales below 0.1 is caused by floating-point truncation error.

Fig. 3 illustrates the advantages of using perspective rather than affine reconstruction, for a camera driving in a 90 degree arc around a scene at various distances. Clearly, the affine approximation introduces a considerable amount of systematic error even for quite distant scenes. Projective factorization is stable and accurate even for distant scenes: even in these cases, the only real advantage of affine factorization is the fact that it is 2-3 times faster.

I have also run the point-based algorithms on several data sequences extracted from real images. Without the ground truth it is hard to be precise, but the final aligned reconstructions seem qualitatively accurate and in good agreement with the results obtained using synthetic data.

6 Discussion & Conclusions

Within the limitations of the factorization paradigm, factorization-based projective reconstruction seems quite successful. For points, the methods studied have proved simple, stable, and surprisingly accurate. For lines the situation is less clear: the methods work, but least-squares refinement often improves the results significantly. As with any line reconstruction, there are always outliers, especially when the cameras are collinear or coplanar.

Fixed-rank factorization works well, although (as might be expected) SVD always produces slightly more accurate results. The savings in run time over SVD probably only become significant for quite large problems (say more than 40 images and 100 points), but in these cases they can become very substantial.

This paper presents only the first few members of a large family of reconstruction techniques, based on the recovery of projective depths or scale factors. Future work will expand on this. There are analogous factorization methods using higher matching tensors, and also methods that reconstruct the projection matrices directly from matching tensors without factorization (and hence do not require tokens to be tracked through every image). All of these allow various trade-offs between redundancy, computation and implementation effort. I am also investigating numerical factorization methods that can handle missing data and incremental updates gracefully, and alternatives to Levenberg-Marquardt refinement (which I feel is not well suited to nonlinear least-squares reconstruction).

Summary: Projective structure and motion can be recovered from multiple perspective images of a scene consisting of points and lines, by estimating fundamental matrices and epipoles from the image data, using these to rescale the image measurements, and then factorizing the resulting rescaled measurement matrix using either SVD or a fast approximate factorization algorithm.

References

[1] P. Beardsley, I. Reid, A. Zisserman, and D. Murray. Active visual navigation using non-metric structure. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 58–64, Cambridge, MA, June 1995.

[2] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[3] O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[4] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[5] R. Hartley. Euclidean reconstruction from multiple views. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.

[6] R. Hartley. Lines and points in three views – an integrated approach. In *Image Understanding Workshop*, Monterey, California, November 1994.

[7] R. Hartley. In defence of the 8-point algorithm. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1064–70, Cambridge, MA, June 1995.

[8] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 761–4, Urbana-Champaign, Illinois, 1992.

[9] P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 314–20, Cambridge, MA, June 1995.

[10] R. Mohr, B. Boufama, and P. Brand. Accurate projective reconstruction. In 2^{nd} *Europe-U.S. Workshop on Invariance*, page 257, Ponta Delgada, Azores, October 1993.

[11] C. J. Poelman and T. Kanade. A parapersective factorization method for shape and motion recovery. In J-O. Eklundh, editor, *European Conf. Computer Vision*, pages 97–108, Stockholm, 1994. Springer-Verlag.

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992. 2^{nd} edition.

[13] A. Shashua. Projective structure from uncalibrated images: Structure from motion and recognition. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 16(8), 1994.

[14] A. Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 17(8):779–89, 1995.

[15] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–20, Cambridge, U.K., 1996. Springer-Verlag.

[16] R. Szeliski and S. B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. In *IEEE Conf. Computer Vision & Pattern Recognition*, page 752, 1993.

[17] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Computer Vision*, 9(2):137–54, 1992.

[18] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to *Int. J. Computer Vision*.

[19] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.

[20] B. Triggs. New methods for projective reconstruction. In preparation, 1996.

Linear Projective Reconstruction from Matching Tensors

Bill Triggs
INRIA Rhône-Alpes,
655 avenue de l'Europe, 38330 Montbonnot St. Martin, France.
Bill.Triggs@inrialpes.fr ◇ *http://www.inrialpes.fr/movi/Triggs*

Abstract

This paper describes initial work on a family of projective reconstruction techniques that compute projection matrices directly and linearly from matching tensors estimated from the image data. The approach is based on 'joint image closure relations' — bilinear constraints between matching tensors and projection matrices, that express the fact that the former derive from the latter. The simplest methods use fundamental matrices and epipoles, alternative ones use trilinear tensors. It is possible to treat all of the image data uniformly, without reliance on 'privileged' images or tokens. The underlying theory is discussed, and the performance of the new methods is quantified and compared with that of several existing ones.

Keywords: Multi-image structure, projective reconstruction, matching tensors.

1 Introduction

Traditional stereo vision systems use carefully calibrated cameras to provide metric reconstruction from a single pair of static images. It has long been clear that the redundancy offered by further images can significantly increase the quality and stability of visual reconstructions, as well as extending their coverage to previously hidden parts of the scene. Furthermore, much of the 3D structure can be recovered without *any* prior camera calibration. Even in the extreme case of several distinct unknown projective cameras viewing the scene from unknown positions, the entire metric scene geometry can be recovered up to just 9 global parameters — 3 scale factors, 3 skews and 3 projective distortions[1] [4, 7, 13]. Various common scene or camera constraints can be used to further reduce this ambiguity, *e.g.* known vanishing points or length ratios, known skew or aspect ratio, motion-constancy of intrinsic parameters, ... [6]. This is especially relevant to applications such as scene modelling for virtual reality or robot navigation, where many images are needed to cover the scene and precise calibration is difficult owing to uncertain camera motions, changes in internal parameters (focus, zooming) or the use of several cameras.

There is a need for visual reconstruction methods with the following characteristics:

1) **Multi-image/multi-point/missing data:** It is hard to match features reliably across many images, especially under large changes of viewpoint. Reconstruction methods requiring long sequences of matches tend to run into missing data problems. For example, factorization methods [26, 25, 29, 24] are very stable and treat all images and points equally, but require completely filled 'blocks' of points *vs.* images. Traditional methods further limit these blocks to small fixed numbers

This paper was published in Image & Vision Computing. An earlier version appeared in BMVC'96. The work was supported by INRIA Rhône-Alpes, the Esprit HCM network and the Esprit LTR grant CUMULI.

[1] If there is lens distortion, this can also (in theory) be recovered up to an unknown image homography.

of images or points. The stability of such methods is critically dependent on the images chosen, and since these must usually be closely-spaced to allow reliable matching, overall accuracy suffers. It is possible to work around gaps in the data by 'patching together' several partial reconstructions, but it would be useful to have methods that handled missing data naturally, without relying on *ad hoc* patching, key points, or key images.

2) **Flexible calibration:** Calibration constraints come in many forms: prior knowledge, calibration images, scene or motion constraints, ... It is not always obvious how to incorporate them into the multi-image reconstruction process. Often it is simpler to ignore them at first, working projectively and only later going back and using them to 'straighten' the recovered projective structure. This 'stratification' school [6] has its critics [32, 20]. In particular, it is felt that stability may be compromised by failing to enforce reasonable camera and motion models at the outset. However as far as I know it is the only approach that has yet produced true multi-image reconstruction algorithms for general cameras and motions [25, 29, 30, 24].

3) **Precision/robustness/stability:** *Precision* means that the method gives accurate results when it works; *robustness* that it works reliably (*e.g.* in the face of mismatches or initialization errors); *stability* that the results are not overly sensitive to perturbations in the input data. Stability is a precondition for precision and robustness, but is easily compromised by degeneracies in either the viewing geometry or the algorithmic formulation used.

For the best precision there is no substitute for rigorous statistical parameter estimation, *e.g.* maximum likelihood. For this, a nonlinear cost reflecting a statistical error model of the image observations must be globally optimized over all unknown 3D structure and calibration parameters. With Gaussian errors, this reduces to covariance-weighted nonlinear least squares. Such statistical 'bundle adjustment' is a truism for photogrammetrists but seems to be tacitly discouraged in computer vision, where the traditional emphasis is on A.I. image understanding rather than precision (however *cf.* [17, 10, 19, 14, 9]). Efficient numerical methods exist for handling large problems, both off-line and in a linearized recursive framework [1, 18].

Rigorous, statistically weighted least squares should not be confused with 'unweighted' or 'linear least squares' minimization of *ad hoc* 'algebraic distances' — sums of squared algebraic constraint violations with no direct relation to measured image residuals. For example the 'linear' method for the fundamental matrix [12], reconstruction by affine and projective factorization [26, 25, 29, 24], and the new 'closure based' methods presented here, all linearize the problem and minimize algebraic distances using linear algebra techniques (*e.g.* SVD). Common characteristics of such methods are: (*i*) they are linear and much simpler to implement than the corresponding statistical methods; (*ii*) no prior initialization is needed; (*iii*) somewhat more than the minimal amount of data is required, to allow nonlinearities to be "linearized away"; (*iv*) they are sensitive to the relative weighting of different components of the error function (but the choice is not too critical once you realize it has to be made); (*v*) with suitable weighting, they give results not too far from (but still worse than) the statistical optimum. Criticisms include: (*i*) ignoring constraints may reduce stability and make the results difficult to interpret; (*ii*) general linear methods are often slower than dedicated nonlinear ones, as large matrices tend to be involved; (*iii*) it is difficult to detect outliers without a clear error model.

Bundle adjustment routines provide all of the desirable features listed above, except robustness against initialization. As they are only iterative improvement techniques, they require initial estimates for all unknown parameters. In practice they are seldom robust against gross errors in these, or even against re-parametrization (*e.g.* convergence tests are notoriously sensitive to this).

Hence, there is still a need for stable and relatively tractable suboptimal reconstruction methods that require no prior initialization, take into account as many as possible of the above properties, and can be used as input to nonlinear methods if more precision is required. Partly in response to this,

there has recently been a significant amount of work on the theoretical foundations of multi-image projection and reconstruction [11, 10, 19, 18, 23, 2, 22, 8, 15, 16, 31, 27, 28, 3]. The problem turns out to have a surprisingly rich mathematical structure and several complementary approaches exist. The field is developing rapidly and there is no space for a survey here, so I will only mention a few isolated results. The epipolar constraint (the geometry of stereo pairs) is now well understood (*e.g.* [5]). Shashua [22] and Hartley [11] developed the theory of the trivalent tensor (three view constraint). Faugeras and Mourrain [8] and I [28] systematically studied the complete family of multi-image constraints (only one was unknown: a quadrilinear one).

As a means to this, I developed a tensorial approach to multi-image vision [28], which nicely unifies the geometric and algebraic aspects of the subject. This lead to the **joint image** picture, in which the combined homogeneous coordinates of all the images of a 3D point are stacked into a single big 'joint image' vector. The geometry of this space can be related to that of the original 3D points via the stacked projection matrices. All of the familiar image entities — points, lines, homographies, matching tensors, *etc* — fall naturally out of this picture as the joint image representatives of the corresponding 3D objects. The approach is also 'dual' (in the sense of Carlsson [3]) to Sparr's 'affine shape' formalism [23, 15, 24], where coordinates are stacked by point rather than by image.

In the MOVI group, we have recently developed several families of projective reconstruction methods based on the joint image approach. The factorization-based 'projective depth recovery' methods [25, 29] use matching tensors to recover a coherent set of projective scale factors for the image points. This gives an implicit reconstruction, which can be concretized by factorizing the matrix of rescaled image points into projection and structure matrices by a process analogous to the Tomasi-Kanade-Poelman method for affine structure [26, 21]. Factorization-based methods give an implicit linear least squares fit to all of the image data. They are simple and extremely stable, but have the serious practical disadvantage that each point must be visible in every image (modulo 'hallucination' [26]). This is unrealistic when there are many images covering a wide range of viewing positions.

The current paper represents a first attempt to overcome this problem. It describes a new family of reconstruction methods that extract projection matrices directly and linearly from estimated matching tensors, after which the scene structure can be recovered linearly by back-projecting the image measurements. The projections are estimated using 'joint image closure relations' — bilinear constraints between projections and their matching tensors, analogous to the depth recovery relations used for projective factorization, but with projection matrices replacing image points.

In principle, the closure based reconstruction methods treat all of the images uniformly, so they have the potential to be significantly more stable than the commonly used approach of initially reconstructing from two key images, then reprojecting into the other ones to estimate the remaining projection matrices. On the other hand, because they only use the image data indirectly via the matching tensors, they are not as stable as factorization based methods. The suggestion is that they will prove good replacements for the 'stereo + reprojection' methods (whose main application is probably to initialize more refined nonlinear least squares iterations), but that when tokens are visible in every image factorization will still be the best linear method.

The rest of the paper outlines the theory of the closure relations, describes the resulting reconstruction algorithms and their implementation, reports on an initial experimental study of their performance, and ends with a short discussion.

2 Theory

This section sketches the theoretical background of multi-image reconstruction, and discusses the 'joint image closure relations' on which the new reconstruction methods are based. The theory is

not difficult, but when more than two images are involved the equations are hard to express without using tensorial notation. We will use ordinary matrix-vector notation except for a few trivalent tensor equations, so you should be able to follow most of the paper without a knowledge of tensors. An *extremely* brief introduction to them follows — see [28, 27] for more details. All quantities are assumed to be projective, expressed in homogeneous coordinates.

Tensors are just multidimensional arrays of components. Vectors (1-index arrays) and matrices (2-index arrays) are examples. Each index is associated with a specific space (the 3D world, image i, ...), and inherits the corresponding change-of-basis law. Many common vector and matrix operations generalize directly to tensors, provided we specify which of the many indices the operation applies to. (For matrices, the index is implicit in the 'juxtaposition = multiplication' rule). To keep track of the indices, we write them out explicitly: $a, b, c \ldots$ for world-space indices and $A_i, B_i, C_i \ldots$ for image i ones. The most common operation is **contraction** — summing a corresponding pair of indices over the range of their values, as in vector dot-product, matrix product or trace. The summation signs are elided: any index that appears twice in a term is implicitly summed over.

A further complication is that in projective geometry each space has a corresponding **dual**, *e.g.* in each image, the space of points is dual to the space of lines (hyperplanes). This means that every index actually comes in two varieties: point-like or **contravariant** and hyperplane-like or **covariant**. These have *different* (complementary) transformation laws under changes of basis, so they must be carefully distinguished: point indices are written as superscripts, hyperplane ones as subscripts. Contractions are only meaningful between covariant-contravariant pairs of indices from the same space, *e.g.* there is *no* meaningful 'dot product' between pairs of projective points — the result would be completely dependent on the basis chosen.

World points $\mathbf{X}^a$ project to image ones $\mathbf{x}^{A_i}$ by contraction with 3×4 projection matrices $\mathbf{P}_a^{A_i}$: $\mathbf{x}^{A_i} \sim \mathbf{P}_a^{A_i} \mathbf{X}^a$ (implicit summation over a). $\mathbf{e}_1^{A_2}$ denotes the epipole of camera 1 in image 2; $\mathbf{F}_{A_1 B_2}$ the fundamental matrix between images 1 and 2; and $\mathbf{G}_{A_1}{}^{B_2 C_3}$ the trivalent tensor between images 2 and 3 based in image 1. (There are also corresponding trivalent tensors based in images 2 and 3). In ordinary matrix-vector notation, $\mathbf{X}$ stands for $\mathbf{X}^a$, $\mathbf{x}_i$ for $\mathbf{x}^{A_i}$, $\mathbf{P}_i$ for $\mathbf{P}_a^{A_i}$, $\mathbf{e}_{ij}$ for $\mathbf{e}_i^{A_j}$, and $\mathbf{F}_{ij}$ for $\mathbf{F}_{A_i B_j}$.

Consider the projections $\lambda_{ip}\mathbf{x}_{ip} = \mathbf{P}_i \mathbf{X}_p$ of n homogeneous world points $\mathbf{X}_p$, $p = 1, \ldots, n$, into m images via 3×4 perspective projection matrices $\mathbf{P}_i$, $i = 1, \ldots, m$. The resulting mn homogeneous image points $\mathbf{x}_{ip}$ are only defined up to unknown scale factors λ_{ip}, called **projective depths**. As each $\mathbf{P}_i$ and $\mathbf{X}_p$ can be arbitrarily rescaled, there is some superficial freedom in the choice of these scales. However there is a strong underlying coherence that embodies the projective structure of the scene: the depths λ_{ip} really do capture the projective part of visual depth. An algebraic result of the coherence is the low rank (four) of the rescaled data matrix:

$$\begin{pmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \cdots & \lambda_{mn}\mathbf{x}_{mn} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_m \end{pmatrix} \begin{pmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_n \end{pmatrix}$$

It is useful to view this column-by-column, as the projection of world points $\mathbf{X}_p$ to $3m$-component **joint image space** vectors via the stacked $3m \times 4$ **joint projection** matrix $\mathbf{P}$:

$$\begin{pmatrix} \lambda_{1p}\mathbf{x}_{1p} \\ \vdots \\ \lambda_{mp}\mathbf{x}_{mp} \end{pmatrix} = \mathbf{P}\,\mathbf{X}_p \qquad \text{where} \qquad \mathbf{P} \equiv \begin{pmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_m \end{pmatrix}$$

The joint projection can be viewed as a projective injection mapping the 3D projective world bijectively to the **joint image** — a 3D projective subspace of $(3m-1)$-D projective joint image space

[28, 27]. This is a faithful projective copy of the world expressed entirely in image coordinates. Projection from it to the individual images is a trivial forgetting of coordinates and scale factors. Projective reconstruction of the joint image amounts to recovering the missing depths λ_{ip}. This is a canonical process[2] up to a once-and-for-all choice of scales for the projections $\mathbf{P}_i$. The four columns of the joint projection matrix form a spanning basis for the joint image. The coordinates of a rescaled joint image point with respect to this basis are exactly the corresponding 3D point's homogeneous world coordinates. But neither the basis nor the world coordinates are canonical: only the geometric position of the point in the joint image is recoverable from the image data.

The above geometry can be converted directly to algebra. The 4×4 minors (submatrix determinants) of the joint projection encode the location of the joint image (and hence the projective camera geometry) in a well-defined algebraic sense: they are its 'Grassmann-Plücker coordinates'. Moreover, the minors turn out to be just the components of the **matching tensors** between the images. These generate the multilinear constraints that tokens in different images must satisfy if they are to be the projections of a single world token. They can also be used for projective depth recovery, and to transfer tokens between images. There are four basic types of matching tensors: **epipoles** $\mathbf{e}_{ij}$ (tensorially: $\mathbf{e}_i^{A_j}$), **fundamental matrices** $\mathbf{F}_{ij}$ ($\mathbf{F}_{A_iB_j}$), **trivalent tensors** $\mathbf{G}_{A_i}{}^{B_jC_k}$ and **quadrivalent tensors** $\mathbf{H}^{A_iB_jC_kD_l}$. These are formed from minors with respectively 3+1, 2+2, 2+1+1, and 1+1+1+1 rows from 2, 2, 3 and 4 images i, j, k, l [22, 8, 28].

The 'joint image closure relations' that underlie the new reconstruction methods are bilinear constraints between projection matrices and the corresponding matching tensors. They guarantee that the projections are coherent with the joint image subspace defined by the tensors. Algebraically, they express the four-dimensionality ("closure") of the joint image. The simplest way to derive them is to append any column of the $3m \times 4$ joint projection matrix to the existing matrix, to form a rank deficient $3m \times 5$ matrix. The 5×5 minors of this matrix vanish. Expand by cofactors in the appended column. The coefficients are matching tensor components (4×4 minors of the original joint projection matrix). Closer examination reveals five basic types of relation. We use only the simplest two here[3]:

$$\mathbf{F}_{ji}\,\mathbf{P}_i + [\mathbf{e}_{ij}]_\times \mathbf{P}_j \;=\; \mathbf{0} \qquad \mathbf{F}\text{-}\mathbf{e} \text{ closure} \tag{1}$$

$$\mathbf{G}_{B_j}{}^{A_iC_k}\,\mathbf{P}_a^{B_j} + \mathbf{e}_j^{A_i}\,\mathbf{P}_a^{C_k} - \mathbf{P}_a^{A_i}\,\mathbf{e}_j^{C_k} \;=\; \mathbf{0} \qquad \mathbf{e}\text{-}\mathbf{G}\text{-}\mathbf{e} \text{ closure} \tag{2}$$

These relations provide constraints between matching tensors (which can be estimated from the image data) and columns of the joint projection matrix. For each column, (1) contains 3 constraints of which 2 are linearly independent, while (2) contains $3 \times 3 = 9$ constraints of which 5 are linearly independent. By accumulating enough of these constraints, we can solve linearly for the four $3m$-component joint projection columns, up to an overall 4×4 linear transformation that amounts to a homography of the reconstructed world space. Geometrically, the joint image (the 4D subspace spanned by the columns of the joint projection) is the null space of the constraints. Given the projections, the scene reconstruction can be completed by linearly back-projecting image structure into the world space, which amounts to solving redundant linear equations

$$\mathbf{x}_{ip} \wedge (\mathbf{P}_i \mathbf{X}_p) = \mathbf{0} \tag{3}$$

for the world points $\mathbf{X}_p$ in terms of their images $\mathbf{x}_{ip}$ and the projection matrices $\mathbf{P}_i$.

The **depth recovery relations** used for projective factorization [25, 29, 27] follow directly from the above closure constraints. Attaching a world point $\mathbf{X}_p$ to each projection gives bilinear

[2] 'Canonical' means that it characterizes the imaging geometry and is characterized uniquely (up to the scales) by it; it does not depend on the world or image coordinate systems used; and it is in some sense the 'natural' arena of action for *any* reconstruction method.

[3] $[\mathbf{x}]_\times$ denotes the skew 3×3 matrix giving the vector cross product: $[\mathbf{x}]_\times \mathbf{y} = \mathbf{x} \wedge \mathbf{y}$.

constraints between the matching tensors and the *correctly rescaled* image points $\lambda_{ip}\mathbf{x}_{ip} \equiv \mathbf{P}_i\mathbf{X}_p$:

$$\mathbf{F}_{ji}\,(\lambda_{ip}\mathbf{x}_{ip}) + \mathbf{e}_{ij} \wedge (\lambda_{jp}\mathbf{x}_{jp}) \;=\; \mathbf{0} \tag{4}$$

$$\mathbf{G}_{B_j}{}^{A_iC_k}\,(\lambda_j\mathbf{x}^{B_j}) - (\lambda_i\mathbf{x}^{A_i})\,\mathbf{e}_j^{C_k} + \mathbf{e}_j^{A_i}\,(\lambda_k\mathbf{x}^{C_k}) \;=\; \mathbf{0} \tag{5}$$

Given the matching tensors, a coherent set of projective depths for the images of each world point can be recovered linearly using these relations. These already contain a virtual projective reconstruction, implicit in the fact that the rescaled data matrix (2) has rank 4. The reconstruction can be consolidated and 'read off' by any convenient matrix factorization algorithm [25, 29].

Another way to express (1) is to note that $\mathbf{F}_{ji}$ has rank 2 and hence can be decomposed (non-uniquely) as $\mathbf{F}_{ji} = \mathbf{u}_j\mathbf{v}_i^\top - \mathbf{v}_j\mathbf{u}_i^\top$. Here, $\mathbf{u}_i \leftrightarrow \mathbf{u}_j$ and $\mathbf{v}_i \leftrightarrow \mathbf{v}_j$ turn out to be corresponding pairs of epipolar line-vectors (with appropriate relative scaling), and hence $\mathbf{e}_{ij} = \mathbf{u}_j \wedge \mathbf{v}_j$, $\mathbf{e}_{ji} = \mathbf{v}_i \wedge \mathbf{u}_i$. Suitable $\mathbf{u}$'s and $\mathbf{v}$'s are easily obtained by rescaling the SVD basis of $\mathbf{F}_{ji}$. Since $[\mathbf{e}_{ij}]_\times = \mathbf{u}_j\mathbf{v}_j^\top - \mathbf{v}_j\mathbf{u}_j^\top$, the combined **F-e** closure constraints from images i-j and j-i have rank just 2 and are spanned by the rows of a 2×6 matrix $\mathbf{U}_{ij}$:

$$\begin{pmatrix} \mathbf{F}_{ji} & [\mathbf{e}_{ij}]_\times \\ [\mathbf{e}_{ji}]_\times & \mathbf{F}_{ij} \end{pmatrix} = \begin{pmatrix} -\mathbf{v}_j & \mathbf{u}_j \\ \mathbf{v}_i & -\mathbf{u}_i \end{pmatrix} \mathbf{U}_{ij} \qquad \text{where} \qquad \mathbf{U}_{ij} = \begin{pmatrix} \mathbf{u}_i^\top & \mathbf{u}_j^\top \\ \mathbf{v}_i^\top & \mathbf{v}_j^\top \end{pmatrix}$$

In fact, the $\mathbf{u}$'s and $\mathbf{v}$'s extracted from the SVD of $\mathbf{F}_{ji}$ combine to form a basis of the 2D orthogonal complement of the i-j joint image. (The space spanned by the 4 columns of the i-j joint projection matrix $\left(\begin{smallmatrix}\mathbf{P}_i\\ \mathbf{P}_j\end{smallmatrix}\right)$, or equivalently by those of the i-j rescaled data matrix $\left(\begin{smallmatrix}\lambda_{i1}\mathbf{x}_{i1} & \cdots & \lambda_{in}\mathbf{x}_{in}\\ \lambda_{j1}\mathbf{x}_{j1} & \cdots & \lambda_{jn}\mathbf{x}_{jn}\end{smallmatrix}\right)$). Hence, another way to obtain the constraint matrix $\mathbf{U}_{ij}$ is to use any two image reconstruction method (*e.g.* factorization) and extract the left null space of the resulting i-j joint projection or rescaled data matrix, *e.g.* by QR or SVD.

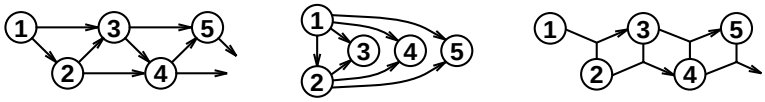
Similarly, the **e-G-e** closure constraint (2) can be written (in 3×3 blocks) as a 9×9 rank 5 matrix

$$\left(\begin{array}{c|c|ccc} -\mathbf{e}_j^{x_k}\,\mathbf{I}_{3\times3} & \mathbf{G}_\bullet{}^{x_k} & \mathbf{e}_{ji} & \mathbf{0} & \mathbf{0} \\ -\mathbf{e}_j^{y_k}\,\mathbf{I}_{3\times3} & \mathbf{G}_\bullet{}^{y_k} & \mathbf{0} & \mathbf{e}_{ji} & \mathbf{0} \\ -\mathbf{e}_j^{z_k}\,\mathbf{I}_{3\times3} & \mathbf{G}_\bullet{}^{z_k} & \mathbf{0} & \mathbf{0} & \mathbf{e}_{ji} \end{array}\right) \begin{pmatrix} \mathbf{P}_i \\ \mathbf{P}_j \\ \mathbf{P}_k \end{pmatrix} = \mathbf{0}$$

Here, the 27 components of $\mathbf{G}_{A_j}{}^{B_iC_k}$ are viewed as three 3×3 matrices, for $C_k = x, y, z$. As before, the rank remains 5 even if further bilinear or trilinear closure constraints are added for the same images taken in a different order (but *cf.* the discussion on scaling below). Any rank 5 decomposition $\mathbf{U}_{ijk}$ of this constraint matrix (*e.g.* by SVD) gives a trivalent equivalent of the above $\mathbf{U}_{ij}$ matrix. For any such $\mathbf{U}_{ijk}$, each of its 5 rows contains three 3-component row vectors which define a matching triplet of image lines, and hence a corresponding 3D line. (If $\{\mathbf{u}_i, \mathbf{u}_j, \mathbf{u}_k\}$ is such a triplet, the closure constraint says that the pulled-back visual planes meet in a common 3D line: $(\mathbf{u}_i\mathbf{P}_i) + (\mathbf{u}_j\mathbf{P}_j) + (\mathbf{u}_k\mathbf{P}_k) = \mathbf{0}$). The 4D projective space of linear combinations of these 5 line-triplet vectors bijectively spans the entire 4D space (Plücker quadric) of lines in 3D, *except* that the correspondence is singular for lines in the trifocal plane.

The complete closure-based reconstruction process runs roughly as follows. A very large number of closure constraints is available, relating the projections of any selection of 2, 3, or even (for higher closure constraints) 4 or 5 images. It would be impractical to enforce all of these, but in any case they are highly redundant and only a small subset of them need be used in practice. The choice must depend on the correspondences and matching tensors available, convenience, and a run time *vs.* redundancy trade-off. To fully constrain the projections, each image (except the first pair) must be related to *at least* two others. This can be done with one **e-G-e** constraint or two **F-e** ones, in either their full or reduced (**U**-matrix) versions. (The experiments below use the full versions).

This paper considers only the simplest possible choices, based on minimal sets of constraints for the first two types of closure relation. Each image is connected to exactly two previous ones in a chain. The following types of chain have been considered



Serial chains connect each image to the two immediately preceding ones, while parallel ones connect each image to two 'key frames'. For the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ chains, the trivalent tensor based in (with covariant index in) the middle image of the triplet is used, *e.g.* , $\mathbf{e}_2^{A_1} - \mathbf{G}_{B_2}{}^{A_1 C_3} - \mathbf{e}_2^{C_3}$ for images 1-2-3. Note that the basic formulation is symmetric in that it allows any pair or triplet of images to be incorporated. Choosing a particular constraint topology breaks this symmetry, but the choice is at least under user control (modulo suitable estimates of the matching tensors).

Each constraint contributes several rows to a big $3m$-column, m image constraint matrix (unused elements are zero). It is essential to choose consistent relative scalings (see below), but once this is done the constraint matrix generically has rank $3m - 4$. Its null space is exactly the joint image (the 4D space spanned by the joint projection columns). Any basis for the null space provides four $3m$-component column vectors that can be regarded as the columns of a valid reconstructed joint projection. The freedom of choice in the basis corresponds to a 4×4 nonsingular mixing of the columns, which amounts to a projective deformation of the reconstructed world coordinates.

The above process enforces a particular relative scaling for the projection matrices, so it is necessary to choose coherent scalings for the overlapping constraint equations. In fact, matching tensors inherit 'natural' scalings from their definitions as minors of projection matrices, but these are lost when they are estimated from image data. The closure relations depend critically on these scalings, so the relevant part of them must be recovered.

It turns out that the scales can be chosen arbitrarily modulo one constraint for each closed loop in the above chains. The same constraints guarantee the existence of consistent choices of depths in the depth recovery equations (4) or (5), and it turns out to be easiest to recover the scalings using this. For each closed loop, scalings are chosen arbitrarily and the depths of (a selection of) measured image points are propagated around the loop by a chain of depth recovery steps (*cf.* [25]). Then, one of the tensor scales is modified to make the average 'closed-loop gain' unity, as it must be for consistency. For the $\mathbf{F}$-$\mathbf{e}$ constraint this involves 3-image loops (*e.g.* $1 \to 2 \to 3 \to 1$), while for the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ one we multiply (5) by $[\mathbf{e}_{21}]_\times$ so that only two terms survive, and then propagate through just two images (*e.g.* $2 \to 3 \to 2$). The required epipoles are also estimated from $\mathbf{G}$ and (5), by multiplying by $[\mathbf{x}_1]_\times$ or $[\mathbf{x}_3]_\times$ and solving. The epipoles and scalings could also be found bilinearly from $\mathbf{G}$ alone, but for maximum stability I prefer to use linear methods based on the image data.

Numerically, once the combined constraint matrix has been assembled there are several ways to calculate its null space. The experiments reported here use the four smallest singular vectors of the SVD, but eigendecomposition of the normal matrix gives similar results. These methods are numerically stable and easily handle redundant constraints, but all of them are rather slow when there are many images, as large matrices with many zeros are involved. With sparse sets of constraints (as here), the null-space could also be estimated using various sparse or recursive methods. These should be much faster than the full SVD, although some stability may be lost — more investigation is needed here.

In fact, it is clear (in retrospect) from the above discussion that one can also view closure-based reconstruction as a means of 'gluing together' many overlapping virtual 2 or 3 image reconstructions into a coherent multi-image whole. Each reconstruction implicitly provides a 6×4 or 9×4

joint projection matrix in some arbitrary world frame. The closure-based framework characterizes these by their 2 or 5 dimensional left null spaces. These have the advantage of being independent of the world frames chosen, and directly extractable from the matching tensors without passing through an explicit intermediate reconstruction. Finally, the accumulated null space constraints are re-inverted to give the combined joint projection matrix. In retrospect, it is unclear whether passing through a large $(3m-4)$-D null space computation is an effective means of patching together several (implicit) 4D partial reconstructions. This must rest as a subject for future work.

In practice, the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ method turns out to be quite a lot slower than the $\mathbf{F}$-$\mathbf{e}$ one, mainly because larger matrices are involved at each step. However it is also significantly more stable. In particular, for a camera moving in a straight line, the fundamental matrices and epipoles of different images coincide. This is a well-known singular case for epipolar-line-based token transfer, and $\mathbf{F}$-$\mathbf{e}$ closure based reconstruction fails here too. The failure is intrinsic to any method based solely on epipolar geometry (rather than image measurements). Camera zooms centred on the unique epipole leave the epipolar geometry unchanged and hence can not be recovered. (The problem still exists for two images, but there it can be absorbed by a 3D homography). In contrast, trivalent transfer and $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ reconstruction are well behaved for aligned centres, as is reconstruction by $\mathbf{F}$-$\mathbf{e}$ depth recovery and factorization. Basically, some information about positions along epipolar lines is needed to stabilize things. This can be provided by trivalent transfer, or even better by anchoring onto explicit image correspondences.

3 Implementation

Now we summarize the reconstruction algorithms, and discuss a few important implementation details. The $\mathbf{F}$-$\mathbf{e}$ closure algorithm has the following steps:

0) Extract and match features between images.
1) Standardize the image coordinates (see below).
2) Estimate fundamental matrices and epipoles connecting each image to at least two others.
3) Correct the scales of the fundamental matrices and epipoles using (4) (*cf.* section 2).
4) Build the constraint matrix of equations (1) and use SVD to find its 4D null space.
5) Extract the projection matrices from the null space column vectors.
6) Back-project and solve for 3D structure using (3).
7) De-standardize the projection matrices (see below).

The $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ closure based method follows the same pattern, except that: (*i*) both point and line features can be used to estimate the trivalent tensors; (*ii*) equation 5 is used to correct the trivalent scaling, and equation (2) to build the constraint matrix.

The current implementations use linear methods to estimate fundamental matrices and trivalent tensors. With properly standardized coordinates these turn out to be very stable and surprisingly accurate [12]. Using a nonlinear least squares iteration to refine the estimates marginally improves the stability of (for example) the long serial chains of the $\mathbf{F}$-$\mathbf{e}$ method, but not enough to change the basic conclusions. The linear method for $\mathbf{F}$ includes a final 3×3 SVD to enforce $\det \mathbf{F} = 0$ and calculate the epipoles. The epipoles for the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ method are found linearly from $\mathbf{G}$ and the image data using (5).

For accurate results it is *essential* to work in a well-adapted coordinate system. This is standard numerical practice, but it is particularly important when there are implicit least-squares trade-offs between redundant constraints, as here. If some components of the input vectors are typically much larger than others — for example when homogeneous pixel coordinates $(x, y, z) \sim (256, 256, 1)$ are used — some constraints have a much higher implicit weight than others and this significantly

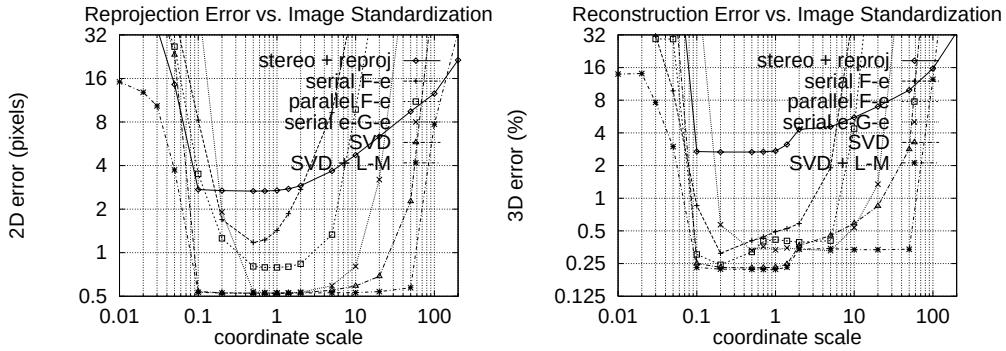


Figure 1: Mean reprojection and reconstruction error *vs.* image coordinate standardization.

distorts the estimated solution. Hartley has underlined the importance of this for fundamental matrix estimation [12], and it is equally true for reconstruction. In practice it makes little difference which of the many possible standardization schemes is used. Here, the pixel coordinates are scaled uniformly into the unit square $[-1,1]\times[-1,1]$, homogenized, and normalized as 3-vectors to norm 1. This is easy, fast, independent of the image, and works equally well for visible and off-image virtual points (*e.g.* distant vanishing points or epipoles). Figure 1 shows the effect of standardization: pixel coordinates (scale ~ 256) give reconstructions hundreds of times worse than well standardized ones (scale ~ 1). The error rises rapidly at scales below 10^{-1} owing to (32 bit) floating point truncation error.

4 Experiments

To help quantify the performance of the algorithms, I have run a series of simulations using synthetic data. The algorithms have also been tested on hand-matched points extracted from real images, and an implementation on 'live' images is in progress. The simulations are based on trial scenes consisting of random 3D points in the unit cube. These are viewed by identical perspective cameras spaced evenly along a 90° arc of radius 2, looking directly at the centre of the scene. These are ideal conditions for accurate reconstruction, but many other configurations have also been tested, including infinitesimal viewing angles and distant scenes with negligible perspective. When cameras are added, their spacing is decreased so that the total range of viewing angles remains the same. The positions of the projected image points are perturbed by uniform random noise. Mean-square (and median and maximum) 2D reprojection and 3D reconstruction errors are accumulated over 50 trials. The 3D error is the residual after projective alignment of the reconstruction with the scene. Unless otherwise stated, default values of 10 views, 50 points and ± 1 pixel noise are used.

Figure 2 summarizes the results, giving image reprojection and 3D reconstruction errors *vs.* image noise, number of points and number of views. The new techniques under test are serial and parallel chain $\mathbf{F}$-$\mathbf{e}$ closure, and serial chain $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ closure. For comparison, several existing techniques are also shown.

Evidently, the most stable techniques are 'SVD' and 'SVD+L-M': SVD-based projective factorization [25, 29], and a Levenberg-Marquardt-like nonlinear least squares algorithm initialized from this. However, remember that these are only applicable when points can be matched across all images, while the other techniques require matches across only 2-3 images[4].

The '2 image' methods simply reconstruct the scene from two images, and then reproject to

[4]To allow fair comparison, the point reconstruction step for each method has been allowed to combine data from all the images using the recovered projections.

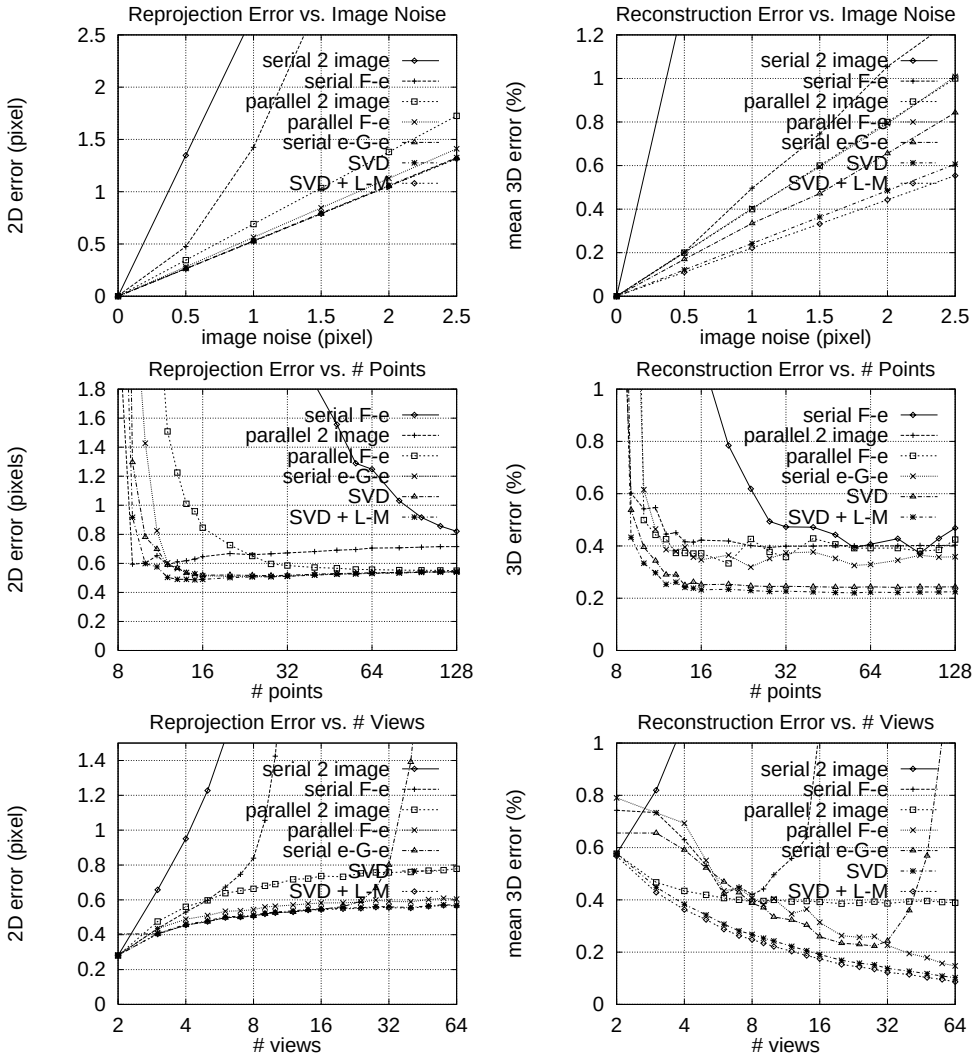


Figure 2: Mean reprojection and reconstruction error *vs.* noise, number of points and number of views.

estimate the projection matrices for the remaining ones. The ‘serial 2 image’ method uses only the first two images, and hence involves a considerable amount of extrapolation. This can be very inaccurate, but it is realistic in the sense that practical two image methods are often restricted to nearby images when tracking is difficult. The serial **F-e** and **e-G-e** closure methods fuse a series of small, inaccurate steps of this sort and still manage to produce significantly better results, despite the potential for accumulation of errors.

In contrast, the ‘parallel 2 image’ method uses the first and last images of the sequence, and hence maintains a constant baseline. The same applies to the ‘parallel **F-e**’ closure method, which links each image to the two end ones. These results require unrealistically wide matching windows, but they provide a clear indication of the “integrating power” of the closure formalism. In particular, adding more images does continue to improve the ‘parallel **F-e**’ closure results, while the ‘parallel 2 image’ results stay roughly constant (as expected). However, the closure method seems to need about 10 images just to overcome the extreme stability of the 2 image factorization method.

All of the methods scale linearly with noise and initially improve as more points are added, but level off after about 20 points. The serial methods eventually worsen as more images are added and their baseline decreases: the ‘2 image’ one immediately (as expected); the **F-e** one after about 10

images; and the **e**-**G**-**e** one after about 30. In general, the trivalent methods are significantly more stable than the fundamental matrix ones. It definitely pays to select images as widely separated as possible for the closure constraints, even if this means having to use several 'key' images. The instabilities arising from long chains seem to be far greater than any biases introduced by working from 'key' images. However, tracking reliability puts strong practical limitations on the separations that can be attained.

All of the methods are stable for both close and distant scenes (modulo straight line motion for **F**-**e** closure), but all of them (especially the fundamental matrix ones) give very poor results for points near the axis of fronto-parallel motion, as there is no stereo baseline there for point reconstruction. (Surface continuity constraints are essential in this case).

One reason for the early failure of **F**-**e** closure is the fact that it is singular whenever three adjacent camera centres are aligned. This happens to an increasing extent as the spacing along the circular baseline decreases, adding to the natural uncertainty associated with the short baseline itself. For this reason, it is advisable to use the **e**-**G**-**e** method (or an equivalent **U** matrix derived from reconstruction of at least 3 images) whenever straight line motions are involved.

The factorization method is notable for being linear yet close to optimal. It is based on **F**-**e** depth recovery (4) — essentially the same equations as the **F**-**e** closure based method, but applied directly to the image points rather than to the projections. Clearly, the direct use of image data gives a significant improvement in accuracy. Unfortunately, factorization is practically limited as it requires every token to be visible in every image: this is why the closure-based methods were developed.

5 Summary

The closure relation based projective reconstruction techniques work reasonably well in practice, except that the **F**-**e** method fails for aligned camera centres. If there are many images, closure is more accurate than the common 'reconstruct from 2 images and reproject for the other projections' paradigm, but it can not compete with projective factorization when features can be tracked through all the images. In principle there is no need to single out 'privileged' features or images. But short chains of closure relations turn out to be significantly more stable than long ones, so in practice it is probably best to relate all of the images to a few 'key' ones (or perhaps hierarchically). The trivalent techniques are slower, but significantly more stable than the fundamental matrix based ones.

Future work will implement the methods on real images, investigate fast recursive solutions of the reconstruction equations, study the stabilizing effects of incorporating redundant constraints, and compare the closure-based methods with direct techniques for merging several partial reconstructions.

References

[1] K. B. Atkinson. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, Roseleigh House, Latheronwheel, Caithness, Scotland, 1996.

[2] S. Carlsson. Multiple image invariance using the double algebra. In J. Mundy, A. Zissermann, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[3] S. Carlsson. Duality of reconstruction and positioning from projective views. In P. Anandan, editor, *IEEE Workshop on Representation of Visual Scenes*. IEEE Press, 1995.

[4] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[5] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

[6] O. Faugeras. Stratification of 3-d vision: Projective, affine, and metric representations. *J. Optical Society of America*, A 12(3):465–84, March 1995.

[7] O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[8] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[9] W. Förstner. 10 pros and cons of performance characterization in computer vision. In *Workshop on Performance Characterization of Vision Algorithms*, Cambridge, U.K., 1996.

[10] R. Hartley. Euclidean reconstruction from multiple views. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.

[11] R. Hartley. Lines and points in three views – an integrated approach. In *Image Understanding Workshop*, Monterey, California, November 1994.

[12] R. Hartley. In defence of the 8-point algorithm. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1064–70, Cambridge, MA, June 1995.

[13] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 761–4, Urbana-Champaign, Illinois, 1992.

[14] R. Hartley and P. Sturm. Triangulation. In *ARPA Image Understanding Workshop*, pages 957–66, Monterey, November 1994.

[15] A. Heyden. Reconstruction from image sequences by means of relative depths. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1058–63, Cambridge, MA, June 1995.

[16] A. Heyden and K. Åström. A canonical framework for sequences of images. In *IEEE Workshop on Representations of Visual Scenes*, Cambridge, MA, June 1995.

[17] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993.

[18] P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 314–20, Cambridge, MA, June 1995.

[19] R. Mohr, B. Boufama, and P. Brand. Accurate projective reconstruction. In 2^{nd} *Europe-U.S. Workshop on Invariance*, page 257, Ponta Delgada, Azores, October 1993.

[20] J. Oliensis and V. Govindu. Experimental evaluation of projective reconstruction in structure from motion. Technical report, NEC Research Institute, 4 Independence Way, Princeton N.J., 1995.

[21] C. J. Poelman and T. Kanade. A parapersective factorization method for shape and motion recovery. In J-O. Eklundh, editor, *European Conf. Computer Vision*, pages 97–108, Stockholm, 1994. Springer-Verlag.

[22] A. Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 17(8):779–89, 1995.

[23] G. Sparr. A common framework for kinetic depth, reconstruction and motion for deformable objects. In J-O. Eklundh, editor, *European Conf. Computer Vision*, pages 471–82, Stockholm, 1994. Springer-Verlag.

[24] G. Sparr. Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In *Int. Conf. Pattern Recognition*, Vienna, 1996.

[25] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–20, Cambridge, U.K., 1996. Springer-Verlag.

[26] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Computer Vision*, 9(2):137–54, 1992.

[27] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to *Int. J. Computer Vision*.

[28] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.

[29] B. Triggs. Factorization methods for projective structure and motion. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 845–51, San Francisco, CA, 1996.

[30] B. Triggs. Autocalibration and the absolute quadric. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[31] M. Werman and A. Shashua. The study of 3D-from-2D using elimination. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 473–9, Cambridge, MA, June 1995.

[32] A. Zisserman and S.J. Maybank. A case against epipolar geometry. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 69–88, Ponta Delgada, Azores, October 1993.

Chapitre 5

Auto-calibrage d'une caméra en mouvement

Les travaux précédents relèvent tous de la structure *projective* – ou implicite en terme de contraintes et de tenseurs d'appariement, ou explicite en terme d'une reconstruction et des caméras projectives. Cette structure contient déjà une description quasi-complète de la scène. Seulement les valeurs de 9 paramètres manquent comme cela a déjà été dit plus haut: 3 pour la déformation projective (location du plan à l'infini) ; 5 pour la déformation affine ; et un facteur d'échelle global qui ne peut jamais être retrouvé sans informations externes. Mais la plupart des applications exigent une structure métrique. Pour retrouver ces derniers 8–9 paramètres, il nous faut des contraintes qui sont situées au delà des images et du modèle projectif non-calibré: elles viennent ou du calibrage interne des caméras, ou de leur mouvement, ou de la scène elle-même. Les trois cas sont intéressants et ont été bien étudiés [MF92, Har93a, MBB93, Fau95, ZF96].

Ici, on se limite au cas de « **contraintes non-mesurées** » sur les calibrages internes – par là on entend toute contrainte qui peut vraisemblablement figurer dans nos connaissances préalables, sans effectuer des mesures précises qui relèvent d'une calibrage classique. On appel « **auto-calibrage** » l'obtention des calibrages et/ou de la structure métrique de la scène (jusqu'au facteur d'échelle près) à partir des contraintes non-mesurées. En particulier, on verra comment la simple connaissance de *constance* du calibrage interne d'une caméra en mouvement peut suffire pour obtenir les valeurs numériques des paramètres des caméra ainsi que la structure 3D euclidienne.

On pourrait éventuellement autoriser à faire varier plusieurs paramètres (ex. focale et point principal), et considérer que certaines connaissances numériques (skew nul, rapport d'échelle égal à un) font partie de l'auto-calibrage, car elles sont très stables et souvent connues par défaut à précision suffisante. Les contraintes décrites ci-dessous s'adaptent facilement à ce genre de problème, mais les articles présentés se limitent au cas des paramètres internes constants et inconnus.

5.1 Resumé de « Autocalibration and the Absolute Quadric » – CVPR'97

Historique

Ce papier fut publié à CVPR'97. Il représente mon travail de base sur l'auto-calibrage. À l'époque, il y avait déjà plusieurs études, soit suivant l'approche originale de Maybank & Faugeras [MF92] liée aux contraintes « **de Kruppa** » entre paires d'images [FLM92, ZF96], soit fondée sur l'estimation préliminaire de la structure affine (par voie ou de la homographie ou du plan à l'infini) [Har93a, BRZM95] – approche qui était nommée plus tard « **stratification** » [Fau95]. On peut ci-

ter aussi plusieurs travaux sur les cas particuliers (mouvements spécifiques des caméras, calibrages partiels, ou structures de scènes particulières comme l'observation de parallélogrammes rectangles) [Har92, Har93b, HCP94, BRZM95, Har94, PGP96].

