Vision 3D multi-images : contribution à l'obtention de solutions globales par optimisation polynomiale et théorie des moments

Florian Bugarin

▶ **To cite this version:**

Florian Bugarin. Vision 3D multi-images : contribution à l'obtention de solutions globales par optimisation polynomiale et théorie des moments. Automatique / Robotique. Institut National Polytechnique de Toulouse - INPT, 2012. Français. NNT : 2012INPT0068 . tel-04275974v2

HAL Id: tel-04275974

https://theses.hal.science/tel-04275974v2

Submitted on 8 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Automatique et Robotique

Présentée et soutenue par :
Florian Bugarin

le : vendredi 5 octobre 2012

Titre :

Vision 3D multi-images : contribution à l'obtention de solutions globales par optimisation polynomiale et théorie des moments

---

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
LAAS

Directeur(s) de Thèse :
Didier Henrion – LAAS-CNRS
Pr. Jean-José Orteu – Institut Clément Ader

Rapporteurs :

Pr. François Glineur – Université catholique de Louvain
Pr. Richard Hartley – Australian National University

Membre(s) du jury :

Pr. Monique Laurent – Centrum Wiskunde & Informatica
Jean-Bernard Lasserre (Encadrant) – LAAS-CNRS
Thierry Sentenac (Encadrant) – Institut Clément Ader
Peter Sturm – INRIA Grenoble

Remerciements

Ma thèse aura été un formidable investissement psychologique et physique, largement ponctué de formidables moments d'amitiés et de doutes que j'ai envie de retracer un peu ici. Je vais sûrement être trop long, mais il faut bien donner de quoi lire à ceux qui ne comprendront rien à ce travail, je pense particulièrement, et affectueusement, à ma famille et à nombre d'amis et de collègues pour qui ce travail restera à jamais énigmatique. Bien que tous se soient montrés curieux de mes travaux, ce fut toujours à grande peine que je leur répondais qu'il m'était difficile d'expliquer simplement ce que je faisais ; que ce texte soit pour eux l'occasion de découvrir, à défaut du contenu, l'environnement de ma thèse.

Ainsi, je remercie tout d'abord messieurs **Gérard Bernhart** et **Thierry Cutard**, directeurs successifs de feu le CROMeP et de l'Institut Clément Ader - Albi, ainsi que la direction d'ARMINES, de m'avoir permis d'effectuer ma thèse au sein du laboratoire. Compte tenu de mon statut particulier, je les remercie d'avoir toujours tout mis en œuvre pour me permettre de travailler dans les meilleures conditions possibles. De manière générale, cette thèse a été possible grâce à l'effort de tous les collègues de mon laboratoire : **Jean-José**, **Fabrice**, **Esther**, **Catherine**, **Yannick**, **Philippe**, **Farhad**, **Denis**, **Luc**, **Florentin**, **Christine**, **Olivier**, **Gilles**, **Pascal**, **Laurent**, **Thierry**, **Vincent**, **Vanessa** , **Mohamed**, **Jean-Paul**, **Didier**, **Sabine**, **Vincent**, **Jean-Michel**, **Thomas**, **Fabrice**, **Karim** et **Serge**. Je les remercie tous chaleureusement pour leurs encouragements et leur support quotidien.

Par ailleurs, je remercie aussi **Isabelle Queinnec** et **Denis Arzelier**, responsables successifs du groupe MAC, de m'avoir accueilli dans leur équipe au sein du LAAS et de m'avoir simplifié bon nombre de démarches administratives parfois tortueuses.

C'est une grande joie pour moi de pouvoir remercier mes directeurs de thèse **Jean-José Orteu** et **Didier Henrion** ainsi que mes encadrants **Jean-Bernard Lasserre** et **Thierry Sentenac**. Après m'avoir recruté au sein du laboratoire en tant qu'ingénieur et initié à la vision artificielle, je suis heureux de pouvoir remercier **Jean-José Orteu**, pour son support et sa patience. En effet, bien que nos caractères soient diamétralement opposés, il a toujours fait le difficile effort d'essayer de me comprendre et pris le temps de m'amener à réfléchir, tant bien que mal, de manière pragmatique. Je remercie mon autre directeur **Didier Henrion**, pour sa disponibilité, ses conseils, son écoute et sa pédagogie. En effet, il a su devancer mes interrogations et toujours leur trouver, malgré la difficulté du sujet et mes maladresses scientifiques, les réponses les plus compréhensibles possible. Je remercie ensuite **Thierry Sentenac**, pour ses encouragements dans les moments de doutes, sa rigueur scientifique et ses remarques toujours pertinentes. Enfin, je remercie **Jean-Bernard Lasserre**, chantre de la mesure, de m'avoir formé à l'optimisation polynomiale et à la théorie des moments et d'avoir fait preuve d'une grande disponibilité pour répondre à mes questions, qui s'avéraient souvent être les mêmes tournées d'une autre manière.