Méthode

La clé de l'auto-calibrage est la façon d'implanter la géométrie euclidienne dans l'espace projectif. Les transformations projectives sont relativement grossières – elles ne préservent ni laissent distinguer qu'une partie de la structure métrique. Pour retrouver la structure perdue de cette façon, on peut se pencher sur une grande variété des connaissances euclidiennes : de la structure 3D observée, des mouvements, ou des calibrages des caméras. Ici, on prend comme base la constance des calibrages caméras, et il faut se limiter aux aspects de la géométrie 3D euclidienne qui s'appliquent indifféremment à toutes les scènes. Les différences intrinsèques entre l'espace euclidien (jusqu'à une facteur d'échelle près) et l'espace projectif pourraient être réduites dans un seul objet géométrique – « **la quadrique absolue duale** » – qui mesure pour l'essentiel les angles entre les vecteurs normaux de plans 3D. Dans un repère euclidien, sa forme matricielle est une matrice 4×4 symétrique de rang 3

$$\mathbf{\Omega} \;=\; \begin{pmatrix} \boldsymbol{I}_{3\times 3} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix}$$

avec pour loi de transformation sous les transformations projectives $\mathbf{X} \rightarrow \mathbf{T}\,\mathbf{X}$

$$\mathbf{\Omega} \;\longrightarrow\; \mathbf{T}\,\mathbf{\Omega}\,\mathbf{T}^{\top}$$

Cette matrice est invariante par toute transformation euclidienne $\mathbf{T} = \left(\begin{smallmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{smallmatrix}\right)$, mais sous des transformations projectives générales elle devient une matrice générale symétrique positive semi-définie de rang 3. Elle résume la structure affine – le plan à l'infini $(\boldsymbol{0}\ 1)$ est son vecteur nul – et aussi la structure métrique angulaire – l'angle θ entre deux plans $\mathbf{p}$ et $\mathbf{q}$ est

$$\cos\theta \;=\; \frac{\mathbf{p}\,\mathbf{\Omega}\,\mathbf{q}^{\top}}{\sqrt{(\mathbf{p}\,\mathbf{\Omega}\,\mathbf{q}^{\top})\,(\mathbf{q}\,\mathbf{\Omega}\,\mathbf{q}^{\top})}}$$

Si on peut localiser $\mathbf{\Omega}$ dans une reconstruction projective d'une scène, il est alors facile de « **rectifier** » l'espace pour obtenir la structure euclidienne.

La projection de $\mathbf{\Omega}$ dans une caméra $\mathbf{P} \;=\; \boldsymbol{K}\,(\boldsymbol{R}\,|\,\boldsymbol{t})$ est « **l'image de la quadrique absolue duale** »

$$\boldsymbol{\omega} \;\simeq\; \mathbf{P}\,\mathbf{\Omega}\,\mathbf{P}^{\top} \;=\; \boldsymbol{K}\boldsymbol{K}^{\top}$$

où $\boldsymbol{K}$ est la matrice de calibrage de la caméra. Si la calibrage $\boldsymbol{K}$ est constant entre images, $\boldsymbol{\omega}$ est aussi constant, ce qui nous donne un système d'équations algébriques qui lie les matrices de projection (par exemple d'une reconstruction projective) et les matrices inconnues $\mathbf{\Omega}$ et $\boldsymbol{\omega}$.

On montre alors que ces équations de projection de $\mathbf{\Omega}$ peuvent être résolues à partir de 3 images. La méthode de résolution préférée est l'optimisation numérique sous contraintes par programmation quadratique séquentielle : l'erreur résiduelle des équations de projection est minimisée, avec pour contrainte le fait que $\mathbf{\Omega}$ soit de rang 3. Cette méthode se révèle en pratique très stable en comparaison avec d'autres méthodes d'auto-calibrage, et semble fiable même avec une initialisation arbitraire. Comme toujours en auto-calibration, les caméras doivent tourner sur deux axes significativement non-parallèles – sinon il y a une ambiguïté dans la structure et les calibrages retrouvées.

5.2 Resumé de « Autocalibration from Planar Scenes » – ECCV'98

Ce papier fut publié à ECCV'98. Il refond le formalisme de la quadrique absolue duale en terme d'une base de vecteurs de directions, et de là il étend la théorie d'auto-calibrage précédente au cas où la scène est plane.

L'apport de la base de directions est plus esthétique que fondamental – pour les « non-initiés » aux tenseurs, elle est moins abstraite et plus intuitive que la quadrique absolue duale, et elle simplifie significativement certaines dérivations.

C'est peut être un peu surprenant que l'auto-calibrage à base d'une scène plane soit même possible : dans ce cas, la structure 3D projective – point de départ pour la méthode 3D ci-dessus – n'est plus disponible, donc l'étape d'initialisation est nettement moins évident. Néanmoins, les contraintes d'auto-calibrage sont toujours actives, et il suffit d'en empiler suffisamment pour rendre le système bien contraint et résoluble. Ce n'est en effet que cela qu'on propose : on raffine l'étape d'initialisation, et applique les contraintes de façon numérique. Un nombre relativement important de vues différentes sont nécessaires, mais au niveau de sa stabilité numérique la méthode semble plus ou moins satisfaisante.

Cette méthode peut être vue comme : (*i*) une généralisation de la méthode de « caméra tournante » de Richard HARTLEY [Har94, Har97], pour le cas où les translations de la caméra sont aussi autorisées ; (*ii*) une généralisation (mais qui était publié avant) des méthodes de « calibrage plan » de STURM & MAYBANK [SM99] et de ZHANG [Zha98] (aussi mentionné par LIEBOWITZ & ZISSERMAN [LZ98]), dans les cas où la structure du plan n'est pas connue. Ces dernières méthodes sont facile à implanter et très efficaces en pratique quand leurs hypothèses respectives sont vérifiées.

Autocalibration and the Absolute Quadric

Bill Triggs
INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot St. Martin, France.
Bill.Triggs@inrialpes.fr ◇ *http://www.inrialpes.fr/movi/people/Triggs*

Abstract

We describe a new method for camera autocalibration and scaled Euclidean structure and motion, from three or more views taken by a moving camera with fixed but unknown intrinsic parameters. The motion constancy of these is used to rectify an initial projective reconstruction. Euclidean scene structure is formulated in terms of the **absolute quadric** — the singular dual 3D quadric (4×4 rank 3 matrix) giving the Euclidean dot-product between plane normals. This is equivalent to the traditional absolute conic but simpler to use. It encodes both affine and Euclidean structure, and projects very simply to the dual absolute image conic which encodes camera calibration. Requiring the projection to be constant gives a bilinear constraint between the absolute quadric and image conic, from which both can be recovered nonlinearly from $m \geq 3$ images, or quasi-linearly from $m \geq 4$. Calibration and Euclidean structure follow easily. The nonlinear method is stabler, faster, more accurate and more general than the quasi-linear one. It is based on a general constrained optimization technique — sequential quadratic programming — that may well be useful in other vision problems.

Keywords: autocalibration, absolute quadric, multiple images, Euclidean reconstruction, constrained optimization.

1 Introduction

Camera calibration is traditionally based on explicit 3D scene or motion measurements, but even for unknown motions in an unknown scene there are strong rigidity constraints relating the calibration to the image, scene and motion. **Autocalibration** is the recovery of calibration and motion from an unknown scene using rigidity. Structure follows easily from this.

With arbitrary cameras, structure can only be recovered up to an overall projectivity. Additional constraints are required to 'Euclideanize' it. We will focus on the traditional case of a single camera with fixed but unknown intrinsic parameters moving arbitrarily in the scene [13, 4, 7], but our formalism easily extends to handle multiple cameras and prior calibration, motion or scene constraints. Alternative approaches restrict the motion to a pure rotation [8] or a plane [1]; handle zoom modulo an initial pre-calibration [15, 16]; or assume a rigidly moving stereo head [22]. For practical applications it is important to exploit any constraints that may be available, as this both increases stability and allows autocalibration from more restricted types of motion.

Used on its own, autocalibration has several notable weaknesses: (*i*) scene scale can not be recovered — small motions in a small scene are indistinguishable from large motions in a large one; (*ii*) *generic* motions — independent rotations and some translation — are required for a unique (up to scale) solution: many common types of motion are degenerate cases; (*iii*) past formulations have tended to be complex and ill-conditioned, often adding further degeneracies of their own; (*iv*) it has been hard to incorporate additional knowledge except during a final bundle adjustment, exacerbating the degeneracy and ill-conditioning problems.

This paper focuses on the last two points, contributing a simpler, more direct problem formulation and a well-behaved numerical algorithm that easily handles additional constraints.

2 The Absolute Quadric

We work in homogeneous coordinates, initially Euclidean, later projective. Finite points and asymp-

This paper appeared in CVPR'97. The work was supported by INRIA Rhône-Alpes and Esprit LTR project CUMULI. I would like to thank P. Sturm for useful discussions and R. Horaud and G. Csurka for supplying calibration data.

totic directions ('points at infinity') are given by column vectors $\mathbf{x} = (\boldsymbol{x}\ 1)^\top$ and $\mathbf{v} = (\boldsymbol{v}\ 0)^\top$. A *row* vector $\mathbf{p} = (\boldsymbol{n}\ d)$ specifies a plane with normal $\boldsymbol{n}$ and offset $-d$. $\mathbf{x}$ lies on $\mathbf{p}$ iff its signed distance from it vanishes: $\mathbf{p}\,\mathbf{x} = \boldsymbol{n} \cdot \boldsymbol{x} + d = 0$. The **plane at infinity** $\mathbf{p}_\infty = (\boldsymbol{0}\ 1)$ contains the infinite points $(\boldsymbol{d}\ 0)$ and no finite ones.

Change-of-basis transformations are 4×4 matrices acting by left multiplication on points ($\mathbf{x} \to \mathbf{T}\,\mathbf{x}$) and by right multiplication by the *inverse* on planes ($\mathbf{p} \to \mathbf{p}\,\mathbf{T}^{-1}$) so that point-plane products are preserved: $\mathbf{p}\,\mathbf{x} = (\mathbf{p}\,\mathbf{T}^{-1})(\mathbf{T}\,\mathbf{x})$. **Euclidean** transformations take the form $\begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix}$ where $\boldsymbol{R}$ is a 3×3 rotation matrix ($\boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}$) and $\boldsymbol{t}$ a translation vector. $\boldsymbol{R}$ becomes a rescaled rotation for **scaled Euclidean** or **similarity** transformations, and an arbitrary nonsingular 3×3 matrix for **affine** ones. For **projective** transformations $\mathbf{T}$ is an arbitrary nonsingular 4×4 matrix.

To distinguish their very different transformation laws, points are called **contravariant**, and planes **covariant**. Matrices and higher dimensional arrays (tensors) have a different transformation law associated with each index. **Contraction** ('projective dot product' or sum over products of components) is only meaningful between contravariant-covariant index pairs (*e.g.* a point and a plane). Otherwise the result is completely basis-dependent.

The **absolute quadric** is the symmetric 4×4 rank 3 matrix $\boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix}$. It is defined to be contravariant (point-like) in each index, so $\boldsymbol{\Omega} \to \mathbf{T}\,\boldsymbol{\Omega}\,\mathbf{T}^\top$ under change-of-basis transforms $\mathbf{x} \to \mathbf{T}\,\mathbf{x}$. It follows that $\boldsymbol{\Omega}$ is invariant under Euclidean transformations, is rescaled under similarities, takes the form $\begin{pmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix}$ (symmetric 3×3 nonsingular $\boldsymbol{Q}$) under affine ones, and becomes an arbitrary symmetric 4×4 rank 3 matrix under projective ones.

Being contravariant, $\boldsymbol{\Omega}$ can be contracted against plane vectors. Given a finite plane $\mathbf{p}$, $\boldsymbol{\Omega}\,\mathbf{p}^\top$ is the point at infinity representing its Euclidean normal direction. The plane at infinity is $\boldsymbol{\Omega}$'s unique null vector: $\boldsymbol{\Omega}\,\mathbf{p}_\infty^\top = \mathbf{0}$. The Euclidean dot product of the normals of two finite planes $\mathbf{p}$ and $\mathbf{p}'$ is $\boldsymbol{n} \cdot \boldsymbol{n}' = \mathbf{p}\,\boldsymbol{\Omega}\,\mathbf{p}'^\top$, and the angle between them is $\cos\theta = (\mathbf{p}\,\boldsymbol{\Omega}\,\mathbf{p}'^\top)/\sqrt{(\mathbf{p}\,\boldsymbol{\Omega}\,\mathbf{p}^\top)(\mathbf{p}'\,\boldsymbol{\Omega}\,\mathbf{p}'^\top)}$. These formulae apply in any basis provided the corresponding $\boldsymbol{\Omega}$ is used. So $\boldsymbol{\Omega}$ is a projective encoding of both scaled Euclidean (angle between planes) and affine (plane at infinity) structure. Using $\boldsymbol{\Omega}$, it is straightforward to define further Euclidean concepts such as spheres, angles between lines and lines or planes, relative distances, and even (fixing a scale) absolute distances.

In contrast to planes, there is no meaningful "Euclidean dot product" between finite points. However, introducing 3-component coordinates on the plane at infinity, the dot product of two direction vectors becomes $\boldsymbol{u} \cdot \boldsymbol{v} = \boldsymbol{u}^\top \boldsymbol{C}\,\boldsymbol{v}$ where the 3×3 symmetric doubly covariant **absolute conic** matrix $\boldsymbol{C}$ becomes $\boldsymbol{I}$ in any Euclidean basis. The need for separate coordinates on $\mathbf{p}_\infty$ is inconvenient. In world coordinates the direction dot product can be written $\mathbf{u}^\top \mathbf{Q}\,\mathbf{v}$, where $\mathbf{Q}$ is *any* doubly covariant symmetric 4×4 matrix of the form $\begin{pmatrix} \boldsymbol{I} & * \\ * & * \end{pmatrix}$. However there is no *canonical* choice of $\mathbf{Q}$: it *cannot* be invariant under translations. Only the upper 3×3 submatrix (the restriction of $\mathbf{Q}$ to $\mathbf{p}_\infty$) is invariant. Such a $\mathbf{Q}$ converts a point at infinity (direction vector) $\mathbf{d}$ into some finite plane $\mathbf{d}^\top \mathbf{Q}$ orthogonal to it, but there is no canonical choice of such a plane.

The absolute quadric is also much simpler to project into images than the absolute conic. Any doubly contravariant world matrix $\mathbf{M}$ can be projected to a doubly contravariant image one $\boldsymbol{m}$ according to $\boldsymbol{m} \sim \mathbf{P}\,\mathbf{M}\,\mathbf{P}^\top$, where $\mathbf{P}$ is the usual 3×4 point projection $\mathbf{x} \to \mathbf{P}\mathbf{x}$. This applies both to skew Plücker line matrices $\mathbf{L}$ and symmetric dual quadric matrices $\mathbf{Q}$. In each case the result represents the actual image of the 3D object (skew matrix representation $[\boldsymbol{l}]_\times$ of image line $\boldsymbol{l}$, and dual image conic $\boldsymbol{q}$ representing the image of the dual quadric $\mathbf{Q}$'s occluding contour). $\boldsymbol{\Omega}$'s projection $\boldsymbol{\omega} \equiv \mathbf{P}\,\boldsymbol{\Omega}\,\mathbf{P}^\top$ is the **dual absolute image conic** — a symmetric 3×3 rank 3 image matrix. Using 3×3 RQ decomposition to expand the projection $\mathbf{P} = \boldsymbol{K}\boldsymbol{R}\,(\boldsymbol{I} \mid -\boldsymbol{t})$ into the traditional upper triangular **calibration matrix** $\boldsymbol{K}$, rotation $\boldsymbol{R}$ and translation to the optical centre $\boldsymbol{t}$, we find that $\boldsymbol{\omega} = \boldsymbol{K}\boldsymbol{K}^\top$ is invariant under rigid motions and encodes the camera's intrinsic parameters. $\boldsymbol{K}$ can be recovered from $\boldsymbol{\omega}$ by Choleski factorization.

The dual and non-dual absolute image conics $\boldsymbol{\omega}$ and $\boldsymbol{\omega}^{-1}$ encode the 3D angular structure implicit in the image measurements. The 3D angle between the visual planes of image lines $\boldsymbol{l}$ and $\boldsymbol{m}$ is $\cos\theta = (\boldsymbol{l}\,\boldsymbol{\omega}\,\boldsymbol{m}^\top)/\sqrt{(\boldsymbol{l}\,\boldsymbol{\omega}\,\boldsymbol{l}^\top)(\boldsymbol{m}\,\boldsymbol{\omega}\,\boldsymbol{m}^\top)}$, while that between the visual rays of image points $\mathbf{x}$ and $\mathbf{y}$ is

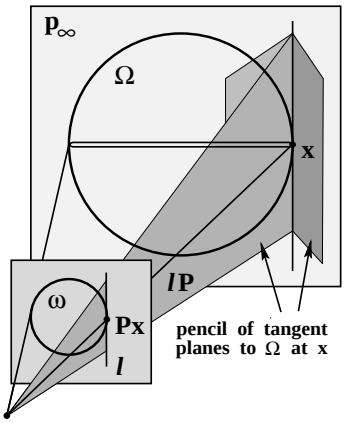


Figure 1: The absolute quadric $\boldsymbol{\Omega}$ is a very flat dual quadric "squashed onto" the plane at infinity, whose rim is the absolute conic $\boldsymbol{C}$.

$$\cos\theta = (\mathbf{x}\,\boldsymbol{\omega}^{-1}\mathbf{y}^\top)/\sqrt{(\mathbf{x}\,\boldsymbol{\omega}^{-1}\mathbf{x}^\top)(\mathbf{y}\,\boldsymbol{\omega}^{-1}\mathbf{y}^\top)}.$$

The above algebra is all we will need to use $\boldsymbol{\Omega}$, but a geometric picture may help intuition. Temporarily allow $\mathbf{x}$ to be complex. Then a symmetric covariant matrix $\mathbf{Q}$ uniquely defines a non-empty **quadric**: a quadratic hypersurface (ellipsoid, hyperboloid,...) given by homogeneous equations $\mathbf{x}^\top\mathbf{Q}\,\mathbf{x} = 0$. The plane $\mathbf{x}^\top\mathbf{Q}$ is called the **dual plane** of $\mathbf{x}$ in $\mathbf{Q}$. $\mathbf{x}$ lies on $\mathbf{Q}$ iff it lies in its own dual plane: $(\mathbf{x}^\top\mathbf{Q})\,\mathbf{x} = 0$. This happens iff $\mathbf{x}^\top\mathbf{Q}$ is tangent to the quadric at $\mathbf{x}$. The **dual** of $\mathbf{Q}$ is the quadric $\mathbf{p}\,\mathbf{Q}^{-1}\mathbf{p}^\top = 0$ in the projective space of all planes. The 'points' of $\mathbf{Q}^{-1}$ are exactly the tangent planes of $\mathbf{Q}$, as is easily seen by replacing $\mathbf{p} \leftrightarrow \mathbf{x}^\top\mathbf{Q}$.

For regular $\mathbf{Q}$ the duality relation is symmetric. For singular $\mathbf{Q}$ the point quadric 'stretches out' to a cone then a plane pair, while in dual-space the quadric collapses onto a plane then a line until only its 'rim' remains (*i.e.* it becomes a dual-space plane conic curve or a point pair). The cone vertex and its dual space supporting plane correspond to the kernel of $\mathbf{Q}$.

Dually, a singular *dual* quadric $\mathbf{Q}^{-1}$ defines a dual-space cone and a point-space conic curve whose dual-space vertex or point-space supporting plane is the null space of $\mathbf{Q}^{-1}$. This is the case with the absolute quadric $\boldsymbol{\Omega}$: it is the degenerate dual-space quadric whose 'rim' is the absolute conic $\boldsymbol{C}$ in $\mathbf{p}_\infty$ (see fig. 1). Dual quadric projection $\mathbf{Q} \rightarrow \mathbf{P}\,\mathbf{Q}\,\mathbf{P}^\top$ is also easy to picture: an image line $\boldsymbol{l}$ is tangent to the image conic iff the pulled back visual plane $\boldsymbol{l}\,\mathbf{P}$ is tangent to the 3D quadric: $\boldsymbol{l}\,(\mathbf{P}\,\mathbf{Q}\,\mathbf{P}^\top)\,\boldsymbol{l}^\top = (\boldsymbol{l}\,\mathbf{P})\,\mathbf{Q}\,(\boldsymbol{l}\mathbf{P})^\top = 0$ (*c.f.* fig. 1).

3 Autocalibration

There are essentially three current approaches to autocalibration, all based on the motion constancy of $\boldsymbol{\omega}$. Multilinear **matching constraints** exist relating 2–10 images of any dual quadric, including $\boldsymbol{\Omega}$. The **Kruppa constraint** is the two image case, originally used to find epipolar geometry for relative orientation from *known* calibration. It essentially says that since epipolar lines correspond via epipolar planes, the above angle-between-visual-planes formula must give the same result for corresponding epipolar lines in either image. A compact derivation applies the **closure identity** [19] $\boldsymbol{F}_{21}\,\mathbf{P}_1 \sim [\boldsymbol{e}_{12}]_\times\,\mathbf{P}_2$ to either side of $\boldsymbol{\Omega}$ to derive the quadric matching constraint $\boldsymbol{F}_{21}\,\boldsymbol{\omega}\,\boldsymbol{F}_{21}^\top \sim [\boldsymbol{e}_{12}]_\times\,\boldsymbol{\omega}\,[\boldsymbol{e}_{12}]_\times^\top$. Allowing for symmetry and rank deficiency, this amounts to 3 linearly or (cross multiplying to eliminate the unknown scale) 2 algebraically independent equations. $\boldsymbol{\omega}$ has 5 d.o.f. so at least 3 images are required. Various resolution procedures exist. Maybank, Faugeras & Luong [13, 4] use algebraic elimination in well-chosen coordinates, Zeller & Faugeras [21] apply least squares optimization over many images, and Hartley (reported in [14]) uses a preliminary SVD based simplification.

The second approach **stratifies** [12, 3] the problem into affine and Euclidean parts. Affine structure is encoded in $\mathbf{p}_\infty$ or the **absolute homography** $\boldsymbol{H}_\infty$ — the inter-image mapping defined by projecting pixels up onto $\mathbf{p}_\infty$. For fixed calibration, $\boldsymbol{H}_\infty = \boldsymbol{K}\,\boldsymbol{R}\,\boldsymbol{K}^{-1}$ is conjugate to a rotation and $\boldsymbol{\omega}$ turns out to be invariant: $\boldsymbol{H}_\infty\,\boldsymbol{\omega}\,\boldsymbol{H}_\infty^\top \sim \boldsymbol{\omega}$ (with equality if $\det(\boldsymbol{H}_\infty) = 1$). This gives a linear constraint on the "Kruppa matrix" $\boldsymbol{\omega}$, sometimes also (misleadingly) called the Kruppa constraint. Since $\boldsymbol{H}_\infty$ fixes the direction $\boldsymbol{d}$ of the rotation axis, $\boldsymbol{\omega} + \lambda\,\boldsymbol{d}\,\boldsymbol{d}^\top$ also satisfies the constraint for any λ. So two rotations with different axes are needed to solve for $\boldsymbol{\omega}$.

If there is negligible translation compared to a visible 'background', $\boldsymbol{H}_\infty$ is an observable inter-image homography so autocalibration is straightforward (but not structure!) [8]. $\boldsymbol{H}_\infty$ can also be found from known vanishing points or 3D parallelism [3]. But for pure autocalibration on finite points, the only constraints on $\mathbf{p}_\infty$ and $\boldsymbol{H}_\infty$ are their relations to $\boldsymbol{\Omega}$,

$\boldsymbol{\omega}$, and $\boldsymbol{K}$. Given a plane $(\boldsymbol{n}\ d)$ and an image projection $\mathbf{P} = \boldsymbol{A}(\boldsymbol{I} \mid -\boldsymbol{t})$, the image-to-plane homography is $\begin{pmatrix} (\boldsymbol{n}\cdot\boldsymbol{t}+d)\,\boldsymbol{I}-\boldsymbol{t}\,\boldsymbol{n} \\ -\boldsymbol{n} \end{pmatrix} \boldsymbol{A}^{-1}$. Specializing to coordinates $\mathbf{P} = (\boldsymbol{I} \mid \boldsymbol{0})$ and projecting into another image $\boldsymbol{A}'(\boldsymbol{I} \mid -\boldsymbol{t}')$ gives a homography $\boldsymbol{H} = \boldsymbol{A}'(d\,\boldsymbol{I} + \boldsymbol{t}'\,\boldsymbol{n})$. If $(\boldsymbol{n}\ d)$ represents $\mathbf{p}_\infty$ in some projective frame, applying this to $\boldsymbol{\omega} \sim \boldsymbol{H}_\infty\,\boldsymbol{\omega}\,\boldsymbol{H}_\infty^\top$ gives equations relating the unknowns $(\boldsymbol{n}\ d)$ and $\boldsymbol{\omega}$. These can be solved iteratively given a reasonable initial guess for $\mathbf{p}_\infty$ or $\boldsymbol{K}$.

Hartley pioneered this sort of approach using bounds on $\mathbf{p}_\infty$ [7]. Most other authors start from an approximate prior calibration [12, 10]. Heyden & Åström's formulation [10] also partially (but independently) foreshadows ours given below. The **modulus constraint** [12, 15] — that $\boldsymbol{H}_\infty = \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{-1}$ being conjugate to a rotation matrix must have the same unit modulus eigenvalues — focuses on $(\boldsymbol{n}\ d)$ by implicitly eliminating $\boldsymbol{\omega}$ or $\boldsymbol{K}$. Armstrong *et. al.* [1] take a more eclectic approach, restricting attention to planar motion and using both parallelism to constrain $\boldsymbol{H}_\infty$ and the motion constancy of the circular points (the 1D analogue of $\boldsymbol{\omega}$).

The Kruppa (epipolar constraint) approach avoids the need to deduce $\boldsymbol{H}_\infty$ indirectly from the constraints, but it can not distinguish $\boldsymbol{\Omega}$ from any other quadric with constant image: planarity (rank $\boldsymbol{\Omega} = 3$) is not directly enforced.

3.1 Absolute Quadric Method

This paper introduces a third approach to autocalibration, which explicitly locates the absolute quadric in an initial projective reconstruction and uses it to 'straighten' the projective structure. $\boldsymbol{\Omega}$ is recovered using the motion constancy of its projection $\boldsymbol{\omega} \sim \mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top$, where $\mathbf{P}_i \sim \boldsymbol{K}\boldsymbol{R}_i(\boldsymbol{I} \mid -\boldsymbol{t}_i)\,\mathbf{T}^{-1}$ for fixed unknown 3×3 and 4×4 transformations $\boldsymbol{K}$ and $\mathbf{T}$ and normalized rotations $\boldsymbol{R}_i$. If we knew the correct relative scaling for the projections, $\boldsymbol{\omega} = \mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top$ would be linear in the unknowns $\boldsymbol{\omega}$ and $\boldsymbol{\Omega}$ and could be solved trivially. Instead, we eliminate the unknown scale by taking ratios of components and cross-multiplying, in much the same way as the point projection $\boldsymbol{x} \sim \mathbf{P}\,\mathbf{x}$ can be rewritten as $\boldsymbol{x} \wedge (\mathbf{P}\,\mathbf{x}) = \mathbf{0}$:

$$\boldsymbol{\omega}^{AB}(\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top)^{CD} - \boldsymbol{\omega}^{CD}(\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top)^{AB} = \mathbf{0}$$

This **absolute quadric projection constraint** is the basis of our autocalibration method. The antisymmetrization interchanges *both* indices AB and CD of the 3×3 symmetric matrices $\boldsymbol{\omega}$ and $\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top$. Viewing these as abstract 6D vectors, we will write this symbolically as

$$\boldsymbol{\omega} \wedge (\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top) = \mathbf{0}$$

For each image, this amounts to $\binom{6}{2} = 15$ bilinear equations (5 linearly independent) in the $10+6 = 16$ independent components of $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$, with coefficients quadratic in the image's reconstructed projection matrix. It can also be written as 9 bilinear equations in $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}^{-1}$ (8 linearly independent):

$$\boldsymbol{\omega}^{-1}\,\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top \;=\; \tfrac{1}{3}\,\mathrm{trace}(\boldsymbol{\omega}^{-1}\,\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top)\cdot\boldsymbol{I}$$

The constraint says that angles between visual planes measured using $\boldsymbol{\Omega}$ must agree with those measured from the corresponding image lines using $\boldsymbol{\omega}$. Roughly speaking, the Kruppa constraint is the projection of the restriction of this to epipolar planes, while the homography constraint $\boldsymbol{\omega} \wedge (\boldsymbol{H}_\infty\,\boldsymbol{\omega}\,\boldsymbol{H}_\infty^\top) = \mathbf{0}$ is the projection of the rotational part of it. At least 3 images are required for a unique solution. For maximum stability it is advisable to include further images, and to enforce $\mathrm{rank}(\boldsymbol{\Omega}) = 3$ (*i.e.* $\det(\boldsymbol{\Omega}) = 0$) and any known scene or calibration constraints.

We will describe two methods of resolving the absolute quadric projection constraints. Both use all $15m$ equations from m images and solve the system in algebraic least squares. The **nonlinear method** uses constrained numerical optimization on $m \geq 3$ images, while the **quasi-linear** method uses SVD based factorization on $m \geq 4$. Only the nonlinear method directly enforces $\det(\boldsymbol{\Omega}) = 0$. It requires a (very approximate) initialization, but turns out to be more accurate, stabler, faster and simpler than the quasi-linear method.

Once $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$ are known, the camera calibration $\boldsymbol{K}$ is easily found by Choleski decomposition of $\boldsymbol{\omega} = \boldsymbol{K}\boldsymbol{K}^\top$. Similarly, a Euclideanizing homography $\mathbf{x} \to \mathbf{T}^{-1}\,\mathbf{x}$, $\mathbf{P} \to \mathbf{P}\,\mathbf{T}$ can be found from the eigen-decomposition $\mathbf{E}\,\boldsymbol{\Lambda}\,\mathbf{E}^\top$ of $\boldsymbol{\Omega} \sim \mathbf{T}\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix}\mathbf{T}^\top$ by setting $\mathbf{T} \sim \mathbf{E}\,\boldsymbol{\Lambda}^{1/2}$ (with the 0 eigenvalue in $\boldsymbol{\Lambda}$ replaced by 1). The columns of $\mathbf{T}$ are an absolute Euclidean basis in projective coordinates

(*i.e.* 3 orthogonal directions and an origin). If required, the rotational part of each rectified projection $\boldsymbol{K}^{-1}\mathbf{P}_i\,\mathbf{T} \sim \boldsymbol{R}_i\,(\boldsymbol{I}\,|\,-\boldsymbol{t}_i)$ can be perturbed to be precisely orthonormal (*e.g.* using quaternions and SVD [11]). As always, a final, close-lying-outlier-insensitive bundle adjustment over all parameters is recommended for precise work.

3.2 Degeneracy

Autocalibration has some intrinsic limitations that apply uniformly to all algorithms. In particular, if the axes of all the camera rotations are parallel (say, vertical), the horizontal-to-vertical aspect ratio of neither the camera nor the scene can be recovered. Intuitively, a narrow scene taken with a wide aspect ratio lens is indistinguishable from a wide scene taken with a narrow lens. This is unfortunate as many real image sequences do preserve a vertical. To avoid this problem, one must either include images with 3 substantially different tilts or cyclotorsions, or rely on prior scene, motion or camera knowledge (*e.g.* aspect ratios). 90° rotations provide the maximum stability, but feature extraction and matching limitations mean that these are usually only possible with pure cyclotorsion.

Formally, if $\mathbf{d} = (\boldsymbol{d}\ \ 0)^\top$ is the 3D direction (common point at infinity) of the rotation axes and $\mathbf{P}_i\,\mathbf{d} = \boldsymbol{K}\boldsymbol{R}_i\,\boldsymbol{d} = \boldsymbol{K}\,\boldsymbol{d}$ (independent of i) is the corresponding image point, adding any multiple of $\mathbf{d}\mathbf{d}^\top$ to $\boldsymbol{\Omega}$ and the same multiple of $(\mathbf{P}\mathbf{d})(\mathbf{P}\mathbf{d})^\top$ to $\boldsymbol{\omega}$ maintains both $\boldsymbol{\omega} \sim \mathbf{P}\,\boldsymbol{\Omega}\,\mathbf{P}^\top$ and $\det(\boldsymbol{\Omega}) = 0$, so it gives another feasible solution. This corresponds to a vertical stretching of both $\boldsymbol{K}$ and the scene.

Pure translation is an even more degenerate case as it fixes *all* points at infinity: affine structure follows easily, but $\boldsymbol{\Omega}$ is essentially arbitrary so autocalibration is impossible. Various other types of motion lead to further degeneracies: Sturm [17] gives a detailed catalog. Such ambiguities must typically be handled by imposing further constraints (known skew, aspect ratio, motion…). This can be difficult with algebraic approaches, but is very easy in our numerical formalism below.

Euclidean structure and motion follow directly from autocalibration, provided only that there is sufficient translation to give a stereo baseline. Translation-neutral internal calibration methods would be useful: Hartley's method [8] requires zero translation, while reconstruction based methods require fairly substantial ones and nonplanar scenes.

3.3 Nonlinear Solution

Now consider how to solve the quadric projection constraints $\boldsymbol{\omega} \wedge (\mathbf{P}_i\,\boldsymbol{\Omega}\,\mathbf{P}_i^\top) = \mathbf{0}$ for $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$, with $\det(\boldsymbol{\Omega}) = 0$. By far the most effective approach turns out to be direct constrained numerical optimization. Numerical approaches are sometimes undervalued in the vision community. Empirically, algebraic elimination on coordinate expressions provides valuable theoretical insight but almost inevitably leads to poor numerical conditioning, while numerical resolution based directly on the original, physically meaningful variables tends to be significantly more stable in practical applications, but too 'opaque' to provide much theoretical insight. At present it is hard to relate the two approaches, but progress in tensorial and Grassmann-Cayley-like formalisms [19, 5] and computational nonlinear algebra (*e.g.* [2]) may soon make this much easier.

Many constrained optimization schemes exist [6]. I will give a brief outline of the simple one used here, as I think that it has considerable potential for other constrained problems in vision. **Sequential Quadratic Programming** [6] is a general numerical scheme for optimizing smooth non-linear cost functions under smooth non-linear constraints. It is Newton-like in that it requires second derivatives of the cost function and potentially provides quadratic convergence. The version presented below is trivial to implement and adequate for our needs. More elaborate versions provide inequality constraints, stabilization and step control schemes.

The goal is to extremize a scalar cost function $f(\mathbf{x})$ subject to a vector of constraints $\mathbf{c}(\mathbf{x}) = \mathbf{0}$. Lagrange multipliers $\mathbf{z}$ give an implicit solution:

$$\nabla f + \mathbf{z}\cdot\nabla\mathbf{c} = \mathbf{0} \qquad \text{with} \qquad \mathbf{c}(\mathbf{x}) = \mathbf{0}$$

Resolve this iteratively starting from some initial guess $\mathbf{x}_0$. Approximate the cost to second order and the constraint to first order at $\mathbf{x}_0$, giving a quadratic optimization subproblem with linear constraints:

$$\min_{\delta\mathbf{x}} \left(\nabla f\cdot\delta\mathbf{x} + \frac{1}{2}\,\delta\mathbf{x}^\top\cdot\nabla^2 f\cdot\delta\mathbf{x}\right)\Big|_{\mathbf{c}+\nabla\mathbf{c}\cdot\delta\mathbf{x}=\mathbf{0}}$$

This subproblem has an exact linear solution:

$$\begin{pmatrix}\nabla^2 f & \nabla\mathbf{c}^\top\\ \nabla\mathbf{c} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\delta\mathbf{x}\\ \mathbf{z}\end{pmatrix} = -\begin{pmatrix}\nabla f\\ \mathbf{c}\end{pmatrix}$$

Solve for $\delta\mathbf{x}$, update $\mathbf{x}_0$ to $\mathbf{x}_1 = \mathbf{x}_0 + \delta\mathbf{x}$, re-estimate derivatives, and iterate to convergence.

In the current application, $\mathbf{x}$ contains the $10+6 = 16$ components of $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$. The cost function is the sum of squared violations of the projection constraints $\sum_i \|\boldsymbol{\omega} \wedge (\mathbf{P}_i \boldsymbol{\Omega} \mathbf{P}_i^\top)\|^2$. The constraint vector $\mathbf{c}$ enforces rank-3-ness $\det(\boldsymbol{\Omega}) = 0$ and normalization $\|\boldsymbol{\omega}\|^2 = \|\boldsymbol{\Omega}\|^2 = 3$. Further knowledge or constraints are easily added (*e.g.* known skew, aspect ratio, principal point,...). A Gauss-Newton approximation (ignoring second derivatives of the quadric projection constraints) was used for the Hessian $\nabla^2 f$.

Initial guesses $\boldsymbol{\Omega}_0$ and $\boldsymbol{\omega}_0$ are required. Using $\boldsymbol{\omega}_0 \wedge (\mathbf{P}\boldsymbol{\Omega}_0\mathbf{P}^\top) = \mathbf{0}$, $\boldsymbol{\Omega}_0$ can be estimated in linear least squares from an approximate calibration $\boldsymbol{\omega}_0 = \boldsymbol{K}_0 \boldsymbol{K}_0^\top$, or $\boldsymbol{\omega}_0$ by projecting an estimated $\boldsymbol{\Omega}_0$ derived from approximate scene constraints. In fact, for $m \geq 4$ images and reasonably well placed cameras (*i.e.* several independent rotations and translations), spurious solutions seem to be rare and any initialization will do. The choices $\boldsymbol{\omega}_0 = \boldsymbol{I}$ and $\boldsymbol{\Omega}_0 = \mathbf{I}$ or $\left(\begin{smallmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{smallmatrix}\right)$ often suffice, although for 3 images, long focal lengths or highly constrained motions they can sometimes lead to local minima.

Convergence is rapid (4–10 iterations) unless the problem is degenerate, and even then failure to converge to *some* feasible solution is rare. It is worth using a fairly accurate (*e.g.* nonlinear least squares) projective reconstruction, especially in the unstable 3 image case. Omitting the $\det(\boldsymbol{\Omega}) = 0$ constraint significantly reduces both accuracy and stability.

3.4 Quasi-Linear Approach

It is also possible to solve the quadric projection constraints using a "quasi-linear" approach. No initialization is required, but at least 4 images are needed and the method is slower, less stable and less accurate than SQP.

The basic idea is to write the independent components of $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$ as vectors and work with the $10 \times 6 = 60$ components of their outer product matrix. The absolute quadric projection constraints are linear and have rank 15 in these variables, so the matrix can be recovered linearly from $m \geq \lceil \frac{59}{15} \rceil = 4$ images. A 10×6 SVD projects the result to rank 1 and factorizes it into vectors $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$. Finally, $\boldsymbol{\Omega}$ (rewritten as a matrix) is projected to rank 3 by annulling its smallest eigenvalue, and the method proceeds with $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$ as above.

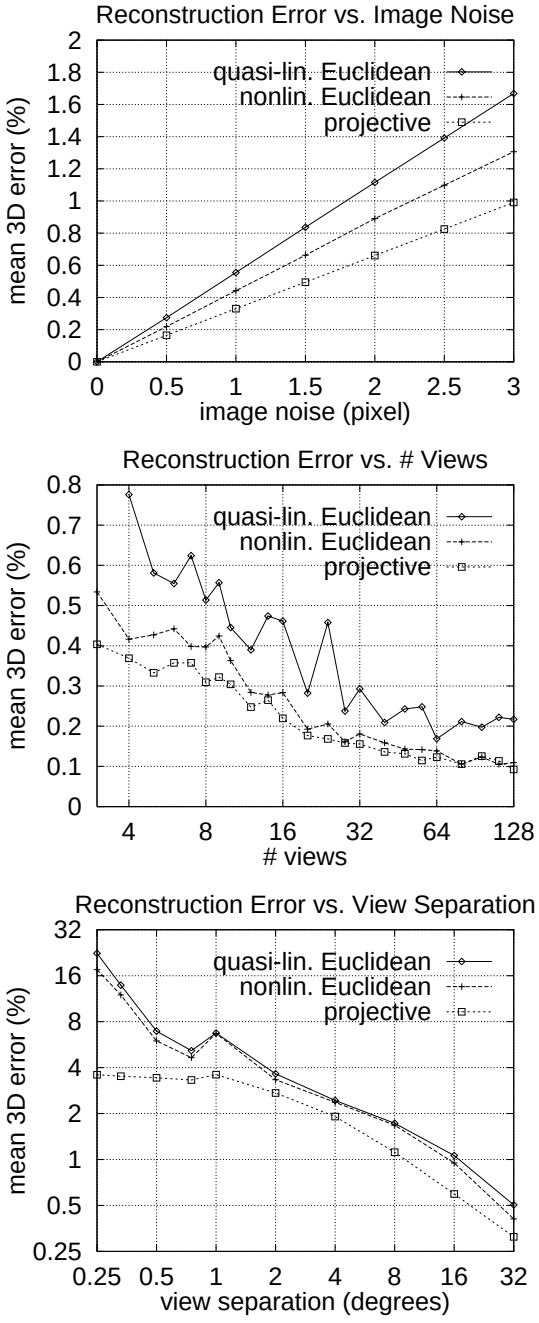


Figure 2: Mean 3D reconstruction error *vs.* image noise, number of images and angular spread of cameras for quasi-linear Euclidean, nonlinear Euclidean and projective reconstructions of point clouds.

Since it only enforces the rank 1 and $\det(\boldsymbol{\Omega}) = 0$ constraints indirectly, the quasi-linear method introduces degeneracies that are not intrinsic to the underlying problem. In particular, it fails whenever *any* point — even a finite one — is fixed in all im-

ages (*e.g.* a fixating camera).

4 Algorithm

The full algorithm for autocalibration and scaled Euclidean reconstruction is as follows:

1) Standardize all image coordinates.

2) Find the projections $\mathbf{P}_i$ by projective reconstruction.

3) Find the absolute quadric $\boldsymbol{\Omega}$ and image conic $\boldsymbol{\omega}$ by solving $15m$ bilinear quadric projection constraints $\boldsymbol{\omega} \wedge (\mathbf{P}_i \boldsymbol{\Omega} \mathbf{P}_i^\top) = \mathbf{0}$ (nonlinear and quasi-linear methods).

4) Recover the camera calibration $\boldsymbol{K}$ by Choleski decomposition of $\boldsymbol{\omega} = \boldsymbol{K}\boldsymbol{K}^\top$.

5) Find a 4×4 Euclideanizing homography $\mathbf{T}$ by eigen-decomposition of $\boldsymbol{\Omega}$.

6) Perturb $\boldsymbol{K}^{-1}\mathbf{P}_i\mathbf{T}^{-1} \sim \boldsymbol{R}_i(\boldsymbol{I} \mid -\boldsymbol{t}_i)$ to be exactly Euclidean.

7) Recover Euclidean structure by $\mathbf{x} \rightarrow \mathbf{T}\mathbf{x}$ or back-projecting with the corrected projections.

8) Optional bundle adjustment.

Standardization rescales image pixel coordinates to lie in the unit box $[-1, 1] \times [-1, 1]$. It is *absolutely indispensable*. Otherwise, different equations of $\|\boldsymbol{\omega} \wedge (\mathbf{P}_i \boldsymbol{\Omega} \mathbf{P}_i^\top)\|^2 \approx 0$ have a difference in scale of (say) $256^6 \approx 10^{14}$. Their numerical conditioning is terrible and severe floating point truncation error leads to further loss of precision. This is perhaps the major reason for the observed instability of some previous autocalibration approaches. Standardization ('preconditioning') is *essential* whenever there is an implicit least squares trade-off (as here), particularly with equations of high degree. It is discussed in every text on numerical methods, but does not seem to have been widely known in vision before Hartley made the point for fundamental matrix estimation [9].

5 Experiments

To give a rough idea of the performance of the algorithm, we briefly report on numerical experiments with synthetic data. Images of random point clouds were taken with identical wide-angle cameras placed randomly within a fixed cone of viewing angles, approximately on a sphere surrounding the scene. Several other configurations have also been tried with success. Uniform random noise was added to the image points. The initial projective reconstruction was projective factorization [18, 20] followed by projective bundle adjustment (not indispensable). The nonlinear method was initialized with a calibration wrong by about 50%. Mean 3D reconstruction error over 10 trials was estimated by projective least squares alignment for projective reconstructions and scaled Euclidean alignment for Euclidean ones. There was no final Euclidean bundle adjustment, although this is recommended for real applications. Default values were ± 1 pixel noise, 6 views, 50 points, with a wide ($\pm 30^\circ$) range of viewing directions and cyclotorsions.

Figure 2(a) shows that all errors scale linearly with noise, and that the un-adjusted nonlinear Euclidean reconstruction (with $3 + 3 + 1 = 7$ free parameters) is very nearly as good as the underlying projective one (with 15). Figure 2(b) suggests that this applies for any number of images, while the quasi-linear method is somewhat less stable. Figure 2(c) shows that the error scales smoothly as the viewing angles are decreased.

In an informal test on real images of a calibration grid, we compared un-bundle-adjusted autocalibration with the scatter of results from conventional calibration using known 3D point positions. It was within: 0.1% (0.3σ) on α_u and α_v; 0.01% (1.5σ) on α_u/α_v; and 5 pixels ($\sim$1–2σ) on u_0 and v_0 (the σ estimates here are very imprecise).

6 Discussion & Conclusions

We have described a new method for autocalibrating a moving camera with fixed but unknown intrinsic parameters, moving arbitrarily in an unknown scene. An initial projective reconstruction is rectified to give calibration and scaled Euclidean structure and motion. The method is based on a new projective encoding of metric structure: the **absolute quadric**. This is equivalent to the absolute conic, but considerably easier to use. It projects very simply to the dual absolute image conic which encodes camera calibration. The absolute quadric and conic are recovered simultaneously using an efficient constrained nonlinear optimization technique (**sequential quadratic programming**) or a quasi-linear method. The results are stable and accurate

for generic camera motions, and the formalism clarifies the reasons for autocalibration's intrinsic degeneracies. A major practical advantage of the nonlinear approach is the ease with which it incorporates any further constraints that may be available, potentially significantly reducing the problems of degeneracy.

Future work will examine several topics. In the one camera case, priorities are techniques to detect and handle degeneracy, and a study of the advantages of incorporating various additional constraints. Problems with several cameras (*i.e.* several $\boldsymbol{\omega}$'s) are easily handled, as are rigidly moving stereo heads ($\boldsymbol{\omega}$ is replaced by a 'local' $\boldsymbol{\Omega}$ in the head frame, invariant under motion induced 4×4 homographies). Non-reconstruction based autocalibration techniques that work whether or not the translations are zero would be useful. Finally, SQP is being successfully applied to several other constrained statistical fitting problems in vision.

References

[1] M. Armstrong, A. Zisserman, and R. Hartley. Self-calibration from image triplets. In B. Buxton and R. Cipolla, editors, *European Conf. Computer Vision*, pages 3–16, Cambridge, U.K., April 1996.

[2] J. Canny. A toolkit for nonlinear algebra. Report available from http://http.cs.berkeley.edu/∼jfc, August 1993.

[3] O. Faugeras. Stratification of 3-d vision: Projective, affine, and metric representations. *J. Optical Society of America*, A 12(3):465–84, March 1995.

[4] O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[5] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[6] P. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, 1981.

[7] R. Hartley. Euclidean reconstruction from multiple views. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.

[8] R. Hartley. Self-calibration from multiple views with a rotating camera. In *European Conf. Computer Vision*, pages 471–8. Springer-Verlag, 1994.

[9] R. Hartley. In defence of the 8-point algorithm. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1064–70, Cambridge, MA, June 1995.

[10] A. Heyden and K. Åström. Euclidean reconstruction from constant intrinsic parameters. In *Int. Conf. Pattern Recognition*, pages 339–43, Vienna, 1996.

[11] B. Horn. Relative orientation. *Int. J. Computer Vision*, 4:59–78, 1990.

[12] Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. Technical Report UCB/CSD-93-772, Dept. EECS, Berkeley, California, 1993.

[13] S. J. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. J. Computer Vision*, 8(2):123–151, 1992.

[14] R. Mohr and B. Triggs. Projective geometry for image analysis. Tutorial given at *Int. Symp. Photogrammetry and Remote Sensing*, July 1996.

[15] M. Pollefeys, L Van Gool, and M. Proesmans. Euclidean 3d reconstruction from image sequences with variable focal length. In B. Buxton and R. Cipolla, editors, *European Conf. Computer Vision*, pages 31–42, Cambridge, U.K., April 1996.

[16] P. Sturm. Self-calibration of a moving camera by pre-calibration. In *British Machine Vision Conference*, pages 675–84, Edinburgh, September 1996. British Machine Vision Association.

[17] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[18] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–20, Cambridge, U.K., 1996. Springer-Verlag.

[19] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to *Int. J. Computer Vision*.

[20] B. Triggs. Factorization methods for projective structure and motion. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 845–51, San Francisco, CA, 1996.

[21] C. Zeller and O. Faugeras. Camera self-calibration from video sequences: the Kruppa equations revisited. Technical Report 2793, INRIA, INRIA Sophia-Antipolis, France, 1996.

[22] Z. Zhang, Q.-T. Luong, and O. Faugeras. Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. Technical Report 2079, INRIA, Sophia-Antipolis, France, 1993.

Autocalibration from Planar Scenes

Bill Triggs
INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot St. Martin, France.
Bill.Triggs@inrialpes.fr ◇ *http://www.inrialpes.fr/movi/people/Triggs*

Abstract

This paper describes a theory and a practical algorithm for the autocalibration of a moving projective camera, from $m \geq 5$ views of a *planar* scene. The unknown camera calibration, and (up to scale) the unknown scene geometry and camera motion are recovered from the hypothesis that the camera's internal parameters remain constant during the motion. This work extends the various existing methods for non-planar autocalibration to a practically common situation in which it is not possible to bootstrap the calibration from an intermediate projective reconstruction. It also extends Hartley's method for the internal calibration of a rotating camera, to allow camera translation and to provide 3D as well as calibration information. The basic constraint is that the projections of orthogonal direction vectors (points at infinity) in the plane must be orthogonal in the calibrated camera frame of each image. Abstractly, since the two circular points of the 3D plane (representing its Euclidean structure) lie on the 3D absolute conic, their projections into each image must lie on the absolute conic's image (representing the camera calibration). The resulting numerical algorithm optimizes this constraint over all circular points and projective calibration parameters, using the inter-image homographies as a projective scene representation.

Keywords: Autocalibration, Euclidean structure, Absolute Conic & Quadric, Planar Scenes.

1 Introduction

This paper describes a method of autocalibrating a moving projective camera with general, unknown motion and unknown intrinsic parameters, from $m \geq 5$ views of a *planar* scene. Autocalibration is the recovery of metric information — for example the internal and external calibration parameters of a moving projective camera — from non-metric information and (metric) self-consistency constraints — for example the knowledge that the camera's internal parameters are constant during the motion, and the inter-image consistency constraints that this entails. Since the seminal work of Maybank & Faugeras [14, 3], a number of different approaches to autocalibration have been developed [5, 6, 1, 27, 26, 2, 13, 9, 16, 15, 21, 10]. For the 'classical' problem of a single perspective camera with constant but unknown internal parameters moving with a general but unknown motion in a 3D scene, the original Kruppa equation based approach [14] seems to be being displaced by approaches based on the 'rectification' of an intermediate projective reconstruction [5, 9, 15, 21, 10]. More specialized methods exist for particular types of motion and simplified calibration models [6, 24, 1, 16]. Stereo heads can also be autocalibrated [27, 11]. Solutions are still — in theory — possible if some of the intrinsic parameters are allowed to vary [9, 15]. Hartley [6] has given a particularly simple internal calibration method for the case of a single camera whose translation is known to be negligible compared to the distances of some identifiable (real or synthetic) points

This revised version of my ECCV'98 paper [23] contains an additional paragraph on the Kruppa instability and an appendix describing an unused (but potentially useful) factorization method for homographies between $m \geq 2$ images. The work was supported by Esprit LTR project CUMULI. I would like to thank P. Sturm and the reviewers for comments, G. Csurka and A. Ruf for the test data, and C. Gramkow for pointing out some missing constants in eqns. (11) and (12).

in the scene, and Faugeras [2] has elaborated a 'stratification' paradigm for autocalibration based on this. The numerical conditioning of classical autocalibration is historically delicate, although recent algorithms have improved the situation significantly [9, 15, 21]. The main problem is that classical autocalibration has some restrictive intrinsic degeneracies — classes of motion for which no algorithm can recover a full unique solution. Sturm [18, 19] has given a catalogue of these. In particular, at least 3 views, some translation and some rotation about at least two non-aligned axes are required.

Planar Autocalibration: All of the existing approaches to classical autocalibration rely on information equivalent to a 3D projective reconstruction of the scene. In the Kruppa approach this is the fundamental matrices and epipoles, while for most other methods it is an explicit 3D reconstruction. For some applications (especially in man-made environments) this is potentially a problem, because planar or near-planar scenes sometimes occur for which stable 3D projective reconstructions (or fundamental matrices, *etc.*) can not be calculated. This well-known failing of projective reconstruction is something of an embarrassment: the *calibrated* reconstruction of planar scenes is not difficult, so it is exactly in this case when autocalibration fails that it would be most useful. The current paper aims to rectify this by providing autocalibration methods that work in the planar case, by 'rectifying' the inter-image homographies induced by the plane. In the longer term, we would like to find ways around the ill-conditioning of projective reconstruction for near-planar scenes, and also to develop 'structure-free' internal calibration methods similar to Hartley's zero-translation one [6], but which work for non-zero translations. The hope is that planar methods may offer one way to attack these problems.

Planar autocalibration has other potential advantages. Planes are very common in man-made environments, and often easily identifiable and rather accurately planar. They are simple to process and allow very reliable and precise feature-based or intensity-based matching, by fitting the homographies between image pairs. They are also naturally well adapted to the calibration of lens distortion as some of the subtleties of 3D geometry are avoided[1].

The main *dis*advantage of planar autocalibration (besides the need for a nice, flat, textured plane) seems to be the number of images required. Generically, $m \geq \lceil \frac{n+4}{2} \rceil$ images are needed for an internal camera model with n free parameters, *e.g.* $m \geq 5$ for the classical 5 parameter projective model (focal length, aspect ratio, skew, principal point), or $m \geq 3$ if only focal length is estimated. However for good accuracy and reliability, at least 8–10 images are recommended in practice. Also, almost any attempt at algebraic elimination across so many images rapidly leads to a combinatorial explosion. Hence, the approach is resolutely numerical, and it seems impracticable to initialize the optimization from a minimal algebraic solution. Although for the most part the numerical domain of convergence seems to be sufficient to allow moderately reliable convergence from a fixed default initialization, and we have also developed a numerical initialization search which may be useful in some cases, occasional convergence to false minima remains a problem.

Organization: Section 2 gives a direction-vector based formulation of the theory of autocalibration, and discusses how both non-planar and planar autocalibration can be approached within this framework. Section 3 describes the statistically-motivated cost function we optimize. Section 4 discusses the numerical algorithm, and the method used to initialize it. Section 5 gives experimental results on synthetic and real images, and section 6 concludes the paper.

Notation will be introduced as required. Briefly we use bold upright $\mathbf{x}$ for homogeneous 3D (4 component) vectors and matrices; bold italic $\boldsymbol{x}$ for 3 component ones (homogeneous image, inhomogeneous 3D, 3-component parts of homogeneous 4-component objects); $\boldsymbol{P}$ for image projections

[1] We will ignore lens distortion throughout this paper. If necessary it can be corrected by a nominal model, or — at least in theory — estimated up to an overall 3×3 projectivity by a bundled adjustment over all the inter-image homographies. (The pixel-pixel mapping induced by geometric homography $\boldsymbol{H}_i$ is $\boldsymbol{D}\boldsymbol{H}_i\boldsymbol{D}^{-1}$ where $\boldsymbol{D}$ is the distortion model).

and $\boldsymbol{H}$ for inter-image homographies; $\boldsymbol{K}$, $\boldsymbol{C} = \boldsymbol{K}^{-1}$ for upper triangular camera calibration and inverse calibration matrices; $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}^*$ for the absolute (hyperplane) quadric and (direction) conic; and $\boldsymbol{\omega} = \boldsymbol{K}\boldsymbol{K}^\top = \boldsymbol{P}\boldsymbol{\Omega}\boldsymbol{P}^\top$ and $\boldsymbol{\omega}^{-1} = \boldsymbol{C}^\top\boldsymbol{C}$ for their images. $[\,\cdot\,]_\times$ denotes the matrix generating the cross product: $[\boldsymbol{x}]_\times\boldsymbol{y} = \boldsymbol{x} \wedge \boldsymbol{y}$.

2 Euclidean Structure and Autocalibration

To recover the metric information implicit in projective images, we need a projective encoding of Euclidean structure. The key to Euclidean structure is the dot product between direction vectors ("points at infinity"), or dually the dot product between (normals to) hyperplanes. The former leads to the stratified "hyperplane at infinity + absolute (direction) conic" formulation (affine + metric structure) [17], the latter to the "absolute (hyperplane) quadric" one [21]. These are just dual ways of saying the same thing. The hyperplane formalism is preferable for 'pure' autocalibration where there is no *a priori* decomposition into affine and metric strata, while the point one is simpler if such a stratification is given.

Generalities: Consider k-dimensional Euclidean space. We will need the cases $k = 2$ (the planar scene and its 2D images) and $k = 3$ (ordinary 3D space). Introducing homogeneous Euclidean coordinates, points, displacement vectors and hyperplanes are encoded respectively as homogeneous $k+1$ component column vectors $\mathbf{x} = (\boldsymbol{x}, 1)^\top$, $\mathbf{t} = (\boldsymbol{t}, 0)^\top$ and row vectors $\mathbf{p} = (\boldsymbol{n}, d)$. Here $\boldsymbol{x}$, $\boldsymbol{t}$ and $\boldsymbol{n}$ are the usual k-D coordinate vectors of the point, the displacement, and the hyperplane normal, and d is the hyperplane offset. Points and displacements on the plane satisfy respectively $\mathbf{p}\cdot\mathbf{x} = \boldsymbol{n}\cdot\boldsymbol{x} + d = 0$ and $\mathbf{p}\cdot\mathbf{t} = \boldsymbol{n}\cdot\boldsymbol{t} = 0$. Displacement directions can be appended to the point space, as a **hyperplane at infinity** $\mathbf{p}_\infty$ of **points at infinity** or **vanishing points**. Projectively, $\mathbf{p}_\infty$ behaves much like any other hyperplane. In Euclidean coordinates, $\mathbf{p}_\infty = (\boldsymbol{0}, 1)$ so that $\mathbf{p}_\infty\cdot\mathbf{t} = 0$ for any displacement $\mathbf{t} = (\boldsymbol{t}, 0)$. **Projective transformations** mix finite and infinite points. Under a projective transformation encoded by an arbitrary nonsingular $(k+1)\times(k+1)$ matrix $\mathbf{T}$, points and directions (column vectors) transform **contravariantly**, *i.e.* by $\mathbf{T}$ acting on the left: $\mathbf{x} \to \mathbf{T}\,\mathbf{x}$, $\mathbf{v} \to \mathbf{T}\,\mathbf{v}$. To preserve the point-on-plane relation $\mathbf{p}\cdot\mathbf{x} = \boldsymbol{n}\cdot\boldsymbol{x} + d = 0$, hyperplanes (row vectors) transform **covariantly**, *i.e.* by $\mathbf{T}^{-1}$ acting on the right: $\mathbf{p} \to \mathbf{p}\,\mathbf{T}^{-1}$.

Absolute Quadric & Conic: The usual Euclidean dot product between hyperplane normals is $\boldsymbol{n}_1\cdot\boldsymbol{n}_2 = \mathbf{p}_1\,\boldsymbol{\Omega}\,\mathbf{p}_2^\top$ where the symmetric, rank k, positive semidefinite matrix

$$\boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{I}_{k\times k} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix}$$

is called the **absolute (hyperplane) quadric**[2]. $\boldsymbol{\Omega}$ encodes the Euclidean structure in projective coordinates. Under projective transformations it transforms contravariantly (*i.e.* like a point) in each of its two indices so that the dot product between plane normals is invariant: $\boldsymbol{\Omega} \to \mathbf{T}\,\boldsymbol{\Omega}\,\mathbf{T}^\top$ and $\mathbf{p}_i \to \mathbf{p}_i\,\mathbf{T}^{-1}$, so $\mathbf{p}_1\,\boldsymbol{\Omega}\,\mathbf{p}_2^\top = \boldsymbol{n}_1\cdot\boldsymbol{n}_2$ is constant. $\boldsymbol{\Omega}$ is invariant under Euclidean transformations, but in a general projective frame it loses its diagonal form and becomes an arbitrary symmetric positive semidefinite rank k matrix. In any frame, the Euclidean angle between two hyperplanes is $\cos\theta = (\mathbf{p}\,\boldsymbol{\Omega}\,\mathbf{p}'^\top)/\sqrt{(\mathbf{p}\,\boldsymbol{\Omega}\,\mathbf{p}^\top)(\mathbf{p}'\,\boldsymbol{\Omega}\,\mathbf{p}'^\top)}$, and the plane at infinity is $\boldsymbol{\Omega}$'s unique null vector: $\mathbf{p}_\infty\,\boldsymbol{\Omega} = \mathbf{0}$. When restricted to coordinates on $\mathbf{p}_\infty$, $\boldsymbol{\Omega}$ becomes nonsingular and can be dualized (inverted) to give the $k\times k$ symmetric positive definite **absolute (direction) conic** $\boldsymbol{\Omega}^*$. This measures dot products between displacement vectors, just as $\boldsymbol{\Omega}$ measures them between hyperplane normals. $\boldsymbol{\Omega}^*$ is defined *only* on direction vectors, not on finite points, and unlike $\boldsymbol{\Omega}$ it has no unique canonical

[2] Abstractly, $\boldsymbol{\Omega}$ can be viewed as a cone (degenerate quadric hypersurface) with no real points in complex projective hyperplane space. But it is usually simpler just to think of it concretely as a symmetric matrix with certain properties.

form in terms of the *un*restricted coordinates. (Anything of the form $\begin{pmatrix} \boldsymbol{I} & \boldsymbol{x} \\ \boldsymbol{x}^\top & y \end{pmatrix}$ can be used, for arbitrary $\boldsymbol{x}, y$).

Direction bases: In Euclidean coordinates, $\boldsymbol{\Omega}$ can be decomposed as a sum of outer products of any orthonormal (in terms of $\boldsymbol{\Omega}^*$) basis of displacement vectors: $\boldsymbol{\Omega} = \sum_{i=1}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\top$ where $\boldsymbol{x}_i \boldsymbol{\Omega}^* \boldsymbol{x}_j = \delta_{ij}$. For example in 2D $\boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{I}_{2\times 2} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix} = \hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top + \hat{\boldsymbol{y}}\hat{\boldsymbol{y}}^\top$ where $\hat{\boldsymbol{x}} = (1,0,0), \hat{\boldsymbol{y}} = (0,1,0)$, are the usual unit direction vectors. Gathering the basis vectors into the columns of a $(k+1) \times k$ orthonormal rank k matrix $\mathbf{U}$ we have $\boldsymbol{\Omega} = \mathbf{U}\,\mathbf{U}^\top$, $\mathbf{p}_\infty \mathbf{U} = \mathbf{0}$ and $\mathbf{U}^\top \boldsymbol{\Omega}^* \mathbf{U} = \boldsymbol{I}_{k\times k}$. The columns of $\mathbf{U}$ span $\mathbf{p}_\infty$. All of these relations remain valid in an arbitrary projective frame $\mathbf{T}$ and with an arbitrary choice of representative for $\boldsymbol{\Omega}^*$, except that $\mathbf{U} \to \mathbf{T}\,\mathbf{U}$ ceases to be orthonormal.

$\mathbf{U}$ is defined only up to an arbitrary $k \times k$ orthogonal mixing of its columns (redefinition of the direction basis) $\mathbf{U} \to \mathbf{U}\,\boldsymbol{R}_{k\times k}$. Even in a projective frame where $\mathbf{U}$ itself is not orthonormal, this mixing freedom remains orthogonal. In a Euclidean frame $\mathbf{U} = \begin{pmatrix} \boldsymbol{V} \\ 0 \end{pmatrix}$ for some $k \times k$ rotation matrix $\boldsymbol{V}$, so the effect of a Euclidean space transformation is $\mathbf{U} \to \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix} \mathbf{U} = \mathbf{U}\,\boldsymbol{R}'$ where $\boldsymbol{R}' = \boldsymbol{V}^\top \boldsymbol{R} \boldsymbol{V}$ is the conjugate rotation: Euclidean transformations of direction bases (*i.e.* on the left) are equivalent to orthogonal re-mixings of them (*i.e.* on the right). This remains true in an arbitrary projective frame, even though $\mathbf{U}$ and the transformation no longer *look* Euclidean. This mixing freedom can be used to choose a direction basis in which $\mathbf{U}$ is orthonormal up to a diagonal rescaling: simply take the SVD $\mathbf{U}' \boldsymbol{D} \boldsymbol{V}^\top$ of $\mathbf{U}$ and discard the mixing rotation $\boldsymbol{V}^\top$. Equivalently, the eigenvectors and square roots of eigenvalues of $\boldsymbol{\Omega}$ can be used. Such orthogonal parametrizations of $\mathbf{U}$ make good numerical sense, and we will use them below.

Circular points: Given any two orthonormal direction vectors $\mathbf{x}, \mathbf{y}$, the complex conjugate vectors $\mathbf{x}_\pm \equiv \frac{1}{\sqrt{2}}(\mathbf{x} \pm i\mathbf{y})$ satisfy $\mathbf{x}_\pm \boldsymbol{\Omega}^* \mathbf{x}_\pm^\top = 0$. Abstractly, these complex directions "lie on the absolute conic", and it is easy to check that any complex projective point which does so can be decomposed into two orthogonal direction vectors, its real and imaginary parts. In the 2D case there is only one such conjugate pair up to complex phase, and these **circular points** characterize the Euclidean structure of the plane. However for numerical purposes, it is usually easier to avoid complex numbers by using the real and imaginary parts $\mathbf{x}$ and $\mathbf{y}$ rather than $\mathbf{x}_\pm$. The phase freedom in $\mathbf{x}_\pm$ corresponds to the 2×2 orthogonal mixing freedom of $\mathbf{x}$ and $\mathbf{y}$.

Theoretically, the above parametrizations of Euclidean structure are equivalent. Which is practically best depends on the problem. $\boldsymbol{\Omega}$ is easy to use, except that constrained optimization is required to handle the rank k constraint $\det \boldsymbol{\Omega} = 0$. Direction bases $\mathbf{U}$ eliminate this constraint at the cost of numerical code to handle their $k \times k$ orthogonal gauge freedom. The absolute conic $\boldsymbol{\Omega}^*$ has neither constraint nor gauge freedom, but has significantly more complicated image projection properties and can only be defined once the plane at infinity $\mathbf{p}_\infty$ is known and a projective coordinate system on it has been chosen (*e.g.* by induction from one of the images). It is also possible to parametrize Euclidean structure by non-orthogonal Choleski-like decompositions $\boldsymbol{\Omega} = \mathbf{L}\,\mathbf{L}^\top$ (*i.e.* the L part of the LQ decomposition of $\mathbf{U}$), but this introduces singularities at maximally non-Euclidean frames unless pivoting is also used.

Image Projections: Since the columns of a 3D direction basis matrix $\mathbf{U}$ are *bona fide* 3D direction vectors, its image projection is simply $\boldsymbol{P}\,\mathbf{U}$, where $\boldsymbol{P}$ is the usual 3×4 point projection matrix. Hence, the projection of $\boldsymbol{\Omega} = \mathbf{U}\,\mathbf{U}^\top$ is the 3×3 symmetric positive definite contravariant image matrix $\boldsymbol{\omega} = \boldsymbol{P}\,\boldsymbol{\Omega}\,\boldsymbol{P}^\top$. Abstractly, this is the image line quadric dual to the image of the absolute conic. Concretely, given any two image lines $\boldsymbol{l}_1, \boldsymbol{l}_2$, $\boldsymbol{\omega}$ encodes the 3D dot product between their 3D visual planes $\mathbf{p}_i = \boldsymbol{l}_i \boldsymbol{P}$: $\mathbf{p}_1\,\boldsymbol{\Omega}\,\mathbf{p}_2^\top = \boldsymbol{l}_1\,\boldsymbol{P}\,\boldsymbol{\Omega}\,\boldsymbol{P}^\top \boldsymbol{l}_2^\top = \boldsymbol{l}_1\,\boldsymbol{\omega}\,\boldsymbol{l}_2^\top$. With the traditional Euclidean decomposition $\boldsymbol{K}\boldsymbol{R}\,(\boldsymbol{I} \,|\, -\boldsymbol{t})$ of $\boldsymbol{P}$ into an upper triangular **internal calibration matrix** $\boldsymbol{K}$, a 3×3 **camera orientation** (rotation) $\boldsymbol{R}$ and an **optical centre** $\boldsymbol{t}$, $\boldsymbol{\omega}$ becomes simply $\boldsymbol{K}\boldsymbol{K}^\top$. Since $\boldsymbol{\Omega}$ is invariant under Euclidean motions, $\boldsymbol{\omega}$ is invariant under camera displacements so long as $\boldsymbol{K}$ remains

constant. $\boldsymbol{K}$ can be recovered from $\boldsymbol{\omega}$ by Choleski decomposition, and similarly the Euclidean scene structure (in the form of a 'rectifying' projective transformation) can be recovered from $\boldsymbol{\Omega}$. The upper triangular **inverse calibration matrix** $\boldsymbol{C} = \boldsymbol{K}^{-1}$ converts homogeneous pixel coordinates to optical ray directions in the Euclidean camera frame. $\boldsymbol{\omega}^{-1} = \boldsymbol{C}^\top \boldsymbol{C}$ is the image of the absolute conic.

Autocalibration: Given several images taken with projection matrices $\boldsymbol{P}_i = \boldsymbol{K}_i \boldsymbol{R}_i (\boldsymbol{I} | -\boldsymbol{t}_i)$, and (in the same Euclidean frame) a orthogonal direction basis $\mathbf{U} = \left(\begin{smallmatrix} \boldsymbol{V} \\ \boldsymbol{0} \end{smallmatrix}\right)$, we find that

$$\boldsymbol{C}_i \boldsymbol{P}_i \mathbf{U} \;=\; \boldsymbol{R}'_i \tag{1}$$

where $\boldsymbol{C}_i = \boldsymbol{K}_i^{-1}$ and $\boldsymbol{R}'_i = \boldsymbol{R}_i \boldsymbol{V}$ is a rotation matrix depending on the camera pose. This is perhaps the most basic form of the autocalibration constraint. It says that the calibrated images (*i.e.* 3D directions in the camera frame) of an orthogonal direction basis must remain orthogonal. It remains true in arbitrary projective 3D and image frames, as the projective deformations of $\mathbf{U}$ *vs.* $\boldsymbol{P}_i$ and $\boldsymbol{P}_i$ *vs.* $\boldsymbol{C}_i$ cancel each other out. However, it is not usually possible to choose the scale factors of projectively reconstructed projections *a priori*, in a manner consistent with those of their unknown Euclidean parents. So in practice this constraint can only be applied up to an unknown scale factor for each image: $\boldsymbol{C}_i \boldsymbol{P}_i \mathbf{U} \;\sim\; \boldsymbol{R}'_i$. As always, the direction basis $\mathbf{U}$ is defined only up to an arbitrary 3×3 orthogonal mixing $\mathbf{U} \rightarrow \mathbf{U}\boldsymbol{R}$.

2.1 Autocalibration for Non-Planar Scenes

The simplest approaches to autocalibration for non-planar scenes are based on the consistency equation (1), an intermediate projective reconstruction $\boldsymbol{P}_i$, and some sort of knowledge about the $\boldsymbol{C}_i$ (*e.g.* classically that they are all the same: $\boldsymbol{C}_i = \boldsymbol{C}$ for some unknown $\boldsymbol{C}$). Nonlinear optimization or algebraic elimination are used to estimate the Euclidean structure $\boldsymbol{\Omega}$ or $\mathbf{U}$, and the free parameters of the $\boldsymbol{C}_i$. Multiplying (1) either on the left or on the right by its transpose to eliminate the unknown rotation, and optionally moving the $\boldsymbol{C}$'s to the right hand side, gives several equivalent symmetric 3×3 constraints linking $\boldsymbol{\Omega}$ or $\mathbf{U}$ to $\boldsymbol{\omega}_i$, $\boldsymbol{K}_i$ or $\boldsymbol{C}_i$

$$\mathbf{U}^\top \boldsymbol{P}_i^\top \boldsymbol{\omega}_i^{-1} \boldsymbol{P}_i \mathbf{U} \;\sim\; \boldsymbol{I}_{3\times 3} \tag{2}$$

$$\boldsymbol{C}_i \boldsymbol{P}_i \boldsymbol{\Omega} \boldsymbol{P}_i^\top \boldsymbol{C}_i^\top \;\sim\; \boldsymbol{I}_{3\times 3} \tag{3}$$

$$\boldsymbol{P}_i \boldsymbol{\Omega} \boldsymbol{P}_i^\top \;\sim\; \boldsymbol{\omega}_i \;=\; \boldsymbol{K}_i \boldsymbol{K}_i^\top \tag{4}$$

In each case there are 5 independent constraints per image on the 8 non-Euclidean d.o.f. of the 3D projective structure[3] and the 5 (or fewer) d.o.f. of the internal calibration. For example, three images in general position suffice for classical constant-$\boldsymbol{C}$ autocalibration. In each case, the unknown scale factors can be eliminated by treating the symmetric 3×3 left and right hand side matrices as $3 \cdot 4/2 = 6$ component vectors, and either (*i*) projecting (say) the left hand sides orthogonally to the right hand ones (hence deleting the proportional components and focusing on the constraint-

[3] These can be counted as follows: 15 for a 3D projective transformation modulo 7 for a scaled Euclidean one; or 12 for a 4×3 $\mathbf{U}$ matrix modulo 1 scale and 3 d.o.f. for a 3×3 orthogonal mixing; or $4 \cdot 5/2 = 10$ d.o.f. for a 4×4 symmetric quadric matrix $\boldsymbol{\Omega}$ modulo 1 scale and 1 d.o.f. for the rank 3 constraint $\det \boldsymbol{\Omega} = 0$; or 3 d.o.f. for $\mathbf{p}_\infty$ and 5 for the $3 \cdot 4/2 = 6$ components of $\boldsymbol{\Omega}^*$ modulo 1 scale.

violating non-proportional ones), or (*ii*) cross-multiplying in the usual way:

$$\begin{array}{c} \mathbf{u}_i \cdot \mathbf{v}_i = \mathbf{u}_i \cdot \mathbf{w}_i = \mathbf{v}_i \cdot \mathbf{w}_i = 0 \\ \|\mathbf{u}_i\|^2 = \|\mathbf{v}_i\|^2 = \|\mathbf{w}_i\|^2 \end{array} \quad \text{where} \quad (\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i) \equiv \boldsymbol{C}_i \boldsymbol{P}_i \mathbf{U} \tag{5}$$

$$\begin{array}{c} (\boldsymbol{C}_i \boldsymbol{P}_i \boldsymbol{\Omega} \boldsymbol{P}_i^\top \boldsymbol{C}_i^\top)^{AA} = (\boldsymbol{C}_i \boldsymbol{P}_i \boldsymbol{\Omega} \boldsymbol{P}_i^\top \boldsymbol{C}_i^\top)^{BB} \\ (\boldsymbol{C}_i \boldsymbol{P}_i \boldsymbol{\Omega} \boldsymbol{P}_i^\top \boldsymbol{C}_i^\top)^{AB} = 0 \end{array} \quad \text{where} \quad A < B = 1 \ldots 3 \tag{6}$$

$$(\boldsymbol{P}_i \boldsymbol{\Omega} \boldsymbol{P}_i^\top)^{AB} \; (\boldsymbol{\omega})^{CD} = (\boldsymbol{\omega})^{AB} \; (\boldsymbol{P}_i \boldsymbol{\Omega} \boldsymbol{P}_i^\top)^{CD} \quad \text{where} \quad A \leq B, C \leq D = 1 \ldots 3 \tag{7}$$

Several recent autocalibration methods for 3D scenes (*e.g.* [21, 9]) are based implicitly on these constraints, parametrized by $\boldsymbol{K}$ or $\boldsymbol{\omega}$ and by something equivalent[4] to $\boldsymbol{\Omega}$ or $\mathbf{U}$. All of these methods seem to work well provided the intrinsic degeneracies of the autocalibration problem [18] are avoided.

In contrast, methods based on the Kruppa equations [14, 3, 26] can not be recommended for general use, because they add a serious additional singularity to the already-restrictive ones intrinsic to the problem. If any 3D point projects to the same pixel and is viewed from the same distance in each image, a 'zoom' parameter can not be recovered from the Kruppa equations. In particular, for a camera moving around an origin and fixating it at the image centre, the focal length can not be recovered[5]. Sturm [19] gives a geometric argument for this, but it is also easy to see algebraically. Let $\boldsymbol{x}$ be the fixed image point, $\boldsymbol{F}$ the fundamental matrix between images 1 and 2, $\boldsymbol{e}$ the epipole of image 2 in image 1, and $\boldsymbol{\omega}$ the constant dual absolute image quadric. Choosing appropriate scale factors for $\boldsymbol{e}$ and $\boldsymbol{F}$, the Kruppa constraint can be written as $\boldsymbol{F}\boldsymbol{\omega}\boldsymbol{F}^\top = [\boldsymbol{e}]_\times \boldsymbol{\omega} [\boldsymbol{e}]_\times^\top$. Since $\boldsymbol{x}$ is fixed, $\boldsymbol{x}^\top \boldsymbol{F} \boldsymbol{x} = 0$ and by the projective depth recovery relations [20] $\boldsymbol{F}\boldsymbol{x} = \lambda [\boldsymbol{e}]_\times \boldsymbol{x}$ where λ is the relative projective depth (projective scale factor) of $\boldsymbol{x}$ in the two images. Hence $\boldsymbol{F}(\boldsymbol{\omega} + \mu \boldsymbol{x}\boldsymbol{x}^\top)\boldsymbol{F}^\top = [\boldsymbol{e}]_\times(\boldsymbol{\omega} + \mu\lambda \boldsymbol{x}\boldsymbol{x}^\top)[\boldsymbol{e}]_\times^\top$. With these normalizations of $\boldsymbol{e}$ and $\boldsymbol{F}$, $\lambda = 1$ iff the *Euclidean* depth of $\boldsymbol{x}$ is the same in each image. If this is the case for all of the images we see that if $\boldsymbol{\omega}$ is a solution of the Kruppa equations, so is $\boldsymbol{\omega} + \mu \boldsymbol{x}\boldsymbol{x}^\top$ for any μ. This means that the calibration can only be recovered up to a zoom centred on $\boldsymbol{x}$. Numerical experience suggests that Kruppa-based autocalibration remains ill-conditioned even quite far from this singularity. This is hardly surprising given that in any case the distinction between zooms and closes depends on fairly subtle 2^{nd}-order perspective effects, so that the recovery of focal lengths is never simple. (Conversely, the effects of an inaccurate zoom-close calibration on image measurements or local object-centred 3D ones are relatively minor).

2.2 Autocalibration from Planar Scenes

Now consider autocalibration from *planar* scenes. Everything above remains valid, except that no intermediate 3D projective reconstruction is available from which to bootstrap the process. However we will see that by using the inter-image homographies, autocalibration is still possible.

The Euclidean structure of the scene plane is given by any one of (*i*) a 3×3 rank 2 absolute line quadric $\boldsymbol{Q}$; (*ii*) a 3 component line at infinity $\boldsymbol{l}_\infty$ and its associated 2×2 absolute (direction) conic matrix; (*iii*) a 3×2 direction basis matrix $\boldsymbol{U} = (\boldsymbol{x}\,\boldsymbol{y})$; (*iv*) two complex conjugate circular points $\boldsymbol{x}_\pm = \frac{1}{\sqrt{2}}(\boldsymbol{x} \pm i\boldsymbol{y})$ which are also the two roots of the absolute conic on $\boldsymbol{l}_\infty$ and the factors of the

[4] If the first camera projection is taken to be $(\boldsymbol{I}|\boldsymbol{0})$ [5, 9], $\mathbf{U}$ can be chosen to have the form $\begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{p}^\top \end{pmatrix} \boldsymbol{K}$ where $\mathbf{p}_\infty \sim (\boldsymbol{p}^\top, 1)$, whence $\boldsymbol{\Omega} \sim \begin{pmatrix} \boldsymbol{\omega} & -\boldsymbol{\omega}\boldsymbol{p} \\ -\boldsymbol{p}^\top\boldsymbol{\omega} & \boldsymbol{p}^\top\boldsymbol{\omega}\boldsymbol{p} \end{pmatrix}$ and $\begin{pmatrix} \boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{p}^\top & 1 \end{pmatrix}$ is a Euclideanizing projectivity.

[5] For most other autocalibration methods, this case is ambiguous only if the fixed point is at infinity (rotation about a fixed axis + arbitrary translation).

absolute line quadric $\boldsymbol{Q} = \boldsymbol{x}\,\boldsymbol{x}^\top + \boldsymbol{y}\,\boldsymbol{y}^\top = \boldsymbol{x}_+\boldsymbol{x}_-^\top + \boldsymbol{x}_-\boldsymbol{x}_+^\top$. In each case the structure is the natural restriction of the corresponding 3D one, re-expressed in the planar coordinate system. In each case it projects isomorphically into each image, either by the usual 3×4 3D projection matrix (using 3D coordinates), or by the corresponding 3×3 world-plane to image homography $\boldsymbol{H}$ (using scene plane coordinates). Hence, each image inherits a pair of circular points $\boldsymbol{H}_i\,\boldsymbol{x}_\pm$ and the corresponding direction basis $\boldsymbol{H}_i\,(\boldsymbol{x}\,\boldsymbol{y})$, line at infinity $\boldsymbol{l}_\infty\,\boldsymbol{H}_i^{-1}$ and 3×3 rank 2 absolute line quadric $\boldsymbol{H}_i\,\boldsymbol{Q}\,\boldsymbol{H}_i^\top$. As the columns of the planar $\boldsymbol{U}$ matrix represent *bona fide* 3D direction vectors (albeit expressed in the planar coordinate system), their images still satisfy the autocalibration constraints (1):

$$\boldsymbol{C}_i\,\boldsymbol{H}_i\,\boldsymbol{U} \sim \boldsymbol{R}_{3\times 2} \tag{8}$$

where $\boldsymbol{R}_{3\times 2}$ contains the first two columns of a 3×3 rotation matrix. Multiplying on the left by the transpose to eliminate the unknown rotation coefficients gives (*c.f.* (2)):

$$\boldsymbol{U}^\top\,\boldsymbol{H}_i^\top\,\boldsymbol{\omega}_i^{-1}\,\boldsymbol{H}_i\,\boldsymbol{U} \;\sim\; \boldsymbol{I}_{2\times 2} \tag{9}$$

Splitting this into components gives the form of the constraints used by our planar autocalibration algorithm:

$$\|\boldsymbol{u}_i\|^2 = \|\boldsymbol{v}_i\|^2, \quad 2\,\boldsymbol{u}_i \cdot \boldsymbol{v}_i = 0 \qquad \text{where} \qquad (\boldsymbol{u}_i, \boldsymbol{v}_i) \;\equiv\; \boldsymbol{C}_i\,\boldsymbol{H}_i\,(\boldsymbol{x},\boldsymbol{y}) \tag{10}$$

These constraints say that any two orthonormal direction vectors in the world plane project under the calibrated world-plane to image homography $\boldsymbol{C}_i\,\boldsymbol{H}_i$ to two orthonormal vectors in the camera frame. Equivalently, the (calibrated) images of the circular points $\boldsymbol{x}_\pm = \frac{1}{\sqrt{2}}(\boldsymbol{x} \pm i\boldsymbol{y})$ lie on the image of the (calibrated) absolute conic:

$$(\boldsymbol{H}_i\,\boldsymbol{x}_\pm)^\top\,\boldsymbol{\omega}^{-1}\,(\boldsymbol{H}_i\,\boldsymbol{x}_\pm) \;=\; \|\boldsymbol{u}_{i\pm}\|^2 \;=\; 0 \qquad \text{where} \qquad \boldsymbol{u}_{i\pm} \;\equiv\; \boldsymbol{C}_i\,\boldsymbol{H}_i\,\boldsymbol{x}_\pm \tag{11}$$

All of the above constraints are valid in arbitrary projective image and world-plane frames, except that $(\boldsymbol{x},\boldsymbol{y})$ are no longer orthonormal. As always, $(\boldsymbol{x},\boldsymbol{y})$ are defined only up to a 2×2 orthogonal mixing, and we can use this gauge freedom to require that $\boldsymbol{x} \cdot \boldsymbol{y} = 0$.

Our planar autocalibration method is based on direct numerical minimization of the residual error in the constraints (10) from several images, over the unknown direction basis $(\boldsymbol{x},\boldsymbol{y})$ and any combination of the five intrinsic calibration parameters f, a, s, u_0 and v_0. The input data is the set of world plane to image homographies $\boldsymbol{H}_i$ for the images, expressed with respect to an arbitrary projective frame for the world plane. In particular, if the plane is coordinatized by its projection into some key image (say image 1), the inter-image homographies $\boldsymbol{H}_{i1}$ can be used as input.

Four independent parameters are required to specify the Euclidean structure of a projective plane: the 6 components of $(\boldsymbol{x},\boldsymbol{y})$ modulo scale and the single d.o.f. of a 2×2 rotation; or the $3 \cdot 4/2 = 6$ components of a 3×3 absolute line quadric $\boldsymbol{Q}$ modulo scale and the rank 2 constraint $\det \boldsymbol{Q} = 0$; or the 2 d.o.f. of the plane's line at infinity, plus the 2 d.o.f. of two circular points on it. Since equations (9), (10) or (11) give two independent constraints for each image, $\lceil \frac{n+4}{2} \rceil$ images are required to estimate the Euclidean structure of the plane and n intrinsic calibration parameters. Two images suffice to recover the structure if the calibration is known, three are required if the focal length is also estimated, four for the perspective f, u_0, v_0 model, and five if all 5 intrinsic parameters are unknown.

2.3 Camera Parametrization

We have not yet made the camera parametrization explicit, beyond saying that it is given by the upper triangular matrices $\boldsymbol{K}$ or $\boldsymbol{C} = \boldsymbol{K}^{-1}$. For autocalibration methods which fix some parameters

while varying others, it makes a difference which parametrization is used. I prefer the following form motivated by a zoom lens followed by an affine image-plane coordinatization:

$$\boldsymbol{K} = \begin{pmatrix} f & f\,s & u_0 \\ 0 & f\,a & v_0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \boldsymbol{C} = \boldsymbol{K}^{-1} = \frac{1}{f\,a} \begin{pmatrix} a & -s & s\,v_0 - a\,u_0 \\ 0 & 1 & -v_0 \\ 0 & 0 & f\,a \end{pmatrix}$$

Here, if standard pixel coordinates are used, $f = \alpha_u$ is the focal length in u-pixels, $s = -\tan\theta_{\text{skew}}$ is the dimensionless geometric skew, $a = \alpha_v/(\alpha_u \cos\theta_{\text{skew}})$ is the dimensionless $v : u$ aspect ratio, and (u_0, v_0) are the pixel coordinates of the principal point. However pixel coordinates are *not* used in the optimization routine below. Instead, a nominal calibration is used to standardize the parameters to nominal values $f = a = 1$, $s = u_0 = v_0 = 0$, and all subsequent fitting is done using the above model with respect to these values.

3 Algebraic *vs.* Statistical Error

Many vision problems reduce to minimizing the residual violation of some vector of nonlinear constraints $\mathbf{e}(\mathbf{x}, \boldsymbol{\mu}) \approx \mathbf{0}$ over parameters $\boldsymbol{\mu}$, given fixed noisy measurements $\mathbf{x}$ with known covariance $\mathbf{V_x}$. Often, heuristic error metrics such as the **algebraic error** $\|\mathbf{e}(\mathbf{x}, \boldsymbol{\mu})\|^2$ are taken as the target for minimization. However, such approaches are statistically sub-optimal and if used uncritically can lead to (*i*) very significant bias in the results and (*ii*) severe constriction of the domain of convergence of the optimization method. Appropriate **balancing** or **preconditioning** (numerical scaling of the variables and constraints, *e.g.* as advocated in [7, 8] or any numerical optimization text) is the first step towards eliminating such problems, but it is not the whole story. In any case it begs the question of what *is* "balanced". It is *not* always appropriate to scale all variables to $\mathcal{O}(1)$. In fact, in the context of parameter estimation, "balanced" simply means "close to the underlying **statistical error measure**"[6]

$$\chi^2_{\mathbf{e}} \;\approx\; \mathbf{e}^\top\, \mathbf{V}_{\mathbf{e}}^{-1}\, \mathbf{e} \quad \text{where} \quad \mathbf{V_e} \;\approx\; \tfrac{\mathbf{De}}{\mathbf{Dx}}\, \mathbf{V_x}\, \tfrac{\mathbf{De}}{\mathbf{Dx}}^\top \quad \text{is the covariance of } \mathbf{e}$$

Ideally one would like to optimize the statistical cost (*i.e.* log likelihood). Unfortunately, this is often rather complicated owing to the matrix products and (pseudo-)inverse, and simplifying assumptions are often in order. I feel that this pragmatic approach is the *only* acceptable way to introduce algebraic error measures — as explicit, controlled approximations to the underlying statistical metric. Given that the extra computation required for a suitable approximation is usually minimal, while the results can be substantially more accurate, it makes little sense to iteratively minimize an algebraic error without such a validation step.

One very useful simplification is to ignore the dependence of $\mathbf{V}_{\mathbf{e}}^{-1}$ on $\boldsymbol{\mu}$ in cost function derivatives. This gives **self-consistent** or **iterative re-weighting** schemes (*e.g.* [12]), where $\mathbf{V_e}$ is treated as a constant within each optimization step, but updated at the end of it. One can show that the missing terms effectively displace the cost derivative evaluation point from the measured $\mathbf{x}$ to a first order estimate of the true underlying value $\mathbf{x}_0$ [22]. For the most part this makes little difference unless the constraints are strongly curved on the scale of $\mathbf{V_x}$.

For our autocalibration method, the statistical error splits into independent terms for each image[7]. For want of a more specific error model, we assume that the components of the data $\mathbf{x}$ (here,

[6] $\mathbf{e}$ is a random variable through its dependence on $\mathbf{x}$. Assuming that the uncertainty is small enough to allow linearization and that $\mathbf{x}$ is centred on some underlying $\mathbf{x}_0$ satisfying $\mathbf{e}(\mathbf{x}_0, \boldsymbol{\mu}_0) = 0$ for some parameter value $\boldsymbol{\mu}_0$, $\mathbf{e}(\mathbf{x}, \boldsymbol{\mu}_0)$ has mean $\mathbf{0}$ and the above covariance. It follows that $\mathbf{e}^\top\, \mathbf{V}_{\mathbf{e}}^{-1}\, \mathbf{e}$ is approximately a $\chi^2_{\text{rank}(\mathbf{e})}$ variable near $\boldsymbol{\mu}_0$, which can be minimized to find a maximum likelihood estimate of $\boldsymbol{\mu}$.

[7] We (perhaps unwisely) ignore the fact that the $\boldsymbol{H}_i$ are correlated through their mutual dependence on the base image. The base image is treated just like any other in the sum.

the $\boldsymbol{H}_i$ in nominally calibrated coordinates) are i.i.d.: $E\left[\Delta\boldsymbol{H}_B^A\,\Delta\boldsymbol{H}_D^C\right] \approx \epsilon\cdot\delta^{AC}\delta_{BD}$ where ϵ is a noise level[8]. From here it is straightforward to find and invert the constraint covariance. For the planar autocalibration constraint (10), and assuming that we have enforced the gauge constraint $\boldsymbol{x}\cdot\boldsymbol{y}=0$, the constraint covariance is

$$4\epsilon\cdot\begin{pmatrix}\boldsymbol{x}^2\,\boldsymbol{a}_i^2+\boldsymbol{y}^2\,\boldsymbol{b}_i^2 & (\boldsymbol{x}^2-\boldsymbol{y}^2)\,\boldsymbol{a}_i\cdot\boldsymbol{b}_i\\ (\boldsymbol{x}^2-\boldsymbol{y}^2)\,\boldsymbol{a}_i\cdot\boldsymbol{b}_i & \boldsymbol{x}^2\,\boldsymbol{b}_i^2+\boldsymbol{y}^2\,\boldsymbol{a}_i^2\end{pmatrix}\qquad\text{where}\quad(\boldsymbol{a}_i,\boldsymbol{b}_i)\;\equiv\;\boldsymbol{C}_i^\top\,(\boldsymbol{u}_i,\boldsymbol{v}_i)\;=\;\boldsymbol{\omega}_i^{-\mathbf{1}}\,\boldsymbol{H}_i\,(\boldsymbol{x},\boldsymbol{y})$$

In this case, numerical experience indicates that the off-diagonal term is seldom more than a few percent of the diagonal ones, which themselves are approximately equal for each image, but differ by as much as a factor of 2–3 between images[9]. Hence, we drop the off-diagonal term to give an autocalibration method based on self-consistent optimization of the diagonal cost function

$$\sum_{i=1}^{m}\left(\frac{(\|\boldsymbol{u}_i\|^2-\|\boldsymbol{v}_i\|^2)^2/4}{\boldsymbol{x}^2\,\|\boldsymbol{C}_i^\top\boldsymbol{u}_i\|^2+\boldsymbol{y}^2\,\|\boldsymbol{C}_i^\top\boldsymbol{v}_i\|^2}+\frac{(\boldsymbol{u}_i\cdot\boldsymbol{v}_i)^2}{\boldsymbol{x}^2\,\|\boldsymbol{C}_i^\top\boldsymbol{v}_i\|^2+\boldsymbol{y}^2\,\|\boldsymbol{C}_i^\top\boldsymbol{u}_i\|^2}\right)\quad\text{where}\;\;(\boldsymbol{u}_i,\boldsymbol{v}_i)\equiv\boldsymbol{C}_i\,\boldsymbol{H}_i\,(\boldsymbol{x},\boldsymbol{y})\tag{12}$$

In our synthetic experiments, this statistically motivated cost function uniformly reduced the ground-truth standard deviation of the final estimates by about 10% as compared to the best carefully normalized algebraic error measures. This is a modest but useful improvement, obtained without any measurable increase in run time. The improvement would have been much larger if the error model had been less uniform in the standardized coordinates. Perhaps most importantly, the statistical cost is almost completely immune to mis-scaling of the variables, which is certainly *not* true of the algebraic ones which deteriorated very rapidly for mis-scaling factors greater than about 3.

4 Planar Autocalibration Algorithm

Numerical Method: Our planar autocalibration algorithm is based on direct numerical minimization of the m-image cost function (12), with respect to the direction basis $\{\boldsymbol{x},\boldsymbol{y}\}$ and any subset of the 5 internal calibration parameters focal length f, aspect ratio a, skew s, and principal point (u_0,v_0). There are 4 d.o.f. in $\{\boldsymbol{x},\boldsymbol{y}\}$ — 6 components defined up to an overall mutual rescaling and a 2×2 orthogonal mixing — so the optimization is over 5–9 parameters in all. Numerically, the 6 component $(\boldsymbol{x},\boldsymbol{y})$ vector is locally projected onto the subspace orthogonal to its current scaling and mixing d.o.f. by Householder reduction (*i.e.* effectively a mini QR decomposition). As mentioned in section 2, the mixing freedom allows us to enforce the gauge condition $\boldsymbol{x}\cdot\boldsymbol{y}=0$. Although not essential, this costs very little (one Jacobi rotation) and we do it at each iteration as an aid to numerical stability.

A fairly conventional nonlinear least squares optimization method is used: Gauss-Newton iteration based on Choleski decomposition of the normal equations. As always, forming the normal equations gives a fast, relatively simple method but effectively squares the condition number of the constraint Jacobian. This is not a problem so long as intermediate results are stored at sufficiently high precision: double precision has proved more than adequate for this application.

[8] This model is undoubtedly over-simplistic. Balancing should make their variances similar, but in reality the components are most unlikely to be independent. We should at very least subtract a diagonal term $\boldsymbol{H}_B^A\boldsymbol{H}_D^C/\|\boldsymbol{H}_B^A\|^2$, as variations proportional to $\boldsymbol{H}$ make no projective difference. However this makes no difference here, as when contracted with $\nabla\mathbf{e}$'s it just gives back $\mathbf{e}(\mathbf{x}_0)$'s which vanish. This *had* to happen: correctly weighted error terms must be insensitive to projective scale factors, and hence have total homogeneity 0 in their projective-homogeneous variables.

[9] This was to be expected, since we chose everything to be well-scaled except that the $\boldsymbol{H}$ normalizations may differ somewhat from their 'correct' Euclidean ones, and our noise model is uniform in an approximately calibrated frame. If any of these conditions were violated the differences would be *much* greater.

As with any numerical method, care is needed to ensure stability should the numerical conditioning become poor. Our parametrization of the problem guarantees that all variables are of $\mathcal{O}(1)$ and fairly well decoupled, so preconditioning is not necessary. The Choleski routine uses diagonal pivoting and Gill & Murray's [4] minimum-diagonal-value regularization to provide local stability. The regularizer is also manipulated in much the same way as a Levenberg-Marquardt parameter to ensure that each step actually reduces the cost function. We also limit the maximum step size for each variable, relatively for the positive, multiplicative parameters f and a and absolutely for the others. Both the regularizer and the step size limits are activated fairly often in practice, the regularizer at any time, and the step limit usually only during the first 1–2 iterations. The method terminates when the step size converges to zero, with additional heuristics to detect thrashing. Convergence within 5–10 iterations is typical.

Prior over Calibrations: We also allow for a simple user-defined prior distribution on the calibration parameters. Even if there is no very strong prior knowledge, it is often advisable to include a weak prior in statistical estimation problems as a form of regularization. If there are unobservable parameter combinations (*i.e.* that make little or no difference to the fit), optimal, unbiased estimates of these are almost always extremely sensitive to noise. Adding a weak prior makes little difference to strong estimates, but significantly reduces the variability of weak ones by biasing them towards reasonable default values. A desire to "keep the results unbiased" is understandable, but limiting the impact of large fluctuations on the rest of the system may be more important in practice.

Default priors are also useful to ensure that parameters retain physically meaningful values. For example, we use heuristic priors of the form $(x/x_0 - x_0/x)^2$ for f and a, to ensure that they stay within their physically meaningful range $(0, \infty)$. This is particularly important for autocalibration problems, where degenerate motions occur frequently. In such cases the calibration can not be recovered uniquely. Instead there is a one or more parameter family of possible solutions, usually including physically unrealizable ones. A numerical method (if it converges at all) will converge to an arbitrary one of these solutions, and for sanity it pays to ensure that this is a physically feasible one not too far from the plausible range of values. A weak default prior is an effective means of achieving this, and seems no more unprincipled than any other method. This is not to say that such degeneracies should be left unflagged, but simply that whatever cleaning up needs to be done will be easier if it starts from reasonable default values.

Initialization: The domain of convergence of the numerical optimization is reasonably large and for many applications it will probably be sufficient to initialize it from fixed default values. The most critical parameters are the focal length f and the number and angular spread of the views. For example, if f can only be guessed within a factor of 2 and all 5 parameters f, a, s, u_0, v_0 are left free, about 9–10 images spread by more than about 10° seem to be required for reliable convergence to the true solution. Indeed, with 5 free parameters and the theoretical minimum of only 5–6 images, even an *exact* initialization is not always sufficient to eliminate false solutions (*i.e.* with slightly smaller residuals than the true one).

These figures assume that the direction basis $\boldsymbol{x}, \boldsymbol{y}$ is completely unknown. Information about this is potentially very valuable and should be used if available. Knowledge of the world-plane's horizon (line at infinity) removes 2 d.o.f. from $\boldsymbol{x}, \boldsymbol{y}$ and hence reduces the number of images required by one, and knowledge of its Euclidean structure (but not the positions of points on it) eliminates another image. Even if not directly visible, horizons can be recovered from known 3D parallelism or texture gradients, or bounded by the fact that visible points on the plane must lie inside them. We will not consider these types of constraints further here.

If a default initialization is insufficient to guarantee convergence, several strategies are possible. One quite effective technique is simply to use a preliminary optimization over $\boldsymbol{x}, \boldsymbol{y}$ or $\boldsymbol{x}, \boldsymbol{y}, f$ to initialize a full one over all parameters. More generally, some sort of initialization search over

f, $\boldsymbol{x}$ and $\boldsymbol{y}$ is required. Perhaps the easiest way to approach this is to fix nominal values for all the calibration parameters except f, and to recover estimates for $\boldsymbol{x},\boldsymbol{y}$ as a function of f from a single pair of images as f varies. These values can then be substituted into the autocalibration constraints for the other images, and the overall most consistent set of values chosen to initialize the optimization routine. The estimation of $\boldsymbol{x}(f),\boldsymbol{y}(f)$ reduces to the classical photogrammetric problem of the relative orientation of two calibrated cameras from a planar scene, as the Euclidean structure is easily recovered once the camera poses are known. In theory this problem could be solved in closed form (the most difficult step being a 3×3 eigendecomposition) and optimized over f analytically. But in practice this would be rather messy and I have preferred to implement a coarse numerical search over f. The search uses a new SVD-based planar relative orientation method (see appendix 1) related to Wunderlich's eigendecomposition approach [25]. The camera pose and planar structure are recovered directly from the SVD of the inter-image homography. As always with planar relative orientation, there is a two-fold ambiguity in the solution, so both solutions are tested. In the implemented routine, the solutions for each image against the first one, and for each f in a geometric progression, are substituted into the constraints from all the other images, and the most consistent overall values are chosen.

If the full 5 parameter camera model is to be fitted, Hartley's 'rotating camera' method [6] can also be used for initialization. It works well *provided* (*i*) the camera translations are smaller than or comparable to the distance to the plane; (*ii*) no point on the plane is nearly fixated from a constant distance. (For such a point $\boldsymbol{x}$, $\boldsymbol{\omega} + \mu\boldsymbol{x}\boldsymbol{x}^\top$ is an approximate solution of Hartley's equation $\boldsymbol{H}\boldsymbol{\omega}\boldsymbol{H}^\top = \boldsymbol{\omega}$ for any μ, *i.e.* $\boldsymbol{\omega}$ can not be estimated uniquely, even for small translations).

5 Experiments

Synthetic data: The method has been implemented in C and tested on both real and synthetic images. For the synthetic experiments, the camera roughly fixates a point on the plane from a constant distance, from randomly generated orientations varying by (by default) $\pm 30^\circ$ in each of the three axes. The camera calibration varies randomly about a nominal focal length of 1024 pixels and unit aspect ratio, by $\pm 30\%$ in focal length f, $\pm 10\%$ in aspect ratio a, ± 0.01 in dimensionless skew s, and ± 50 pixels in principal point (u_0, v_0). (These values are standard deviations of log-normal distributions for f, a and normal ones for s, u_0, v_0). The scene plane contains by default 40 visible points, projected into the 512×512 images with a Gaussian noise of ± 1 pixel. Before the homographies are estimated and the method is run, the pixel coordinates are centred and scaled to a nominal focal length of 1: $(u, v) \rightarrow (u - 256, v - 256)/1024$. The output is classed as a 'success' or 'failure' according to fixed thresholds on the size of its deviation from the true value. Only successes count towards the accuracy estimates. The usual mode of failure is convergence to a false solution with extremely short focal length (say < 50 pixels). However when the angular spread of the views is small or there are only a few images, random fluctuations sometimes take a "correct" but highly variable solution outside the (generously set) thresholds. Conversely, there is occasionally convergence to a false solution within the threshold. Thus, when the failure rate is high, neither it nor the corresponding error measure (nor, for that matter, the results!) are accurate. The optimization typically converges within 5–10 iterations, although more may be needed for degenerate problems. The run time is negligible: on a Pentium 133, about 0.5 milliseconds per image if the default initialization is used, or 2.0 with a fairly fine initialization search over f.

Figure 1 gives some illustrative accuracy and reliability results, concentrating on the estimation of focal length f. First consider the plots where all 5 calibration parameters are estimated. The error scales roughly linearly with noise and inversely with the angular spread of the views. It drops rapidly as the first few images are added, but levels off after about 10 images. The failure rate increases rapidly for more than about 2–3 pixels noise, and is also unacceptably high for near-

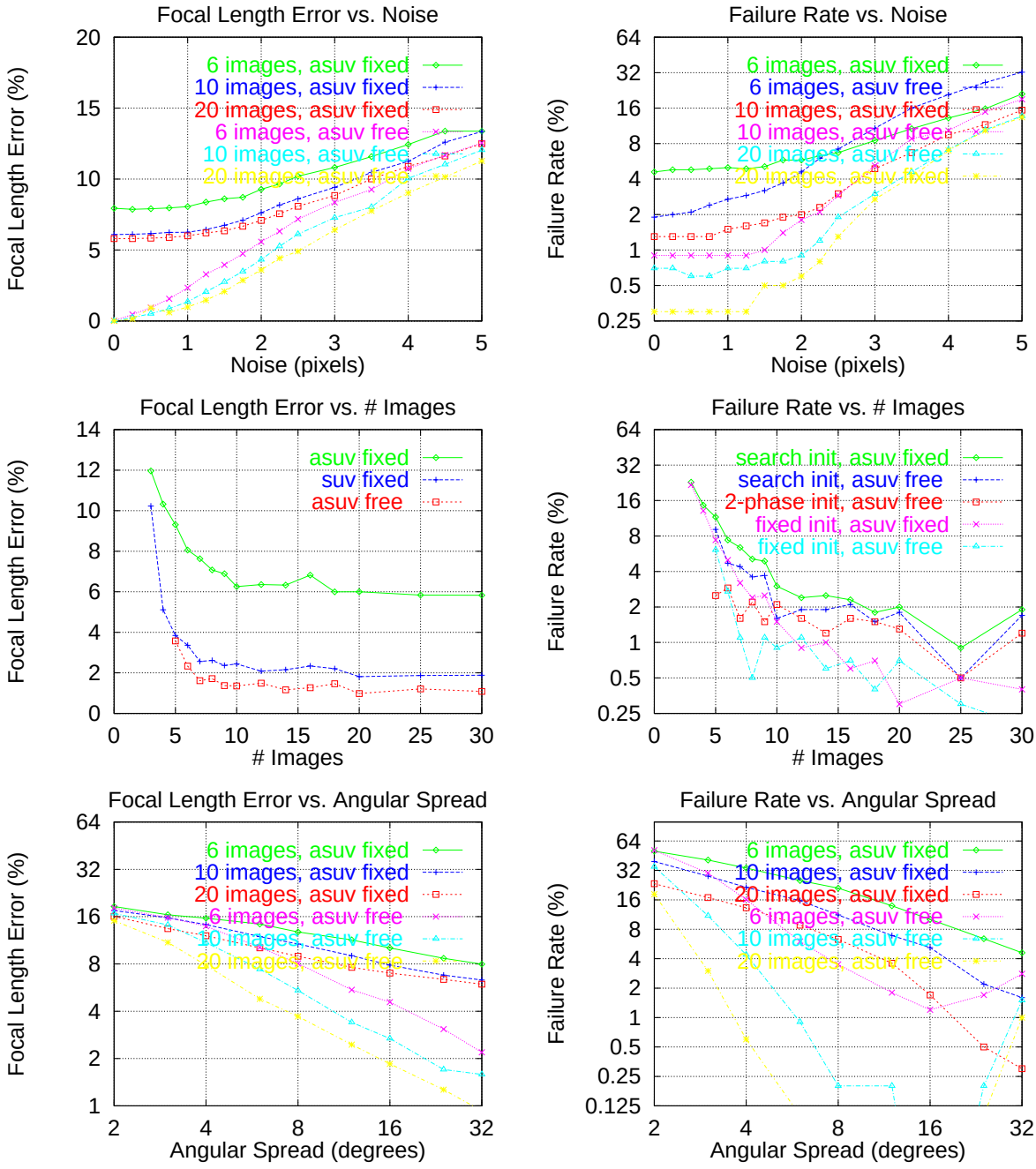


Figure 1: Error in estimated focal length f and failure rate *vs.* image noise, number of images and angular spread of cameras. Each value is the average of 1000 trials. The aspect ratio a, skew s, and principal point (u_0, v_0) are either fixed at their nominal values, or allowed to vary freely, as indicated. The method is initialized from the nominal calibration, except that in the failure *vs.* images plot we also show the results for initialization by numerical search over f, and by a preliminary fit over f alone ('2-phase').

minimal numbers of images (within 1–2 of the minimum) and small angular spreads (less than about 10°). however, it decreases rapidly as each of these variables is increased. It seems to be difficult to get much below about 1% failure rate with the current setup. Some of these failures are probably the result of degeneracies in the randomly generated problems, but most of them are caused by convergence to a false solution with implausible parameters, either very small f (less than about 50) or a far from 1. The initialization method has little impact on the reliability. In fact,

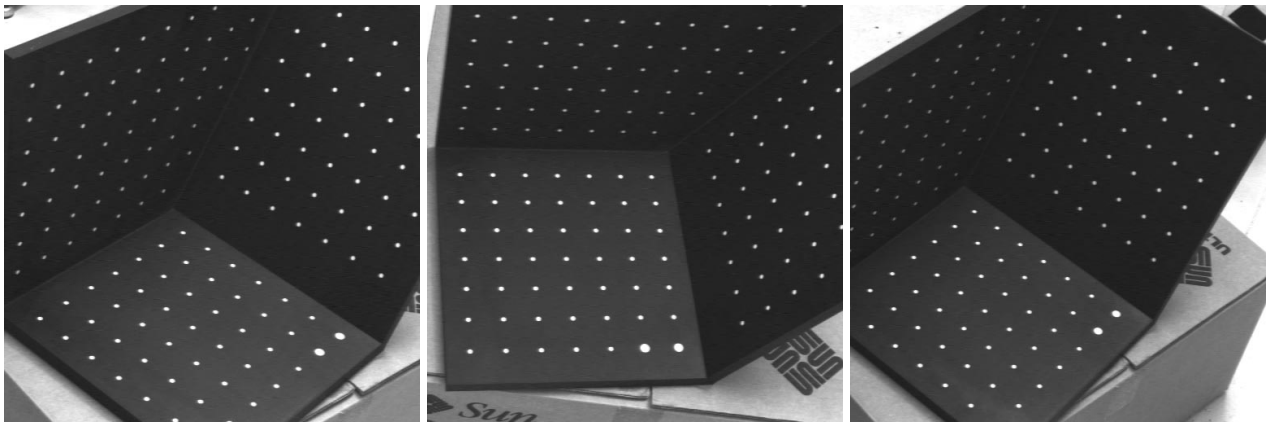


Figure 2: Several images from our calibration sequence.

in these experiments the default initialization proved more reliable than either numerical search over f, or an initial optimization over f alone. The reason is simply that we do not assume prior knowledge of *any* of the calibration parameters. An initialization search over f must fix a, s, u_0, v_0 at their inaccurate nominal values, and this is sometimes enough to make it miss the true solution entirely. This also explains the poor performance of the methods which hold a, s, u_0, v_0 fixed and estimate f alone. As the graphs of error *vs.* noise and number of images show, errors in a, s, u_0, v_0 lead to a significant bias in f, but most of this can be eliminated by estimating a as well as f. The initialization search over f also becomes much more reliable (*e.g.* 0.05% failure rate for 10 images, 30° spread and 1 pixel noise) if a and s are accurate to within a few percent. Here and elsewhere, it is only worthwhile to fix parameters if they are reliably known to an accuracy better than their measured variabilities, *e.g.* here for 1 pixel noise and 10 images, to about 0.003 for a, s or 20 pixels for u_0, v_0.

For conventional calibration, f is often said the most difficult parameter to estimate, and also the least likely to be known *a priori*. In contrast, a and s are said to be estimated quite accurately, while u_0 and v_0 — although variable — are felt to have little effect on the overall results. A more critical, quantitative view is to compare the *relative* accuracy $|\Delta f/f|$ to the dimensionless quantities $|\Delta a|$, $|\Delta s|$, $|\Delta u_0/f|$ and $|\Delta v_0/f|$. Errors in these contribute about equally to the overall geometric accuracy (*e.g.* reconstruction errors of 3D visual ray directions). Conversely, other things being equal, geometric constraints such as the autocalibration ones typically constrain each of these quantities to about the same extent. Hence a good rule of thumb is that for autocalibration (and many other types of calibration) $|\Delta u_0/f|$ and $|\Delta v_0/f|$ are of the same order of magnitude as $|\Delta f/f|$, while $|\Delta a|$ and $|\Delta s|$ are usually somewhat smaller if there is cyclotorsion or other aspect ratio constraints, but larger if there are none (*e.g.* if the rotation axis direction is almost constant). These rules are well borne out in all the experiments reported here: we always find $|\Delta u_0| \approx |\Delta v_0| \approx |\Delta f|$, while $|\Delta a|$ and $|\Delta s|$ are respectively about one fifth, one half, and one tenth of $|\Delta f/f|$ for the synthetic experiments, the real experiments below, and the Faugeras-Toscani calibration used in the real experiments.

Real data: We have run the method on several non-overlapping segments of a sequence of about 40 real images of a calibration grid (see fig. 2). Only the 49 (at most) points on the base plane of the grid are used. (It would be straightforward to extend the algorithm to handle several planes, but there seems little point as a non-planar autocalibration method could be used in this case). The motion was intended to be general within the limits of the 5 d.o.f. robot used to produce it, but is fairly uniform within each subsequence. Visibility considerations limited the total angular displacement to about $40°$, and significantly less within each subsequence. The sample means and standard

deviations over a few non-overlapping subsequences for (*i*) f alone, and (*ii*) all 5 parameters, are as follows (the errors are observed sample scatters, *not* estimates of absolute accuracy):

	f only	f	a	s	u_0	v_0
calibration	-	1515 ± 4	0.9968 ± 0.0002	-	271 ± 3	264 ± 4
6 images	1584 ± 63	1595 ± 63	0.9934 ± 0.0055	0.000 ± 0.001	268 ± 10	271 ± 22
8 images	1619 ± 25	1614 ± 42	0.9890 ± 0.0058	-0.005 ± 0.005	289 ± 3	320 ± 26
10 images	1612 ± 19	1565 ± 41	1.0159 ± 0.0518	-0.004 ± 0.006	273 ± 5	286 ± 27

The 'calibrated' values are the averaged results of several single-image Faugeras-Toscani calibrations using all visible points on the grid. Looking at the table, the results of the autocalibration method seem usable but not quite as good as I would have expected on the basis of the synthetic experiments. This may just be the effect of the small angular range within each subsequence, but the estimates of f seem suspiciously high and it may be that some small systematic error has occurred during the processing. Further work is required to check this. Note that in this case, fixing a, s, u_0, v_0 appears to have the desired effect of decreasing the variability of the estimated f without perturbing its value very much.

6 Summary

In summary, we have shown how autocalibration problems can be approached using a projective representation of orthogonal 3D direction frames, and used this to derive a practical numerical algorithm for the autocalibration of a moving projective camera viewing a planar scene. The method is based on the 'rectification' of inter-image homographies. It requires a minimum of 3 images if only the focal length is estimated, or 5 for all five internal parameters. Adding further images significantly increases both the reliability and the accuracy, up to a total of about 9–10. An angular spread between the cameras of at least 10–20° is recommended.

The priorities for future work are the initialization problem and the detection of false solutions (or possibly the production of multiple ones). Although the current numerical method is stable even for degenerate motions (and hence gives *a* possible solution), it does not attempt to detect and flag the degeneracy. This could be done, *e.g.* by extracting the null space of the estimated covariance matrix. It would also be useful to have autocalibration methods that could estimate lens distortion. This should be relatively simple in the planar case, as distortion can be handled during homography estimation.

Appendix: Homography Factorization

Our planar autocalibration approach is based on scene plane to image homographies $\boldsymbol{H}_i$. In practice we can not estimate these directly, only the inter-image homographies $\boldsymbol{H}_{ij} = \boldsymbol{H}_i \boldsymbol{H}_j^{-1}$ induced by them. In theory this is not a problem as the formalism is invariant to projective deformations of the input frame, so we can choose scene plane coordinates derived from a key image (say image 1) and use the $\boldsymbol{H}_{i1}$ in place of the $\boldsymbol{H}_i$ (*i.e.* the unknown direction vectors $\boldsymbol{x}, \boldsymbol{y}$ are parametrized by their coordinates in the key image). This works reasonably well in practice, but there is a risk that inaccurate measurements or poor conditioning in the key image will have an undue influence on the overall numerical accuracy or stability of the method, since they potentially contribute coherently to all the $\boldsymbol{H}$'s. It would be useful to find a homography representation that does not single out a

specific key image, but instead averages the uncertainty over all of them. This can be achieved by a factorization method analogous to factorization-based projective structure and motion [20, 21][10].

This appendix describes the homography factorization algorithm. However note that it is *not* used in the final planar autocalibration routine as it turns out to give slightly *worse* results in practice. I am not sure why this happens. It may be that the scaling required for the homographies induces less than ideal error averaging, or that the resulting frame is in some way less well adapted to the calibration problem. In any case, it suggests that the use of a key image does not introduce too much bias in the calibration. Despite this, I have included a description of the factorization method here as I still think it is potentially useful for other applications.

Suppose we have estimated inter-image homographies $\boldsymbol{H}_{ij}$ between each pair of m images of a plane. In terms of some coordinate system on the plane which induces plane to image homographies $\boldsymbol{H}_i$ we have $\lambda_{ij}\boldsymbol{H}_{ij} \approx \boldsymbol{H}_i\boldsymbol{H}_j^{-1} + \text{noise}$, where the λ_{ij} are unknown scale factors. Write this as a big $(3m) \times (3m)$ rank 3 matrix equation

$$\begin{pmatrix} \lambda_{11}\boldsymbol{H}_{11} & \lambda_{12}\boldsymbol{H}_{12} & \cdots & \lambda_{1m}\boldsymbol{H}_{1m} \\ \lambda_{21}\boldsymbol{H}_{21} & \lambda_{22}\boldsymbol{H}_{22} & \cdots & \lambda_{2m}\boldsymbol{H}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1}\boldsymbol{H}_{m1} & \lambda_{m2}\boldsymbol{H}_{m2} & \cdots & \lambda_{mm}\boldsymbol{H}_{mm} \end{pmatrix} \approx \begin{pmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \vdots \\ \boldsymbol{H}_m \end{pmatrix} \begin{pmatrix} \boldsymbol{H}_1^{-1} & \boldsymbol{H}_2^{-1} & \cdots & \boldsymbol{H}_m^{-1} \end{pmatrix} + \text{noise}$$

As in the projective structure case, if we can recover a self-consistent set of scale factors λ_{ij}, the left hand side can be factorized to rank 3 using (*e.g.*) SVD or a fixed-rank power iteration method: $\mathbf{H}_{3m\times 3m} = \mathbf{U}_{3m\times 3}\mathbf{V}_{3\times 3m}$. Any such rank 3 factorization has the required noise-averaging properties and represents some 'numerically reasonable' choice of projective coordinates on the plane. For our purposes we need not insist that the 3×3 submatrices of $\mathbf{U}$ are exactly the inverses of those of $\mathbf{V}$, although — given that $\boldsymbol{H}_{ii} = \boldsymbol{I}$ — the inverse property is always approximately satisfied up to scale.

A suitable set of scale factors λ_{ij} can be found very simply by choosing a key image 1 and noting that up to scale $\boldsymbol{H}_{ij} \approx \boldsymbol{H}_{i1}\boldsymbol{H}_{1j}$. Resolving this approximate matrix proportionality by projecting it along $\boldsymbol{H}_{ij}$, we find that the quantities

$$\lambda_{ij} \equiv \frac{\text{Trace}\left((\boldsymbol{H}_{i1}\boldsymbol{H}_{1j}) \cdot \boldsymbol{H}_{ij}^\top\right)}{\text{Trace}(\boldsymbol{H}_{ij} \cdot \boldsymbol{H}_{ij}^\top)}$$

are an approximately self-consistent set of scale factors. As in the projective structure case, the matrix of scale factors λ_{ij} is only defined up to independent overall rescalings of each row and each column. Numerically, it is highly advisable to balance the matrix so that all its elements are of order $\mathcal{O}(1)$ before applying it to the $\boldsymbol{H}_{ij}$'s and factorizing. Our balancing algorithm proceeds by alternate row and column normalizations as in the projective structure case [20], and converges within 2-3 iterations.

It may seem that using a key image to find the scale factors is likely to spoil the noise averaging properties of the factorization, but this is not so. Perturbations of the scales of $\boldsymbol{H}_{i1}$ and $\boldsymbol{H}_{1j}$ introduce no inconsistency, while other perturbations of order $\mathcal{O}(\epsilon)$ introduce errors only at $\mathcal{O}(\epsilon^2)$ in the projection of $\boldsymbol{H}_{i1}\boldsymbol{H}_{1j}$ along $\boldsymbol{H}_{ij}$ — and hence in the scale factors — as these matrices are proportional up to noise. At normal noise levels $\epsilon \ll \frac{1}{m}$, these errors are swamped by the $\mathcal{O}(\epsilon)$ ones arising from the explicit $\boldsymbol{H}_{i1}$ and $\boldsymbol{H}_{1j}$ terms in the factorization, so each image has roughly the same total influence on the result (*provided* the λ_{ij} have been balanced appropriately). The same phenomenon is observed in the projective structure method: errors in the fundamental matrices and epipoles used to estimate the scales have very little effect.

[10] Analogous methods also exist for 3D homographies (projective structure alignment, rank=4) and, more interestingly, for finding coherent sets of fundamental matrices or line projections (rank=6).

References

[1] M. Armstrong, A. Zisserman, and R. Hartley. Self-calibration from image triplets. In B. Buxton and R. Cipolla, editors, *European Conf. Computer Vision*, pages 3–16, Cambridge, U.K., April 1996.

[2] O. Faugeras. Stratification of 3-d vision: Projective, affine, and metric representations. *J. Optical Society of America*, A 12(3):465–84, March 1995.

[3] O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[4] P. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, 1981.

[5] R. Hartley. Euclidean reconstruction from multiple views. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.

[6] R. Hartley. Self-calibration from multiple views with a rotating camera. In *European Conf. Computer Vision*, pages 471–8. Springer-Verlag, 1994.

[7] R. Hartley. In defence of the 8-point algorithm. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1064–70, Cambridge, MA, June 1995.

[8] R. Hartley. Minimizing algebraic error. In *IEEE Int. Conf. Computer Vision*, Bombay, January 1998.

[9] A. Heyden and K. Åström. Euclidean reconstruction from constant intrinsic parameters. In *Int. Conf. Pattern Recognition*, pages 339–43, Vienna, 1996.

[10] A. Heyden and K. Åström. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[11] R. Horaud and G. Csurka. Self-calibration and euclidean reconstruction using motions of a stereo rig. In *IEEE Int. Conf. Computer Vision*, January 1998.

[12] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, Amsterdam, 1996.

[13] Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. Technical Report UCB/CSD-93-772, Dept. EECS, Berkeley, California, 1993.

[14] S. J. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. J. Computer Vision*, 8(2):123–151, 1992.

[15] M. Pollefeys and L Van Gool. A stratified approach to metric self-calibration. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 407–12, San Juan, Puerto Rico, June 1997.

[16] M. Pollefeys, L Van Gool, and M. Proesmans. Euclidean 3d reconstruction from image sequences with variable focal length. In B. Buxton and R. Cipolla, editors, *European Conf. Computer Vision*, pages 31–42, Cambridge, U.K., April 1996.

[17] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1952.

[18] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[19] P. Sturm. *Vision 3D non calibrée : contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage*. Ph.D. Thesis, Institut National Polytechnique de Grenoble, December 1997.

[20] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–20, Cambridge, U.K., 1996. Springer-Verlag.

[21] B. Triggs. Autocalibration and the absolute quadric. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[22] B. Triggs. A new approach to geometric fitting. Available from http://www.inrialpes.fr/movi/people/Triggs, 1997.

[23] B. Triggs. Autocalibration from planar scenes. In *European Conf. Computer Vision*, pages I 89–105, Freiburg, June 1998.

[24] T. Viéville and O. Faugeras. Motion analysis with a camera with unknown and possibly varying intrinsic parameters. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 750–6, Cambridge, MA, June 1995.

[25] W. Wunderlich. Rechnerische rekonstruktion eines ebenen objekts aus zwei photographien. In *Mittsilungen Geodät. Inst. TU Gras*, Folge 40 (festschrift K. Rimmer sum 70. Geburstag), pages 365–77, 1982.

[26] C. Zeller and O. Faugeras. Camera self-calibration from video sequences: the Kruppa equations revisited. Technical Report 2793, INRIA, INRIA Sophia-Antipolis, France, 1996.

[27] Z. Zhang, Q.-T. Luong, and O. Faugeras. Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. Technical Report 2079, INRIA, Sophia-Antipolis, France, 1993.

Chapitre 6

Perspectives et problèmes ouverts

> If the fool would persist in his folly
> he would become wise.
>
> William BLAKE
> *The Marriage of Heaven and Hell*

La recherche – et toute particulièrement une thèse – étant une exemplaire hors pair de la persistance en sa folie, il faut à l'occasion se demander si on est déjà devenu sage ... et sinon, combien de temps et comment persister ? – Cette chapitre propose, sous la forme de problèmes ouverts, quelques perspectives sur la vision géométrique engendré par nos travaux pendant la periode de cette thèse.

Considérons d'abord quelques problèmes techniques de la vision géométrique.

Reconstruction des scènes complexes : Malgré tous nos efforts, la reconstruction visuelle de scènes complexes représente toujours un défi majeur. La mise en correspondance est loin d'être résolue, particulièrement quand les prises de vue des images sont très écartées [SMB98, PZ98, TGDK99]. Le choix d'une paramétrisation et son initialisation automatique ne sont pas plus évidentes, quand les primitives et les contraintes sont complexes. Enfin l'ajustement de faisceaux pour de grands modèles reste très coûteux, en particulier pour des scènes dynamiques, où l'existence de paramètres de mouvements indépendants à chaque image de la séquence peut augmenter énormément le nombre de paramètres à estimer.

Concernant l'ajustement de faisceaux, nos experiences initiales semblent indiquer que dans un cas réaliste où chaque primitive n'est vue que par un nombre constant de images (donc le nombre de primitives augmente linéairement avec le nombre d'images n), toutes les méthodes connues risquent d'être à peu près de l'ordre $\mathcal{O}\left(n^3\right)$. Ceci en dépit de tout effort de prendre en compte l'aspect très creux du système, ou encore de le résoudre par des méthodes itératives (qui sont de l'ordre $\mathcal{O}\left(n\right)$ ou $\mathcal{O}\left(n^2\right)$ par itération, mais qui semblent prendre un nombre exorbitant d'itérations quand n augmente). Des meilleures méthodes numériques pour résoudre ce problème d'optimisation sont à mettre au point – ou des méthodes directes qui gèrent de façon plus efficace l'aspect creux du système, ou encore des méthodes itératives qui prennent mieux en compte sa structure enchaînée ...caméra–primitive–caméra...

Méthodes d'initialisation fiables : Si on commence à maîtriser l'étape d'optimisation pour la plupart de nos problèmes, trouver une solution approximative initiale reste difficile, et plus particulièrement quand le conditionnement géométrique est délicat et/ou il y a un grand nombre de valeurs

aberrantes dans les données. On ne demande pas que l'initialisation soit parfaite, mais seulement qu'elle tombe avec consistance dans la zone où l'algorithme d'optimisation converge vers la solution optimale du problème, et non vers un autre minimum local, ou dans une zone qui est inadmissible. À présent, dans les cas où il manque des valeurs par défaut qui suffiront à l'initialisation, on cherche souvent à initialiser par une solution quasi-linéaire (c.-à-d., qui ne prend pas en compte quelques contraintes nonlinéaires qui auraient normalement dû être imposées), basée sur un modèle d'erreur algébrique ou linéarisé. On sait convertir (par exemple) un système général de polynômes dans une telle forme, mais : (*i*) le résultat risque d'être lourd ; (*ii*) le modèle d'erreur approché qui est implicit en la construction peut être très biaisé, et ceci et l'élision de contraintes nonlinéaires engendrent souvent une initialisation imprécise voire même fausse ; (*iii*) il y a toujours des singularités, qui correspondent très souvent aux cas où on voudrait appliquer l'algorithme. Des méthodes de réduction plus fiables et plus légères seraient bienvenues pour l'initialisation, particulièrement si elles peuvent prendre en compte le modèle d'erreur statistique et les zones de convergence de la méthode d'optimisation nonlinéaire. Dans le cas où les minima locaux du problème ont une structure typique (ce qui est peut être le cas pour la reconstruction), ce serait aussi intéressant de développer des heuristiques pour « sauter » d'un minimum à (la zone d'attraction d')un autre, afin de trouver le minimum global.

Un autre aspect de l'initialisation est l'utilisation de méthodes d'échantillonnage aléatoire comme RANSAC [FB81] pour contourner le problème de valeurs aberrantes. Avec de telles valeurs, l'essence est de trouver une *cohérence* – un sous-ensemble des données qui soient cohérentes avec le même modèle, et qui ne le seraient que très rarement par hasard, si elles ne correspondent pas réellement à un tel modèle. RANSAC et ses cousins nous semblent des méthodes effectives, mais assez primaires pour ce type de travail. Il doit y avoir des méthodes moins coûteuses et plus sûres pour trouver la cohérence, mais de toute façon un échantillonnage entièrement aléatoire nous semble trop simpliste, particulièrement si la dimension ou la probabilité d'aberrance sont grandes, ou si plusieurs modèles différents sont nécessaires pour décrire la scène. Il serait intéressant de développer des méthodes de tirage qui prennaient mieux en compte les informations préalables sur la distribution de primitives, y inclut les principes de support local, et d'exclusion dans le cas où plusieurs correspondances sont possibles. La recherche de correspondence / cohérence géométrique a un aspect optimisation combinatoire qui est loin d'être épuisé.

Une exemple notable d'initialisation est la reconstruction par factorisation. La factorisation (SVD ou autre) trouve automatiquement « par magie » les caméras et la structure 3D – sans aucune étape d'initialisation explicite, et sans problèmes apparents de minima locaux ou de convergences fausses. « C'est plus fiable que l'optimisation » ... mais en effet, au coeur de la SVD il y a précisément une méthode d'optimisation itérative, relativement délicate – en effet, il a fallu 30 ans d'expérience pour la perfectionner – et avec sa propre étape d'initialisation interne. On appel de telles méthodes « directes » – elles sont itératives et en principe faillibles, mais en pratique si sûres et de convergence si régulière que on les considère égales aux méthodes finies comme l'élimination gaussienne. Nous sommes convaincus que des méthodes d'initialisation d'une fiabilité pareille sont possibles en vision, par exemple pour la factorisation avec des données manquantes, mais aussi pour bien d'autres problèmes. Seulement, la méthode de réduction matricielle qui donne la première étape de la SVD semble difficilement généralisable aux données manquantes, donc il faut chercher une autre façon de procéder.

L'auto-calibrage depuis trois images : Un problème ouvert notable est de trouver les contraintes d'appariement entre les trois images d'une quadrique. Ce problème est important principalement par son apport à l'auto-calibrage – de telles contraintes appliquées à la quadrique absolue duale seraient l'analogue en 3 images des contraintes de Kruppa [MF92] pour le cas de 2 images. En principe le problème est simple : la méthode de dérivation des contraintes d'appariement des points s'applique

directement aux quadriques, avec des matrices 6×10 de projection qui sont quadratiques aux entrées des matrices ordinaires 3×4. Seulement, les tenseurs d'appariement quadriques sont issue des déterminants 10×10, et les récrire en terme de tenseurs standards (voir même les développer tels quels), est un problème fort lourd. Nous l'avons attaqué de plusieurs façons et à plusieurs reprises pendant la période de cette thèse, sans jamais aboutir, mais sans tomber très loin non plus. Nous allons poursuivre, bien qu'il ne soit pas clair que la résolution doit mener à une méthode pratique, car les contraintes risquent d'être elles mêmes fort complexes.

Une autre façon d'aborder le même sujet serait d'étudier le tenseur trifocal calibré – l'analogue en 3 images de la matrice essentielle en 2, et qui a la représentation $\boldsymbol{R} \otimes \boldsymbol{e}' - \boldsymbol{e} \otimes \boldsymbol{R}'$, où $\boldsymbol{R}$,$\boldsymbol{R}'$ sont des matrices 3×3 de rotation.

L'algèbre géométrique multi-images : Alors que de nombreux chercheurs en vision ont étudié intensivement la vision géométrique, aucun d'eux n'était expert en géomètrie algébrique moderne de par sa formation. Pour l'instant il semble que nous avons poussé les outils plus classiques presque aussi loin que possible, avec de bons résultats, mais toujours avec une explosion de complexité qui en limite l'horizon. Cependant, il nous semble que les rares exploration de la vision par les géomètres algébriques professionnels [Dem88, Buc92] ont promis des avances théoriques significatives, si toutefois ils l'étudieraient plus systématiquement avec des outils abstraits modernes ((co)homologie, résolutions libres, classes caractéristiques, outils de dénouement de singularités ...). Surtout, et en dehors de son intérêt théorique, une telle étude pourrait apporter sur les problèmes d'initialisation, des minima locaux, de paramétrisation effective de la géométrie multi-caméras, et des singularités de la reconstruction et l'auto-calibrage.

Tournons maintenant vers des perspectives plus larges. Il nous semble que – quoique le filon de la géométrie pure qui a tant apporté récemment soit bien loin d'être épuisé comme certains le prédisent – la recherche en vision entre maintenant une période plus synthétique et applicative, où l'ingénierie et l'interdisciplinarité vont présider. C'est à dire, il va falloir que ceux qui travaillent sur la reconstruction deviennent un peu photogrammètres [1], et ceux qui travaillent aux applications en synthèse d'images deviennent un peu plus graphistes.

Les deux domaines suivants nous semblent particulièrement susceptibles de subir un progrès significatif pour les années qui viennent :

Reconstruction de modèles effectifs de rendu graphique : La géométrie n'est qu'une partie d'un modèle de scène graphique générative. Il faut y ajouter [FvDFH91, WW92] des modèles d'illumination et de réflectance (matériel, BRDF), d'ombres, de transparence et d'effets atmosphériques pour permettre un rendu de qualité, et des partitions géométriques et des échelles multiples pour l'accélérer ... Dans ce genre d'application, tout est permis pourvu que les images de sortie soient convaincantes, légères à générer, et facile à manier ou à modifier. Le modèle peut être un mélange de sous-modèles physiques, heuristiques, locaux, en 3D ou 2D ; la géométrie peut être imprécise, simplifiée, implicite ou même inexistante ; tous les raccourcis sont autorisés – imposteurs, couches, bump maps, carrelages, textures stochastiques. Il faut bien sûr commencer avec des choses simples, mais les environnements à reconstruire – « imiter vraisemblablement à base d'images » serait peut être une description plus précise – peuvent au fur et à mesure devenir très complexes, avec géométrie et photométrie détaillées, mouvement, matériaux non-rigides ...

Certes on fait déjà l'affichage de cartes de texture sur des facettes planes, ce qui donne des résultats plus ou moins bons. Mais les transitions entre facettes et les bords d'objets restent particu-

1. L'ignorance quasi-totale en vision des éléments de base de la photogrammétrie nous semble inadmissible, mais être mauvais ingénieur, et redécouvrir les débuts des méthodes déjà bien connues et développées d'ailleurs, semble toujours presque un point de fierté chez de nombreux visionneurs ...

lièrement difficiles à rendre correctement, et de tels modèles ont en général une apparence « plate » ... comme les images affichées sur les plans. Pour créer des modèles qui soient visuellement plus « vifs », il va falloir aussi capter les micro-effets de la surface – l'interaction de la texture 3D et des petits reliefs, avec l'illumination locale, les ombres et les reflets, et les micro-parallaxes. Tous ces effets nous donnent d'importants indices perceptuels, mais qui sont marginaux par rapport à la géométrie globale de la scène.

En effet, la géométrie classique de points, droites, facettes planes isolés n'est pas toujours suffisante pour le graphisme. Même dans les environnements rectilinéaires, il faut souvent ajouter des couches multi-échelles ou de partitionnement spatial afin d'accélérer le rendu. Dans les environnements plus naturels, il faut considérer des modèles de géométrie (ou de photométrie) plus flexibles : des surfaces splines ou implicites ; des modèles génératifs stochastiques, fractals ou d'ondelettes pour les arbres, l'herbe, les surfaces texturées. Tout modèle graphique étant par définition un modèle génératif, on peut espérer optimiser un tel modèle depuis une estimation initiale, en minimisant itérativement les différences entre l'ensemble d'images observées et les mêmes images synthétisées. Mais les modèles réalistes ont tant de paramètres – à la fois discrets et continus – que la paramétrisation, l'initialisation et l'optimisation d'un tel modèle risquent d'être extrêmement complexes. Il y aura certainement un grand nombre de paramètres qui sont difficilement estimables, et pour lesquels l'information préalable, la régularisation, et les approximations joueraient un rôle critique. En plus, la mesure à optimiser est la similarité perceptive, ce qui n'est pas évident, surtout pour les modèles génératifs ou effectifs qui ne peuvent pas espérer à reproduire l'image en détail, mais seulement son apparence générale.

Compréhension de scène : On dispose à présent des technologies de structuration d'information et d'apprentissage (c'est à dire, estimation) statistique qui n'était que naissantes il y a 10 ans. De plus, on a la puissance de calcul qu'il faut pour alimenter de telles méthodes pour des images. Il nous semble que ces méthodes, genre réseaux bayesiens et modèles de markov implicites (HMM) [Per88, Lau96, CDLS99, RJ86], vont catalyser pendant les 10 ou 20 ans qui viennent un changement profond dans nos capacités de modéliser et de manier la réalité visuelle (changement qui était prédite il y a quelques années dans la communauté de « vision active » [BY92, Alo93], mais qui n'est pas encore là ...). Cette révolution va se produire d'abord par des systèmes qui observent en continu un environnement ou une classe d'activités (par voie de descripteurs 2D ou 3D adaptées, extrait des images en temps réel), et qui apprennent plus ou moins automatiquement (mais selon une architecture préprogrammée) une réponse désiré aux événements qui ont lieu. Les méthodes d'échantillonnage aléatoire (RANSAC, MCMC, Condensation) vont jouer un rôle ici, mais au centre seront les représentations probabilistes *structurées*. C'est la structuration et la modularisation intensive à tous les niveaux qui rendraient possible l'élaboration de tels « systèmes experts de vision », et l'apprentissage des milliers de paramètres nécessaires à leur bonne fonctionnement. Sur le plan des réseaux probabilistes, la modularité, l'apprentissage d'un grand nombre de paramètres, et l'interopérabilité entre les représentations géométriques–continues et les représentations sémantiques–discrètes restent difficiles mais vont émerger. Sur le plan vision, les aspects de moyen niveau (le couplage entre les descripteurs d'image de bas niveau et la représentation plus sémantique de réseau probabiliste) ne sont toujours pas évidents, mais ils sont susceptibles d'être attaqué par la même méthode de réseaux probabilistes.

Annexe A

Autres papiers

Cette appendice regroupe plusieurs autres travaux qui n'ont pas été inclus dans le corps du texte, car s'éloignant un peu de l'axe défini pour cette thèse, ou étant plus marginaux par rapport aux références principales citées.

A.1 Resumé de « Matching Constraints and the Joint Image » – ICCV'95

Ce papier est la version courte de « » dans le corps du texte. J'inclus cette version ici seulement pour référence – il est notamment plus compact à lire, et il reste à ce jour la seule version publiée de ce travail (dans ICCV'95 [Tri95]).

A.2 Resumé de « A Fully Projective Error Model for Visual Reconstruction »

Ce papier – un essai pour un modèle projectif d'erreurs, qui est analogue à la Gaussienne dans le cas affine – fut écrit en 1995, mais n'a pas encore été publié (il était soumis au workshop ICCV'95 « Representations of Visual Scenes » à l'époque). Le travail fut commencé il y a longtemps, en 1992 quand Kenichi KANATANI était en sabbatique en Oxford (où j'étais roboticien) et écrivait son livre « Geometric Computation for Machine Vision » [Kan93]. Il a donné un cours sur quelques chapitres de son ouvrage et auquel j'ai assisté. Nous avons discuté ensemble les modèles d'erreurs. Je n'ai pas été convaincu qu'une forme affine des erreurs images était toujours et strictement « la chose correcte » pour un processus essentiellement projectif comme la formation des images. J'ai voulu créer un modèle « projectif de base », où les choses projectives auraient la forme simple et invariante sous les transformations et les projections projectives. La forme de base de la distribution aux points projectifs fut vite trouvée, mais l'avancement de l'idée était très lent car il fallait rechercher : (*i*) une forme analytique ou une approximation convenable pour exprimer les résultats d probabilistes ; (*ii*) une généralisation de la théorie des points à d'autres sous-espace linéaires, théorie qui engloberait et généraliserait les algèbres Grassmann-Cayley au cas incertain [1].

L'article référencée ci-dessous ne montre qu'une partie assez réduite de cette programme ambitieux. Il évite les questions analytiques (où j'ai des résultats partiels mais pas satisfaisants) et se

1. Seitz & Anandan ont récemment publié un autre tentative de modèle des sous-espaces affines [SA99] incertains, où une Gaussienne (dans l'espace des positions des points) est ajustée sur l'ensemble des points mesurés, et ses axes principaux les plus grands définissent le sous-espace « optimal ». A mon avis ce modèle (que j'avais considéré et rejeté à l'époque) est trop simpliste. Il ne reproduit pas l'ajustement standard moindre carrés d'un sous-espace sur une ensemble des points Gaussiens, et il représente un modèle réduit d'incertitude, ayant seulement $\binom{n}{2}$ paramètres de covariance au place de $\binom{(n-k)k}{2}$ (n,k = dimension de l'espace, sous-espace).

focalise sur la forme algébrique du modèle, et ce pour les points et pour les sous-espaces linéaires de plus haute dimension. Les aspects Grassmann-Cayley – intersection et union des sous-espace, ajustement de sous-espace sur des points – ne sont pas abordés.

L'essentiel consiste à introduire, en contrepartie de la forme quadratique de la log-vraisemblance Gaussienne standard, un dénominateur de la même forme. Donc le modèle de distribution de probabilité devient l'exponentiel d'une forme quadratique rationnelle homogène. Le fait que il soit rationnel homogène donne une invariance de forme sous les transformations projectives, et aussi permet le début d'une généralisation aux sous-espaces linéaires généraux. Le nombre de paramètres libres est en principe multiplié par deux par cette homogénéisation. La forme en « cloche elliptique » d'une Gaussienne compacte peut être maintenue quasi-globalement, mais d'autres formes deviennent possibles, notamment la « cloche × cône » des pré-images possibles d'un point image incertain. Les distributions aux dénominateurs différentes peuvent localement être recombinées, mais pas globalement de façon aussi simple qu'avec les Gaussiennes. La loi devient l'exponentiel d'une *somme* de termes quadratiques rationnels incombinables, ce qui est plus difficile à manipuler (à intégrer, à maximiser) qu'une seule quadratique. Donc même si on a réussi à créer une forme de distribution qui est invariante sous les transformations projectives, les calculs pratiques ont tendance à générer des sommes de distributions incombinables, et il faut assez tôt passer de l'analytique au numérique.

A.3 Resumé de « Critical Motions in Euclidian Structure from Motion » – CVPR'99

Ce papier avec Frederik KAHL, doctorant à Lund en Suède, fut publié en CVPR'99 [KT99]. Il fut écrit lors de son séjour chez nous en automne 1998, dans le cadre de notre projet européen commun ESPRIT LTR 21914 CUMULI. Le but général du programme est de caractériser rigoureusement les cas où l'auto-calibrage faillit. Peter STURM avait déjà publié une excellente étude de ces cas pour l'auto-calibrage aux paramètres internes constants inconnus [Stu97a, Stu97b]. Fredrik a voulu étendre l'étude aux autres cas, ou avec connaissances préalables sur certains paramètres (skew, rapport d'échelle), ou avec d'autres paramètres variables (focale). Il existait déjà des algorithmes pratiques pour plusieurs de ces cas, par exemple pour l'estimation des deux longueurs focales à partir d'une matrice fondamentale [Har92, Har93b, NHBP96, Bou98]. Mais toutes ces méthodes ont des singularités qui se montrent souvent gênantes en pratique, et on a voulu caractériser lesquelles étaient intrinsèques au problème, et lesquelles seraient évitables par des meilleures formulations.

Les preuves « intuitives-géométriques » de Sturm semblaient à l'époque difficilement généralisables à ces cas parfois « plus simples » mais toujours moins symétriques, donc nous avons pris une route plus algébrique, fondée sur la géométrie algébrique effective. En principe, on travaille dans l'espace de caméras (poses et calibrages) et des structures euclidiennes 3D possibles – espace qui peut être paramétré de plusieurs façons. L'essentiel, c'est que dans cet espace, chaque suite de contraintes d'auto-calibrage découpe la variété algébrique de caméras/structures qui les vérifient. On suppose que ces contraintes sont notre seul moyen de déceler la vraie calibrage/structure face à des solutions alternatives fausses. Donc le problème d'auto-calibrage ne peut être résolu de façon unique que si cette variété est réduite à un seul point, *i.e.* que si il n'existe pas d'autres caméras/structures qui vérifient les contraintes. On étude la variété par voie de « **l'ideal** » (ensemble de tous les polynômes) engendrée par les contraintes. Chaque ideal peut – et pour les calculs effectifs, souvent doit – être caractérisé par certains sous-ensembles « exhaustifs » dit « **bases de Gröbner** ».

En principe ces calculs sont « standards », mais en pratique ils ont une forte tendance à exploser de façon incontrôlable. Dans ce premier article on n'a abordé que des situations relativement simples, mais déjà les calculs ont été assez lourds, même pour des outils de calcul de base de Gröb-

ner récents comme MACAULAY 2 et SINGULAR. Donc toute l'astuce consiste à trouver une bonne paramétrisation, ce qui relève encore de l'intuition géométrique ...

A.4 Resumé de « Camera Pose and Calibration from 4 or 5 Known 3D Points » – ICCV'99

Ce papier fut publié à ICCV'99. Il décrit encore un travail fait pour notre projet européen CUMULI, en ce cas sur l'initialisation des caméras. Il donne plusieurs méthodes permettant de retrouver la pose (position et orientation) et quelques paramètres internes d'une caméra, à partir d'une seule image d'un minimum de 4 ou 5 points 3D connus. Toutes ces méthodes sont basées sur les matrices multiresultantes – façon de résoudre un système de polynômes redondante avec l'algèbre linéaire. Leur avantage est que – comme la « **transformée directe linéaire** » classique pour 6 points – elles sont quasi-linéaires, donc relativement facile à implanter et donnent une solution unique.

Le papier raconte aussi la théorie de base des multiresultants dans une forme concrète et applicable aux calculs numériques – chose rare dans la littérature, où l'accent est toujours mis sur les aspects formels qui n'ont qu'un impact très relatif sur la construction des matrices multiresultantes compactes et stables.

Matching Constraints and the Joint Image

Bill Triggs
LIFIA, INRIA Rhône-Alpes,
46 avenue Félix Viallet, 38031 Grenoble, France.
Bill.Triggs@imag.fr

Abstract

This paper studies the geometry of multi-image perspective projection and the **matching constraints** that this induces on image measurements. The combined image projections define a 3D **joint image subspace** of the space of combined homogeneous image coordinates. This is a complete projective replica of the 3D world in image coordinates. Its location encodes the imaging geometry and is captured by the 4 index **joint image Grassmannian** tensor. Projective reconstruction in the joint image is a canonical process requiring only a simple rescaling of image coordinates. Reconstruction in world coordinates amounts to a choice of basis in the joint image. The matching constraints are multilinear tensorial equations in image coordinates that tell whether tokens in different images could be the projections of a single world token. For 2D images of 3D points there are exactly three basic types: the epipolar constraint, Shashua's trilinear one, and a new quadrilinear 4 image one. For images of lines Hartley's trilinear constraint is the only type. The coefficients of the matching constraints are tensors built directly from the joint image Grassmannian. Their complex algebraic interdependency is captured by quadratic structural simplicity constraints on the Grassmannian.

Keywords: multi-image stereo, projective reconstruction, matching constraints, tensor calculus, geometric invariants.

1 Introduction

Multi-image reconstruction is currently a topic of lively interest in the vision community. This paper uncovers some rather beautiful geometric structure that underlies multi-image projection, and applies it to the problem of projective reconstruction. There is only space for a brief sketch of the theory here: more detail can be found in [8]. The mathematics and notation may be a little unfamiliar, but the main conclusions are fairly straightforward: The homogeneous coordinates for all the images can be gathered into a single vector and viewed as a point in an abstract projective space called **joint image space**. The combined projection matrices define a 3D projective subspace of joint image space called the **joint image**. This is an exact projective replica of the 3D world in image coordinates. Up to an arbitrary choice of scale factors its position encodes the imaging geometry. The combined projection matrices can be viewed either as a set of image projections or as a projective basis for the joint image. Algebraically, the location of the joint image is encoded by the antisymmetric four index **joint image Grassmannian** tensor, whose components are 4×4 minors built from projection matrix rows. Projective scene reconstruction is a canonical process only in the joint image, where it reduces to a simple rescaling of image coordinates. World-space reconstruction amounts to the choice of a projective basis for the joint image. The essence of reconstruction is the recovery of a coherent set of scalings for the image coordinates of different tokens, modulo a single arbitrary overall choice of scale factors. The multilinear tensorial **matching constraints** tell whether tokens in different images could possibly be the projections of a single world token. For 2D images of 3D points there are exactly three basic types: the bilinear epipolar constraint; Shashua's trilinear one [7]; and a new quadrilinear four image one. The sequence stops at four because homogenized 3D space has four dimensions. For images of lines the only type of matching constraint is Hartley's trilinear one [4]. The matching constraints are a direct algebraic reflection of the location of the joint image. Their coefficients are tensors built from com-

This paper appeared in ICCV'95. The work was supported by the European Community through Esprit programs HCM and SECOND.

ponents of the joint image Grassmannian. Up to a choice of scale factors the Grassmannian is linearly equivalent to the matching tensors. The matching tensors and constraints are linearly independent but algebraically highly redundant. The redundancy is encapsulated by a set of 'structural simplicity' constraints on the Grassmannian, that induce a large set of quadratic identities among the matching tensors. For m 2D images of 3D space there are $\binom{3m}{4}$ linearly independent matching tensor components, but only $11m - 15$ of these are algebraically independent. We introduce an 'industrial strength' tensorial notation that (even though it may seem a little opaque at first sight) makes these and many other complex vision calculations *much* easier. The traditional matrix-vector notation is simply not powerful enough to express most of the concepts described here.

The geometry of the joint image was suggested by the original projective reconstruction papers of Faugeras, Luong & Maybank [1, 2], but its algebraic expression was only provoked by the recent work of Shashua [7] and Hartley [4] on the trilinear constraint and Luong & Viéville on canonic decompositions [5]. Independently of the current work, Faugeras & Mourrain [3] and Werman & Shashua [10] also discovered the quadrilinear constraint and some of the related structure (but not the 'big picture' — the full joint image geometry). The tensorial notation and the general spirit of the approach owe a very deep debt to the Oxford mathematical physics research group led by Roger Penrose [6].

2 Conventions & Notation

We will assume an uncalibrated perspective (pinhole camera) imaging model and work projectively in homogeneous coordinates. The development will be purely theoretical: there will be 'too many equations, no algorithms and no images'. Divine intervention (or more likely a graduate student with a mouse) will be invoked for low-level token extraction and matching. Measurement uncertainty will be ignored (but *c.f.* [9]).

Fully tensorial notation will be used, with all indices written out explicitly [6]. Writing out indices is tedious for simple expressions but makes complicated ones *much* clearer. Many equations apply only up to scale, denoted "$\sim$". Different types of index denote different spaces: $a, b, \ldots = 0, \ldots, d$ and $A_i, B_i, \ldots = 0, \ldots, D_i$ respectively denote homogeneous coordinates in the d-dimensional projective world space $\mathcal{P}^a$ and the D_i-dimensional i^{th} image $\mathcal{P}^{A_i}$. Usually, $d = 3$ and $D_i = 2$ but other cases do have applications. $i, j, \ldots = 1, \ldots, m$ are non-tensorial image labels. Section 3 introduces a $(D + m - 1)$-dimensional projective **joint image space** $\mathcal{P}^\alpha$ that combines the homogeneous coordinates of all of the images, indexed by Greek indices $\alpha, \beta, \ldots = 0_1, \ldots, D_i, 0_{i+1}, \ldots, D_m$ ($D \equiv \sum_{i=1}^m D_i$). Index 0 is used for homogenization, so the default inclusion of an affine vector $(x^1, \ldots, x^d)^\top$ in projective space is $(1, x^1, \ldots, x^d)$.

Superscripts denote contravariant (point) indices and subscripts covariant (hyperplane) ones. These transform inversely under changes of basis, so that the **contraction** (dot product or sum over all values) of a covariant-contravariant pair is invariant. We adopt the **Einstein summation convention** in which indices repeated in covariant and contravariant positions denote contractions (implicit summations). The same base symbol is used for analogous things in different spaces, with $\mathbf{x}, \mathbf{y}, \ldots$ standing for points and $\mathbf{P}$ for projection matrices. For example $\mathbf{x}^{A_i} \sim \mathbf{P}_a^{A_i} \mathbf{x}^a$ represents the the projection up to scale of a world point $\mathbf{x}^a$ to the corresponding i^{th} image point $\mathbf{x}^{A_i}$ via the matrix-vector product $\sum_{\mathrm{a}=0}^{d} \mathbf{P}_{\mathrm{a}}^{A_i} \mathbf{x}^{\mathrm{a}}$ with the i^{th} projection matrix $\mathbf{P}_a^{A_i}$. Since the indices themselves give the contraction, the order of factors is irrelevant.

$\mathbf{T}^{[ab\ldots c]}$ denotes the **antisymmetrization** of $\mathbf{T}^{ab\ldots c}$ over all permutations of the indices $ab\ldots c$, with a minus sign for odd permutations, *e.g.* $\mathbf{T}^{[ab]} \equiv \frac{1}{2!}(\mathbf{T}^{ab} - \mathbf{T}^{ba})$. In a d-dimensional projective space there is a unique-up-to-scale $d+1$ index antisymmetric tensor $\boldsymbol{\varepsilon}^{[a_0 a_1 \cdots a_d]}$ and its dual $\boldsymbol{\varepsilon}_{[a_0 a_1 \cdots a_d]}$. Up to scale, these have components ± 1 and 0 as $a_0 a_1 \ldots a_d$ is respectively an even or odd permutation of $01 \ldots d$, or not a permutation at all. Any antisymmetric $k+1$ index contravariant tensor $\mathbf{T}^{[a_0 \ldots a_k]}$ can be **dualized** to an antisymmetric $d-k$ index covariant one $(*\mathbf{T})_{a_{k+1} \cdots a_d} \equiv \frac{1}{(k+1)!}\, \boldsymbol{\varepsilon}_{a_{k+1} \cdots a_d b_0 \cdots b_k} \mathbf{T}^{b_0 \ldots b_k}$, and vice versa $\mathbf{T}^{a_0 \ldots a_k} = \frac{1}{(d-k)!}\, (*\mathbf{T})_{b_{k+1} \cdots b_d}\, \boldsymbol{\varepsilon}^{b_{k+1} \cdots b_d a_0 \cdots a_k}$, without losing information. This is effectively just a reshuffling of components: both forms have $\binom{d+1}{k+1}$ linearly independent components.

Later we will need to characterize the location of a projective d-dimensional subspace algebraically, without reference to a particular choice of basis in the subspace. This can be done by specifying an antisymmetric $(d+1)$-index **Grassmann tensor** whose components are the **Grassmann coordinates** of the subspace. These generalize the Plücker coordinates of a 3D line to arbitrary subspaces. An appendix sketches the details.

3 The Joint Image

The basic idea of the joint image is very simple. Suppose we are given m homogeneous projection matrices $\mathbf{P}_a^{A_i}$ from a d-dimensional world space $\mathcal{P}^a$ to m D_i-dimensional images $\mathcal{P}^{A_i}$. The matrices can be stacked into a big $(D+m)\times(d+1)$ dimensional **joint projection matrix** $(D=\sum_i D_i)$

$$\mathbf{P}_a^{\alpha} \equiv \begin{pmatrix} \mathbf{P}_a^{A_1} \\ \vdots \\ \mathbf{P}_a^{A_m} \end{pmatrix} \qquad \mathbf{x}^{\alpha} \equiv \begin{pmatrix} \mathbf{x}^{A_1} \\ \vdots \\ \mathbf{x}^{A_m} \end{pmatrix}$$

This maps world points $\mathbf{x}^a$ to $(D+m)$-component homogeneous vectors $\mathbf{x}^\alpha$. These can be viewed as elements of an abstract $(D+m-1)$-dimensional projective **joint image space** $\mathcal{P}^\alpha$. Joint image space points can be projected into the images by trivial coordinate selection, and conversely any set of homogeneous image vectors (one from each image) determines a unique point in joint image space.

The joint projection matrix can be viewed as a projective mapping from world space to joint image space, which composes with the trivial projections to give back the original projection matrices $\mathbf{P}_a^{A_i}$. It maps $\mathcal{P}^a$ onto a projective subspace of $\mathcal{P}^\alpha$ that we will call the **joint image** $\mathcal{PI}^\alpha$. If the joint projection mapping is singular, different world points map to the same point in joint image space and therefore in the individual images, and unique reconstruction from image measurements is impossible. So from now on we will assume that the joint projection matrix $\mathbf{P}_a^\alpha$ has full rank $(d+1)$. In this case *the joint image is a faithful projective replica of the d-dimensional world in image coordinates.*

The joint image is defined canonically by the imaging geometry, up to an arbitrary choice of scale factors for the underlying projection matrices. The truly canonical structure is the set of equivalence classes of joint image space points under arbitrary rescalings, but that has a complicated stratified structure that makes it difficult to handle. So from now on we will assume that some choice of scalings has been made and work with the joint image.

The joint projection matrix can be viewed in two ways: *(i)* as a set of m world-to-image projection matrices; *(ii)* as a set of $d+1$ $(D+m)$-component column vectors that specify a projective basis for the joint image subspace $\mathcal{PI}^\alpha$ in $\mathcal{P}^\alpha$. Hence, a coordinate vector $(x^0,\ldots,x^d)$ can be viewed either as the coordinates of a world point $\mathbf{x}^a$ or as the coordinates of a joint image point with respect to the basis $\{\mathbf{P}_{\mathrm{a}}^{\alpha}|\mathrm{a}=0,\ldots,d\}$. *Any reconstruction in world coordinates can equally be viewed as a reconstruction in the joint image*. However, modulo a once-and-for-all choice of the overall scale factors, *reconstruction in the joint image is a canonical geometric process requiring only a simple rescaling of image coordinates*. The m-tuples of image points that correspond to some world point are exactly those that *can* be rescaled to lie in the joint image [8]. No choice of basis is needed and there is no arbitrariness apart from the overall scale factors. A basis is needed only to transfer the final results from the joint image to world space. In fact, the portion of the world that can be recovered from image measurements is *exactly* the abstract joint image geometry.

Since the joint image is a d dimensional projective subspace its location can be specified algebraically by giving its $(d+1)$-index Grassmann coordinate tensor, the **joint image Grassmannian**. This is an intrinsic property of the joint image geometry independent of any choice of coordinates, but in terms of the projection matrices it becomes

$$\mathbf{I}^{\alpha_0\cdots\alpha_d} \equiv \tfrac{1}{(d+1)!}\mathbf{P}_{a_0}^{\alpha_0}\cdots\mathbf{P}_{a_d}^{\alpha_d}\boldsymbol{\varepsilon}^{a_0\cdots a_d} \sim \mathbf{P}_0^{[\alpha_0}\cdots\mathbf{P}_d^{\alpha_d]}$$

Here each α_i runs through the combined coordinates of all the images, and the components of the tensor are just the $(d+1)\times(d+1)$ minors of the $(D+m)\times(d+1)$ joint projection matrix $\mathbf{P}_a^\alpha$. We will see that these are equivalent to the complete set of matching tensor components.

As a simple example of joint image geometry [8], for two 2D images of 3D space the fundamental matrix $\mathbf{F}_{A_1A_2}$ has rank 2 and can therefore be decomposed as $\mathbf{u}_{A_1}\mathbf{v}_{A_2}-\mathbf{v}_{A_1}\mathbf{u}_{A_2}$ where $\mathbf{u}_{A_1}\leftrightarrow\mathbf{u}_{A_2}$ and $\mathbf{u}_{A_1}\leftrightarrow\mathbf{u}_{A_2}$ turn out to be corresponding pairs of

independent epipolar lines. Combining these into joint image space row vectors $\mathbf{u}_\alpha \equiv (\mathbf{u}_{A_1}\ \mathbf{u}_{A_2})$ and $\mathbf{v}_\alpha \equiv (\mathbf{v}_{A_1}\ \mathbf{v}_{A_2})$, the constraints $\mathbf{u}_\alpha \mathbf{x}^\alpha = 0 = \mathbf{v}_\alpha \mathbf{x}^\alpha$ define a 3D projective subspace of the 5D joint image space that turns out to be exactly the joint image. All joint image points satisfy the epipolar constraint $\mathbf{F}_{A_1A_2}\mathbf{x}^{A_1}\mathbf{x}^{A_2} = 0$, and all image points that satisfy the epipolar constraint can be rescaled to lie in the joint image.

4 Basic Reconstruction Equations

Given m images $\mathbf{x}^{A_i} \sim \mathbf{P}_a^{A_i}\mathbf{x}^a$ of an unknown point $\mathbf{x}^a$, we can introduce variables λ_i to represent the unknown scale factors and combine the resulting equations $\mathbf{P}_a^{A_i}\mathbf{x}^a - \lambda_i \mathbf{x}^{A_i} = \mathbf{0}$ into a single $(D+m)\times(d+1+m)$ homogeneous linear system, the **basic reconstruction equations**:

$$\left(\begin{array}{c|cccc} & \mathbf{x}^{A_1} & \mathbf{0} & \cdots & \mathbf{0} \\ & \mathbf{0} & \mathbf{x}^{A_2} & \cdots & \mathbf{0} \\ \mathbf{P}_a^\alpha & \vdots & \vdots & \ddots & \vdots \\ & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{x}^{A_m} \end{array}\right) \left(\begin{array}{c} \mathbf{x}^a \\ \hline -\lambda_1 \\ -\lambda_2 \\ \vdots \\ -\lambda_m \end{array}\right) = \mathbf{0}$$

Any nonzero solution of these equations gives a reconstructed world point consistent with the image measurements, and also provides the unknown scale factors λ_i.

Alternatively, assuming (or relabelling so that) $\mathbf{x}^{0_i} \neq 0$ we can use the 0^{th} components to eliminate the λ's and combine the remaining equations into a compact $D \times (d+1)$ homogeneous system of **reduced reconstruction equations**:

$$\left(\begin{array}{c} \mathbf{x}^{0_1}\,\mathbf{P}_a^{A_1} - \mathbf{x}^{A_1}\,\mathbf{P}_a^{0_1} \\ \vdots \\ \mathbf{x}^{0_m}\,\mathbf{P}_a^{A_m} - \mathbf{x}^{A_m}\,\mathbf{P}_a^{0_m} \end{array}\right) \mathbf{x}^a = \mathbf{0} \qquad {\scriptstyle (A_i=1,\ldots,D_i)}$$

The basic and reduced systems are ultimately equivalent, but we will work with the basic one as its greater symmetry simplifies many derivations.

In either case, if there are more measurements than world dimensions ($D > d$) the system is usually overspecified and a solution exists only when certain constraints between the projection matrices $\mathbf{P}_a^{A_i}$ and the image measurements $\mathbf{x}^{A_i}$ are satisfied. We will call these relations **matching constraints** and the inter-image tensors they generate **matching tensors**. The simplest example is the epipolar constraint.

On the other hand, if $D < d$ there will be at least two more free variables than equations and the solution (if it exists) will not be unique. Similarly, if the joint projection matrix $\mathbf{P}_a^\alpha$ has rank less than $d+1$ the solution will not be unique because any vector in the kernel of $\mathbf{P}_a^\alpha$ can be added to a solution without changing the projections at all. So from now on we will require $D \geq d$ and $\mathrm{Rank}(\mathbf{P}_a^\alpha) = d+1$. These conditions are necessary but not generally sufficient. However in the usual 3D to 2D case where the 3×4 rank 3 projection matrices have 1D kernels (the centres of projection), $\mathrm{Rank}(\mathbf{P}_a^\alpha) = 4$ implies that there are at least two distinct centres of projection and is also *sufficient* for a unique reconstruction.

Recalling that the joint projection columns $\mathbf{P}_a^\alpha$ ($a = 0,\ldots,d$) form a basis for the joint image $\mathcal{PI}^\alpha$ and treating the $\mathbf{x}^{A_i}$ as vectors in $\mathcal{P}^\alpha$ whose other components vanish, we can interpret the reconstruction equations as the geometrical statement that the space spanned by the image vectors $\{\mathbf{x}^{A_i} |\ i = 1,\ldots,m\}$ in $\mathcal{P}^\alpha$ must intersect $\mathcal{PI}^\alpha$. At the intersection there is a point of $\mathcal{P}^\alpha$ that can be expressed: *(i)* as a rescaling of the image measurements $\sum_i \lambda_i \mathbf{x}^{A_i}$; *(ii)* as a point of $\mathcal{PI}^\alpha$ with coordinates $\mathbf{x}^a$ in the basis $\{\mathbf{P}_a^\alpha |\ a = 0,\ldots,d\}$; *(iii)* as the projection into $\mathcal{PI}^\alpha$ of a world point $\mathbf{x}^a$ under $\mathbf{P}_a^\alpha$. This construction is important because although neither the coordinate system in $\mathcal{P}^a$ nor the columns of $\mathbf{P}_a^\alpha$ can be recovered from image measurements, the joint image $\mathcal{PI}^\alpha$ *can* be recovered (up to a relative rescaling). In fact the content of the matching constraints is *precisely* the location of the joint image in $\mathcal{P}^\alpha$. This gives a completely geometric and almost canonical projective reconstruction technique in $\mathcal{P}^\alpha$ that requires only the rescaling of image measurements. A choice of basis in $\mathcal{PI}^\alpha$ is necessary only to map the construction back into world coordinates.

5 Matching Constraints

Now we briefly sketch the derivation [8] of the matching constraints from the basic reconstruction equations. We assume that there are redundant measurements $D > d$ and that the combined projection matrix $\mathbf{P}_a^\alpha$ has full rank $(d+1)$. The equations have a nonzero solution if and only if the

$(D+m)\times(d+m+1)$ coefficient matrix is rank deficient, which happens if and only if all of its $(d+m+1)\times(d+m+1)$ minors vanish. The matching constraints are precisely the conditions for this to happen.

Each minor involves an antisymmetrization over every column of the system matrix, so the minors are homogeneous multilinear functions linear in each $\mathbf{x}^{A_i}$, with coefficients that are antisymmetrized products of projection matrix elements of the form $\mathbf{P}_0^{[\alpha_0}\mathbf{P}_1^{\alpha_1}\cdots\mathbf{P}_d^{\alpha_d]}$ for some choice of $\alpha_0\ldots\alpha_d$. This implies that *the final matching constraint equations will be linear tensorial equations in the coordinates of each image that appears in them, with coefficient tensors that are exactly the Grassmann coordinates* $\mathbf{I}^{\alpha_0\cdots\alpha_d}$ *of the joint image subspace in* $\mathcal{P}^\alpha$. This is no accident: the Grassmann coordinates are the only quantities that *could* have appeared if the equations were to be properly invariant under projective changes of basis in world space.

Each minor involves all m images, but the system matrix is rather sparse and there are many degeneracies. In fact, any minor that involves only a single row A_i from image i simply contains a constant overall factor of $\mathbf{x}^{\mathrm{A}_i}$. These factors can be eliminated to reduce the system to irreducible factors involving at least two rows from each of between 2 and $d+1$ images. For 2D images of 3D space the possibilities are as follows ($i\neq j\neq k\neq l=1,\ldots,m$):

$$\begin{aligned}\mathbf{0} &= \mathbf{I}^{[A_iB_iA_jB_j}\,\mathbf{x}^{C_i}\mathbf{x}^{C_j]}\\ \mathbf{0} &= \mathbf{I}^{[A_iB_iA_jA_k}\,\mathbf{x}^{C_i}\mathbf{x}^{B_j}\mathbf{x}^{B_k]}\\ \mathbf{0} &= \mathbf{I}^{[A_iA_jA_kA_l}\,\mathbf{x}^{B_i}\mathbf{x}^{B_j}\mathbf{x}^{B_k}\mathbf{x}^{B_l]}\end{aligned}$$

These represent respectively the bilinear epipolar constraint, Shashua's trilinear one [7] and a new quadrilinear four image one. Here, $\mathbf{x}^{A_i}$ represents a $\mathcal{P}^\alpha$ vector whose non-image-i components vanish, so it is enough to antisymmetrize over the indices from each image separately. Each constraint is discussed in detail below. Recall that the Grassmannian can be expressed as $\mathbf{I}^{\alpha\beta\gamma\delta} \equiv \frac{1}{4!}\mathbf{P}_a^\alpha\mathbf{P}_b^\beta\mathbf{P}_c^\gamma\mathbf{P}_d^\delta\,\varepsilon^{abcd}$.

5.1 Bilinear Constraints

The epipolar constraint corresponds to a 6×6 minor containing three rows each from two images and (antisymmetrizing separately over each image) can be written $\mathbf{x}^{[A_1}\,\mathbf{I}^{B_1C_1][B_2C_2}\,\mathbf{x}^{A_2]} = \mathbf{0}$. Dualizing both sets of skew indices by contracting with $\varepsilon_{A_1B_1C_1}\,\varepsilon_{A_2B_2C_2}$ gives the equivalent but more familiar form

$$\begin{aligned}0 &= \mathbf{F}_{A_1A_2}\,\mathbf{x}^{A_1}\mathbf{x}^{A_2}\\ &= \tfrac{1}{4\cdot 4!}\left(\varepsilon_{A_1B_1C_1}\mathbf{P}_a^{A_1}\mathbf{P}_b^{B_1}\mathbf{x}^{C_1}\right)\cdot\\ &\qquad\cdot\left(\varepsilon_{A_2B_2C_2}\mathbf{P}_c^{A_2}\mathbf{P}_d^{B_2}\mathbf{x}^{C_2}\right)\varepsilon^{abcd}\end{aligned}$$

where the $3\times 3=9$ component bilinear constraint tensor or **fundamental matrix** $\mathbf{F}_{A_1A_2}$ is defined by

$$\begin{aligned}\mathbf{F}_{A_1A_2} &\equiv \tfrac{1}{4}\,\varepsilon_{A_1B_1C_1}\,\varepsilon_{A_2B_2C_2}\,\mathbf{I}^{B_1C_1B_2C_2}\\ &= \tfrac{1}{4\cdot 4!}\left(\varepsilon_{A_1B_1C_1}\mathbf{P}_a^{B_1}\mathbf{P}_b^{C_1}\right)\cdot\\ &\qquad\cdot\left(\varepsilon_{A_2B_2C_2}\mathbf{P}_c^{B_2}\mathbf{P}_d^{C_2}\right)\varepsilon^{abcd}\\ \mathbf{I}^{B_1C_1B_2C_2} &= \mathbf{F}_{A_1A_2}\,\varepsilon^{A_1B_1C_1}\,\varepsilon^{A_2B_2C_2}\end{aligned}$$

The constraint can be viewed geometrically as follows. An image point $\mathbf{x}^A$ can be dualized to $\varepsilon_{ABC}\,\mathbf{x}^C$. Roughly speaking, this represents the point as the pencil of lines through it: for any two lines $\mathbf{l}_A$ and $\mathbf{m}_A$ through $\mathbf{x}^A$, the tensor $\mathbf{l}_{[A}\mathbf{m}_{B]}$ is proportional to $\varepsilon_{ABC}\,\mathbf{x}^C$. Any covariant image tensor can be 'pulled back' through the projection $\mathbf{P}_a^A$ to a covariant tensor in 3D space. An image line $\mathbf{l}_A$ pulls back to the 3D plane $\mathrm{l}_a = \mathrm{l}_A\mathbf{P}_a^A$ through the projection centre that projects to the line. The tensor $\varepsilon_{ABC}\,\mathbf{x}^C$ pulls back to the 2 index covariant tensor $\mathbf{x}_{[ab]} \equiv \varepsilon_{ABC}\,\mathbf{P}_a^A\mathbf{P}_b^B\,\mathbf{x}^C$. This is the covariant representation of a line in 3D: the optical ray through $\mathbf{x}^A$. The requirement that two 3D lines $\mathbf{x}_{[ab]}$ and $\mathbf{y}_{[ab]}$ intersect can be written $\mathbf{x}_{ab}\,\mathbf{y}_{cd}\,\varepsilon^{abcd}=0$. So the bilinear constraint really *is* the standard epipolar one, *i.e.* the requirement that the optical rays of the two image points intersect.

5.2 Trilinear Constraints

The trilinear constraints $\mathbf{I}^{[B_1C_1B_2B_3}\,\mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3]} = \mathbf{0}$ correspond to 7×7 basic reconstruction minors formed by selecting all three rows from one image and two each from two others. Dualizing with $\varepsilon_{A_1B_1C_1}$ gives the equivalent constraint

$$\mathbf{x}^{A_1}\mathbf{x}^{[A_2}\,\mathbf{G}_{A_1}{}^{B_2][B_3}\,\mathbf{x}^{A_3]} = \mathbf{0}$$

where the $3\times 3\times 3=27$ component trilinear tensor is

$$\begin{aligned}\mathbf{G}_{A_1}{}^{A_2A_3} &\equiv \tfrac{1}{2}\,\varepsilon_{A_1B_1C_1}\,\mathbf{I}^{B_1C_1A_2A_3}\\ &= \tfrac{1}{2\cdot 4!}\left(\varepsilon_{A_1B_1C_1}\mathbf{P}_a^{B_1}\mathbf{P}_b^{C_1}\right)\,\mathbf{P}_c^{A_2}\,\mathbf{P}_d^{A_3}\,\varepsilon^{abcd}\\ \mathbf{I}^{A_1B_1A_2A_3} &= \mathbf{G}_{C_1}{}^{A_2A_3}\,\varepsilon^{C_1A_1B_1}\end{aligned}$$

Dualizing the image 2 and 3 indices re-expresses the constraint as

$$\begin{aligned}\mathbf{0} &= \varepsilon_{A_2B_2C_2}\,\varepsilon_{A_3B_3C_3}\cdot\mathbf{G}_{A_1}{}^{B_2B_3}\cdot\mathbf{x}^{A_1}\mathbf{x}^{C_2}\mathbf{x}^{C_3}\\ &= \tfrac{1}{2.4!}\left(\varepsilon_{A_1B_1C_1}\,\mathbf{P}_a^{A_1}\mathbf{P}_b^{B_1}\mathbf{x}^{C_1}\right)\cdot\\ &\quad\cdot\left(\varepsilon_{A_2B_2C_2}\,\mathbf{P}_c^{B_2}\mathbf{x}^{C_2}\right)\left(\varepsilon_{A_3B_3C_3}\,\mathbf{P}_d^{B_3}\mathbf{x}^{C_3}\right)\varepsilon^{abcd}\end{aligned}$$

These equations hold for all $3\times 3 = 9$ values of the free indices A_2 and A_3. However when A_2 is projected along the $\mathbf{x}^{A_2}$ direction or A_3 is projected along the $\mathbf{x}^{A_3}$ direction the equations are tautological because, for example, $\mathbf{x}^{[A_2}\mathbf{x}^{B_2]}\equiv 0$. So for any particular vectors $\mathbf{x}^{A_2}$ and $\mathbf{x}^{A_3}$ there are actually only $2\times 2 = 4$ linearly independent scalar constraints among the $3\times 3 = 9$ equations, corresponding to the two image 2 directions 'orthogonal' to $\mathbf{x}^{A_2}$ and the two image 3 directions 'orthogonal' to $\mathbf{x}^{A_3}$. The trilinear constraint can also be written in matrix notation (*c.f.* [7]) as

$$[\mathbf{x}_2]_\times\,[\mathbf{G}\,\mathbf{x}_1]\,[\mathbf{x}_3]_\times \;=\; \mathbf{0}_{\{3\times 3\}}$$

Here, $[\mathbf{x}]_\times$ is the usual 'cross product' representation of a 3-component vector $\mathbf{x}$ as a skew-symmetric matrix, and the contraction $\mathbf{x}^{A_1}\mathbf{G}_{A_1}{}^{A_2A_3}$ is viewed as a 3×3 matrix $[\mathbf{G}\,\mathbf{x}_1]$. The projections along $\mathbf{x}_2^\top$ (on the left) and $\mathbf{x}_3$ (on the right) vanish identically, so again there are only 4 linearly independent equations.

Two index antisymmetrizations ('cross products') vanish only for parallel vectors, so the trilinear constraint $\mathbf{x}^{A_1}\mathbf{x}^{[A_2}\,\mathbf{G}_{A_1}{}^{B_2][B_3}\,\mathbf{x}^{A_3]} = \mathbf{0}$ also implies that for all values of the free indices $[A_2B_2]$

$$\mathbf{x}^{A_3} \;\sim\; \mathbf{x}^{A_1}\mathbf{x}^{[A_2}\,\mathbf{G}_{A_1}{}^{B_2]A_3}$$

(More precisely, for *matching* $\mathbf{x}^{A_1}$ and $\mathbf{x}^{A_2}$ the quantity $\mathbf{x}^{A_1}\mathbf{x}^{[A_2}\,\mathbf{G}_{A_1}{}^{B_2]A_3}$ can always be factorized as $\mathbf{T}^{[A_2B_2]}\,\mathbf{x}^{A_3}$ for some $\mathbf{x}^{A_i}$-dependent tensor $\mathbf{T}^{[A_2B_2]}$). By fixing suitable values of $[A_2B_2]$, these equations can be used to **transfer** points from images 1 and 2 to image 3, *i.e.* to directly predict the projection in image 3 of a 3D point whose projections in images 1 and 2 are known, without any intermediate 3D reconstruction step.

Geometrically, the trilinear constraints can be interpreted as follows. As above, $\varepsilon_{ABC}\,\mathbf{P}_a^A\mathbf{P}_b^B\,\mathbf{x}^C$ is the optical ray through $\mathbf{x}^A$ in covariant 3D coordinates. For any $\mathbf{y}^A$ the quantity $\varepsilon_{ABC}\,\mathbf{P}_a^A\mathbf{x}^B\mathbf{y}^C$ defines the 3D plane through the optical centre that projects to the image line through $\mathbf{x}^A$ and $\mathbf{y}^A$. All such planes contain the optical ray of $\mathbf{x}^A$, and as $\mathbf{y}^A$ varies the entire pencil of planes through this line is traced out. The constraint then says that for any plane through the optical ray of $\mathbf{x}^{A_2}$ and any other plane through the optical ray of $\mathbf{x}^{A_3}$, the 3D line of intersection of these planes meets the optical ray of $\mathbf{x}^{A_1}$. A little geometry shows that this implies that all three of the optical rays meet in a point, so the three pairwise epipolar constraints between the images follow from the trilinear one.

The constraint tensor $\mathbf{G}_{A_1}{}^{A_2A_3} \equiv \varepsilon_{A_1B_1C_1}\,\mathbf{I}^{B_1C_1A_2A_3}$ treats image 1 specially and there are analogous image 2 and image 3 tensors $\mathbf{G}_{A_2}{}^{A_3A_1}$ and $\mathbf{G}_{A_3}{}^{A_1A_2}$. These turn out to be linearly independent of $\mathbf{G}_{A_1}{}^{A_2A_3}$ and give further linearly independent trilinear constraints on $\mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3}$. Together, the 3 constraint tensors contain $3\times 27 = 81$ linearly independent components (including 3 arbitrary scale factors) and naïvely give $3\times 9 = 27$ scalar trilinear constraint equations, of which $3\times 4 = 12$ are linearly independent for any given triple $\mathbf{x}^{A_1}\mathbf{x}^{A_2}\mathbf{x}^{A_3}$.

However, although there are no *linear* relations between the 81 trilinear and $3\times 9 = 27$ bilinear matching tensor components for the three images, the tensors are certainly not *algebraically* independent of each other. There are many *quadratic* relations between them inherited from the structural simplicity constraints on the joint image Grassmannian tensor $\mathbf{I}^{\alpha_0\cdots\alpha_3}$. In fact, the number of algebraically independent degrees of freedom in the $\binom{3m}{4}$-component Grassmann tensor (and therefore in the complete set of matching tensor coefficients) is only $11m-15$ (*i.e.* 18 for $m=3$). Similarly, there are only $2m-3=3$ *algebraically* independent scalar constraint equations among the *linearly* independent $3\times 4 = 12$ trilinear and $3\times 1 = 3$ bilinear constraints on each matching triple of points.

One of the main advantages of the Grassmann formalism is the extent to which it clarifies the rich algebraic structure of this matching constraint system. The constraint tensors are essentially just the Grassmann coordinates of the joint image, and Grassmann coordinates are always linearly independent but quadratically redundant. Generically, three bilinear constraints or any three components of a trilinear one are enough to imply all of the remaining con-

straints for three images, although numerically and for degenerate imaging situations it turns out that the trilinear constraints are somewhat more robust than the bilinear ones [7, 3].

5.3 Quadrilinear Constraints

Finally, the quadrilinear, four image Grassmannian constraint $\mathbf{I}^{[A_1A_2A_3A_4}\,\mathbf{x}^{B_1}\mathbf{x}^{B_2}\mathbf{x}^{B_3}\mathbf{x}^{B_4]} = \mathbf{0}$ corresponds to an 8×8 basic reconstruction minor selecting two rows from each of four images. In this case the simplest form of the constraint tensor is just a direct selection of $3^4 = 81$ components of the Grassmannian itself

$$\begin{aligned}\mathbf{H}^{A_1A_2A_3A_4} &\equiv \mathbf{I}^{A_1A_2A_3A_4} \\ &= \tfrac{1}{4!}\,\mathbf{P}_a^{A_1}\mathbf{P}_b^{A_2}\mathbf{P}_c^{A_3}\mathbf{P}_d^{A_4}\,\boldsymbol{\varepsilon}^{abcd}\end{aligned}$$

Dualizing the indices from each image separately gives the quadrilinear constraint

$$\begin{aligned}\mathbf{0} &= \boldsymbol{\varepsilon}_{A_1B_1C_1}\,\boldsymbol{\varepsilon}_{A_2B_2C_2}\,\boldsymbol{\varepsilon}_{A_3B_3C_3}\,\boldsymbol{\varepsilon}_{A_4B_4C_4}\cdot \\ &\qquad\cdot\,\mathbf{H}^{B_1B_2B_3B_4}\;\mathbf{x}^{C_1}\mathbf{x}^{C_2}\mathbf{x}^{C_3}\mathbf{x}^{C_4} \\ &= \tfrac{1}{4!}\,\left(\boldsymbol{\varepsilon}_{A_1B_1C_1}\mathbf{P}_a^{B_1}\mathbf{x}^{C_1}\right)\left(\boldsymbol{\varepsilon}_{A_2B_2C_2}\mathbf{P}_b^{B_2}\mathbf{x}^{C_2}\right)\cdot \\ &\quad\cdot\left(\boldsymbol{\varepsilon}_{A_3B_3C_3}\mathbf{P}_c^{B_3}\mathbf{x}^{C_3}\right)\left(\boldsymbol{\varepsilon}_{A_4B_4C_4}\mathbf{P}_d^{B_4}\mathbf{x}^{C_4}\right)\,\boldsymbol{\varepsilon}^{abcd}\end{aligned}$$

This must hold for each of the $3^4 = 81$ values of $A_1A_2A_3A_4$. Again the constraints with A_i along the direction $\mathbf{x}^{A_i}$ for any $i = 1,\ldots,4$ vanish identically, so for any given quadruple of points there are only $2^4 = 16$ linearly independent constraints among the $3^4 = 81$ equations.

Together, these constraints say that for every possible choice of four planes, one through the optical ray of $\mathbf{x}^{A_i}$ for each $i = 1,\ldots,4$, the planes meet in a point. By fixing three of the planes and varying the fourth we immediately find that each of the optical rays passes through the point, and hence that they all meet. This brings us back to the two and three image sub-cases.

Again, there is nothing algebraically new here. The $3^4 = 81$ components of the quadrilinear constraint tensor are *linearly* independent of each other and of the $4\times 3\times 27 = 324$ trilinear and $6\times 9 = 54$ bilinear tensor components; and the $2^4 = 16$ linearly independent quadrilinear scalar constraints are *linearly* independent of each other and of the linearly independent $4\times 3\times 4 = 48$ trilinear and $6\times 1 = 6$ bilinear constraints. However there are only $11m - 15 = 29$ *algebraically* independent tensor components in total, which give $2m - 3 = 5$ *algebraically* independent constraints on each 4-tuple of points. The quadrilinear constraint is algebraically equivalent to various different combinations of two and three image constraints, and vice versa.

5.4 Further Results

The Grassmann tensor also contains the **epipoles** in the form

$$\begin{aligned}\mathbf{e}_i{}^{A_j} &\equiv \tfrac{1}{d!}\,\boldsymbol{\varepsilon}_{A_iB_iC_i}\,\mathbf{I}^{A_jA_iB_iC_i} \\ \mathbf{I}^{A_jA_iB_iC_i} &= \mathbf{e}_i{}^{A_j}\,\boldsymbol{\varepsilon}^{A_iB_iC_i}\end{aligned}$$

This exhausts the $\binom{3m}{4}$ components of the Grassmannian, so modulo a choice of scalings the Grassmannian can be reconstructed linearly from the complete set of matching tensors and epipoles.

The Grassmann structural simplicity relations $\mathbf{I}^{\alpha_0\alpha_1\alpha_2[\beta_0}\,\mathbf{I}^{\beta_1\beta_2\beta_3\beta_4]} = 0$ induce a rich set of quadratic identities between the matching tensors of up to 8 images. The simplest is just $\mathbf{F}_{A_1A_2}\mathbf{e}_1{}^{A_2} = 0$. Many more are listed in [8].

The formalism also extends to lines and other types of subspace. For any number of 2D images of 3D lines the only type of matching constraint is Hartley's trilinear one [4]. The relationships between trilinear line and point constraints emerge very clearly from this approach. One can also derive the theory of homographic images of planes (2D worlds) and matching constraints for 1D (linear) cameras in this way.

Matching constraints are closely associated with **minimal reconstruction techniques** that reconstruct world objects from the absolute minimum amount of image data. In 3D there are bilinear and trilinear minimal reconstruction techniques for points and bilinear ones for lines. Reprojection of the reconstructions gives matching tensor based methods for the **transfer** of structure between images.

Finally, given a sufficient set of matching tensors one can exhibit 'reconstruction' techniques that work directly in the joint image without reference to any world space or basis. The 'reconstructions' are somewhat implicit, but they really do contain all of the relevant structure and with a choice of basis

they reduce to more familiar coordinate-based techniques.

6 Summary

The combined homogeneous coordinates of a set of m perspective images of a 3D scene define an abstract projective **joint image space** containing a 3D projective subspace called the **joint image**. This is a faithful projective replica of the scene in image coordinates defined intrinsically by the imaging geometry. Projective reconstruction is a canonical geometric process in the joint image, requiring only a rescaling of image coordinates. A choice of basis in the joint image allows the reconstruction to be transferred to world space.

There are multilinear **matching constraints** between the images that determine whether a set of image points could be the projection of a single world point. For images of 3D points only three types of constraint exist: the bilinear epipolar one, Shashua's trilinear three-image one and a new quadrilinear four-image one. For 3D lines the only type of constraint is Hartley's trilinear three-image one.

All of the constraints fit into a single geometric object, the 4 index **joint image Grassmannian** tensor. This is an algebraic encoding of the location of the joint image. The matching constraints are linearly independent but algebraically dependent: structural constraints on the Grassmannian tensor induce a rich family of quadratic identities between them.

Appendix: Grassmann Coordinates

A k dimensional projective subspace in d-dimensions can be specified by choosing a $k+1$ element basis $\{\mathbf{u}_i^a | i = 0, \ldots, k\}$ of vectors that span it, or dually by giving a $d-k$ element basis $\{\mathbf{w}_a^i | i = k+1, \ldots, d+1\}$ of linear forms orthogonal to it (*i.e.* the subspace is $\{\mathbf{x}^a | \mathbf{w}_a^i \mathbf{x}^a = 0, i = k+1, \ldots, d+1\}$). Given a choice of basis for the embedding space, the $\mathbf{u}$'s can be thought of as the columns of a $(d+1) \times (k+1)$ rank $k+1$ matrix $\mathbf{U}$ and the $\mathbf{w}$'s as the rows of a $(d-k) \times (d+1)$ rank $d-k$ matrix $\mathbf{W}$. Up to scale, the $(k+1) \times (k+1)$ minors of $\mathbf{U}$ are exactly the components of the antisymmetric **Grassmann tensor** $\mathbf{u}^{a_0 \cdots a_k} \equiv \mathbf{u}_0^{[a_0} \cdots \mathbf{u}_k^{a_k]}$. Similarly, the $(d-k) \times (d-k)$ minors of $\mathbf{W}$ are the components of the **dual Grassmann tensor** $\mathbf{w}_{a_{k+1} \cdots a_{d+1}} \equiv \mathbf{w}^{k+1}_{[a_{k+1}} \cdots \mathbf{w}^{d+1}_{a_{d+1}]}$. By the rank conditions on $\mathbf{U}$ and $\mathbf{W}$, neither of these tensors vanish. The usual determinant-of-a-product rule implies that under a $(k+1) \times (k+1)$ linear redefinition $\mathbf{u}_i^a \rightarrow \sum_{j=0}^{k} \mathbf{u}_j^a \Lambda_i^j$ of the spanning basis $\mathbf{u}_i^a$, the components of $\mathbf{u}^{a_0 \cdots a_k}$ are simply rescaled by $\mathrm{Det}(\Lambda)$. Similarly, $\mathbf{w}_{a_{k+1} \cdots a_{d+1}}$ is invariant up to scale under $(d-k) \times (d-k)$ redefinitions of $\mathbf{w}_a^i$. A point $\mathbf{x}^a$ lies in the subspace if and only if the $(k+2) \times (d+1)$ matrix formed by appending the column vector of $\mathbf{x}^a$ to $\mathbf{U}$ is rank deficient, *i.e.* if and only if $\mathbf{u}^{[a_0 \cdots a_k} \mathbf{x}^{a_{k+1}]} = 0$. Dually, $\mathbf{x}^a$ lies in the subspace if and only if $\mathbf{w}_a^i \mathbf{x}^a = 0$ for all $i = k+1, \ldots, d+1$ and this is equivalent to $\mathbf{w}_{a_{k+1} \cdots a_d b} \mathbf{x}^b = 0$. Finally, it turns out that up to scale $\mathbf{u}^{a_0 \cdots a_k}$ and $\mathbf{w}_{a_{k+1} \cdots a_{d+1}}$ are tensor duals of one another.

In summary, *up to scale the antisymmetric Grassmann tensor* $\mathbf{u}^{a_0 \cdots a_k}$ *(or dually* $\mathbf{w}_{a_{k+1} \cdots a_{d+1}}$*) uniquely characterizes the subspace and is characterized by it, independent of the basis chosen to span the subspace.* This can be used to algebraize projective geometric relationships. For example the union (span) of two nonintersecting subspaces is just $\mathbf{u}^{[a_0 \cdots a_k} \mathbf{v}^{b_0 \cdots b_l]}$ and dually the intersection of two minimally intersecting subspaces is $\mathbf{w}_{[a_{k+1} \cdots a_{d+1}} \mathbf{x}_{b_{l+1} \cdots b_{d+1}]}$.

However, although each subspace specifies a unique antisymmetric tensor, very few tensors specify subspaces. Those that do are called **simple** because they can be factorized in the form $\mathbf{u}_0^{[a_0} \cdots \mathbf{u}_k^{a_k]}$ for some set of $\mathbf{u}_i^a$. This occurs exactly when either of the following equivalent quadratic **Grassmann simplicity relations** are satisfied

$$\begin{aligned} \mathbf{u}^{a_0 \cdots [a_k} \mathbf{u}^{b_0 \cdots b_k]} &= \mathbf{0} \\ (*\mathbf{u})_{a_{k+1} \cdots a_d c} \mathbf{u}^{c b_1 \cdots b_k} &= \mathbf{0} \end{aligned}$$

These structural relations obviously hold for any simple tensor because some vector always appears twice in an antisymmetrization. One can also show that they do not hold for any non-simple one. They restrict the $\binom{d+1}{k+1}$-dimensional space of $(k+1)$-index skew tensors to a $(k+1)(d-k)$ dimensional quadratic subvariety that exactly parameterizes the possible subspaces. *Grassmann coordinates are linearly independent but quadratically highly redundant.*

References

References

[1] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[2] O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[3] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[4] R. Hartley. Lines and points in three views – an integrated approach. In *Image Understanding Workshop*, Monterey, California, November 1994.

[5] Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. Technical Report UCB/CSD-93-772, Dept. EECS, Berkeley, California, 1993.

[6] R. Penrose and W. Rindler. *Spinors and space-time. Vol. 1, Two-spinor calculus and relativistic fields*. Cambridge University Press, 1984.

[7] A. Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 17(8):779–89, 1995.

[8] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to *Int. J. Computer Vision*.

[9] B. Triggs. A fully projective error model for visual reconstruction. Submitted to *IEEE Workshop on Representations of Visual Scenes*, Cambridge, MA, June 1995.

[10] M. Werman and A. Shashua. The study of 3D-from-2D using elimination. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 473–9, Cambridge, MA, June 1995.

A Fully Projective Error Model for Visual Reconstruction

Bill Triggs
LIFIA, INRIA Rhône-Alpes,
46 avenue Félix Viallet, 38031 Grenoble, France.
Bill.Triggs@imag.fr

Abstract

Measurement uncertainty is a recurrent concern in visual reconstruction. Image formation and 3D structure recovery are essentially projective processes that do not quite fit into the classical framework of affine least squares, so intrinsically projective error models must be developed. This paper describes initial theoretical work on a fully projective generalization of affine least squares. The result is simple and projectively natural and works for a wide variety of projective objects (points, lines, hyperplanes, and so on). The affine theory is contained as a special case, and there is also a canonical probabilistic interpretation along the lines of the classical least-squares/Gaussian/approximate log-likelihood connection. Standard linear algebra often suffices for practical calculations.

1 Introduction

For reliable reconstruction of 3D geometry from image measurements it is essential to take account of measurement uncertainties. Image formation and reconstruction are essentially projective processes and the errors they generate do not quite fit into the classical linear framework of error models such as affine least squares. In the absence of fully projective error models, uncertainty is currently handled on a rather *ad hoc* basis, often by simply feeding quasi-linear phenomenological error estimates into a general nonlinear least squares routine. This produces numerical answers, but it obscures the underlying geometric structure of the problem and makes further theoretical (*i.e.* algebraic) development impossible.

This unpublished paper dates from 1995. The work was supported by the European Community projects Second and HCM.

This paper describes initial work on a fully projective generalization of affine least squares. The resulting theory is relatively simple and projectively natural, and it extends to a wide variety of projective objects: points, lines, hyperplanes and so forth. Given a choice of 'plane at infinity', the classical affine theory is contained as a special case. There is a canonical probabilistic interpretation along the lines of the potent least-squares/Gaussian/approximate log-likelihood connection, and standard linear algebra often suffices for practical calculations.

The notion that projective geometry should be 'simpler' than affine geometry is central to this work. Several aspects of projective naturality played key rôles in the development of the theory:

- It should look simple and natural in homogeneous coordinates and work equally well at all points of a projectivized space, from the origin right out to the hyperplane at infinity.
- It should generalize easily from points to hyperplanes, lines and other projective subspaces, and perhaps even to higher-degree projective varieties like quadrics and cubics.
- For projective subspaces, it should be simply expressible in terms of Grassmann coordinates (*i.e.* 'the natural parameterization').
- It should behave naturally under point/hyperplane — and hence Grassmann/dual Grassmann — duality, and also under projective transformations.

We will use tensorial notation with all indices written out explicitly, as in [7, 9]. Most of the development will apply to general projective spaces,

but when we refer to the computer vision case of 2D projective images of a 3D projective world we will use indices $a = 0, \ldots, 3$ for homogeneous world vectors and $A = 0, 1, 2$ for homogeneous image vectors. The Einstein summation convention applies to corresponding covariant/contravariant index pairs, so for example $\mathbf{T}^a_b \, \mathbf{x}^b$ stands for matrix-vector multiplication $\sum_b \mathbf{T}^a_b \, \mathbf{x}^b$.

Probability densities will be denoted $\mathbf{dp}(\mathbf{x}^a|\text{Evidence})$ to emphasize that they are densities in $\mathbf{x}^a$ rather than functions. A **relative likelihood** is a function defined by dividing a probability density by a (sometimes implicit) prior $\mathbf{dp}(\mathbf{x}^a)$ or volume form ('uniform prior') $\mathbf{dV}$. **Log-unlikelihood** means -2 times the logarithm of a relative likelihood, defined up to an additive constant. $\boldsymbol{\chi}^2$ variables are log-unlikelihoods.

Although many specific error models have appeared in the literature there have been very few attempts to unify different aspects of the field. Zhang & Faugeras [10, 1] and Luong *et al* [3] respectively provide linearized least squares models for 3D point and line reconstruction and fundamental matrix estimation. Mohr *et al* [5] formulate multi-image reconstruction as a batch-mode nonlinear least squares problem, and more recently McLauchlan & Murray [4] describe a suboptimal but practically efficient linearized incremental framework for several types of reconstruction.

2 Homogenized Affine Least Squares

To motivate the projective model we will re-express classical least squares for affine points in homogeneous coordinates. Consider a random vector $\mathbf{x} = (x^1, \ldots, x^d)^\top$ in a d-dimensional affine space, subject to some probability distribution with mean $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{X}$. We can homogenize $\mathbf{x}$ and embed it in d dimensional projective space by adding an extra component $\mathbf{x}^0 \equiv 1$ to make a $d+1$ component homogeneous vector $\mathbf{x}^a = (1, x^1, \ldots, x^d)^\top$. The mean and covariance are neatly contained in the expectation value $\left\langle \mathbf{x}^a \mathbf{x}^b \right\rangle$:

$$\left\langle \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \begin{pmatrix} 1 & \mathbf{x}^\top \end{pmatrix} \right\rangle = \begin{pmatrix} 1 \\ \bar{\mathbf{x}} \end{pmatrix} \begin{pmatrix} 1 & \bar{\mathbf{x}}^\top \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{X} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & \bar{\mathbf{x}}^\top \\ \bar{\mathbf{x}} & \bar{\mathbf{x}}\bar{\mathbf{x}}^\top + \mathbf{X} \end{pmatrix}$$

Inverting this **homogenized covariance matrix** gives an equally simple **homogenized information matrix**:

$$\begin{pmatrix} 1 & \bar{\mathbf{x}}^\top \\ \bar{\mathbf{x}} & \bar{\mathbf{x}}\bar{\mathbf{x}}^\top + \mathbf{X} \end{pmatrix}^{-1} = \begin{pmatrix} 1 + \bar{\mathbf{x}}^\top \mathbf{X}^{-1} \bar{\mathbf{x}} & -\bar{\mathbf{x}}^\top \mathbf{X}^{-1} \\ -\mathbf{X}^{-1} \bar{\mathbf{x}} & \mathbf{X}^{-1} \end{pmatrix}$$

Finally, contracting the information matrix with $\mathbf{x}^a$ and $\mathbf{x}^b$ gives (up to an additive constant) the chi-squared/Mahalanobis distance/Gaussian exponent/approximate log-unlikelihood of $\mathbf{x}$ with respect to $\bar{\mathbf{x}}$ and $\mathbf{X}$:

$$1 + \boldsymbol{\chi}^2(\mathbf{x}|\bar{\mathbf{x}}, \mathbf{X}) = 1 + (\mathbf{x} - \bar{\mathbf{x}})^\top \mathbf{X}^{-1} (\mathbf{x} - \bar{\mathbf{x}})$$

$$= \begin{pmatrix} 1 & \mathbf{x}^\top \end{pmatrix} \begin{pmatrix} 1 + \bar{\mathbf{x}}^\top \mathbf{X}^{-1} \bar{\mathbf{x}} & -\bar{\mathbf{x}}^\top \mathbf{X}^{-1} \\ -\mathbf{X}^{-1} \bar{\mathbf{x}} & \mathbf{X}^{-1} \end{pmatrix} \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix}$$

The determinants of the homogenized covariance and information matrices are simply $\mathrm{Det}(\mathbf{X})$ and $\mathrm{Det}(\mathbf{X}^{-1})$.

The moral is that homogenization makes many Gaussian and affine least squares notions even simpler and more uniform. In fact, it is a nice way to work even when there is no question of projective space, because the parameters of the Gaussian are all kept together in one matrix. Derivations and coding become easier because equations for means fall out of those for covariances.

3 Projective Point Distributions

Now we briefly sketch the key elements of the projective least squares error model for a single projective point. For a more complete development of the theory see [8].

Consider an arbitrary probability density $\mathbf{dp}(\mathbf{x}^a)$ for an uncertain point in a d dimensional projective space $\mathcal{P}^a$. To be projectively well defined, the density must be *scale invariant*: $\mathbf{dp}(\mathbf{x}^a) = \mathbf{dp}(\lambda \mathbf{x}^a)$ for all $\mathbf{x}^a$ and all $\lambda \neq 0$. Integration against $\mathbf{dp}(\cdot)$ induces a linear expectation value operator $\langle \cdot \rangle$ on the scale-invariant functions on $\mathcal{P}^a$:

$$\langle f \rangle \equiv \int_{\mathcal{P}^a} f(\mathbf{x}^a) \, \mathbf{dp}(\mathbf{x}^a)$$

The homogenized affine analysis given above suggests that we should try to evaluate a *homogeneous covariance tensor* $\mathbf{X}^{ab} \sim \left\langle \mathbf{x}^a \mathbf{x}^b \right\rangle$, invert

it to produce a *homogeneous information tensor* $\mathbf{M}_{ab} \equiv (\mathbf{X}^{-1})_{ab}$, and then take $1 + \boldsymbol{\chi}^2(\mathbf{x}^a|\mathbf{X}^{ab}) \sim \mathbf{M}_{ab}\mathbf{x}^a\mathbf{x}^b$ as a measure of normalized squared error or approximate log-unlikelihood. Unfortunately, this can not quite work as it stands because $\langle\cdot\rangle$ is only defined for *scale invariant* functions of $\mathbf{x}^a$ and the moment monomials $\mathbf{x}^{a_1}\cdots\mathbf{x}^{a_k}$ all depend on the scale of $\mathbf{x}^a$. On a general projective space there is no canonical way to fix this scale, so classical means and covariances are simply not defined.

This problem can be resolved by introducing an auxiliary normalization tensor $\mathbf{N}_{ab}$ and homogenizing with respect to it, so that quantities of the form $\left(\mathbf{M}_{ab}\mathbf{x}^a\mathbf{x}^b\right)$ are replaced by homogeneous scale invariant quantities $\left(\mathbf{M}_{ab}\mathbf{x}^a\mathbf{x}^b\right) / \left(\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d\right)$. We will call such functions **biquadrics** because their level surfaces are quadric: $\left(\mathbf{M}_{ab}\mathbf{x}^a\mathbf{x}^b\right) / \left(\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d\right) = \lambda$ implies $(\mathbf{M}_{ab} - \lambda\,\mathbf{N}_{ab})\,\mathbf{x}^a\mathbf{x}^b = 0$. As an example, the affine normalization condition $x^0 = 1$ can be relaxed if we divide through by $\mathbf{N}^{\text{aff}}_{ab}\mathbf{x}^a\mathbf{x}^b = (\mathbf{p}^\infty_a\mathbf{x}^a)^2 = (x^0)^2$, where $\mathbf{N}^{\text{aff}}_{ab} \equiv \mathbf{p}^\infty_a\mathbf{p}^\infty_b = \left(\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right)$ and $\mathbf{p}^\infty_a = (1\ 0\ \ldots\ 0)$ is the plane at infinity, At first sight the normalizer simply provides a fiducial scaling $\mathbf{N}_{ab}\mathbf{x}^a\mathbf{x}^b = 1$ with respect to which the error model can be defined, but ultimately $\mathbf{N}$ is on a par with $\mathbf{M}$ and tends to play an equally active rôle in the theory.

3.1 Basic Equations

Given a projective probability distribution $\mathbf{dp}(\mathbf{x}^a)$ and an arbitrary symmetric positive semidefinite **normalization tensor** $\mathbf{N}_{ab}$ on a projective space $\mathcal{P}^a$, we can define the **homogeneous covariance tensor**

$$\mathbf{X}^{ab} \equiv \left\langle \frac{\mathbf{x}^a\mathbf{x}^b}{\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d} \right\rangle$$

Note that $\mathbf{X}$ is symmetric, positive semidefinite and independent of the scale of $\mathbf{x}^a$, but it does depend on the value and scale of $\mathbf{N}$. If $\mathbf{N}$ has null directions it should be compatible with $\mathbf{dp}(\cdot)$ in the sense that the above expectation value is finite, *i.e.* the distribution should not have too much weight in the vicinity of the null space of $\mathbf{N}$. Since $\langle\cdot\rangle$ is linear, if $\mathbf{dp}(\cdot)$ is correctly normalized we have the following **covariance normalization consistency condition** on $\mathbf{X}$

$$\mathbf{N}_{ab}\mathbf{X}^{ab} = \left\langle \frac{\mathbf{N}_{ab}\mathbf{x}^a\mathbf{x}^b}{\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d} \right\rangle = \langle 1\rangle = 1$$

Viewing $\mathbf{X}$ and $\mathbf{N}$ as matrices, this can be written $\mathrm{Trace}(\mathbf{NX}) = 1$. If $\mathbf{dp}(\cdot)$ is not correctly normalized, we can normalize by dividing through by $\mathbf{N}_{ab}\mathbf{X}^{ab} = \langle 1\rangle \neq 1$. The normalized covariance tensor is then

$$\left\langle \frac{\mathbf{x}^a\mathbf{x}^b}{\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d} \right\rangle / \langle 1\rangle = \frac{\mathbf{X}^{ab}}{\mathbf{N}_{cd}\mathbf{X}^{cd}}$$

Usually, one can arrange to work with normalized quantities and ignore the scale factor, *i.e.* $\mathbf{N}_{ab}\mathbf{X}^{ab} = 1$.

By analogy with the homogenized affine case and assuming for the moment that $\mathbf{X}$ is nonsingular, we can invert it to produce a **homogeneous information tensor** $\mathbf{M}_{ab} \equiv (\mathbf{X}^{-1})_{ab}$ and define a corresponding **homogeneous** $1 + \boldsymbol{\chi}^2$ **function**

$$1 + \boldsymbol{\chi}^2(\mathbf{x}^a|\mathbf{X},\mathbf{N}) \equiv \frac{\mathbf{M}_{ab}\mathbf{x}^a\mathbf{x}^b}{\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d}$$

It is not immediately obvious that these definitions make sense, but one can argue [8] that they do in fact lead to a coherent theory of approximate least squares estimation. Two key approximations are required, both of which are *exact* in the affine case and generally accurate whenever the uncertainty is small compared to the scale of projective space. (And it is only in the limit of small uncertainty that *any* least squares technique becomes a good approximation to the more rigorous maximum relative likelihood theory).

As in the affine case, it is often useful to regard the information as the primitive quantity and derive the covariance from it. The quadratic (Gaussian) exponent $\boldsymbol{\chi}^2(\mathbf{x}|\bar{\mathbf{x}},\mathbf{X}) = (\mathbf{x}-\bar{\mathbf{x}})^\top\mathbf{X}^{-1}(\mathbf{x}-\bar{\mathbf{x}})$ is the keystone of affine estimation theory because it is the leading term in the **central moment expansion** of an arbitrary distribution. The **central limit theorem** (which guarantees the asymptotic dominance of this term given 'reasonable' behaviour of the underlying distributions) is the ultimate probabilistic justification for affine least squares techniques.

Similarly, **biquadric** exponents $1 + \boldsymbol{\chi}^2(\mathbf{x}^a|\mathbf{X},\mathbf{N}) \equiv \left(\mathbf{M}_{ab}\mathbf{x}^a\mathbf{x}^b\right) / \left(\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d\right)$ lie at the heart of projective least squares. In particular,

they are likely to be good asymptotic approximations to arbitrary projective log-unlikelihood functions, so that estimation theory based on them should 'work' in much the same way that conventional least squares 'works' in affine space. Given this, the uncertainties of projective points can be modelled with **biquadric probability distributions**

$$\mathbf{dp}(\mathbf{x}^a) \sim \exp\left(-\frac{1}{2}\frac{\mathbf{M}_{ab}\mathbf{x}^a\mathbf{x}^b}{\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d}\right)\mathbf{dV}$$

much as affine uncertainties can be modelled with Gaussians.

Several approximations are required here. Firstly, there is no canonical volume form $\mathbf{dV}$ on projective space, so it is necessary to make an 'arbitrary but reasonable' choice of this 'uniform prior'. This is annoying, but it is not specifically a problem with projective least squares: implicitly or explicitly, *every* least squares theory makes such choices. The mere existence of a uniform volume form on affine space does not make it a universally acceptable prior.

Secondly, biquadric distributions are somewhat less tractable than Gaussian ones and (except in the limit of affine normalization) there does not seem to be a closed form for their integrals. This means that we do not know the exact functional form of the relation $\mathbf{X} = \mathbf{X}(\mathbf{M}, \mathbf{N})$ between the covariance and the information and normalization. However with an appropriate choice of projective basis the integral can be approximated by a Gaussian [8], with the result that *for properly normalized distributions* the 'classical' homogenized affine formula $\mathbf{X} \approx \mathbf{M}^{-1}$ is still approximately valid. Here properly normalized means that the covariance normalization condition $\mathbf{N}_{ab}\mathbf{X}^{ab} = 1$ holds for $\mathbf{X} \equiv \mathbf{M}^{-1}$, so that the distribution in $\mathbf{M}$ and $\mathbf{N}$ is approximately normalized in the sense that $\langle 1 \rangle \approx 1$.

It is often necessary to normalize an unnormalized biquadric distribution. Rescaling the density function amounts to **shifting** the information $\mathbf{M}$ by a multiple of $\mathbf{N}$: $\mathbf{M} \to \mathbf{M} - \lambda\mathbf{N}$. We will say that $\mathbf{M}$ is **correctly shifted** if $\mathbf{M}^{-1}$ has the correct normalization to be a covariance: $\mathbf{N}_{ab}(\mathbf{M}^{-1})^{ab} = 1$. The correct shift factor can be found by solving the nonlinear **normalizing shift equation**

$$\mathbf{N}_{ab}\left((\mathbf{M} - \lambda\mathbf{N})^{-1}\right)^{ab} = 1$$

This amounts to a polynomial of degree $\mathrm{Rank}(\mathbf{N})$ in λ, linear in the case of affine normalization. The desired solution is the smallest real root, which can be roughly approximated by the **approximate shift solution**

$$\lambda \approx \left(\mathbf{N}_{ab}(\mathbf{M}^{-1})^{ab}\right)^{-1} - 1$$

The two main approximations required to make projective least squares 'work' are the covariance estimate $\mathbf{X} \approx \mathbf{M}^{-1}$ and the approximate shift solution. Both are *exact* for affine normalization and generally accurate for small uncertainties, but neither is very good for distributions that spread across the entire width of projective space. However, least squares is not really suitable for weak evidence (wide distributions) in any event. It makes too many assumptions about the uniformity of priors and the asymptotic shapes of distributions to be competitive with the more rigorous maximum relative likelihood theory in this case. Its main strengths are simplicity and asymptotic correctness in the limit where many moderate pieces of evidence combine to make a single strong one. And it is in exactly this limit that the additional approximations made by projective least squares become accurate.

To define a meaningful distribution, $\mathbf{M}$ and $\mathbf{N}$ need to be **non-negative**, but it is practically useful to allow them to have null directions. To guarantee the normalization condition $\mathbf{N}_{ab}(\mathbf{M}^{-1})^{ab} = 1$, we will impose a **null space compatibility condition**: the null space of $\mathbf{M}$ must be contained in that of $\mathbf{N}$. This ensures that any pseudo-inverse of a singular $\mathbf{M}$ can be used to evaluate $\mathbf{N}_{ab}(\mathbf{M}^{-1})^{ab}$ (it makes no difference which). However, the covariance tensor $\mathbf{X} \approx \mathbf{M}^{-1}$ is only defined for nonsingular $\mathbf{M}$.

3.2 Normalizations

If we take $\mathbf{N}$ to be the **affine normalization** $\mathbf{N}^{\mathrm{aff}}_{ab} \equiv \mathbf{p}^\infty_a\mathbf{p}^\infty_b = \left(\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right)$ where $\mathbf{p}^\infty_a = (1\ 0\ \ldots\ 0)$ is the hyperplane at infinity, the biquadric distribution reduces to the homogenized affine case we started from. In this case the covariance normalization condition is simply $\mathbf{X}^{00} = \mathbf{N}^{\mathrm{aff}}_{ab}\mathbf{X}^{ab} = 1$ and (to the extent that the underlying distribution is well modelled by a Gaussian) the homogeneous $1 + \boldsymbol{\chi}^2$ function is one plus a genuine classical $\boldsymbol{\chi}^2$ variable.

On the other hand, if $\mathbf{N}$ is taken to be the identity matrix in some projective basis we have a **spherical normalization** $\mathbf{N}_{ab}\mathbf{x}^a\mathbf{x}^b = \sum_{a=0}^{d}(x^a)^2 = 1$ and the error model reduces to a spherical analogue

of linear least squares, with 'distances' measured by sines of angles on the unit sphere. The two normalizations coincide for points near the origin but differ significantly near the hyperplane at infinity. The affine normalization vanishes on the plane at infinity and points there are infinitely improbable, whereas the spherical normalization is regular and well behaved for all points, including those at infinity.

These are just two of the infinitely many possible choices for $\mathbf{N}$. There is no universally 'correct' or 'canonical' normalizer. Ideally, $\mathbf{N}$ should be chosen to reflect the mechanism that generates the experimental uncertainty, although in practice numerical expediency is also a factor.

With the spherical normalization it is natural to take an eigenvalue expansion of $\mathbf{X}$ in an 'orthonormal' projective basis. The mode (maximum likelihood value) of the distribution is the maximum-eigenvalue eigenvector of $\mathbf{X}$ and the remaining eigenvectors give the principal axes of the uncertainty ellipsoids. Small eigenvalues correspond to directions with little uncertainty, while for large ones (those near the modal eigenvalue) the distribution spreads across the entire width of projective space. For the 'uniform' distribution, $\mathbf{X} = \frac{1}{d+1}\mathbf{I}$.

More generally, given any $\mathbf{M}$ and $\mathbf{N}$ there is always some projective basis in which they are in **canonical form**, *i.e.* simultaneously diagonal with $\mathbf{N}$ having entries $+1$ or 0. In this basis the global minimum of $1 + \boldsymbol{\chi}^2$ is at the minimum eigenvalue eigenvector of $\mathbf{M}$ along a '1' direction of $\mathbf{N}$, and a correctly normalized distribution has $\sum 1/\lambda_i = 1$ where the sum is over the inverse eigenvalues of $\mathbf{M}$ along '1' directions of $\mathbf{N}$.

3.3 Homogeneous Chi-Squared

Except in the case of affine normalization and an underlying Gaussian distribution, the homogeneous $\boldsymbol{\chi}^2$ variable is unlikely to have a classical $\boldsymbol{\chi}^2$ distribution. However, the "$\boldsymbol{\chi}^2$" variables used in statistics seldom *do* have exact $\boldsymbol{\chi}^2$ distributions and that does not stop them being useful error measures. Several familiar properties of the traditional $\boldsymbol{\chi}^2$ do continue to hold. Our $\boldsymbol{\chi}^2$ is nonnegative ($1+\boldsymbol{\chi}^2 \geq 1$), and for nonsingular $\mathbf{M}$ its expectation value is the number of independent degrees of freedom, *i.e.* the dimension d of the projective space:

$$\begin{aligned}\left\langle 1+\boldsymbol{\chi}^2(\mathbf{x}^a)\right\rangle &= \mathbf{M}_{ab}\left\langle \frac{\mathbf{x}^a\mathbf{x}^b}{\mathbf{N}_{cd}\mathbf{x}^c\mathbf{x}^d}\right\rangle \\ &= \mathbf{X}^{-1}_{ab}\mathbf{X}^{ab} = d+1\end{aligned}$$

Moreover, we have already seen that — in analogy to the error ellipsoids of the classical $\boldsymbol{\chi}^2$ — the level surfaces of $1+\boldsymbol{\chi}^2$ are always quadric. Near a minimum of $1+\boldsymbol{\chi}^2$ these surfaces will be ellipsoidal, but further away they may cut the plane at infinity and hence appear hyperboloidal rather than ellipsoidal.

3.4 Homogeneous Taylor Approximation

To get an idea of why biquadric functions should appear in projective least squares, consider an arbitrary smooth scale invariant function on projective space: $f(\mathbf{x}^a) = f(\lambda\mathbf{x}^a)$. $f(\cdot)$ can be approximated with a conventional Taylor series at a point, but this is not very satisfactory as the resulting truncated Taylor polynomials are not exactly scale invariant and depend on the scale of the homogeneous vector at which the derivatives are evaluated. What is needed is a *projectively invariant* analogue of the Taylor series. Once again homogenization with respect to a normalizer $\mathbf{N}_{ab}$ makes this possible.

Consider the scale-invariant function

$$f(\mathbf{x}^a) = \frac{\mathbf{M}_{a_1a_2\cdots a_{2k}}\mathbf{x}^{a_1}\mathbf{x}^{a_2}\cdots\mathbf{x}^{a_{2k}}}{(\mathbf{N}_{ab}\mathbf{x}^a\mathbf{x}^b)^k}$$

where $\mathbf{M}$ and $\mathbf{N}$ are arbitrary symmetric tensors. Multiplying out and differentiating $2k$ times using the usual iterated chain rule gives

$$\begin{aligned}\mathbf{M}_{a_1a_2\cdots a_{2k}} &= \frac{1}{(2k)!}\frac{\partial^{2k}\left[(\mathbf{N}_{ab}\mathbf{x}^a\mathbf{x}^b)^k\cdot f(\mathbf{x}^a)\right]}{\partial\mathbf{x}^{a_1}\cdots\partial\mathbf{x}^{a_{2k}}} \\ &= \sum_{j=0}^{2k}\frac{\partial^j f}{\partial\mathbf{x}^{(a_1}\cdots\partial\mathbf{x}^{a_j}}\cdot\frac{\partial^{2k-j}(\mathbf{N}_{ab}\mathbf{x}^a\mathbf{x}^b)^k}{\partial\mathbf{x}^{a_{j+1}}\cdots\partial\mathbf{x}^{a_{2k})}}\end{aligned}$$

Here, $(a_1\cdots a_{2k})$ means 'take the symmetric part'. The factorial weights of the familiar iterated chain rule are subsumed by this symmetrization.

This formula gives $\mathbf{M}$ in terms of $\mathbf{N}$ and the first $2k$ derivatives of $f(\cdot)$. Now choose any $\mathbf{N}$ and let $f(\cdot)$ stand for an arbitrary scale-invariant function. The resulting $\mathbf{M}$ defines a function $(\mathbf{M}_{a_1\cdots a_{2k}}\mathbf{x}^{a_1}\cdots\mathbf{x}^{a_{2k}})/(\mathbf{N}_{ab}\mathbf{x}^a\mathbf{x}^b)^k$ that is guaranteed to agree with $f(\cdot)$ to order $2k$ at $\mathbf{x}^a$. We will

say that $\mathbf{M}$ and $\mathbf{N}$ define a $(2k)^{th}$-order **homogeneous Taylor approximation** to $f(\cdot)$ at $\mathbf{x}^a$. The 'Taylor coefficients' pack neatly into the single homogeneous tensor $\mathbf{M}_{a_1 \cdots a_{2k}}$. For example adding a constant to $f(\cdot)$ amounts to adding a multiple of $\mathbf{N}_{(a_1 a_2} \cdots \mathbf{N}_{a_{2k-1} a_{2k})}$ to $\mathbf{M}$. With affine normalization $\mathbf{N} \equiv \mathbf{N}^{\text{aff}}$, the homogeneous Taylor series reduces to the usual inhomogeneous affine one at $\mathbf{x} = 0$.

In the present case we are mainly interested in approximating projective log-unlikelihood functions to second order near their peaks, by analogy with the Gaussian approximation to the peak of an affine distribution. The second order homogeneous Taylor approximation is a biquadric with

$$\mathbf{M}_{ab} = \frac{1}{2} \cdot (\mathbf{N}_{cd} \mathbf{x}^c \mathbf{x}^d) \cdot \frac{\partial^2 f}{\partial \mathbf{x}^a \partial \mathbf{x}^b} + \frac{\partial f}{\partial \mathbf{x}^a} \cdot \mathbf{N}_{bc} \mathbf{x}^c + \frac{\partial f}{\partial \mathbf{x}^b} \cdot \mathbf{N}_{ac} \mathbf{x}^c + \mathbf{N}_{ab} \cdot f$$

4 Projective Least Squares for Points

We are finally ready to describe how projective least squares can be used to estimate the position of an uncertain projective point. Suppose we have collected several independent estimates of the point's position that can be summarized by a set of biquadric distributions

$$1 + \boldsymbol{\chi}^2(\mathbf{x}^a \,|\, \text{Evidence}_i) = \frac{\mathbf{M}^{(i)}_{ab} \mathbf{x}^a \mathbf{x}^b}{\mathbf{N}^{(i)}_{cd} \mathbf{x}^c \mathbf{x}^d} \quad i = 1, \ldots, k$$

Just as one might summarize the uncertainty of an experimental measurement in affine space by specifying its mean and covariance, the uncertainty of a projective measurement can be summarized by a homogeneous information tensor $\mathbf{M}$ (or alternatively by the covariance $\mathbf{X} = \mathbf{M}^{-1}$). The corresponding normalization tensor $\mathbf{N}$ should be chosen to reflect the source of the uncertainty. For example, in computer vision a spherical normalization might be appropriate for uncertainty in the 3D angular position of an incoming visual ray relative to the camera, whereas affine normalization would probably be a better model for errors due mainly to uncertainty in the measured projection of the ray on the flat image plane (*e.g.* quantization error). However when the uncertainty is small the choice of normalization is not too critical.

Since the biquadric $1 + \boldsymbol{\chi}^2$ functions represent log-unlikelihoods, the proper way to combine them into a single estimate of the position of $\mathbf{x}^a$ is to add them and then correct the constant offset term to normalize the combined distribution. First consider the **commensurable** case in which all of the normalizations $\mathbf{N}^{(i)} = \mathbf{N}$ are identical. The sum of log-unlikelihoods reduces to a sum of information tensors, as in the affine theory:

$$1 + \sum_{i=1}^{k} \boldsymbol{\chi}^2(\mathbf{x}^a \,|\, \text{Evidence}_i) = \frac{\mathbf{M}_{ab} \mathbf{x}^a \mathbf{x}^b}{\mathbf{N}_{cd} \mathbf{x}^c \mathbf{x}^d}$$

where

$$\mathbf{M}_{ab} \equiv \sum_{i=1}^{k} \mathbf{M}^{(i)}_{ab} - (k-1)\, \mathbf{N}_{ab}$$

The term $(k-1)\,\mathbf{N}$ prevents the '1's of the $1 + \boldsymbol{\chi}^2$ terms from accumulating, but a further correction to the shift of $\mathbf{M}$ is still needed. This can be found by solving the normalizing shift equation either exactly or approximately. The shifted $\mathbf{M}$ then defines a correctly normalized posterior distribution for $\mathbf{x}^a$ given all the evidence, and its inverse $\mathbf{X} = \mathbf{M}^{-1}$ gives the covariance in the usual way. The mode (maximum likelihood estimate) for $\mathbf{x}^a$ is the global minimum of the biquadric, *i.e.* the minimum eigenvalue eigenvector of $\mathbf{M}$ along a non-null direction of $\mathbf{N}$. The shift correction step is dispensable if only the mode is required.

Now consider the **incommensurable** case in which all of the normalizers $\mathbf{N}^{(i)}$ are different. This case is much less tractable. In general the combined log-unlikelihood is a complicated rational function and analytical or numerical approximations are required.

Many nonlinear optimization techniques can be used to find the mode. One possible way to proceed is to make a commensurable reapproximation of the combined distribution by choosing some suitable common normalization $\mathbf{N}$ and approximating each log-unlikelihood to second order with a biquadric in $\mathbf{N}$. This is straightforward except for the choice of the point(s) at which the approximations are to be based. To ensure self-consistency, the log-unlikelihoods should ideally be expanded about

the true mode of the combined distribution. Since this is not known until the end of the calculation, it is necessary to start with approximations based at some sensible estimate of the mode (or perhaps at the mode of each distribution), find the resulting approximate combined mode, and if necessary iterate, at each step basing a new approximation at the latest mode estimate. Each iteration must accumulate a new approximate unshifted information tensor $\mathbf{M}$ from the component distributions and find its minimum eigenvalue eigenvector (the updated estimate of the combined mode). There is no need adjust the shift of $\mathbf{M}$ until the end of the calculation. Once the mode has been found, a second order biquadric reapproximation gives an estimate of the combined information and covariance.

There is no guarantee that this nonlinear procedure will converge correctly. Indeed, combinations of incommensurable distributions are often multi-modal, although the secondary peaks are usually negligible unless there is strongly conflicting evidence. However preliminary experiments suggest that convergence is reasonable in some realistic cases. A possible explanation for this is the fact that biquadrics are typically convex within quite a wide radius of their global minimum. They become non-convex near their non-minimal eigenvectors, but these critical points are usually far from the minimum in the standard projective basis unless $\mathbf{N}$ is particularly 'squashed'.

It might be suggested that the need to resort to approximations in the incommensurable case is a flaw of the projective least squares method, but that is not quite fair. It arises because the biquadric form is significantly richer than the Gaussian one, and even 'linear' least squares produces nonlinear equations in all but the simplest situations (*e.g.* orthogonal regression, *c.f.* section 6). In fact, except for problems with the nonlinear normalizing shift equation, the projective model is not significantly less tractable than the affine one. And even for incommensurable distributions, projective least squares provides an attractive intermediate analytical form for problems that might otherwise have produced completely 'opaque' end-to-end numerical formulations.

5 Behaviour under Projections

Now we discuss the behaviour of projective least squares under projective mappings. First consider a general situation in which some event $\mathbf{x}$ 'causes' an event $\mathbf{y}$ in the sense that $\mathbf{y} = \mathbf{f}(\mathbf{x})$ for some function $\mathbf{f}(\cdot)$, and $\mathbf{y}$ in turn gives rise to some measured evidence E. The conditional independence of E on $\mathbf{x}$ given $\mathbf{y}$ results in the classical Bayesian formula

$$\frac{\mathbf{dp}(\mathbf{x} \,|\, E)}{\mathbf{dp}(\mathbf{x})} = \frac{\mathbf{dp}(\mathbf{y} \,|\, E)}{\mathbf{dp}(\mathbf{y})}\bigg|_{\mathbf{y}=\mathbf{f}(\mathbf{x})}$$

which says that E augments the prior likelihood $\mathbf{dp}(\mathbf{x})$ of $\mathbf{x}$ to the same degree that it enhances that of $\mathbf{y} = \mathbf{f}(\mathbf{x})$. In other words, the relative likelihood function on $\mathbf{y}$-space simply pulls back to the correct relative likelihood function on $\mathbf{x}$-space under $\mathbf{f}(\cdot)$. If several $\mathbf{x}$ are mapped to the same $\mathbf{y}$, their relative weightings are determined by the prior $\mathbf{dp}(\mathbf{x})$.

If $\mathbf{f}(\cdot)$ has unknown internal parameters μ, *i.e.* $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mu)$, the data space $\mathbf{x}$ can be extended to include these and the above $\mathbf{dp}(\mathbf{x} \,|\, \cdot)$ factors become $\mathbf{dp}(\mathbf{x}, \mu \,|\, \cdot)$. Integrating over all possible values of $\mathbf{x}$ and applying the conditional probability definition $\mathbf{dp}(\mathbf{x}, \mu) = \mathbf{dp}(\mathbf{x} \,|\, \mu) \cdot \mathbf{dp}(\mu)$ gives

$$\begin{aligned}
\frac{\mathbf{dp}(\mu \,|\, E)}{\mathbf{dp}(\mu)} &= \int_{\mathbf{x}} \frac{\mathbf{dp}(\mathbf{x}, \mu \,|\, E)}{\mathbf{dp}(\mu)} \\
&= \int_{\mathbf{x}} \mathbf{dp}(\mathbf{x} \,|\, \mu) \cdot \frac{\mathbf{dp}(\mathbf{x}, \mu \,|\, E)}{\mathbf{dp}(\mathbf{x}, \mu)} \\
&= \int_{\mathbf{x}} \mathbf{dp}(\mathbf{x} \,|\, \mu) \cdot \frac{\mathbf{dp}(\mathbf{y} \,|\, E)}{\mathbf{dp}(\mathbf{y})}\bigg|_{\mathbf{y} = \mathbf{f}(\mathbf{x}, \mu)}
\end{aligned}$$

This says that the posterior likelihood for μ is proportional to the total probability for *any* corresponding $\mathbf{x}$ to give the observation via $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mu)$. In other words the log-unlikelihood of μ given E is proportional to the logarithmic 'shift factor' required to normalize the distribution of $\mathbf{x}$ given μ and E.

The above analysis applies directly to a projective mapping $\mathbf{x}^a \rightarrow \mathbf{y}^A = \mathbf{P}^A_a \mathbf{x}^a$ between projective spaces $\mathcal{P}^a$ and $\mathcal{P}^A$. If we assume that the relative likelihood on $\mathcal{P}^A$ can be approximated by a biquadric $\{\mathbf{M}_{AB}, \mathbf{N}_{AB}\}$ and that the prior on $\mathcal{P}^a$ is sufficiently 'uniform', the pulled back density on $\mathcal{P}^a$

is the biquadric

$$\mathbf{dp}(\mathbf{x}\,|\,\text{Evidence}) \approx \exp\left(-\frac{1}{2}\frac{(\mathbf{M}_{AB}\,\mathbf{P}_a^A\,\mathbf{P}_b^B)\,\mathbf{x}^a\,\mathbf{x}^b}{(\mathbf{N}_{CD}\,\mathbf{P}_c^C\,\mathbf{P}_d^D)\,\mathbf{x}^c\,\mathbf{x}^d}\right)\,\mathbf{dV}$$

In matrix notation, the information $\mathbf{M}$ and normalization $\mathbf{N}$ are pulled back respectively to $\mathbf{P}^\top\mathbf{M}\mathbf{P}$ and $\mathbf{P}^\top\mathbf{N}\mathbf{P}$. The preservation of the biquadric functional form under projective transformations implies that image space error models are directly pulled back to source space ones. However it should also be clear that there is little hope of obtaining commensurable distributions when combining observations pulled back from distinct image spaces $\mathcal{P}^{A_1},\ldots,\mathcal{P}^{A_k}$: the pulled-back normalizations $\mathbf{N}_{A_iB_i}\mathbf{P}_a^{A_i}\mathbf{P}_b^{B_i}$ will usually all be different.

In general the pulled-back $\mathbf{M}$ needs to be shifted by a multiple of the pulled-back $\mathbf{N}$ to produce a correctly normalized probability density on $\mathcal{P}^a$. The shift required is proportional to the logarithm of the total probability for *any* point in $\mathcal{P}^a$ to project to the observation, and hence depends on $\mathbf{P}_a^A$. As mentioned above, if the transformation is uncertain the posterior log-unlikelihood for a particular value $\mathbf{P}_a^A$ given the observation $\{\mathbf{M}_{AB},\mathbf{N}_{AB}\}$ is proportional to the shift $\lambda(\mathbf{P}_a^A)$ required to normalize the pulled-back distribution. In the next section we will use this to derive estimation techniques for uncertain projective subspaces, but for the remainder of this section we assume that $\mathbf{P}_a^A$ is a fixed known transformation.

Now let us examine the characteristics of the pulled-back distributions a little more closely. If $\mathbf{P}_a^A$ is a projective isomorphism — a nonsingular mapping between spaces of the same dimension, possibly from $\mathcal{P}^a$ to itself — its effect is analogous to that of a projective change of basis and there are no essentially new features.

If $\mathbf{P}_a^A$ is a nonsingular *injection* — *i.e.* a one-to-one mapping of $\mathcal{P}^a$ onto a projective subspace of $\mathcal{P}^A$ — the pulled-back likelihood is isomorphic to the restriction of the parent likelihood to the range subspace in $\mathcal{P}^A$. The only new feature is that the injected subspace may happen to 'miss' the mode of the parent distribution by a substantial margin, so that the pulled back likelihood has a shape and range of values much attenuated compared to those of the parent function on $\mathcal{P}^A$.

Finally, consider the case where $\mathbf{P}_a^A$ is a singular surjection onto a projective space of lower dimension. In this case each point of $\mathcal{P}^A$ has a nontrivial 'preimage' in $\mathcal{P}^a$ (*i.e.* the projective subspace of $\mathcal{P}^a$ that projects onto it), and $\mathcal{P}^a$ also necessarily contains a null subspace of points that project to nothing at all: $\mathbf{P}_a^A\mathbf{x}^a = 0$. The pulled-back likelihood is constant on each preimage space but is undefined on the null space as the pulled back $\mathbf{M}$ and $\mathbf{N}$ both vanish there. The pulled back equi-probability surfaces are degenerate quadrics with singularities on the null space, and generally look more like elliptical cones than ellipsoids.

The singular surjective situation occurs for the usual 3D→2D perspective projection in computer vision. In that case the null space is the centre of projection, the preimage spaces are the optical rays, and the equi-probability surfaces — the sets of world points that are equally likely to have produced the given image measurement — are elliptical cones centred on the centre of projection and generated by the optical rays, that project to the experimental error ellipses in the image plane. The considerable representative power of the projective least squares framework is illustrated by its ability to deal with error models for perspective projection out-of-hand.

It was to accommodate surjective projections that we insisted on allowing $\mathbf{M}$ to be *semi*-definite. Note that the null space compatibility condition is maintained: if the null space of $\mathbf{M}_{AB}$ is a subset of that of $\mathbf{N}_{AB}$, the same is true of the pulled-back tensors $\mathbf{M}_{AB}\mathbf{P}_a^A\mathbf{P}_b^B$ and $\mathbf{N}_{AB}\mathbf{P}_a^A\mathbf{P}_b^B$. The normalization condition $\mathbf{N}_{AB}(\mathbf{M}^{-1})^{AB} = 1$ (with $\mathbf{M}^{-1}$ interpreted as a pseudo-inverse) is also preserved under surjective pull-backs, so the shift factor of $\mathbf{M}$ does not usually need to be corrected in this case.

6 Subspace Estimation

The results of the previous section can be used to develop projective least squares error models for projective subspaces. Given a number of uncertain points, we are interested in 'fitting' a projective subspace to them and estimating its uncertainty.

Suppose we have measured a single point $\mathbf{x}^a$, whose uncertainty is characterized by a biquadric distribution in $\mathbf{M}_{ab}$ and $\mathbf{N}_{ab}$. A k dimensional projective subspace in d dimensions can be specified by choosing a set of $k+1$ independent points that span it, *i.e.* by giving a $(d+1)\times(k+1)$ rank

$k+1$ matrix $\mathbf{U}_A^a$ whose columns span the subspace ($A = 0, \ldots, k$). $\mathbf{U}_A^a$ can be thought of as a non-singular projective injection from an abstract k dimensional projective space $\mathcal{P}^A$ to $\mathcal{P}^a$. As discussed in the previous section, if $\mathbf{U}$ is uncertain its relative likelihood given the observation $\{\mathbf{M}, \mathbf{N}\}$ is proportional to the total probability in the subspace it generates, and hence to the total probability in the pulled back distribution on $\mathcal{P}^A$. In fact, up to an additive constant the log-unlikelihood of $\mathbf{U}$ given $\{\mathbf{M}, \mathbf{N}\}$ is precisely the shift factor $\lambda(\mathbf{U}^\top\mathbf{M}\mathbf{U}, \mathbf{U}^\top\mathbf{N}\mathbf{U})$ required to normalize the pulled back distribution $\{\mathbf{U}^\top\mathbf{M}\mathbf{U}, \mathbf{U}^\top\mathbf{N}\mathbf{U}\}$:

$$1 + \boldsymbol{\chi}^2(\mathbf{U} \,|\, \mathbf{M}, \mathbf{N}) \overset{+\text{const}}{\approx} 1 + \lambda(\mathbf{U}^\top\mathbf{M}\mathbf{U}, \mathbf{U}^\top\mathbf{N}\mathbf{U})$$

At this point our approximate shift solution $1 + \lambda(\mathbf{M}, \mathbf{N}) \approx \text{Trace}^{-1}(\mathbf{N}\mathbf{M}^{-1})$ comes into its own. Without a tractable analytic approximation to $\lambda(\mathbf{U}^\top\mathbf{M}\mathbf{U}, \mathbf{U}^\top\mathbf{N}\mathbf{U})$ it would be impossible to develop explicit methods for the least squares fitting of subspaces. The abstract theory would still exist, but there would be no closed-form formulae. Adopting this approximation we have the remarkably simple estimate

$$\begin{aligned} &1 + \boldsymbol{\chi}^2(\mathbf{U} \,|\, \mathbf{M}, \mathbf{N}) \\ &\quad \overset{+\text{const}}{\approx} \text{Trace}^{-1}\left(\mathbf{U}^\top\mathbf{N}\mathbf{U} \cdot (\mathbf{U}^\top\mathbf{M}\mathbf{U})^{-1}\right) \\ &\quad = \text{Trace}^{-1}\left(\mathbf{N} \cdot \mathbf{U}(\mathbf{U}^\top\mathbf{M}\mathbf{U})^{-1}\mathbf{U}^\top\right) \end{aligned}$$

Note the invariance of this formula under redefinitions $\mathbf{U} \to \mathbf{U}\mathbf{A}$ of the spanning basis of the subspace, where $\mathbf{A}$ is any nonsingular $(k+1) \times (k+1)$ matrix.

Dually, a subspace can be specified as the intersection of $d-k$ hyperplanes, *i.e.* by a $(d-k)\times(d+1)$ rank $d-k$ matrix $\mathbf{W}_a^C$ that determines a set of $d-k$ independent homogeneous linear equations $\mathbf{W}_a^C\mathbf{x}^a = 0$ ($C = k+1, \ldots, d+1$). $\mathbf{W}$ and $\mathbf{U}$ specify the same subspace if and only if $\mathbf{W}\mathbf{U} = 0$ and the $(d+1) \times (d+1)$ matrix $\binom{\mathbf{U}^\top}{\mathbf{W}}$ is nonsingular. For any such pair $\{\mathbf{U}, \mathbf{W}\}$ and any nonsingular symmetric $(d+1) \times (d+1)$ matrix $\mathbf{X}$ we have the standard decomposition

$$\begin{aligned} \mathbf{X} \;=\; &\mathbf{U}\left(\mathbf{U}^\top\mathbf{X}^{-1}\mathbf{U}\right)^{-1}\mathbf{U}^\top \\ &+ \mathbf{X}\mathbf{W}^\top\left(\mathbf{W}\mathbf{X}\mathbf{W}^\top\right)^{-1}\mathbf{W}\mathbf{X} \end{aligned}$$

Applying this at the covariance $\mathbf{X} \equiv \mathbf{M}^{-1}$ gives the approximate log-unlikelihood of the subspace in terms of dual coordinates

$$\begin{aligned} &1 + \boldsymbol{\chi}^2(\mathbf{W} \,|\, \mathbf{X}, \mathbf{N}) \overset{+\text{const}}{\approx} \\ &\quad \text{Trace}^{-1}\left[\mathbf{N} \cdot \left(\mathbf{X} - \mathbf{X}\mathbf{W}^\top(\mathbf{W}\mathbf{X}\mathbf{W}^\top)^{-1}\mathbf{W}\mathbf{X}\right)\right] \end{aligned}$$

Since $\mathbf{X} = \mathbf{M}^{-1}$ is normalized, the leading term is just $\text{Trace}(\mathbf{N} \cdot \mathbf{X}) = 1$.

6.1 Affine Limit

In the affine case the approximate shift formula is exact and the biquadric distributions become Gaussians, so the projective error model reduces to the standard affine one. Making the standard decompositions

$$\mathbf{M} \equiv \begin{pmatrix} 1 + \bar{\mathbf{x}}^\top\mathbf{X}^{-1}\bar{\mathbf{x}} & -\bar{\mathbf{x}}^\top\mathbf{X}^{-1} \\ -\mathbf{X}^{-1}\bar{\mathbf{x}} & \mathbf{X}^{-1} \end{pmatrix}, \quad \mathbf{N} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} = \mathbf{U}\begin{pmatrix} 1 \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} 1 \\ \mathbf{A}\mathbf{y} + \mathbf{b} \end{pmatrix}, \quad \mathbf{U} \equiv \begin{pmatrix} 1 & 0 \\ \mathbf{b}^\top & \mathbf{A} \end{pmatrix}$$

we have

$$\begin{aligned} \mathbf{U}^\top\mathbf{M}\mathbf{U} \;=\; &\begin{pmatrix} 1+(\bar{\mathbf{x}}-\mathbf{b})^\top\mathbf{X}^{-1}(\bar{\mathbf{x}}-\mathbf{b}) & -(\bar{\mathbf{x}}-\mathbf{b})^\top\mathbf{X}^{-1}\mathbf{A} \\ \mathbf{A}^\top\mathbf{X}^{-1}(\bar{\mathbf{x}}-\mathbf{b}) & \mathbf{A}^\top\mathbf{X}^{-1}\mathbf{A} \end{pmatrix} \\ \mathbf{U}^\top\mathbf{N}\mathbf{U} \;=\; &\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \end{aligned}$$

Using the fact that an incorrectly shifted affine information tensor has an inverse with 00 coefficient $1/(1+\lambda)$:

$$\begin{aligned} &\begin{pmatrix} 1 + \lambda + \bar{\mathbf{x}}^\top\mathbf{X}^{-1}\bar{\mathbf{x}} & -\bar{\mathbf{x}}^\top\mathbf{X}^{-1} \\ -\mathbf{X}^{-1}\bar{\mathbf{x}} & \mathbf{X}^{-1} \end{pmatrix}^{-1} \\ &\qquad = \frac{1}{1+\lambda}\begin{pmatrix} 1 \\ \bar{\mathbf{x}} \end{pmatrix}\begin{pmatrix} 1 & \bar{\mathbf{x}}^\top \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{X} \end{pmatrix} \end{aligned}$$

a short calculation gives

$$\begin{aligned} &\boldsymbol{\chi}^2(\mathbf{U} \,|\, \mathbf{X}, \bar{\mathbf{x}}) \overset{+\text{const}}{=} (\bar{\mathbf{x}}-\mathbf{b})^\top \cdot \\ &\cdot \left(\mathbf{X}^{-1} - \mathbf{X}^{-1}\mathbf{A}(\mathbf{A}^\top\mathbf{X}^{-1}\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{X}^{-1}\right)(\bar{\mathbf{x}}-\mathbf{b}) \end{aligned}$$

This is the standard affine formula for the log-unlikelihood of an affine subspace $\mathbf{A}\mathbf{y} + \mathbf{b}$ given

an uncertain observation of a point on it. The matrix vanishes on vectors $\mathbf{Ay}$ in the subspace and hence measures the 'orthogonal Mahalanobis distance' of the mean $\bar{\mathbf{x}}$ from the subspace.

In terms of dual coordinates the affine subspace is

$$\mathbf{W}\begin{pmatrix}1\\ \mathbf{x}\end{pmatrix} \;=\; \mathbf{Dx}+\mathbf{c} \;=\; 0 \qquad \mathbf{W} \;\equiv\; \begin{pmatrix}\mathbf{c} & \mathbf{D}\end{pmatrix}$$

where $\mathbf{DA} = 0$ and $\mathbf{c} = -\mathbf{Db}$. In this case the affine log-unlikelihood is simply

$$\boldsymbol{\chi}^2(\mathbf{W}\,|\,\mathbf{X},\bar{\mathbf{x}}) \stackrel{+\text{const}}{=} (\mathbf{D}\bar{\mathbf{x}}+\mathbf{c})^\top \Big(\mathbf{DXD}^\top\Big)^{-1}(\mathbf{D}\bar{\mathbf{x}}+\mathbf{c})$$

This is easily verified from the non-dual-form affine log-unlikelihood given above, or with a little more effort from the projective dual-form log-unlikelihood. Basically, it says that the information in constraint violation space is measured by the inverse of the classical constraint covariance matrix.

6.2 Grassmann Coordinates

We promised that projective least squares would look natural in Grassmann coordinates, and now we verify this. The k dimensional projective subspace spanned by the column vectors of $\mathbf{U}^a_A$ has Grassmann coordinates [2, 7]

$$\mathbf{u}^{[a_0\cdots a_k]} \;\sim\; \mathbf{U}^{a_0}_{A_0}\cdots\mathbf{U}^{a_k}_{A_k}\,\boldsymbol{\varepsilon}^{A_0\cdots A_k}$$

Alternatively, a k dimensional subspace can be specified by $d-k$ linear constraints $\mathbf{W}^C_a\mathbf{x}^a = 0$ (the rows of the matrix $\mathbf{W}$, labelled by $C = k+1,\ldots,d+1$) to give dual Grassmann coordinates

$$\mathbf{w}_{[a_{k+1}\cdots a_d]} \;\sim\; \boldsymbol{\varepsilon}_{C_{k+1}\cdots C_d}\,\mathbf{W}^{C_{k+1}}_{a_{k+1}}\cdots\mathbf{W}^{C_d}_{a_d}$$

Here, $\mathbf{u}^{a_0\cdots a_k}$ and $\mathbf{w}_{a_{k+1}\cdots a_d}$ are respectively the $(k+1)\times(k+1)$ minors of $\mathbf{U}$ and the $(d-k)\times(d-k)$ minors of $\mathbf{W}$. They are only defined up to scale, and if $\mathbf{U}$ and $\mathbf{W}$ specify the same subspace they are tensor duals of one another.

The subspace log-unlikelihood

$$\begin{aligned}&1+\boldsymbol{\chi}^2(\mathbf{U}\,|\,\mathbf{M},\mathbf{N})\\ &\quad\approx\; \text{Trace}^{-1}\Big(\mathbf{N}\cdot\mathbf{U}(\mathbf{U}^\top\mathbf{MU})^{-1}\mathbf{U}^\top\Big)\end{aligned}$$

can be rewritten in terms of the Grassmann coordinates $\mathbf{u}^{a_0\cdots a_k}$ by expanding the inverse $(\mathbf{U}^\top\mathbf{MU})^{-1}$ by cofactors and rearranging. The result is

$$1+\boldsymbol{\chi}^2(\mathbf{u}\,|\,\mathbf{M},\mathbf{N}) \;\stackrel{+\text{const}}{\approx}\; \frac{\mathbf{M}_{a_0b_0}\cdots\mathbf{M}_{a_kb_k}\cdot\mathbf{u}^{a_0\cdots a_k}\,\mathbf{u}^{b_0\cdots b_k}}{(k+1)\,\mathbf{N}_{c_0d_0}\mathbf{M}_{c_1d_1}\cdots\mathbf{M}_{c_kd_k}\cdot\mathbf{u}^{c_0c_1\cdots c_k}\,\mathbf{u}^{d_0d_1\cdots d_k}}$$

Once again we recognize the familiar form of the biquadric, this time in the Grassmann coordinates $\mathbf{u}^{a_0\cdots a_k}$ rather than the point coordinates $\mathbf{x}^a$, with information[1] $\mathbf{M}_{[\![a_0b_0}\cdots\mathbf{M}_{a_kb_k]\!]}$ and normalization $(k+1)\cdot\mathbf{N}_{[\![a_0b_0}\mathbf{M}_{a_1b_1}\cdots\mathbf{M}_{a_kb_k]\!]}$. If $k=0$ we get back the original point distribution, as would be expected.

The space of k dimensional projective subspaces in d dimensions is locally parameterized by $(d-k)\times(k+1)$ matrices and therefore has dimension $(d-k)(k+1)$. The Grassmann coordinatization embeds it as a projective subvariety of the $\binom{d+1}{k+1}$ dimensional homogeneous space $\mathcal{P}^{[a_0\cdots a_k]}$ of $k+1$ index skew tensors. The constraint equations that determine this subvariety are the quadratic **Grassmann simplicity constraints**

$$\mathbf{u}^{a_0\cdots a_{k-1}[a_k}\mathbf{u}^{b_0\cdots b_k]} = 0$$

Hence, although the Grassmann coordinates $\mathbf{u}^{a_0\cdots a_k}$ are linearly independent, they are quadratically highly redundant.

The subspace information and normalization tensors can be viewed as symmetric matrices on the large $\binom{d+1}{k+1}$ dimensional space $\mathcal{P}^{[a_0\cdots a_k]}$. They are nonsingular whenever the underlying $\mathbf{M}_{ab}$ and $\mathbf{N}_{ab}$ are, however there are linear (non-matricial) relations among their components that enforce the Grassmann simplicity constraints. Any product of symmetric tensors of the form

$$\mathbf{T}^{[a_0a_1\cdots a_k]\cdot[b_0b_1\cdots b_k]} \;\equiv\; \mathbf{M}_0^{[\![a_0b_0}\mathbf{M}_1^{a_1b_1}\cdots\mathbf{M}_k^{a_kb_k]\!]}$$

is 'simple' in the sense that $\mathbf{T}^{a_0\cdots a_{k-1}[a_k\,\cdot\,b_0\cdots b_k]} = 0$ because the antisymmetrization always includes a pair of symmetric indices. A biquadric built with such 'simple' Grassmann tensors "projects on to the simple part of $\mathbf{u}^{a_0\cdots a_k}$" in the sense that it is insensitive to the 'non-simple part' $\mathbf{u}^{a_0\cdots a_{k-1}[a_k}\mathbf{u}^{b_0\cdots b_k]} \neq 0$.

[1] For convenience we introduce the notation $[\![a_0b_0a_1b_1\cdots a_kb_k]\!]$ to denote $[a_0a_1\cdots a_k][b_0b_1\cdots b_k]$ on the index pairs a_ib_i of a set of 2 index tensors, *i.e.* antisymmetrize separately over the first indices and the second indices of the pairs.

A similar process can be applied to the dual-form matricial log-unlikelihood $1 + \boldsymbol{\chi}^2(\mathbf{W} \,|\, \mathbf{X}, \mathbf{N})$ given above, to derive the dual Grassmann log-unlikelihood

$$1 + \boldsymbol{\chi}^2(\mathbf{w} \,|\, \mathbf{X}, \mathbf{N}) \overset{+\text{const}}{\approx} \frac{\mathbf{X}^{a_{k+1}b_{k+1}} \dots \mathbf{X}^{a_d b_d} \cdot \mathbf{w}_{a_{k+1}\cdots a_d}\, \mathbf{w}_{b_{k+1}\cdots b_d}}{(\mathbf{X} - (d-k)\mathbf{X}\mathbf{N}\mathbf{X})^{c_{k+1}d_{k+1}}\, \mathbf{X}^{c_{k+2}d_{k+2}} \dots \mathbf{X}^{c_d d_d} \cdot \mathbf{w}_{c_{k+1}\cdots c_d}\, \mathbf{w}_{d_{k+1}\cdots d_d}}$$

where $\mathbf{X} \equiv \mathbf{M}^{-1}$ and $\mathbf{N}_{ab}\mathbf{X}^{ab} = 1$. Once again the log-unlikelihood has the biquadric form, this time in the dual coordinates $\mathbf{w}_{a_{k+1}\cdots a_d}$. The information and normalization tensors are again 'simple' in the Grassmann sense. This can also be derived by tensor dualization of the contravariant Grassmann formula. Note that in the affine case $\mathbf{X}\mathbf{N}\mathbf{X} = \binom{1}{\bar{\mathbf{x}}}(1 \;\; \bar{\mathbf{x}}^\top)$.

6.3 Hyperplanes

Hyperplanes (codimension one subspaces) are a particularly important special case of the above. The log-unlikelihood for the location of a hyperplane $\mathbf{w}_a\mathbf{x}^a = 0$ given an uncertain point on it follows immediately from the above dual-form matrix or Grassmann formulae:

$$1 + \boldsymbol{\chi}^2(\mathbf{w}_a \,|\, \mathbf{X}, \mathbf{N}) \overset{+\text{const}}{\approx} \frac{\mathbf{X}^{ab}\, \mathbf{w}_a \mathbf{w}_b}{(\mathbf{X} - \mathbf{X}\mathbf{N}\mathbf{X})^{cd}\, \mathbf{w}_c \mathbf{w}_d}$$

Dually to the point case, the log-unlikelihood is a biquadric in the hyperplane coordinates. For an affine distribution this becomes

$$\boldsymbol{\chi}^2(\mathbf{w}_a \,|\, \mathbf{X}, \mathbf{N}) \overset{+\text{const}}{=} \frac{\left(\mathbf{d}^\top \bar{\mathbf{x}} + c\right)^2}{\mathbf{d}^\top \bar{\mathbf{X}} \mathbf{d}}$$

where $\mathbf{w}_a$ is $(c \;\; \mathbf{d}^\top)$, and $\bar{\mathbf{x}}$ and $\bar{\mathbf{X}}$ are the classical mean and covariance.

The denominator plays a much more active rôle in hyperplane and k-subspace estimation than it did in the point fitting problem. Let us examine the hyperplane case a little more closely to find out why.

First of all, there is nothing intrinsically wrong with hyperplane distributions with 'simple' normalizers $\overline{\mathbf{N}}^{ab}$. It is just that in the case of point-plane fitting the correct answer can not be quite so simple. Consider a hyperplane distribution with a 'slowly varying' denominator $\overline{\mathbf{N}}^{ab}\mathbf{w}_a\mathbf{w}_b$. For example, $\overline{\mathbf{N}}^{ab}$ could be the $(d+1)\times(d+1)$ unit matrix in some basis, or the affine hyperplane normalizer $\left(\begin{smallmatrix} 0 & 0 \\ 0 & \mathbf{I} \end{smallmatrix}\right)$, where $\mathbf{I}$ is the $d \times d$ unit matrix[2]. If the plane passes exactly through the mode of the point distribution, we would expect its likelihood to depend only weakly on its orientation: any plane passing right through the observation should be about equally good as far as the least squares error is concerned. Since the denominator was chosen to be almost independent of orientation, the numerator must also depend only weakly on orientation. But this implies that the rate of decay of the likelihood as the plane moves away from the point is *also* independent of orientation: the only remaining parameter is a direction-independent scalar peak width. However in general the point distribution is not spherically symmetric and the rate of decay of the plane distribution ought to be different in different directions. In summary, it is not possible to have all three of: *(i)* an isotropic likelihood *at* the observation; *(ii)* an anisotropic decay *away from* the observation; *(iii)* an isotropic normalizer $\overline{\mathbf{N}}^{ab}$. The first two are essential to represent the data correctly, so we are forced to deal with non-isotropic normalizers $\mathbf{N}^{ab}$ and hence (if the plane is being fitted to several points) incommensurable distributions. This is not simply a problem with the projective theory: classical affine least squares also gives incommensurable distributions for subspace fitting (*e.g.* orthogonal regression). In fact, the projective point of view makes the situation clearer by unifying the classically separate theories of point and plane fitting.

6.4 Normalization & Covariance

The above formulae for subspace log-unlikelihoods are only correct up to an additive constant. The modes (maximum likelihood values) of the subspace distributions can be found directly from the unshifted information tensors, but if subspace covariances are required the correct shift factors must be estimated.

In fact, it is straightforward to show that the normalization sum $\mathrm{Trace}(\mathbf{N} \cdot \mathbf{M}^{-1})$ for the subspace-fitted-to-point distributions is always $\binom{d}{k}$ instead of 1. The reason is simply that even when the point distribution is narrow the resulting subspace distribution always has a $\binom{d}{k}$ dimensional modal (*i.e.* maximum likelihood) subspace in $\mathcal{P}^{[a_0\cdots a_k]}$, corresponding to the $\binom{d}{k}$ different 'directions' in which the k-

[2] This gives the conventional normalization for Euclidean hyperplanes, with a constant offset and a unit direction vector.

subspace can pass right through the centre of the point distribution[3].

Since $\mathrm{Trace}(\mathbf{N}\cdot\mathbf{M}^{-1}) = \binom{d}{k}$ for the $\binom{d+1}{k+1}$ dimensional subspace information tensor, the approximate shift equation predicts a normalizing shift of $\mathbf{M} \rightarrow \mathbf{M} + (\binom{d}{k} - 1)\cdot\mathbf{N}$. However this approximation is not recommended as it is likely to be quite inaccurate for such large shift factors. On the other hand, whenever subspace-through-point likelihoods from several points are combined, the resulting distribution tends to be much better localized because the null directions from different points tend to cancel each other out, leaving a single reasonably well defined mode. In this case (and modulo the usual correction for the accumulation of the '1's in the $(1+\boldsymbol{\chi}^2)$'s) the shift factor required to normalize the combined subspace distribution tends to be much smaller.

The $\binom{d+1}{k+1}$ dimensional Grassmann parameterization is global but redundant and it is often convenient to re-express the mode and covariance in terms of some minimal local parameterization, say $\mathbf{z}^\alpha$ where $\alpha = 1,\ldots,(d-k)(k+1)$. Given Grassmann information and normalization matrices $\mathbf{M}$ and $\mathbf{N}$, the Grassmann mode can be found by the usual minimum eigenvector procedure. The expression $\mathbf{u}^{a_0\cdots a_k}(\mathbf{z}^\alpha)$ for the Grassmann parameterization in terms of $\mathbf{z}^\alpha$ must then be inverted at the Grassmann mode to find the $\mathbf{z}^\alpha$-space mode. (This may require the solution of nonlinear equations). Finally, the $\mathbf{z}^\alpha$-space information matrix can be found by evaluating the second derivatives of $1+\boldsymbol{\chi}^2\left(\mathbf{u}^{a_0\cdots a_k}(\mathbf{z}^\alpha)\,|\,\mathbf{M},\mathbf{N}\right)$ at the $\mathbf{z}^\alpha$-space mode.

7 Fundamental Matrix Estimation

As another example of the use of projective least squares, consider the problem [3, 1] of estimating the fundamental matrix between two images from a set of corresponding point pairs. Given any two 2D projective images $\mathcal{P}^A$ and $\mathcal{P}^{A'}$ of a 3D scene taken from different positions, a pair $\{\mathbf{x}^A,\mathbf{x}^{A'}\}$ of image points corresponds to some 3D point if and only if the **epipolar constraint** $\mathbf{F}_{AA'}\,\mathbf{x}^A\mathbf{x}^{A'} = 0$ is satisfied, where $\mathbf{F}_{AA'}$ is the 3×3 rank 2 **fundamental matrix**. The point $\mathbf{x}^A$ gives rise to a corresponding **epipolar line** $\mathbf{F}_{AA'}\mathbf{x}^A$ in the opposite image $\mathcal{P}^{A'}$ and all potentially matching $\mathbf{x}^{A'}$ lie on this line. The epipolar lines all pass through a point called the **epipole** $\mathbf{e}^{A'}$. This is the image of the projection centre of the opposite camera and satisfies $\mathbf{F}_{AA'}\mathbf{e}^{A'} = 0$. Similarly for $\mathbf{x}^{A'}$, $\mathbf{F}_{AA'}\mathbf{x}^{A'}$ and $\mathbf{e}^A$.

We can estimate $\mathbf{F}$ from a set of corresponding uncertain point pairs by viewing the epipolar constraint from each pair as a single linear constraint on $\mathbf{F}$. Intuitively, the smaller the deviations $|\mathbf{F}_{AA'}\mathbf{x}^A\mathbf{x}^{A'}|$ are, the better the fit will be, but we want to make this into a more rigorous approximate maximum likelihood estimate. The situation is analogous to that of hyperplane estimation: $\mathbf{F}_{AA'}$ can be viewed as defining a projective hyperplane in the $3\times 3-1 = 8$ dimensional projective space of tensors $\mathcal{P}^{AA'}$, and the data can be mapped bilinearly into this space via $\{\mathbf{x}^A,\mathbf{x}^{A'}\} \rightarrow \mathbf{x}^A\mathbf{x}^{A'}$. In fact, it turns out that we can re-use our projective least squares equations for hyperplanes.

Suppose that the uncertainties in the positions of $\mathbf{x}^A$ and $\mathbf{x}^{A'}$ can be modelled by independent normalized biquadric distributions $\{\mathbf{M}_{AB},\mathbf{N}_{AB}\}$ and $\{\mathbf{M}_{A'B'},\mathbf{N}_{A'B'}\}$ with covariances $\mathbf{X}^{AB} = (\mathbf{M}^{-1})^{AB}$ and $\mathbf{X}^{A'B'} = (\mathbf{M}^{-1})^{A'B'}$. Since the distributions are independent their moments can be factorized. In particular

$$\begin{aligned}
&\left\langle \frac{(\mathbf{x}^A\mathbf{x}^{A'})(\mathbf{x}^B\mathbf{x}^{B'})}{(\mathbf{N}_{CD}\mathbf{N}_{C'D'})(\mathbf{x}^C\mathbf{x}^{C'})(\mathbf{x}^D\mathbf{x}^{D'})}\right\rangle \\
&\qquad = \left\langle \frac{\mathbf{x}^A\mathbf{x}^B}{\mathbf{N}_{CD}\mathbf{x}^C\mathbf{x}^D}\cdot\frac{\mathbf{x}^{A'}\mathbf{x}^{B'}}{\mathbf{N}_{C'D'}\mathbf{x}^{C'}\mathbf{x}^{D'}}\right\rangle \\
&\qquad = \left\langle \frac{\mathbf{x}^A\mathbf{x}^B}{\mathbf{N}_{CD}\mathbf{x}^C\mathbf{x}^D}\right\rangle\cdot\left\langle \frac{\mathbf{x}^{A'}\mathbf{x}^{B'}}{\mathbf{N}_{C'D'}\mathbf{x}^{C'}\mathbf{x}^{D'}}\right\rangle \\
&\qquad = \mathbf{X}^{AB}\,\mathbf{X}^{A'B'}
\end{aligned}$$

Viewing $\mathbf{M}_{AB}\mathbf{M}_{A'B'}$, $\mathbf{N}_{AB}\mathbf{N}_{A'B'}$ and $\mathbf{X}^{AB}\mathbf{X}^{A'B'}$ as 9×9 homogeneous symmetric matrices on the 8 dimensional projective space $\mathcal{P}^{AA'}$, we have $\mathbf{N}_{AB}\mathbf{N}_{A'B'}\cdot\mathbf{X}^{AB}\mathbf{X}^{A'B'} = 1$ and (since $\mathbf{X}^{AB}\mathbf{X}^{A'B'}\cdot\mathbf{M}_{BC}\mathbf{M}_{B'C'} = \boldsymbol{\delta}^A_C\boldsymbol{\delta}^{A'}_{C'}$ is the identity operator on $\mathcal{P}^{AA'}$) $\mathbf{X}^{AB}\mathbf{X}^{A'B'} = (\mathbf{M}_{AB}\mathbf{M}_{A'B'})^{-1}$. So rather remarkably, $\mathbf{M}_{AB}\mathbf{M}_{A'B'}$ and $\mathbf{N}_{AB}\mathbf{N}_{A'B'}$

[3] The 'directions' of the modal subspace are generated by the $\binom{d}{k}$ choices of k directions $\mathbf{u}^a_1,\ldots,\mathbf{u}^a_k$ among the d in any hyperplane not passing through the point mode $\hat{\mathbf{x}}^a$. The corresponding k-subspace is the span $\hat{\mathbf{x}}^{[a_0}\mathbf{u}^{a_1}_1\cdots\mathbf{u}^{a_k]}_k$. The $\binom{d}{k}$ dimensional modal subspace intersects the $(d-k)(k-1)$ dimensional Grassmann variety of k-subspaces (*i.e.* simple tensors) in the $(d-k)(k-2)$ dimensional variety of k-subspaces through $\hat{\mathbf{x}}^a$.

define a correctly shifted biquadric distribution with covariance $\mathbf{X}^{AB}\mathbf{X}^{A'B'}$ on $\mathcal{P}^{AA'}$, that correctly models the uncertainty of the tensor-product image point $\mathbf{x}^A\mathbf{x}^{A'}$ to second order. This is notwithstanding the fact that the space of all possible $\mathbf{x}^A\mathbf{x}^{A'}$ is only a 4 dimensional quadratic subvariety of the 8 dimensional projective tensor space $\mathcal{P}^{AA'}$. Since the epipolar constraint $\mathbf{F}_{AA'}\mathbf{x}^A\mathbf{x}^{A'} = 0$ defines a projective hyperplane in $\mathcal{P}^{AA'}$ and we know how to fit projective hyperplanes to points, we can immediately write down the log-unlikelihood of $\mathbf{F}$ given $\mathbf{x}^A$ and $\mathbf{x}^{A'}$:

$$1 + \boldsymbol{\chi}^2(\mathbf{F}_{AA'} \,|\, \mathbf{x}^A, \mathbf{x}^{A'}) \overset{+\text{const}}{\approx} \frac{\mathbf{X}^{AB}\mathbf{X}^{A'B'}\,\mathbf{F}_{AA'}\mathbf{F}_{BB'}}{\left(\mathbf{X}^{CD}\mathbf{X}^{C'D'} - (\mathbf{X}\mathbf{N}\mathbf{X})^{CD}(\mathbf{X}'\mathbf{N}'\mathbf{X}')^{C'D'}\right)\cdot \cdot\mathbf{F}_{CC'}\mathbf{F}_{DD'}}$$

Writing the information and normalization tensors as 9×9 symmetric matrices on the 9 dimensional space of components of $\mathbf{F}$, the biquadric log-unlikelihoods for different point pairs can be combined in the usual way. As in the hyperplane case, they are always incommensurable so nonlinear techniques are required.

If both of the points have affine distributions, converting to 3×3 matrix notation and denoting the homogeneous mean $\binom{1}{\bar{\mathbf{x}}}$ by $\hat{\mathbf{x}}$ and the homogeneous affine covariance $\left(\begin{smallmatrix} 0 & 0 \\ 0 & \bar{\mathbf{X}} \end{smallmatrix}\right)$ by $\hat{\mathbf{X}}$, we can re-express this as follows:

$$\boldsymbol{\chi}^2(\mathbf{F} \,|\, \hat{\mathbf{x}}, \hat{\mathbf{X}}, \hat{\mathbf{x}}', \hat{\mathbf{X}}') \overset{+\text{const}}{=} \frac{(\hat{\mathbf{x}}^\top\mathbf{F}\hat{\mathbf{x}}')^2}{\hat{\mathbf{x}}^\top\mathbf{F}\hat{\mathbf{X}}'\mathbf{F}^\top\hat{\mathbf{x}} + \hat{\mathbf{x}}'^\top\mathbf{F}^\top\hat{\mathbf{X}}\mathbf{F}\mathbf{x}' + \mathrm{Trace}(\mathbf{F}\hat{\mathbf{X}}'\mathbf{F}^\top\hat{\mathbf{X}})}$$

This formula can also be derived by classical maximum likelihood calculations. The term $\mathrm{Trace}\left(\mathbf{F}\hat{\mathbf{X}}'\mathbf{F}^\top\hat{\mathbf{X}}\right)$ is second order in the uncertainty and is often ignored relative to the first order terms: with this approximation the formula has been used for nonlinear estimation of the fundamental matrix with good results [3]. Roughly, it says that the 'primitive' error measure $(\hat{\mathbf{x}}'^\top\mathbf{F}\hat{\mathbf{x}})^2$ needs to be normalized by dividing by the sum of the variance of each measured point orthogonal to the opposite epipolar line. When one or both of the measured points lie near an epipole, the second order trace term is sometimes significant relative to the other terms and tends to have a stabilizing effect on the fit, so it should probably not be omitted if the epipoles lie within the images (*e.g.* frontal motion).

8 Discussion & Future Work

The results we have presented are obviously still at the theoretical level and it remains to be seen how useful projective least squares will turn out to be in practice. However, it is becoming clear that error modelling will become a central issue in visual reconstruction, not only to ensure the accuracy of the final results, but also because the efficiency of intermediate stages such as correspondence and database indexing depends critically on the uncertainties involved. Given that projective least squares is both 'projectively correct' and relatively tractable (notwithstanding the length of some of the equations we have written), it seems likely that it will have a part to play in all this.

On the technical level there are still many loose ends. Analytical work is needed to clarify the status of the two approximations made in deriving the basic error model, and the development of a 'central moment expansion' based on the homogeneous Taylor series could be mathematically fruitful. More practically it would be useful to have projective least squares methods for quadrics and higher order projective varieties, and for further types of subspace-subspace intersection and union (*e.g.* intersection of subspaces at a point). It is also unclear how to extend the fundamental matrix estimation model to the trilinear and quadrilinear constraints that exist when there are additional images [6, 7, 9]. Although the relation between the multilinear data tensors $\mathbf{x}^{A_i}\mathbf{x}^{A_j}\cdots\mathbf{x}^{A_k}$ and the corresponding constraint tensor is still linear, it is no longer a simple scalar and it is not yet clear how to capture it correctly in a projective least squares error model.

References

[1] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

[2] W. V. D. Hodge and D. Pedoe. *Methods of Algebraic Geometry*, volume 1. Cambridge University Press, 1947.

[3] Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimen-

tal results. Technical Report RR-1894, INRIA, Sophia Antipolis, France, 1993.

[4] P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 314–20, Cambridge, MA, June 1995.

[5] R. Mohr, B. Boufama, and P. Brand. Accurate relative positioning from multiple images. Technical Report RT-102, LIFIA, INRIA Rhône-Alpes, Grenoble, France, 1993. Submitted to *Journal of Artificial Intelligence*.

[6] A. Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 17(8):779–89, 1995.

[7] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to *Int. J. Computer Vision*.

[8] B. Triggs. Least squares estimation in projective spaces. To appear, 1995.

[9] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.

[10] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis*. Springer-Verlag, 1992.

Critical Motions in Euclidean Structure from Motion

Fredrik Kahl
Department of Mathematics
Lund University
Box 118, S-221 00 Lund, Sweden
fredrik@maths.lth.se

Bill Triggs
INRIA Rhône-Alpes
655 ave. de l'Europe, 38330 Montbonnot, France
http://www.inrialpes.fr/movi/people/Triggs
Bill.Triggs@inrialpes.fr

Abstract

We investigate the motions that lead to ambiguous Euclidean scene reconstructions under several common calibration constraints, giving a complete description of such *critical motions* for: (*i*) internally calibrated orthographic and perspective cameras; (*ii*) in two images, for cameras with unknown focal lengths, either different or equal. One aim of the work was to evaluate the potential of modern algebraic geometry tools for rigorously proving properties of vision algorithms, so we use ideal-theoretic calculations as well as classical algebra and geometry. We also present numerical experiments showing the effects of near-critical configurations for the varying and fixed focal length methods.

Keywords: structure from motion, critical motions, autocalibration, algebraic geometry.

1 Introduction

'Structure from Motion' (SFM) is the problem of recovering 3D scene geometry from several images. Using projective image measurements, it is only possible to recover structure, camera poses ('motion') and camera internal parameters ('calibrations') up to an unknown 3D projectivity [8, 5]. With additional scene, motion or calibration constraints, one can reduce the ambiguity to a Euclidean similarity [13, 4, 12, 7]. **Autocalibration** is the recovery of Euclidean structure, motion and calibration using partial (often qualitative) constraints on the camera calibrations, *e.g.* vanishing skew or equal focal lengths between images. It is useful because cameras often obey such constraints rather well, whereas — especially for hand-held cameras viewing unknown scenes — motion or structure assumptions are often rather dubious. Unfortunately, most autocalibration methods have situations in which they fail or are exceptionally weak. Practically, it is important to characterize and avoid these **critical sets**. Criticality is often independent of the specific camera calibrations, in which case we speak of **critical motions**.

'Classical' autocalibration assumes a moving projective camera with constant but unknown intrinsic parameters [4, 18, 1, 23, 17]. Sturm [19, 20] categorizes both the intrinsic and some algorithm-specific critical motions for this. The uniformity of the constraints makes this case relatively simple to analyze. But it is also somewhat unrealistic: it is often reasonable, *e.g.* to assume that the constant skew actually vanishes (a stronger constraint), whereas focal length often varies between images (a weaker constraint). Also, although he characterizes the degeneracies fully, Sturm only manages to give a rather implicit description of the corresponding critical motions. For practical purposes a more explicit description would be useful.

This paper derives explicit critical motions for Euclidean SFM under several simple two image 'unknown focal length' calibration constraints [6, 16, 24, 2, 9]. However, we start by giving a complete description of criticality for *known* calibrations, for both perspective and orthographic cameras in multiple images. Although this analysis does not result in any new ambiguities, it rules out the possibility of any further unknown ones.

A second goal of our work — one aspect of our European project CUMULI — was to investigate the use of formal algebraic reasoning tools to deduce rigorous properties of vision algorithms. Sturm [19]

This paper appeared in CVPR'99. The work was supported by Esprit LTR project CUMULI.

relies mainly on geometric intuition. This is unreliable in our less symmetrical situation and we have used a mixture of geometry, classical algebra, and ideal-theoretic algebraic geometry calculations (Gröbner bases, ideal quotient, radical and decomposition) in MAPLE and MACAULAY 2. However we will focus on giving geometric interpretations of our algebraic results whenever possible.

We consider only autocalibration degeneracies: scene and motion constraints are explicitly excluded from consideration. Also, for both projective and Euclidean reconstruction there are certain scene geometries for which SFM is inherently ambiguous [12, 15, 11, 10]. We exclude such **critical surfaces** by assuming that the scene is generic enough to allow unambiguous recovery of projective structure. Hence, *criticality occurs iff the calibration constraints admit alternative Euclidean 'interpretations' of the given projective structure.*

2 Background

Image projection: We assume familiarity with the modern projective formulation of vision geometry [3, 12, 23]. A **perspective (pinhole) camera** is modeled in homogeneous coordinates by the projection equation $\boldsymbol{x} \simeq \mathbf{P}\mathbf{X}$ where $\mathbf{X} = (X,Y,Z,W)^\top$ is a 3D world point, $\boldsymbol{x} = (x,y,z)^\top$ is its 2D image and $\mathbf{P}$ is the 3×4 camera **projection matrix**. In a Euclidean frame $\mathbf{P}$ can be decomposed

$$\mathbf{P} = \boldsymbol{K}\boldsymbol{R}\,(\boldsymbol{I}_{3\times 3}|-\boldsymbol{t}) \qquad \boldsymbol{K} = \begin{pmatrix} f & fs & u_0 \\ 0 & fa & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

into a rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ encoding the camera's 3D pose (**extrinsic parameters**), and a 3×3 upper triangular **calibration matrix** $\boldsymbol{K}$ encoding its internal geometry. Here, f is the **focal length**, a the **aspect ratio**, s the **skew** and (u_0,v_0) the **principal point**.

Absolute Conic: Projective geometry encodes only collinearity and incidence. Affine structure (parallelism) is encoded projectively by singling out a **plane at infinity** Π_∞ of **direction vectors** or **points at infinity**, and Euclidean (similarity) structure by a proper virtual conic on Π_∞. This **absolute conic** Ω_∞ gives dot products between direction vectors. Its dual, the **dual absolute conic** Ω^*_∞, gives those between plane normals. Ω^*_∞ is a 4×4 symmetric rank 3 positive semidefinite contravariant matrix. $\Omega^*_\infty = \mathrm{diag}(1,1,1,0)$ in any Euclidean frame. Π_∞ is Ω^*_∞'s unique null vector: $\Omega^*_\infty \Pi_\infty = 0$. Ω^*_∞'s image projection is $\omega^*_\infty \equiv \mathbf{P}\Omega^*_\infty\mathbf{P}^\top = \boldsymbol{K}\boldsymbol{K}^\top$, a dual image conic that encodes the camera calibration. $\boldsymbol{K}$ is recoverable from ω^*_∞ or its dual image point conic $\omega_\infty = {\omega^*_\infty}^{-1}$ by Cholesky factorization. ω^*_∞ and ω_∞ are proper virtual (positive definite) so long as the camera centre is finite. In calibrated image coordinates $\boldsymbol{K} = \boldsymbol{I}$, $\omega^*_\infty = \omega_\infty = \boldsymbol{I}$. We often use the abbreviations (D)(I)AC for (*D*ual)(*I*mage) *A*bsolute *C*onic.

False absolute conics: Given only a 3D projective reconstruction derived from uncalibrated images, the true absolute conic Ω_∞ is not distinguished in any way from any other proper virtual planar conic in projective space. In fact, given any such conic Ω^*, it is easy to find a 'rectifying' projective transformation that converts it to the Euclidean DAC form $\Omega^*_\infty = \mathrm{diag}(1,1,1,0)$ and hence defines a false Euclidean structure. To recover the true structure, we need constraints that single out the true Ω_∞ and Π_∞ from all possible 'false' ones. In this paper we will constrain only the camera intrinsic parameters $\boldsymbol{K}_i$, or equivalently the images of the true absolute conic $\omega^*_{\infty i} = \boldsymbol{K}_i\boldsymbol{K}_i^\top$. The constraints may apply to individual image conics (*e.g.* vanishing skew $s=0$), or link them as a group (*e.g.* equal but unknown focal lengths $f_i = f$ for all i). Ambiguity arises only if some non-absolute conic and its images satisfy the constraints. We call such conics **potential** or **false absolute conics**. They correspond one-to-one with possible false Euclidean structures for the scene. Ω denotes a potential 3D absolute conic, Ω^* its dual, ω its image and ω^* its dual image. True absolute conics are denoted $\Omega_\infty, \Omega^*_\infty, \omega_\infty, \omega^*_\infty$.

Affine camera: A camera whose optical plane coincides with Π_∞ is **affine** [14]. This is a good approximation for distant (and therefore large focal length) cameras viewing small objects. All visual rays except those on Π_∞ become parallel and the dual image absolute conic ω^*_∞ degenerates to rank 2. An **orthographic** camera is a calibrated affine one and has $\omega^*_\infty = \mathrm{diag}(1,1,0)$.

Kruppa constraints: Given image conics in several images, there may or may not be a 3D quadric having them as image projections. Constraints which guarantee this in two images are called **Kruppa constraints**. Any proper image conic is tangent to exactly two epipolar lines (possibly complex and/or coincident). It turns out [12, 3, 24] that

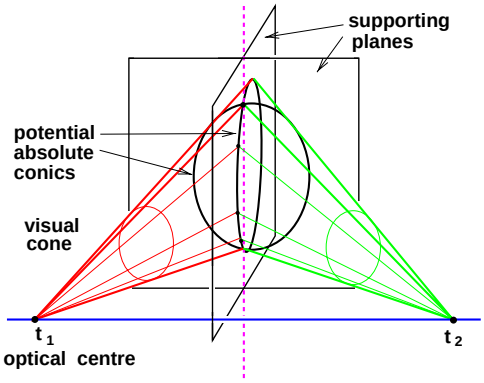


Figure 1: Intersecting the visual cones of two image conics satisfying the Kruppa constraints generates a pair of 3D conics.

there is a corresponding 3D quadric iff the tangent lines in the two images are in epipolar correspondence (see fig. 1). In fact, for non-coincident image centres and proper image conics satisfying the Kruppa constraints, there is always a linear one parameter family of 3D dual quadrics with these images. This family contains exactly two planar (rank 3) dual quadrics, and also the rank 2 one defined by (the symmetric outer product of) the two camera centres. If the image conics are virtual, the planar 3D quadrics are too and hence can serve as potential absolute conics. Thus: *In two images with distinct finite centres, a pair of proper virtual conics defines a potential 3D absolute conic iff it satisfies the Kruppa constraints, and in this case it always defines exactly two potential 3D absolute conics*[1].

The Kruppa constraints have several algebraic formulations [12, 3, 24]. Below we will use the following 3×3 symmetric rank 2 matrix version linking the two dual image conics, the fundamental matrix and one epipole:

$$\boldsymbol{F}^\top \omega_2^* \boldsymbol{F} \;\simeq\; [\boldsymbol{e}]_\times \omega_1^* [\boldsymbol{e}]_\times^\top$$

This vanishes when dotted with the epipole and only holds up to scale, so it gives only two independent constraints.

[1]With more than two images the situation is more delicate and the pairwise Kruppa constraints are *not* always sufficient to guarantee the existence of a corresponding 3D quadric.

3 Approach

We want to explicitly characterize the **critical motions** (relative camera placements) for which particular calibration constraints are insufficient to uniquely determine Euclidean 3D structure. We assume that projective structure is available. Alternative Euclidean structures correspond one-to-one with possible locations for the absolute conic in the projective reconstruction. Any proper virtual projective plane conic is potentially absolute, so we look for such conics Ω whose images also satisfy the given calibration constraints. There is ambiguity iff more than one such conic exists. We want *Euclidean* critical motions, so we work in a Euclidean frame where the true absolute conic Ω_∞ has its standard coordinates.

Several general properties help to simplify the problem:

Calibration invariance: The calibration constraints we use assert either equality between images, or that certain parameters have their 'calibrated' values $(f,a,s,u,v) = (1,1,0,0,0)$. They are satisfied for a set of cameras iff they are also satisfied when each image is premultiplied by its true inverse calibration $\boldsymbol{K}_i^{-1}$. Hence, we are free to assume that each camera is actually calibrated, $\boldsymbol{K}_i = \boldsymbol{I}$. The only difference from the fully calibrated case is that our weaker knowledge does not allow every false conic with $\omega_i^* \neq \boldsymbol{I}$ to be excluded outright.

Rotation invariance: For known-calibrated cameras $\omega_\infty^* = \boldsymbol{I}$, the image of any false AC must be identical to the image of the true one which is invariant to camera rotations. Hence, *criticality depends only on the camera centres, not on their orientations*. More generally, any camera rotation that leaves the calibration constraints intact is irrelevant. For example, arbitrary rotations about the optical axis and 180° flips about any axis in the optical plane are irrelevant if (a,s) is either $(1,0)$ or unconstrained, and (u_0,v_0) is either $(0,0)$ or unconstrained.

Translation invariance: For true or false absolute conics on the plane at infinity, translations are irrelevant so criticality depends only on camera orientation.

In essence, Euclidean structure recovery in projective space is a matter of parametrizing all of the possible proper virtual plane conics, then using the calibration constraints on their images to alge-

braically eliminate parameters until only the unique true absolute conic remains. More abstractly, if $\mathbf{C}$ parametrizes the possible conics and $\mathbf{X}$ the camera geometries, the constraints cut out some algebraic variety in $(\mathbf{C},\mathbf{X})$ space. A constraint set is useful for Euclidean SFM only if this variety generically intersects the subspaces $\mathbf{X}=\mathbf{X}_0$ in one (or at most a few) points $(\mathbf{C},\mathbf{X}_0)$, as each such intersection represents an alternative Euclidean structure for the reconstruction from that camera geometry. A set of camera poses $\mathbf{X}$ is **critical** for the constraints if it has exceptionally (*e.g.* infinitely) many intersections.

For elimination calculations, algebraic varieties are described by **ideals** (the sets of polynomials that vanish on them), which in turn are characterized by certain 'exhaustive' polynomial sets called **Gröbner bases**. Varieties can also be **decomposed** into irreducible components — a generalization of polynomial factorization that we often use as an aid to interpreting results. These are all 'standard' algebraic geometry calculations available in specialized tools like MACAULAY 2 (http://www.math.uiuc.edu/Macaulay2/) and SINGULAR, and in slightly less powerful form in general systems like MAPLE.

Potential absolute conics can be represented in several ways. The following parametrizations have all proven relatively tractable:

(*i*) Choose a Euclidean frame in which Ω^* is diagonal, and express all camera poses w.r.t. this [19, 20]. This is symmetrical w.r.t. all the images and usually gives the simplest equations, but in a frame that changes as Ω^* does. To find explicit critical motions, one must revert to camera-based coordinates which is sometimes delicate. The finite and Π_∞ cases must also be treated separately, *e.g.* $\Omega^*=\mathrm{diag}(c_1,c_2,c_3,c_4)$ with either c_3 or c_4 zero.

(*ii*) Work in the first camera frame, encoding Ω^* by its first image ω_1^* and supporting plane $(\boldsymbol{n}^\top,1)$. Subsequent images $\omega_i^* \simeq \boldsymbol{H}_i\,\omega_1^*\,\boldsymbol{H}_i^\top$ are given by the inter-image homographies $\boldsymbol{H}_i=\boldsymbol{R}_i+\boldsymbol{t}_i\boldsymbol{n}^\top$ where $(\boldsymbol{R}_i|-\boldsymbol{t}_i)$ is the i^{th} camera pose. The output is in the first camera frame and remains well-defined even if the conic tends to infinity, but the algebra required is significantly heavier.

(*iii*) Parametrize Ω^* implicitly by two images ω_1^*, ω_2^* subject to the Kruppa constraints. In the 2 image case this approach is both relatively simple and rigorous — as above, two proper virtual dual image conics satisfy the Kruppa constraints iff they define a (pair of) corresponding 3D potential absolute conics — but it does not extend so easily to multiple images.

4 Calibrated Cameras

We start with fully calibrated perspective cameras:

Theorem 4.1 *Given projective structure and calibrated perspective cameras at $m\geq 3$ distinct finite camera centres, Euclidean structure can always be recovered uniquely. With $m=2$ distinct centres there is always exactly a 2-fold ambiguity corresponding to a 'twisted pair'.*

Proof: The camera orientations are irrelevant because any false absolute conic has the same rotation invariant images as the true one. Assuming that $\boldsymbol{K}=\boldsymbol{I}$ does not change the critical motions. Calibrated cameras never admit false absolute conics on Π_∞, as the (known) visual cone of each camera intersects Π_∞ in a unique conic, which is the true AC. Given a finite false AC, work in a frame in which it is diagonal and supported on the $z=0$ plane: $\Omega^*\equiv\mathrm{diag}(c_1,c_2,0,c_4)$. Since the cameras are calibrated and their orientations are irrelevant, the conic projection in each camera becomes $(\boldsymbol{I}|-\boldsymbol{t})\,\Omega^*\,(\boldsymbol{I}|-\boldsymbol{t})^\top\simeq\boldsymbol{I}$. It is easy to show that the only solutions to this are $\Omega^*\simeq\mathrm{diag}(1,1,0,1/z^2)$ and $\boldsymbol{t}_\pm=(0,0,\pm z)^\top$ for some $z>0$. Hence, ambiguity implies that there are at most two camera centres, and the false AC is a circle of imaginary radius iz, centred in the plane bisecting the two centres.

This two-fold ambiguity corresponds exactly to the well-known **twisted pair** duality [11, 10, 15], where one of the cameras is rotated by 180° around the axis joining their two centres. The improper self-inverse projective transformation

$$\mathbf{T}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&z\\0&0&1/z&0\end{pmatrix}$$

interchanges the true and false DACs $\mathbf{T}\,\Omega^*\,\mathbf{T}^\top\simeq\Omega_\infty^*$ and takes the projection matrices $\mathbf{P}_\pm=\boldsymbol{R}_\pm\,(\boldsymbol{I}|-\boldsymbol{t}_\pm)$ to $\mathbf{P}_-\mathbf{T}^{-1}=\mathbf{P}_-$ and $\mathbf{P}_+\mathbf{T}^{-1}=-\mathbf{P}_+\boldsymbol{U}$ where $\boldsymbol{U}=\mathrm{diag}(-1,-1,1,1)$ is a 180° twisted pair rotation about the z axis. The 'twist' $\mathbf{T}$ represents a very strong projective deformation which cuts the scene

in half, moving the plane between the cameras to infinity. By considering twisted *vs.* non-twisted optical ray intersections, one can also show that it reverses the relative signs of the projective depths [21] of each correspondence, *e.g.* as recovered by the equation $\lambda_1 \boldsymbol{F}\boldsymbol{x}_1 = \lambda_2 (\boldsymbol{e} \wedge \boldsymbol{x}_2)$. Moreover, *any* proper virtual Kruppa geometry (fig. 1) has such a 'twisted pair' projective involution symmetry, so *calibrated or not, two image Euclidean structures always occur in twisted pairs*. However the twist is a simple 180° rotation only for axisymmetric DIACs.

Theorem 4.2 *Given projective structure and $m \geq 3$ scaled orthographic cameras with distinct projective centres (*i.e. *viewing directions, with diametrically opposite ones identified), Euclidean structure can always be recovered uniquely. With only $m = 2$ distinct centres there is a one parameter family of possible structures corresponding to the bas relief ambiguity [11, 10, 15, 22].*

Proof: Choose coordinates in which camera 1 has orientation $\boldsymbol{R}_1 = \boldsymbol{I}$. Orthographic and affine cameras have Π_∞ as their optical planes, so Π_∞ is known and any potential AC must lie on it. Potential DACs have the form $\Omega^* = \left(\begin{smallmatrix} \boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{smallmatrix}\right)$ for symmetric 3×3 $\boldsymbol{C}$. The orthographic calibration constraint is that $\boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^\top \simeq \mathrm{diag}(1,1)$ where $\boldsymbol{U}$ is the first two rows of $\boldsymbol{R}$. In image 1 this gives $\boldsymbol{C}_{11} - \boldsymbol{C}_{22} = \boldsymbol{C}_{12} = 0$ and two analogous constraints in image 2. Representing $\boldsymbol{R}_2$ by a quaternion $\mathbf{q}$ and eliminating $\boldsymbol{C}_{11}$ between these constraints gives

$$\begin{aligned}&\left((q_0 q_1 + q_2 q_3)\boldsymbol{C}_{13} + (q_0 q_2 - q_1 q_3)\boldsymbol{C}_{23} \right) \cdot \\ &\qquad \cdot (q_0^2 + q_3^2)(q_1^2 + q_2^2) \;=\; 0\end{aligned}$$

This must hold for any motion satisfying the constraints. The first two terms correspond to optical axis rotations and 180° flips that leave the optical centre fixed, and are therefore excluded by the statement. Solving for $\boldsymbol{C}$ in terms of $\mathbf{q}$ using the final term gives a linear family of solutions $\boldsymbol{C} \simeq \alpha \boldsymbol{I} + \beta (\mathbf{o}_1 \mathbf{o}_2^\top + \mathbf{o}_2 \mathbf{o}_1^\top)$ where $\mathbf{o}_1 = (0,0,1)^\top$ and $\mathbf{o}_2 =$ (the third row of $\boldsymbol{R}$) are the optical centres, and (α, β) are arbitrary parameters. Given $\boldsymbol{I}$ and any false DAC $\boldsymbol{C} \not\simeq \boldsymbol{I}$, we can uniquely recover the family and its two camera centres (the three rank 2 members of the family each decompose into point pairs, but only one of these is real). Since each family encodes its centres, families with distinct centres never coincide. By linearity, they therefore intersect in at most one conic. All families intersect in the true DAC $\boldsymbol{C} = \boldsymbol{I}$, so no other intersection is possible. *I.e.* false structures are impossible for orthographic images from ≥ 3 distinct centres. That the one parameter ambiguity for two cameras corresponds to the *bas relief* 'flattening' is well known [11, 10, 15, 22].

Two image orthographic absolute conic geometry is easily understood in terms of the Kruppa constraints. These are well behaved as the cameras tend to infinity, and hence still define a one parameter family of dual quadrics. However as the cameras recede and their focal length increases, their DIACs become progressively flatter and this constrains the 3D family to be flatter too, until in the limit all members of the family become infinitely flat rank 3 disk quadrics squashed onto Π_∞.

5 Focal Lengths from 2 Images

For two cameras, projective geometry is encapsulated in the 7 d.o.f. fundamental matrix, and Euclidean geometry in the 5 d.o.f. essential matrix. Hence, from 2 projective images we might hope to estimate Euclidean structure plus two additional calibration parameters. Hartley [6] gave a method for the case where the only unknown calibration parameters are the focal lengths of the two cameras. This was later elaborated by Newsam *et.al.* [16], and Zeller & Faugeras and Bougnoux [24, 2]. Hippisley-Cox & Porrill [9] give a related method for equal but unknown focal lengths and aspect ratios. All of these methods are Kruppa-based. We will give a unified presentation and derive the critical motions for the Hartley-Newsam-Bougnoux (unequal f) and Newsam (equal f) case.

Suppose that we can write all pairs of dual image conics satisfying the calibration constraints as a parametric family $(\omega_1^*(\lambda), \omega_2^*(\lambda))$. As they already obey the calibration constraints, pairs of nonsingular conics in this family represent possible 3D absolute conics iff they also satisfy the Kruppa constraints, $\boldsymbol{F}^\top \omega_2^*(\lambda) \boldsymbol{F} = \mu [\boldsymbol{e}]_\times \omega_1^*(\lambda) [\boldsymbol{e}]_\times^\top$ for some scalar μ. Solving these equations for λ, μ gives the possible image DIACs and hence 3D absolute conics. If $\omega_i^*(\lambda)$ are linear in their parameters λ, the system is bilinear in λ, μ. In particular, for zero skew and known principal point $\boldsymbol{p}_i$, $\omega_i^*(\lambda)$ is linear in f_i^2

and $(a_i\, f_i)^2$. For known a_i and unconstrained f_i, this gives fully linear equations in $\mu\, f_1^2$, μ and f_2^2:

$$\boldsymbol{F}^\top \left(f_2^2 \boldsymbol{D} + \boldsymbol{p}_2 \boldsymbol{p}_2^\top \right) \boldsymbol{F} = [\boldsymbol{e}]_\times \left((\mu\, f_2^2) \boldsymbol{D} + \mu \boldsymbol{p}_1 \boldsymbol{p}_1^\top \right) [\boldsymbol{e}]_\times^\top$$

where $\boldsymbol{D} \equiv \mathrm{diag}(1,1,0)$. Writing the 3×3 symmetric rank 2 matrices $\boldsymbol{F}^\top \boldsymbol{D} \boldsymbol{F}, \ldots, [\boldsymbol{e}]_\times \boldsymbol{p}_1 \boldsymbol{p}_1^\top [\boldsymbol{e}]_\times^\top$ as 6 vectors gives a 6×4 rank 3 homogeneous linear system $\mathbf{M}_{6\times 4}\, (f_2^2, 1, \mu\, f_1^2, \mu)^\top = 0$. This can easily be solved for μ, f_1, f_2. There are multiple solutions for f_i — and hence ambiguous Euclidean structures — iff the coefficient matrix $\mathbf{M}_{6\times 4}$ has rank ≤ 2. We will study this case below. Newsam *et.al.* [16] use the SVD of $\boldsymbol{F}$ to project 3 independent rows out of this system. Bougnoux [2] uses properties of fundamental matrices to solve it in closed form:

$$f_2^2 = -\frac{(\boldsymbol{p}_2^\top \boldsymbol{F} \boldsymbol{D} [\boldsymbol{e}]_\times \boldsymbol{p}_1)\, (\boldsymbol{p}_2^\top \boldsymbol{F} \boldsymbol{p}_1)}{\boldsymbol{p}_1^\top \boldsymbol{F}^\top \boldsymbol{D} \boldsymbol{F} \boldsymbol{D} [\boldsymbol{e}]_\times \boldsymbol{p}_1}$$

If the focal lengths are known to be equal, $f_1 = f_2 = f$, the system takes the form $\mathbf{M}_{6\times 2}(\mu) \begin{pmatrix} f^2 \\ 1 \end{pmatrix} = 0$ where $\mathbf{M}_{6\times 2}(\mu)$ is linear in μ and generically has rank 2. This system has a nontrivial solution iff all of its 2×2 minors vanish — a set of quadratic constraints on μ. If the focal lengths really are equal, at most two of these quadratics are linearly independent and we can generically eliminate the μ^2 term between them, solve linearly for μ, substitute into $\mathbf{M}_{6\times 2}$ (which then has rank 1) and solve uniquely for f^2. This fails iff all of the quadratics are: (*i*) proportional — in which case the single quadratic gives exactly two possible solutions for μ and f; (*ii*) zero — in which case $\mathbf{M}_{6\times 2} = 0$ and any f is possible. We will return to these cases below. Finally (*c.f.* [9]), equal but unknown aspect ratios and focal lengths $a_1 = a_2 = a$, $f_1 = f_2 = f$, give a 6×3 rank 3 system $\mathbf{M}_{6\times 3}(\mu)\, (f^2, (a\, f)^2, 1)^\top = 0$, which has a solution iff the determinant of any of its nontrivial 3×3 minors vanishes — a single cubic in μ, giving at most 3 solutions for μ, f, a.

Now consider the critical motions of the above methods. Assume finite a, f and $\boldsymbol{t} \neq 0$.

Theorem 5.1 *For the known a, unequal f problem, the critical motions for the Hartley, Newsam and Bougnoux methods are all identical and intrinsic to any method for this problem. In fact, they are exactly the two evident singularities of Bougnoux' equations: (*i*) $\boldsymbol{p}_2^\top \boldsymbol{F} \boldsymbol{p}_1 = 0$ and (*ii*) $\boldsymbol{p}_2^\top \boldsymbol{F} \boldsymbol{D} [\boldsymbol{e}]_\times \boldsymbol{p}_1 = 0$.*

Case (*i*) occurs when the principal points are in epipolar correspondence, *i.e.* the optical axes intersect. (*ii*) occurs whenever the point $\boldsymbol{D} [\boldsymbol{e}]_\times \boldsymbol{p}_1$ on the line at infinity in the first camera lies on the epipolar line $\boldsymbol{F}^\top \boldsymbol{p}_2$ of the other principal point. This condition is actually symmetric between the images. If $\boldsymbol{p}_1 = \boldsymbol{p}_2 = (0,0,1)^\top$, (*ii*) occurs whenever $\boldsymbol{F}^\top \boldsymbol{p}_2$ contains the direction orthogonal to the epipolar line $[\boldsymbol{e}]_\times \boldsymbol{p}_1$, *i.e.* whenever the epipolar plane of optical axis $\boldsymbol{p}_1$ is *orthogonal* to that of axis $\boldsymbol{p}_2$ [16]. If either principal point coincides with an epipole, both (*i*) and (*ii*) apply and a second order singularity occurs.

Theorem 5.2 *For the known a equal f problem, there is a unique solution for f everywhere outside the critical variety of the unequal f method. On this variety there are generically exactly two solutions corresponding to the two roots of the single surviving quadratic in μ. Both solutions may be real, or one may be imaginary ($f^2 < 0$). There are more than two real solutions (in fact any f is possible) only on the following subvarieties of the corresponding-principal-point variety $\boldsymbol{p}_2^\top \boldsymbol{F} \boldsymbol{p}_1 = 0$, where $(\boldsymbol{R}(\mathbf{q}), \boldsymbol{t})$ is the relative pose of the second camera with quaternion* $\mathbf{q}$*:*
(i) $t_3 q_2 - t_2 q_3 + t_1 q_0 = 0$ *and* $t_3 q_1 - t_1 q_3 - t_2 q_0 = 0$
(ii) $t_1 q_1 + t_2 q_2 + t_3 q_3 = 0$ *and* $t_2 q_1 - t_1 q_2 + t_3 q_0 = 0$
(iii) $q_1 = 0$ *and* $q_2 = 0$
(iv) $q_3 = 0$ *and* $q_0 = 0$

Each of these subvarieties has codimension 2 in the space of all motions, and codimension 1 in the corresponding principal point variety. (*iii*) and (*iv*) correspond to parallel optical axes (axis rotations, and 180° flips about any axis in the optical plane, plus arbitrary translation). (*ii*) requires both planar motion $\mathbf{q} \cdot \boldsymbol{t} = 0$ and corresponding principal points. The intersection of these two varieties has two components: (*a*) arbitrary planar motions when the optical axes lie in the plane (*e.g.* a driving car with forwards-pointing camera), and (*b*) 'turntable rotations' about the intersection point of the two optical axes, when these do not lie in the plane. Subvariety (*ii*) corresponds to case (*b*). Case (*a*) has two solutions for f but is generically nonsingular.

The above results are straightforward but fairly heavy to prove using the automated algebraic tools

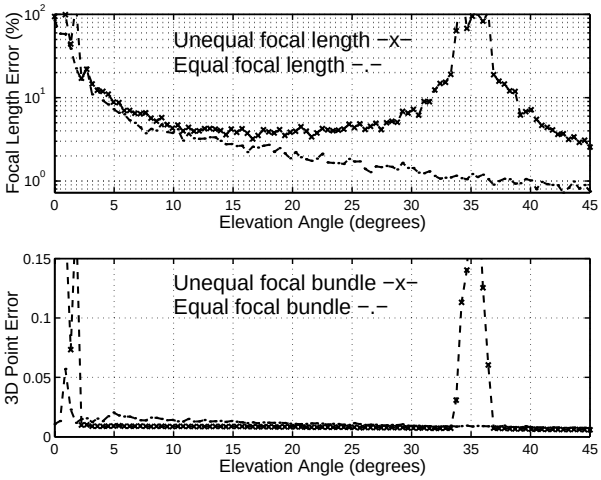


Figure 2: Relative errors in quasi-linear f and bundle-based 3D structure *vs.* camera elevation, for unequal and equal f methods.

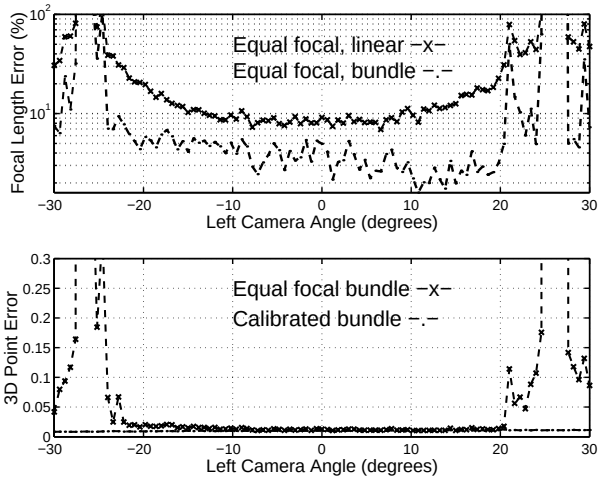


Figure 3: Errors in quasi-linear and bundle-based f, and 3D structure with unknown and known f, for equal f methods.

we are studying here. (Newsam *et.al.* [16] — a reference we were unaware of while completing this work — give a fairly simple SVD-based proof for their unequal f method, but an incomplete result for the equal f one). Since we were initially sceptical that the general Kruppa approach and Bougnoux' detailed manipulations [2] introduced no spurious ambiguities, we proved the results twice: once in a fundamental matrix / Kruppa constraint based parametrization, and once in an image conic / plane homography based one. In each case, given the parametrization we can more or less mechanically calculate and decompose the variety on which the constraints degenerate using MACAULAY 2. The calculations are 'routine', although the homography based ones are near the limits of the current system.

6 Experiments

We have performed some synthetic experiments to evaluate the effects of critical motions. We will focus on the question of how far from critical two cameras must be to get reasonable estimates of focal length and Euclidean 3D structure. The first experiment studies the unequal f case, the second the equal f one. For both experiments, two unit focal length perspective cameras view 25 points distributed uniformly within the unit sphere. Gaussian noise of 1 pixel standard deviation was added to the 512×512 images. For each pose, an optimal projective structure and fundamental matrix is estimated by projective bundle adjustment, the focal length(s) are estimated quasi-linearly as above, Euclidean bundle adjustment is applied to get Euclidean structure, and the resulting 3D error is calculated by Euclidean alignment. Means over 100 trials are shown. The Bougnoux and Newsam unequal f methods give essentially identical results: only the latter is plotted.

In the first experiment, cameras at $(-2,-2,0)$ and $(2,-2,0)$ focus on the origin. Their elevation angles are then varied, upwards for the left camera and downwards for the right one, so that their optical axes are skewed and no longer meet. Quasi-linear focal lengths and bundle adjusted Euclidean structures are estimated, both with and without the equal f constraint. Fig. 2 shows the resulting RMS errors as a function of elevation angle. At zero elevation, the optical axes intersect and the cameras are equidistant from this intersection, so both equal and unequal f methods are critical. This can be seen clearly in the graphs. The unequal f method also breaks down when the epipolar planes of the optical axes become orthogonal at around 35° elevation — the second component of the unequal f critical variety, but non-critical for the equal f method. For geometries more than about 5-10° from criticality, the unequal and equal f bundles both give results very similar to the optimal 3D structure obtained

with *known* calibration.

In the second experiment, cameras at $(-1,-2,0)$ and $(1,-2,0)$ focus on the origin, then the left camera is rotated so that its optical axis sweeps the world plane $z=0$. This is always critical for the unequal f method and the equal f one always gives two possible solutions. But in these trials, one is always tiny or imaginary and can safely be discarded. In fig. 3, the upper graph compares the quasi-linear equal f result with that obtained after optimal equal f bundle adjustment. The lower graph compares the structures obtained with equal f and known-calibration bundle adjustments. At rotation angles of around -27° the camera axes are parallel, and at around $+27^\circ$ their intersection is equidistant from both cameras. These are intrinsic equal f degeneracies, clearly visible in the graphs. Moving about 5-10° from criticality suffices to ensure reasonably accurate focal lengths and Euclidean structure.

7 Conclusions

We have explicitly described the critical motions for a number of simple calibration constraints, ranging from unknown focal lengths to fully calibrated cameras. Numerical experiments studying the effects of near-critical configurations were also presented.

One of our aims was to see what could be achieved in vision with formal ideal-theoretic calculations. It is clear that although automated tools for this (MACAULAY 2, SINGULAR, COCOA) have progressed significantly in recent years, they can not yet replace geometric intuition. Even when a calculation terminates — and the 'ceiling' for this is still frustratingly low — the geometric interpretation of the results remains a difficult 'inverse problem'. However when it comes to rigorously proving formal properties of systems of equations we have found these tools a powerful computational aid and a good deal more reliable than 'proof by intuition'. Hence, we feel that these methods do have a place in vision, particularly for studying singularities of simple algebraic (auto)calibration and camera pose methods.

We are currently investigating critical motions where even less is known about the calibration, *e.g.* cameras having zero skew and unit aspect ratio, but with the other parameters unknown and possibly varying.

References

[1] K. Åström and A. Heyden. Euclidean reconstruction from constant intrinsic parameters. In *Int. Conf. Pattern Recognition*, pages 339–343, 1996.

[2] S. Bougnoux. From projective to Euclidean space under any practical situation, a criticism of self-calibration. In *Int. Conf. Computer Vision*, 1998.

[3] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, Mass, 1993.

[4] O. Faugeras, Q.-T. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *European Conf. Computer Vision*, pages 321–334. Springer-Verlag, 1992.

[5] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *European Conf. Computer Vision*, pages 563–578. Springer-Verlag, 1992.

[6] R. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conf. Computer Vision*, pages 579–87. Springer-Verlag, 1992.

[7] R. Hartley. Euclidean reconstruction from multiple views. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.

[8] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Int. Conf. Computer Vision & Pattern Recognition*, pages 761–4, 1992.

[9] S. Hippisley-Cox and J. Porrill. Auto-calibration, Kruppa's equations and the intrinsic parameters of a camera. In *British Machine Vision Conference*, pages 771–7, 1994.

[10] B. K. P. Horn. Motion fields are hardly ever ambiguous. *Int. J. Computer Vision*, 1, 1987.

[11] H. C. Longuet-Higgins. Multiple interpretations of a pair of images of a surface. *Proc. Roy. Soc. London*, A:418, 1988.

[12] S. Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag, Berlin, Heidelberg, New York, 1993.

[13] S. J. Maybank and O. D. Faugeras. A theory of self calibration of a moving camera. *Int. J. Computer Vision*, 8(2):123–151, 1992.

[14] J. L. Mundy and A. Zisserman, editors. *Geometric invariance in Computer Vision*. MIT Press, Cambridge Ma, USA, 1992.

[15] S. Negahdaripour. Multiple interpretations of the shape and motion of objects from two perspective images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(11):1025–1039, 1990.

[16] G. Newsam, D.Q. Huynh, M. Brooks, and H.-P. Pan. Recovering unknown focal lengths in self-calibration: An essentially linear algorithm and degenerate configurations. In *Int. Arch. Photogrammetry & Remote Sensing*, volume XXXI-B3, pages 575–80, Vienna, 1996.

[17] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Int. Conf. Computer Vision*, 1998.

[18] M. Pollefeys, L. Van Gool, and M. Oosterlinck. The modulus constraint: A new constraint for self-calibration. In *Int. Conf. Pattern Recognition*, pages 349–353, 1996.

[19] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *Proc. Conf. Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 1100–1105, 1997.

[20] P. Sturm. *Vision 3D Non Calibrée: Contributions à la Reconstruction Projective et Étude des Mouvements Critiques pour l'Auto-Calibrage*. PhD thesis, Institut National Polytechnique de Grenoble, 1997.

[21] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–720, 1996.

[22] R. Szeliski and S. B. Kang. Shape ambiguities in structure from motion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(5), May 1997.

[23] B. Triggs. Autocalibration and the absolute quadric. In *Int. Conf. Computer Vision & Pattern Recognition*, 1997.

[24] C. Zeller and O. Faugeras. Camera self-calibration from video sequences: the Kruppa equations revisited. Technical Report 2793, INRIA, Sophia-Antipolis, France, 1996.

Camera Pose and Calibration from 4 or 5 known 3D Points

Bill Triggs
INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot St. Martin, France.
Bill.Triggs@inrialpes.fr ◇ *http://www.inrialpes.fr/movi/people/Triggs*

Abstract

We describe two direct quasilinear methods for camera pose (absolute orientation) and calibration from a single image of 4 or 5 known 3D points. They generalize the 6 point 'Direct Linear Transform' method by incorporating partial prior camera knowledge, while still allowing some unknown calibration parameters to be recovered. Only linear algebra is required, the solution is unique in non-degenerate cases, and additional points can be included for improved stability. Both methods fail for coplanar points, but we give an experimental eigendecomposition based one that handles both planar and nonplanar cases. Our methods use recent polynomial solving technology, and we give a brief summary of this. One of our aims was to try to understand the numerical behaviour of modern polynomial solvers on some relatively simple test cases, with a view to other vision applications.

Keywords: Camera Pose & Calibration, Direct Linear Transform, Polynomial Solving, Multiresultants, Eigensystems.

1 Introduction

This paper describes two quasilinear methods for camera pose (absolute orientation) and calibration from a single image of 4 or 5 known 3D points. The methods are 'direct' (non-iterative) and quasilinear, so: (*i*) only linear algebra is required; (*ii*) they give a unique solution in non-degenerate cases; (*iii*) additional points are easily included to improve stability; and (*iv*) all points are on an equal footing. The classical 'Direct Linear Transform' (DLT) [1, 16] recovers the 5 internal and 6 pose parameters of a fully projective camera from the images of 6 known 3D points. The new methods are analogous to the DLT, but adopt more restrictive calibration models so that: (*i*) the minimum number of points required is reduced to 4 or 5; (*ii*) the results for a given number of points are (at least potentially) more stable, as there is more prior knowledge and hence fewer unknowns to estimate from the input data. The implemented '4 point' method assumes that the focal length f is the only unknown calibration parameter, the '5 point' one that the unknowns are focal length f and principal point (u_0, v_0). Other one (4 point) or three (5 point) parameter linear calibration models could easily be implemented using the same techniques. There are also associated multi-solution methods capable of handling one additional calibration parameter apiece: at most $2^4 = 16$ solutions for pose plus 2 calibration parameters in the 4 point case, $4^2 = 16$ for 4 in the 5 point one. We will not consider these here as they yield too many solutions to be practically useful, and numerical stability is likely to be poor. However we *will* consider a related modification of the quasilinear 4 point method, which has fewer degeneracies but which may return 2 or at most 4 solutions.

Notation: $\mathbf{X}$ denotes 3D points and $\boldsymbol{x}$ image ones. We use homogeneous coordinates and the full projective camera model $\mathbf{P} = \boldsymbol{K}\boldsymbol{R}(\boldsymbol{I}|-\boldsymbol{t})$ where: $\mathbf{P}$ is the camera's 3×4 projection matrix; the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ give its orientation and position; and $\boldsymbol{K} = \begin{pmatrix} a & s & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1/f \end{pmatrix}$ is its internal calibration matrix. The calibration parameters $f, a, s, (u_0, v_0)$ are called *effective focal length*, *aspect ratio*, *skew* and *normalized principal point*. Numerically, we will assume well-normalized image coordinates based on some nominal focal length and principal point (*e.g.* the image centre). Fixed parameters are assumed to have their nominal values $(a, s, u_0, v_0) = (1, 0, 0, 0)$.

Rationale & Existing Work: Our methods use some prior calibration knowledge, and are best seen as intermediate between classical 3–4 point pose-

This paper appeared in ICCV'99. The work was supported by Esprit LTR project CUMULI. I would like to thank Peter Sturm for comments.

with-known-calibration algorithms [16, 8, 15], and ≥ 6 point DLT-like ones which assume completely unknown calibration [1, 16]. They were motivated mainly by the need for approximate camera pose + calibration to initialize bundle adjustment in close range industrial photogrammetry problems. User convenience dictates the use of as few reference points as possible: accurate 3D references are troublesome and expensive to acquire and maintain, and application constraints often mean that only a few points are visible from any given location. As the bundle adjustment can correct quite a lot of residual error, stability is more important than high precision. This suggests the use of simple approximate camera models with minimal free parameters. Aspect ratio a and skew s are both stable and easily measured, so they can usually be pre-calibrated. In contrast, the 'optical scale' parameters focal length f and principal point (u_0, v_0) are difficult to pre-calibrate. Even with a fixed lens they vary slightly with focus, aperture, mechanical/thermal motion of the lens mount, and (with lens distortion) image position. Radial lens distortion is also significant in many close range applications, but we will not consider it here as it is difficult to handle in our DLT-like framework. See [16, 2] for extensions of the DLT which partially account for lens distortion.

Degeneracy is a significant problem for all calibration methods using near-minimal data: for certain relative positionings of the points and camera, there are infinitely many solutions and the method fails. Coplanar reference objects are especially easy to manufacture and measure. But all 6 point DLT-like methods fail for planar scenes, and *any* method with free focal length (including all of ours) fails for *frontoparallel* planes, as forward motion is indistinguishable from zoom. This is problematic as near-planarity and frontoparallelism are common in practice. A planar scene gives only two constraints on the calibration ("the images of the plane's two circular points must lie on the image of the absolute conic" [20, 11, 18, 22]). As there are 5 calibration parameters, at least 3 prior constraints are required to recover from planarity. Our 5 point method has only 2 prior constraints, so it must (and does) fail for planes. The 4 point quasilinear method should do better, but in fact it also fails owing to an algorithm-specific rank deficiency. In contrast, relatively simple homography-based methods [21, 10, 18, 22][1] solve the 4 point planar pose + focal length problem rather stably (barring fronto- and other axis parallelisms). Unfortunately, these methods fail for more than about 5% non-coplanarity, so it would be useful to develop algorithms for the difficult (but practically common) near-planar case. I will describe a preliminary version of such a 4 point method below, which uses recent eigenvector-based polynomial solving technology to separate the true root from the false ones. The underlying technique is worth knowing about as it potentially applies to many other vision problems with degeneracies and/or multiple roots.

Contents: §2 outlines our general approach, §3 covers the necessary material on polynomial solving, §4 summarizes the algorithms and gives implementation details, §5 describes experimental tests, and §6 concludes.

2 Approach

Each image of a known 3D point gives two linear constraints on the projection matrix $\mathbf{P}$, or equivalently two *non*linear ones on the camera pose and calibration. So from $n \geq 3$ points we can estimate at most the 6 pose parameters and $2n - 6$ calibration ones. These minimal cases lead to polynomial systems with multiple solutions. But we will see that by estimating one fewer parameter, we can convert such problems to linear null space computations which generically yield a unique solution. Hence, we can estimate pose plus $2n - 7 = 1, 3, 5$ calibration parameters quasilinearly from $4, 5, 6$ points. 6 points is the standard DLT, so we focus on the 4 and 5 point cases. For 4 points we develop methods for pose + focal length f; for 5 points, pose + f + principal point (u_0, v_0). Other selections of 1–3 of the 5 linear camera parameters f, a, s, u_0, v_0 can be handled analogously. The basic idea is to enforce the constraint that the remaining entries of (a, s, u_0, v_0) have their default values $(1, 0, 0, 0)$. '4' and '5 point' really denote the calibration model assumed, not just the minimum number of points required. All of our methods can incorporate further points on an equal footing, if available.

[1] For full 5 parameter calibration from several known planes, [18], [22] and (slightly later) I myself all independently developed essentially the same method, which is highly recommended.

Direct formulations in terms of camera calibration $\boldsymbol{K}$ and (*i*) pose $(\boldsymbol{R}, \boldsymbol{t})$ (using *e.g.* quaternions for $\boldsymbol{R}$), or (*ii*) the camera-point distances (*c.f.* [8, 15]), are possible, but seem to lead to rather unwieldy matrices. Instead, we proceed indirectly as follows: (*i*) find the linear space of 3×4 projection matrices consistent with the given points; (*ii*) recover the estimated projection matrix $\mathbf{P}$ quasilinearly from this subspace using the calibration constraints; (*iii*) extract the calibration and pose $\boldsymbol{K}, \boldsymbol{R}, \boldsymbol{t}$ from $\mathbf{P}$ in the usual way. We focus mainly on step (*ii*) which is the novel contribution.

Step 1 is very similar to the standard 6 point DLT [1, 16]. Given a 3D point $\mathbf{X}$ and its image $\lambda \boldsymbol{x} = \mathbf{P}\,\mathbf{X}$, eliminate the unknown depth λ by forming the cross-product $\boldsymbol{x} \wedge (\mathbf{P}\,\mathbf{X}) = 0$, and select two independent homogeneous linear constraints on $\mathbf{P}$ from this. (In fact, I project $\mathbf{P}\,\mathbf{X}$ orthogonal to $\boldsymbol{x}$ using $\boldsymbol{x}$'s 3×3 Householder matrix. This is slightly different, but the overall effect is similar). The constraints from n points can be assembled into a $2n \times 12$ matrix which generically has rank $\min(2n, 11)$. With the standard DLT, $n \geq 6$, the rank is generically 11, and the 12 components of the unique null vector directly give the corresponding projection matrix $\mathbf{P}$. For $n = 4, 5$ the rank is generically 8,10 leaving a $d = 4, 2$ dimensional null space. In the noiseless case, this still contains the true projection: $\mathbf{P} = \mathbf{P}(\mu) \equiv \sum_{i=1}^{d} \mu_i \mathbf{P}_i$ where the $\mathbf{P}_i$ are 3×4 projections corresponding to the d vectors of a null space basis, and μ_i are unknown parameters. The null space is calculated numerically by SVD. Even if $n > 4, 5$ and the rank is clearly greater than 8,10, we still take the $d = 4, 2$ smallest singular vectors to span the space $\mathbf{P}(\mu)$ used in the next step.

Step 2 recovers $\mathbf{P}(\mu)$ from the $\mathbf{P}_i$ by estimating μ using the calibration constraints. By the decomposition $\mathbf{P} \simeq \boldsymbol{K}\boldsymbol{R}\,(\boldsymbol{I} \mid -\boldsymbol{t})$, the 4×4 Euclidean invariant **absolute dual quadric** matrix $\boldsymbol{\Omega} \equiv \left(\begin{smallmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{smallmatrix}\right)$ projects to the **dual image of the absolute quadric** (DIAC) [19, 9, 13]

$$\boldsymbol{\omega} \equiv \mathbf{P}\,\boldsymbol{\Omega}\,\mathbf{P}^\top \simeq \boldsymbol{K}\boldsymbol{K}^\top \tag{1}$$

We use this to convert constraints on the calibration $\boldsymbol{K}$ into ones on candidate projections $\mathbf{P}(\mu)$ or their associated DIAC's $\boldsymbol{\omega} = \boldsymbol{\omega}(\mu) \equiv \mathbf{P}(\mu)\,\boldsymbol{\Omega}\,\mathbf{P}(\mu)^\top$. For the 4 point method the only unknown calibration parameter is f. The remaining parameters take their default values $a = 1$, $s = u_0 = v_0 = 0$ so $\boldsymbol{K} = \mathrm{diag}(f, f, 1)$, $\boldsymbol{K}\boldsymbol{K}^\top = \mathrm{diag}(f^2, f^2, 1)$ and the constraints (1) become

$$\boldsymbol{\omega}_{11} = \boldsymbol{\omega}_{22} \qquad \boldsymbol{\omega}_{12} = \boldsymbol{\omega}_{13} = \boldsymbol{\omega}_{23} = 0 \tag{2}$$

This overconstrained system of 4 homogeneous quadratics in 4 variables $\mu_1, \ldots, \mu_4$ generically has at most one solution. We will see below how to convert such a system into a rectangular multiresultant matrix $\mathbf{R}$ whose unique null vector encodes the solution. We can then estimate the null vector numerically (*e.g.* using SVD), extract the corresponding μ, substitute into $\mathbf{P}(\mu)$ to obtain $\mathbf{P}$, and decompose $\mathbf{P}$ to obtain full camera pose + calibration. In this case the resultant matrix turns out to be 80×56 — large, but still tractable.

The 5 point method is similar. It recovers (μ_1, μ_2) using the calibration constraints $a = 1$, $s = 0$. These are no longer linear in the entries of $\boldsymbol{K}\boldsymbol{K}^\top$, but fortunately they *are* linear in those of $\boldsymbol{\omega}^{-1} \simeq (\boldsymbol{K}\boldsymbol{K}^\top)^{-1}$, whose upper 2×2 submatrix is proportional to $\left(\begin{smallmatrix} 1 & -s \\ -s & a^2+s^2 \end{smallmatrix}\right)$. $\boldsymbol{\omega}^{-1}$ is proportional to the matrix of cofactors of $\boldsymbol{\omega}$, and hence quadratic in $\boldsymbol{\omega} = \boldsymbol{\omega}(\mu)$ or quartic in μ. The system $a = 1, s = 0$ or $\boldsymbol{\omega}_{11}^{-1} = \boldsymbol{\omega}_{22}^{-1}$, $\boldsymbol{\omega}_{12}^{-1} = 0$ becomes

$$\begin{aligned} \boldsymbol{\omega}_{22}\,\boldsymbol{\omega}_{33} - \boldsymbol{\omega}_{23}^2 &= \boldsymbol{\omega}_{11}\,\boldsymbol{\omega}_{33} - \boldsymbol{\omega}_{13}^2 \\ \boldsymbol{\omega}_{21}\,\boldsymbol{\omega}_{33} - \boldsymbol{\omega}_{23}\,\boldsymbol{\omega}_{31} &= 0 \end{aligned} \tag{3}$$

This overconstrained system of two homogeneous quartics in (μ_1, μ_2) yields an 8×8 (Sylvester) resultant matrix whose null vector again gives the solution quasilinearly.

Notes: The globally optimal $\mathbf{P}$ lies somewhere in the nonlinear variety of projection matrix space cut out by the d calibration constraints. It has low error so it is usually not far from the space spanned by the smallest few singular vectors of the DLT constraint matrix $\mathbf{A}$. This motivates the choice of the subspace $\mathbf{P}(\mu)$. But with noisy data $\mathbf{P}(\mu)$ rarely contains the exact global optimum. In fact, the calibration system has 1 redundant d.o.f. on $\mathbf{P}(\mu)$, so it seldom has *any* exact solution there, let alone an optimal one. Worst still, step 2 finds its "unique" near-solution by roughly minimizing some highly twisted heuristic form of the constraint residual, *regardless of the resulting image error*. The measured data points contribute *only* to the estimation of the "null" space $\mathbf{P}(\mu)$ in step 1. This is fine for minimal point sets where $\mathbf{P}(\mu)$ is the true null space of

the DLT constraints. But for noisy, non-minimal, well-conditioned data $\mathbf{P}(\mu)$ generally contains several far-from-null directions and there is a risk that step 2 will return a solution with quite large residual. In summary, the multiresultant solution neither exactly satisfies the constraints, nor minimizes the fitting error even within the $\mathbf{P}(\mu)$ subspace, let alone outside it. Experimentally this is verified: (*i*) nonlinear refinement significantly reduces the residual of the multiresultant solutions; (*ii*) the multiresultant methods are most suited to near-minimal data — as more data is added their performance improves comparatively little, so for well-conditioned high-redundancy data the 6 point DLT is preferable.

3 Solving Polynomial Systems

This section briefly sketches the multiresultant theory required to understand our algorithms. Part of this material is classical, but it has seen a significant revival lately and we will use some recent results. There is no space for details here, but the material deserves to be better known in the vision community as large-scale polynomial solving is rapidly becoming a feasible proposition. See, *e.g.* [4, 14] for references and further reading.

A **polynomial** $\mathbf{p}(\mathbf{x}) = \sum p_\alpha \mathbf{x}^\alpha$ in variables $\mathbf{x} = (x_1, \dots, x_n)$ is a finite sum of **coefficients** p_α times **monomials** $\mathbf{x}^\alpha \equiv \prod_{i=1}^n x_i^{\alpha_i}$, with integer **exponents** $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}^n$. For **homogeneous** polynomials, all exponents have the same **degree** $|\alpha| \equiv \sum_i \alpha_i$. Any polynomial can be **homogenized** by including an extra variable x_0 at a suitable power in each term, and de-homogenized by setting $x_0 = 1$. The product of polynomials $\mathbf{p}, \mathbf{q}$ is $(\mathbf{p}\,\mathbf{q})(\mathbf{x}) = \sum_\alpha \left(\sum_\beta p_{\alpha-\beta}\, q_\beta \right) \mathbf{x}^\alpha$. By choosing some sufficiently large list of working exponents $\mathcal{A}$ (to be specified below), we can represent polynomials as row vectors $\mathbf{p}_\mathcal{A} \equiv (\dots p_\alpha \dots)$ and monomials as columns $\mathbf{x}^\mathcal{A} \equiv (\dots \mathbf{x}^\alpha \dots)^\top$, so that $\mathbf{p}(\mathbf{x}) = \mathbf{p}_\mathcal{A} \cdot \mathbf{x}^\mathcal{A}$ is the usual row-column dot product. All of the nonlinearity is hidden in the "simple" monomial evaluation mapping $\mathbf{x} \to \mathbf{x}^\mathcal{A}$. Polynomial multiplication can be represented by matrices $\mathbf{M}_\mathcal{A}(\mathbf{q})$ acting on the right on row vectors $\mathbf{p}$: $(\mathbf{p}\,\mathbf{q})_\mathcal{A} = \mathbf{p}_\mathcal{A}\, \mathbf{M}_\mathcal{A}(\mathbf{q})$. Row α of $\mathbf{M}_\mathcal{A}(\mathbf{q})$ contains the row vector of $\mathbf{x}^\alpha\, \mathbf{q}(\mathbf{x})$, *i.e.* the coefficients of $\mathbf{q}$ 'shifted along' by α. Coefficients shifted outside of $\mathcal{A}$ are truncated, but we will use only untruncated rows.

We want to find the roots of a polynomial system $\{\mathbf{p}_1(\mathbf{x}), \dots, \mathbf{p}_m(\mathbf{x})\}$, *i.e.* the points $\mathbf{x}$ at which all $\mathbf{p}_i(\mathbf{x}) = 0$. It follows that $\sum_i \mathbf{p}_i(\mathbf{x})\, \mathbf{q}_i(\mathbf{x})$ also vanishes at all roots $\mathbf{x}$, for any other polynomials $\mathbf{q}_i(\mathbf{x})$. As row vectors, such sums are linear combinations of rows $\mathbf{x}^\alpha\, \mathbf{p}_i(\mathbf{x})$ from the multiplication matrices $\mathbf{M}_\mathcal{A}(\mathbf{p}_i)$. Gather the (untruncated) rows of these into a big 'multiresultant' matrix $\mathbf{R}$. The vanishing of $\mathbf{x}^\alpha\, \mathbf{p}_i$ at roots implies that the monomial vector $\mathbf{x}^\mathcal{A}$ of any root is *orthogonal* to all rows of $\mathbf{R}$: *The linear subspace of monomial vectors spanned by the root vectors $\mathbf{x}^\mathcal{A}$ is contained in the right null space of $\mathbf{R}$.* It turns out that by making $\mathcal{A}$ larger, this null space can often be made to 'close in' on the space spanned by the roots, until they eventually coincide. If there is only one root $\mathbf{x}$, $\mathbf{x}^\mathcal{A}$ can then be recovered (modulo scale) as the unique null vector of $\mathbf{R}$. $\mathbf{x}$ then follows easily by taking suitable ratios of components, with at most some trivial root extractions. For numerical accuracy, large-modulus components of $\mathbf{x}^\mathcal{A}$ should be selected for these ratios.

For homogeneous polynomials, roots are counted projectively in the homogeneous variables $(x_0, \dots, x_n)$. Bezout's theorem says that a system of n such polynomials of degrees d_i has either exactly $\prod_{i=1}^n d_i$ such complex roots (counted with appropriate multiplicities), or (non-generically) an infinite number. Adding further polynomials gives an overconstrained system that generically has no roots at all. But if it does have one it is generically unique and can be recovered by the above construction. In particular, for **dense** homogeneous polynomials (ones whose coefficients of the given degrees are all nonzero and generic), Macaulay's classical multiresultant [12] chooses $\mathcal{A}$ to contain all monomials of degree $D = 1 + \sum_{i=1}^{n+1}(d_i - 1)$.

Taking all untruncated rows of the multiplication matrices as above generally gives a rectangular matrix $\mathbf{R}$. Macaulay gave a prescription for choosing a *minimal* set of rows (a square $\mathbf{R}$) that (generically) suffices to generate the null space. This is useful for theory and most current multiresultant codes adopt it. But numerically it is ill-advised as nothing says that the selected rows are particularly well-conditioned. I prefer to include all available rows and use a stable numerical null space routine, either pivoting to select suitable rows, or using an orthogo-

nal decomposition like QR or SVD that averages errors over all of them. This also allows any available additional polynomials to be included on an equal footing for better stability and/or reduced degeneracy, simply by adding the appropriate rows of their multiplication matrices to $\mathbf{R}$. If some of the polynomial coefficients vanish the Macaulay construction may fail. Sparse 'Newton' multiresultants are available in such cases [7, 6, 4].

The above is all we need for the quasilinear 4 and 5 point methods, as the $\mathbf{P}_i$ and hence (2), (3) are usually dense. However, as mentioned above, the 4 point method fails unnecessarily for coplanar points. $\mathbf{R}$ develops 3 additional null vectors in this case, corresponding roughly to infinite and zero focal lengths (though not necessarily to coherent roots). The true root monomial still lies in this 4D null space, but it is no longer isolated by the null space computation alone. This failure is annoying, as coplanarity is not actually an intrinsic degeneracy of the 4 point problem. Indeed, stable specialized methods exist for the planar case [21, 10, 18, 22]. Unfortunately, these fail even for mildly non-coplanar scenes. It would be useful to develop a method that handled both cases simultaneously, and in particular the difficult near-planar region. To do this we need some more theory.

The columns of the resultant matrix $\mathbf{R}$ are labelled by the exponent set $\mathcal{A}$. If we partition $\mathcal{A}$ into subsets $\mathcal{A}_1 + \mathcal{A}_0$, $\mathbf{R}$ can be partitioned conformably (after column permutation) as $\mathbf{R} = (\mathbf{R}_1 \,|\, \mathbf{R}_0)$. Choose the partition so that: (*i*) $\mathbf{R}_1$ has full column rank $N_1 = |\mathcal{A}_1|$; (*ii*) $\mathcal{A}_0$ is relatively small and compact in the sense given below. For any left pseudoinverse[2] $\mathbf{R}_1^\dagger$ of $\mathbf{R}_1$, the column span of the $N \times N_0$ matrix $\mathbf{U} = \begin{pmatrix} -\mathbf{R}_1^\dagger \mathbf{R}_0 \\ \mathbf{I} \end{pmatrix}$ contains the null space of the columns of $\mathbf{R}$. In fact, $\mathbf{U}$ regenerates null vectors $\mathbf{v}$ from their $\mathcal{A}_0$ components: $\mathbf{R}\,\mathbf{v} = \mathbf{R}_1\,\mathbf{v}_1 + \mathbf{R}_0\,\mathbf{v}_0 = 0$ implies $\mathbf{U}\,\mathbf{v}_0 = \begin{pmatrix} -\mathbf{R}_1^\dagger \mathbf{R}_0\,\mathbf{v}_0 \\ \mathbf{v}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1^\dagger \mathbf{R}_1\,\mathbf{v}_1 \\ \mathbf{v}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_0 \end{pmatrix} = \mathbf{v}$.

Now choose a non-constant polynomial $\mathbf{q}(\mathbf{x})$ such that the row vectors $\mathbf{x}^\alpha\,\mathbf{q}$ are untruncated in $\mathcal{A}$ for all $\alpha \in \mathcal{A}_0$. (It is to avoid truncation here that $\mathcal{A}_0$ needs to be small and compact. $\mathbf{q}$ can have negative exponents if necessary). Assemble these $\mathcal{A}_0$ rows of $\mathbf{M}_\mathcal{A}(\mathbf{q})$ into an $N_0 \times N$ matrix $\mathbf{M} = (\mathbf{M}_1 \,|\, \mathbf{M}_0)$, and form the $N_0 \times N_0$ **reduced multiplication matrix**

$$\mathbf{M}_{\mathcal{A}_0}(\mathbf{q}\,|\mathbf{p}_1 \ldots \mathbf{p}_m) \;\equiv\; \mathbf{M}\,\mathbf{U} \;=\; \mathbf{M}_0 - \mathbf{M}_1\,\mathbf{R}_1^\dagger\,\mathbf{R}_0$$

What is happening here is that the polynomials $\mathbf{x}^\beta\,\mathbf{p}_i$ (the rows of $\mathbf{R}$, acting via $\mathbf{R}_1^\dagger$) have been used to eliminate the $\mathcal{A}_1$ exponents of the polynomials $\mathbf{x}^\alpha\,\mathbf{q}$, leaving a matrix on the reduced exponent set $\mathcal{A}_0$ representing *multiplication by* $\mathbf{q}$ *followed by reduction modulo (multiples of) the* $\mathbf{p}_i$. The reduction leaves the value of $\mathbf{q}$ unchanged at all roots $\mathbf{x}$ of the $\mathbf{p}_i$, as multiples of $\mathbf{p}_i(\mathbf{x}) = 0$ are added to it. Hence, using the above regeneration property, ***for any root*** $\mathbf{x}$ ***of the system*** $\{\mathbf{p}_1 \ldots \mathbf{p}_m\}$***, the monomial vector*** $\mathbf{x}^{\mathcal{A}_0}$ ***is an eigenvector of*** $\mathbf{M}_{\mathcal{A}_0}(\mathbf{q})$ ***with eigenvalue*** $\mathbf{q}(\mathbf{x})$ ***:***

$$\mathbf{M}_{\mathcal{A}_0}(\mathbf{q})\,\mathbf{x}^{\mathcal{A}_0} \;=\; \mathbf{M}\,\mathbf{x}^{\mathcal{A}} \;=\; \mathbf{q}(\mathbf{x})\,\mathbf{x}^{\mathcal{A}_0}$$

Even if we can't reduce the null space of $\mathbf{R}$ to a single vector owing to multiple roots, ill conditioning, *etc*, we can still obtain roots by solving a nonsymmetric eigenvalue problem. Given $\mathbf{x}^{\mathcal{A}_0}$ we can recover $\mathbf{x}$ as before, if necessary regenerating $\mathbf{x}^{\mathcal{A}} = \mathbf{U}\,\mathbf{x}^{\mathcal{A}_0}$ to do so. Possible problems with this construction are: (*i*) it may be impossible to find an $\mathcal{A}_0$ with well-conditioned $\mathbf{R}_1^\dagger$ and non-constant, untruncated $\mathbf{q}$; (*ii*) if the chosen $\mathbf{q}$ takes similar values at several roots, the eigenvalue routine may fail to separate the corresponding eigenspaces cleanly, leading to inaccurate results; (*iii*) post-processing is required, as some of the recovered eigenvectors may be garbage (*i.e.* vectors that define valid linear forms on polynomials, but whose components do not correspond to the monomials of any root). Beware that nonsymmetric eigenproblems are intrinsically rather delicate, and in this application can become spectacularly unstable for ill-conditioned $\mathbf{R}_1$ or ill-chosen $\mathbf{q}$. This is *not* immediately obvious from the recovered eigenvalues or eigenvectors. However the condition number of the eigenvector matrix is a fairly reliable indicator.

This multiplication matrix approach to numerical root-finding is quite recent [17, 14, 4], although its roots go back a century. So far as I know, the observation that it continues to work when $\mathcal{A}_0$ and $\mathbf{U}$ span more than the null space of $\mathbf{R}$ is new. This is numerically useful, as it allows eigensystem size

[2] *I.e.* , $\mathbf{R}_1^\dagger\,\mathbf{R}_1 = \mathbf{I}_{N_1 \times N_1}$. Such $\mathbf{R}_1^\dagger$ are easily calculated from most numerical decompositions of $\mathbf{R}_1$.

to be traded against elimination stability. This approach can be used to find all of Bezout's projective roots of a dense n polynomial system by building a Macaulay matrix with $D = 1 + \sum_{i=1}^{n}(d_i - 1)$ and choosing $\mathcal{A}_0$ to contain all monomials $\mathbf{x}^\alpha$ with $0 \leq \alpha_i < d_i$. Here, $\mathbf{R}_1$ generically spans the column space of $\mathbf{R}$, so there are no extraneous eigenvalues. Sparse analogues also exist.

We will use the eigenvector method to stabilize the 4 point quasilinear one against near-planar scenes. Coplanarity increases the null space dimension of the 4 point multiresultant $\mathbf{R}$ from 1 to 4. So we need to choose four exponents of $\mathcal{A}$ for the reduced exponent set $\mathcal{A}_0$, and the routine will return at most four potential roots. Currently I use the four lowest degree exponents $(\mu_1, \mu_2, \mu_3, 1)$ (where $\mu_4 = 1$ is the homogenizing variable). This choice parametrizes the true root and at least one false null vector stably, but it is not ideal as the remaining 1–2 false null vectors are mainly supported on 'high' exponents deep within $\mathcal{A}_1$. I know of no way around this dilemma: the supports of the null vectors are too widely separated to gather into an $\mathcal{A}_0$ supporting an untruncated $\mathbf{q}$, even if we could isolate which exponents were needed. With the heuristics discussed below, the modified 4 point routine performs tolerably well despite the fact that both $\mathbf{R}_1^\dagger$ and the eigenvalue problem are often fairly ill-conditioned, but a cleaner solution would be desirable.

4 Implementation

The steps of the new pose + calibration algorithms are as follows, where $d = 4, 2$ for the 4,5 point method:

1. Use SVD to estimate the d-D null space $\mathbf{P}(\mu) = \sum_{i=1}^{d} \mu_i \mathbf{P}_i$ of the DLT constraints $\boldsymbol{x} \wedge (\mathbf{P}\,\mathbf{X}) = 0$.

2. Substitute $\mathbf{P}(\mu)$ into the 4 quadratic calibration constraints (2) (4 point) or 2 quartic ones (3) (5 point).

3. Form the rectangular multiresultant matrix $\mathbf{R}$ of the resulting polynomials, use SVD to recover its unique null vector $\mu^{\mathcal{A}}$, and extract μ. For the eigenvector method, choose a splitting $\mathcal{A}_0$ and a compatible random polynomial $\mathbf{q}(\mu)$, use $\mathbf{R}_1^\dagger$ to form $\mathbf{q}$'s reduced multiplication matrix, extract eigenvectors $\mu^{\mathcal{A}_0}$, and recover the solutions μ.

4. (Optional) Refine the recovered roots μ by Newton iteration against the original calibration constraints.

5. Calculate the camera projection matrix $\mathbf{P}(\mu)$ and decompose it as usual to get pose + calibration.

The routines have been implemented in OCTAVE/MATLAB. The necessary multiresultant matrices were calculated using a MAPLE routine similar to [14] (available from the author). The null space methods are straightforward to implement, but the eigenvector one requires some care. The choice of the 'pivoting' exponent set $\mathcal{A}_0$ is critical, and I am not happy with the current heuristic. In fact, I have tried only the μ_4-based exponent set, but varied which of the projection matrices $\mathbf{P}_i$ (the d smallest right singular vectors of the DLT equations) is assigned to μ_4. I tried various permutations and also random orthogonal mixings. None are wholly satisfactory and a more effective pivoting strategy is clearly required before the eigenvalue approach can be routinely used to rescue resultants from multi-root degeneracies. For 4 points and near-planar scenes, making $\mathbf{P}_4$ correspond to the *greatest* of the 4 singular values is by far the best choice. But it performs erratically for non-coplanar scenes and $n > 4$ points. Changing strategies makes enormous differences to the conditioning of $\mathbf{R}_1$, but does not necessarily stop the routine from working. Slight ($\mathcal{O}(10^{-10})$) damping of the pseudoinverse is also essential with the current $\mathcal{A}_0$, as $\mathbf{R}_1$ actually becomes singular for coplanar points.

Another issue for the eigenvector method is the choice of multiplier polynomial $\mathbf{q}(\mathbf{x})$. For simplicity I have used a linear $\mathbf{q}$, although anything up to 4^{th} order could be handled. For maximum stability, it is important that $\mathbf{q}$ should take well-separated values at different roots. In practice, I randomly choose a few $\mathbf{q}$'s and take the one that gives the best conditioned eigensystem. The cost is negligible compared to the calculation of $\mathbf{R}_1^\dagger$.

The current implementations use SVD for all null space computations. This is perhaps overkill, but it guarantees the stablest possible results. Speed is

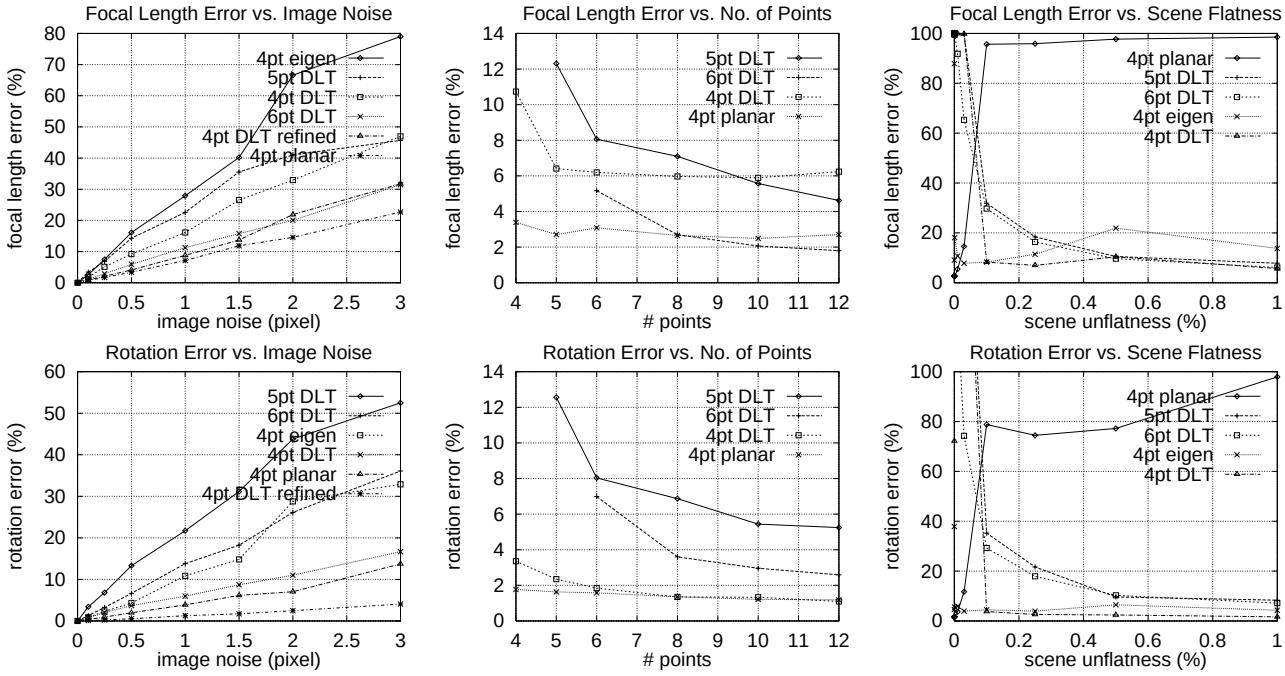


Figure 1: *Left:* Focal length & rotation error *vs.* noise, for each method's minimal point number and preferred scene flatness. *Middle:* Error *vs.* number of points for 0.5 pixels noise. *Right:* Error *vs.* scene flatness for minimal point numbers.

adequate (< 1 second), but might become an issue if the 4 point methods were used in a RANSAC loop.

The roots μ are recovered by selecting suitable large-modulus components of $\mu^{\mathcal{A}}$ and taking their ratios. Optionally, they may then be 'refined' by a simple Newton iteration that minimizes the error in the calibration polynomials (2),(3) over μ. For the best results the original calibration constraints should be used, not their resultant matrix $\mathbf{R}$. Full Newton rather than Gauss-Newton iteration is advisable here, owing to the nonlinearity of the constraints.

5 Experiments

The graphs show some simple experimental tests on synthetic data. The 3D test points are well spread and by default non-coplanar. They are viewed from about 5 scene diameters by a 512×512 camera with $f \approx 1000 \pm 400$ and a default Gaussian noise of 0.5 pixels (which is easily obtainable with marked target points). Median errors over 300 trials are reported. For flat scenes, the plane is viewed at about $30 \pm 15^\circ$ from normal to avoid the frontoparallel degeneracy, which all of the algorithms here suffer from.

The graphs show that all methods are quite sensitive to noise, but all scale linearly with it up to at least 50% relative error. The planar 4 point f-only method [21, 10, 18, 22] is both simpler and intrinsically stabler than the 3D ones, but it can not tolerate more than about 5% non-coplanarity. Plane + parallax might be an interesting approach for pose + calibration from flat scenes. The 5 and 6 point DLT's fail for scenes within about 20% of planarity, whereas the 4 point DLT one (whose failure is algorithmic not intrinsic) continues to work down to around 10%. The 4 point eigenvector method works even for planar scenes, but overall it is somewhat erratic. (*E.g.* it gives better results for near-planar scenes, and for 4 points rather than $n > 4$). As above, this is due to the lack of a good policy for the choice of the residual exponent set $\mathcal{A}_0$.

The performance of the 5 point DLT is somewhat disappointing. The traditional 6 point DLT is always preferable when there are $n \geq 6$ points, and for $n \geq 10$ even beats the 4 point DLT on f (but not on orientation). In general the relative rankings depend somewhat on the error measure chosen. The fact that the 6 point DLT does better than the 4–5 point ones for large numbers of points is annoying but not unexpected. As discussed in section 2, it

happens because the multiresultant step blindly minimizes some sort of twisted constraint residual over the subspace $\mathbf{P}(\mu)$, without any consideration of the image errors produced. For redundant data $\mathbf{P}(\mu)$ usually contains projections with significant image error, hence the problem. I am currently working on this, but for now the 4 and 5 point methods are most useful for minimal and near-minimal data.

The '4pt DLT refined' method runs Newton's method on the output of the linear 4 point one, to minimize the RMS error of the calibration constraints. Such nonlinear refinement is highly recommended, as it reduces the overall residual error by a factor of 2–5. A mini bundle adjustment over the resulting pose estimate would do even better, as it would not be restricted to the d-D 'null space' of the DLT constraints. The large reduction in residual suggests that there is considerable scope for improving the heuristic least squares error function embodied in the multiresultant root estimate. However, except for the initial DLT step, simple rescaling has little effect: the multiresultant is insensitive to the scaling of its input data over a range of at least $10^{\pm 2}$.

Use of the rectangular multiresultant is recommended, as it makes the results significantly more consistent, allows additional points to be incorporated, and reduces errors by 20–40% compared to the square Macaulay resultant.

All of the methods give more accurate relative results as f grows larger and the camera recedes, simply because a larger magnification camera with the same pixel noise is a more accurate angle measurer. Conversely, for small f angular errors and perspective become large and the problem becomes very nonlinear: spurious roots near $f \approx 0$ are common in (auto-)calibration problems. This makes it clear that $1/f$ is a natural expansion parameter, and suggests that pseudo-affine initialization may be a good implementation strategy for pose + calibration methods, *c.f.* [5, 3].

6 Summary and Conclusions

The 4 point quasilinear pose method performs reasonably well considering how much information it extracts from such a small amount of input data. The 5 point method is less good and is probably best reserved for special situations. Both methods are most useful for minimal or near-minimal data. Neither competes with the traditional 6 point DLT when there are ≥ 6 well-spaced points, and hence neither realizes my hopes that calibration constraints could be used to stabilize the 6 point method. The reason is basically the splitting of the problem into 'DLT' and 'multiresultant' parts with different, incompatible error metrics. This sort of subdivision is commonplace in vision geometry, but it is clear that it prevents the data and constraints from being combined very effectively. I am currently reflecting on better ways to handle this. Also, the whole issue of scaling, pivoting, and the effective error metric used by polynomial methods like multiresultants remains very unclear. But the numerical side of this field is very recent, and significant improvements are to be expected over the next few years.

The use of oversized, rectangular multiresultant matrices $\mathbf{R}$ improves the numerical conditioning and also allows redundant data to be included, so it should help to make polynomial-based initialization of many vision optimization problems more feasible. For more difficult cases where there are multiple near-roots and other degeneracies, the eigenvector method has considerable potential. However, if my current experience with the 4 point eigenvector method is any guide, more work on pivoting/exponent choice strategies is essential to make numerically trustworthy.

References

[1] Y. Abdel-Aziz and H. Karara. Direct linear transformation from comparator to object space coordinates in close-range photogrammetry. In *ASP Symp. Close-Range Photogrammetry*, pages 1–18, Urbana, Illinois, 1971.

[2] K. B. Atkinson. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, Roseleigh House, Latheronwheel, Caithness, Scotland, 1996.

[3] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 18(11):1098–1104, 1996.

[4] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Graduate Texts in Mathematics, vol.185. Springer Verlag, 1998.

[5] D. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *Int. J. Computer Vision*, 15(1/2):123–141, 1995.

[6] I. Emiris and J. Canny. Efficient incremental algorithms for the sparse resultant and the mixed volume. *J. Symbolic Computation*, 20:117–49, 1995.

[7] I. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants*. Birkhaüser, Boston, 1994.

[8] R.M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Analysis and solutions of the three point perspective pose estimation problem. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 592–8, 1991.

[9] A. Heyden and K. Åström. Euclidean reconstruction from constant intrinsic parameters. In *Int. Conf. Pattern Recognition*, pages 339–43, Vienna, 1996.

[10] K. Kanatani. Camera calibration using a planar surface. Unpublished, November 1998.

[11] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 482–8, 1998.

[12] F. Macaulay. *The Algebraic Theory of Modular Systems*. Cambridge University Press, 1916.

[13] S. J. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. J. Computer Vision*, 8(2):123–151, 1992.

[14] B. Mourrain. An introduction to linear algebra methods for solving polynomial equations. In *HERMCA'98*, 1998. See also: http://www-sop.inria.fr/saga/logiciels/multires.html.

[15] L. Quan and Z.D. Lan. Linear $n \geq 4$-point pose determination. In *IEEE Int. Conf. Computer Vision*, 1998.

[16] C.C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.

[17] H. J. Stetter. Eigenproblems are at the heart of polynomial system solving. *SIGSAM Bulletin*, 30:22–5, 1996.

[18] P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *IEEE Conf. Computer Vision & Pattern Recognition*, 1999.

[19] B. Triggs. Autocalibration and the absolute quadric. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[20] B. Triggs. Autocalibration from planar scenes. In *European Conf. Computer Vision*, pages I 89–105, Freiburg, June 1998.

[21] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robotics & Automation*, 3(4):323–344, 1987.

[22] Zhengyou Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, December 1998.

Annexe B

Autres activités scientifiques

Ici je regroupe brièvement quelques autres indications de mes activités scientifiques récentes :

- conférencier invité à :
  - MICROSOFT RESEARCH, Seattle (2 fois) ;
  - workshop CVPR'99 « MView'99 – Multi-View Modeling and Analysis of Visual Scenes », Fort Collins, Colorado ;
  - workshop « J.-O Eklundh » en honneur du 60^{e} anniversaire de Jan-Olof EKLUNDH, Stockholm, Suède ;
- animateur du workshop majeur « Vision Algorithms : Theory and Practice » à ICCV'99, avec Andrew ZISSERMAN et Richard SZELISKI ;
- responsable député de (et collaborateur scientifique sur) le projet Esprit LTR 21914 CUMULI ;
- membre du comité de programme des conférences internationaux CVPR et ECCV et de divers workshops ;
- relecteur pour plusieurs journaux et autres conférences internationales (IJRR, IJCV, PAMI, CVGIP, ICCV, SIGGRAPH, RFIA).

Bibliographie

[ÅH96] K. Åström and A. Heyden. Multilinear constraints in the infinitessimal-time case. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 833–8, San Francisco, 1996.

[ÅH98] K. Åström and A. Heyden. Continuous time matching constraints for image streams. *Int. J. Computer Vision*, 28(1):85–96, 1998.

[Alo93] Y. Aloimonos, editor. *Active Perception*. L. Erlbaum Associates, 1993.

[Bou98] S. Bougnoux. From projective to Euclidean space under any practical situation, a criticism of self-calibration. In *IEEE Int. Conf. Computer Vision*, 1998.

[BRZM95] P. Beardsley, I. Reid, A. Zisserman, and D. Murray. Active visual navigation using non-metric structure. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 58–64, Cambridge, MA, June 1995.

[Buc92] T. Buchanan. Critical sets for 3D reconstruction using lines. In *European Conf. Computer Vision*, pages 730–738. Springer, 1992.

[BY92] A. Blake and A. Yuille, editors. *Active Vision*. MIT Press, 1992.

[Car95] S. Carlsson. Duality of reconstruction and positioning from projective views. In P. Anandan, editor, *IEEE Workshop on Representation of Visual Scenes*. IEEE Press, 1995.

[CDLS99] R.G. Cowell, A.P. Dawid, S. Lauritzen, and D.J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

[Dem88] M. Demazure. Sur deux problèmes de reconstruction. Technical report, INRIA, 1988.

[Fau92] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[Fau93] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

[Fau95] O. Faugeras. Stratification of 3-d vision: Projective, affine, and metric representations. *J. Optical Society of America*, A 12(3):465–84, March 1995.

[FB81] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Computer Graphics and Image Processing*, 24(6):381–395, 1981.

[FLM92] O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[FM95a] O. Faugeras and B. Mourrain. About the correspondences of points between n images. In *IEEE Workshop on Representations of Visual Scenes*, pages 37–44, Cambridge, MA, June 1995.

[FM95b] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[FvDFH91] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1991.

[HÅ95] A. Heyden and K. Åström. A canonical framework for sequences of images. In *IEEE Workshop on Representations of Visual Scenes*, Cambridge, MA, June 1995.

[Har92] R. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conf. Computer Vision*, pages 579–87. Springer-Verlag, 1992.

[Har93a] R. Hartley. Euclidean reconstruction from multiple views. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.

[Har93b] R. Hartley. Extraction of focal lengths from the fundamental matrix. Unpublished manuscript, 1993.

[Har94] R. Hartley. Self-calibration from multiple views with a rotating camera. In *European Conf. Computer Vision*, pages 471–8. Springer-Verlag, 1994.

[Har95a] R. Hartley. In defence of the 8-point algorithm. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1064–70, Cambridge, MA, June 1995.

[Har95b] R. Hartley. A linear method for reconstruction from lines and points. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 882–7, Cambridge, MA, June 1995.

[Har97] R.I. Hartley. Self calibration of stationary cameras. *Int. J. Computer Vision*, 22(1):5–23, 1997.

[HCP94] S. Hippisley-Cox and J. Porrill. Auto-calibration, Kruppa's equations and the intrinsic parameters of a camera. In *British Machine Vision Conference*, pages 771–7, 1994.

[Hey95] A. Heyden. Reconstruction from image sequences by means of relative depths. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1058–63, Cambridge, MA, June 1995.

[HGC92] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 761–4, Urbana-Champaign, Illinois, 1992.

[HM93] R. Horaud and O. Monga. *Vision par ordinateur : outils fondamentaux*. Hermes, Paris, 1993.

[HO93] K. Hanna and N. Okamoto. Combining stereo and motion analysis for direct estimation of scene structure. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 357–65, 1993.

[HP47] W. V. D. Hodge and D. Pedoe. *Methods of Algebraic Geometry*, volume 1. Cambridge University Press, 1947.

[Jac97] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to caharacterizing intensity images. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 206–212, Puerto Rico, 1997.

[JW76] F. Jenkins and H. White, editors. *Fundamentals of Optics*. McGraw-Hill, 1976.

[Kan93] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993.

[Kle39] F. Klein. *Elementary Mathematics from an Advanced Standpoint*. Macmillan, New York, 1939. (2 vols).

[KT99] F. Kahl and B. Triggs. Critical motions in euclidean structure from motion. In *IEEE Conf. Computer Vision & Pattern Recognition*, Fort Collins, Colorado, 1999.

[Lau96] S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[LF94] S. Laveau and O. Faugeras. 3d scene representation as a collection of images and fundamental matrices. Technical Report RR-2205, INRIA, Sophia Antipolis, France, February 1994.

[LZ98] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 482–8, 1998.

[MBB93] R. Mohr, B. Boufama, and P. Brand. Accurate relative positioning from multiple images. Technical Report RT-102, LIFIA, INRIA Rhône-Alpes, Grenoble, France, 1993. Submitted to *Journal of Artificial Intelligence*.

[MF92] S. J. Maybank and O. Faugeras. A theory of self calibration of a moving camera. *Int. J. Computer Vision*, 8(2):123–151, 1992.

[NHBP96] G. Newsam, D.Q. Huynh, M. Brooks, and H.-P. Pan. Recovering unknown focal lengths in self-calibration: An essentially linear algorithm and degenerate configurations. In *Int. Arch. Photogrammetry & Remote Sensing*, volume XXXI-B3, pages 575–80, Vienna, 1996.

[Per88] J. Perl. *Probabilistic Reasoning in Intelligent Systems*. Morgan-Kaufmann, 1988.

[PGP96] M. Pollefeys, L Van Gool, and M. Proesmans. Euclidean 3d reconstruction from image sequences with variable focal length. In B. Buxton and R. Cipolla, editors, *European Conf. Computer Vision*, pages 31–42, Cambridge, U.K., April 1996.

[PZ98] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *IEEE Int. Conf. Computer Vision*, pages 754–760, Bombay, January 1998.

[RJ86] L.R. Rabiner and B.H. Juang. An introduction to hidden Markov models. *IEEE Acoustics, Speech and Signal Processing Magazine*, pages 4–16, January 1986.

[SA99] S. Seitz and P. Anandan. Implicit representation and scene reconstruction from probability density functions. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 28–34, 1999.

[SIR95] H.Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 17(9):854–867, 1995.

[SK52] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1952.

[Sla80] C.C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.

[SM99] P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *IEEE Conf. Computer Vision & Pattern Recognition*, 1999.

[SMB98] C. Schmid, R. Mohr, and Ch. Bauckhage. Comparing and evaluating interest points. In *IEEE Int. Conf. Computer Vision*, pages 230–235, January 1998.

[SS97] G. Stein and A. Shashua. Model-based brightness constraints: On direct estimation of structure and motion. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 400–406, 1997.

[ST96] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–20, Cambridge, U.K., 1996. Springer-Verlag.

[Sto91] J. Stolfi. *Oriented Projective Geometry*. Academic Press, 1991.

[Stu97a] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *IEEE Conf. Computer Vision & Pattern Recognition*, Puerto Rico, 1997.

[Stu97b] P. Sturm. *Vision 3D non calibrée : contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage*. Ph.D. Thesis, Institut National Polytechnique de Grenoble, December 1997.

[SW95] A. Shashua and M. Werman. On the trilinear tensor of three perspective views and its underlying geometry. In *IEEE Int. Conf. Computer Vision*, Boston, MA, June 1995.

[TGDK99] T. Tuytelaars, L. Van Gool, L. D'haene, and R. Koch. Matching affinely invariant regions for visual servoing. In *IEEE Int. Conf. Robotics and Automation*, May 1999.

[TK92] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Computer Vision*, 9(2):137–54, 1992.

[Tri] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to *Int. J. Computer Vision*.

[Tri95] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.

[Tri96a] B. Triggs. Factorization methods for projective structure and motion. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 845–51, San Francisco, CA, 1996.

[Tri96b] B. Triggs. Linear projective reconstruction from matching tensors. In *British Machine Vision Conference*, pages 665–74, Edinburgh, September 1996.

[Tri97a] B. Triggs. Linear projective reconstruction from matching tensors. *Image & Vision Computing*, 15(8):617–26, August 1997.

[Tri97b] B. Triggs. A new approach to geometric fitting. Available from http://www.inrialpes.fr/movi/people/Triggs, 1997.

[Tri98] B. Triggs. Optimal estimation of matching constraints. In R. Koch and L. Van Gool, editors, *Workshop on 3D Structure from Multiple Images of Large-scale Environments SMILE'98*, Lecture Notes in Computer Science. Springer, 1998.

[Tri99] B. Triggs. Differential matching constraints. In *IEEE Int. Conf. Computer Vision*, pages 370–6, 1999.

[VF95] T. Viéville and O. Faugeras. Motion analysis with a camera with unknown and possibly varying intrinsic parameters. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 750–6, Cambridge, MA, June 1995.

[VF96] T. Viéville and O. Faugeras. The first order expansion of motion equations in the uncalibrated case. *Computer Vision and Image Understanding*, 64(1):128–46, 1996.

[Wib76] T. Wiberg. Computation of principal components when data are missing. In J. Gordesch and P. Naeve, editors, *Proc. 2nd Symposium on Computational Statistics*, pages 229–236, Berlin, 1976.

[WW92] A. Watt and M. Watt. *Advanced Animation and Rendering Techniques*. ACM Press, 1992.

[ZF96] C. Zeller and O. Faugeras. Camera self-calibration from video sequences: the Kruppa equations revisited. Technical Report 2793, INRIA, INRIA Sophia-Antipolis, France, 1996.

[Zha98] Zhengyou Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, December 1998.