D'une manière générale, je vous remercie tous les quatre pour votre encadrement humain et la patience incroyable dont vous avez fait preuve face à un doctorant qui n'avait de cesse de faire le contraire de ce que vous lui disiez. J'espère que je vous aurais rendu fier du travail que nous avons fait ensemble.

Je tiens aussi à remercier **Michel Devy** qui fut à l'origine de ce travail et pour sa bien-

veillance tout au long de ma thèse.

Je voudrais ensuite remercier les membres de mon jury de thèse, et tout d'abord les rapporteurs de ce travail : monsieur **François Glineur**, dont les remarques méticuleuses m'ont permis de corriger un grand nombre d'imprécisions et monsieur **Richard Hartley** qui, outre ses précieux retours, a eu la gentillesse de s'être déplacé jusqu'à Albi et la patience de supporter mon débit de parole (beaucoup) trop rapide durant la soutenance. Je remercie aussi madame **Monique Laurent** pour sa lecture pertinente du mémoire et ses critiques constructives et monsieur **Peter Sturm** pour m'avoir fait l'honneur de présider ce jury. Enfin, je remercie Adrien Bartoli pour sa disponibilité, ses nombreux conseils et pour m'avoir reçu au sein de son laboratoire afin que je puisse y exposer mes travaux.

Je me suis tourné vers mon domaine mathématique, l'optimisation, grâce à plusieurs enseignants de l'Université Paul Sabatier : **Jean-Baptiste Hiriart-urruty**, **Marcel Mongeau** et **Sophie Jan**. Tranchant avec l'attitude, volontiers élitiste, du milieu de la recherche mathématique, ils ont su épauler l'élève moyen que j'étais et m'aider à trouver ma voie. Je les remercie tous les trois pour leur soutien, avec une pensée particulière pour **Marcel** que je remercie chaleureusement pour son amitié, son enthousiasme inébranlable et sa bonne humeur communicative.

Je remercie aussi les enseignants que j'ai eu la chance de croiser à l'ENSEEIHT : **Joseph Noailles**, **Joseph Gergaud** et **Jean-Baptiste Caillaud** pour leur aide et leur soutien tout au long de mon cursus.

Enfin, je remercie **Gérard Soubry** sans qui rien n'aurait était possible.

En tant qu'ingénieur, j'ai été amené à travailler avec plusieurs doctorants : **Nicolas**, **Jacques**, **Benoit** et **Rémi**. Grâce à leur enthousiasme à communiquer leurs recherches et leurs idées, je leur dois une grande partie des connaissances qui m'ont permis d'écrire ce manuscrit et les remercie pour tout le temps qu'ils ont passé à me les transmettre.

Le bon déroulement d'une thèse est intimement lié à l'environnement humain que l'on côtoie quotidiennement. Ainsi, je remercie :

- les nombreux doctorants et post-doctorants croisés au sein de l'ICA-A qui, au mépris de mon caractère que je qualifierais pudiquement « de légèrement renfrogné », m'ont permis de partager de bons moments. Sans être totalement exhaustifs (les oubliés voudront bien me pardonner), je citerai : **Pauline**, **Anthony**, **Fabio**, **Claire**, **Ines**, **Raffaele**, **Maxime**, **Anaïs**, **Cédric**, **Medhi**, **Myriam**$\cdots$
- **Jean-Louis**, mon compagnon d'enseignement depuis huit ans.
- mes collègues du Groupe MICS : **Yannick**, **Jean-Noël**, **Philippe**, **Marianne** et **Marie-Laetitia** pour la bonne humeur et l'ambiance cordiale qu'ils font régner au sein du groupe. Une mention spéciale à **Yannick** qui, malgré mon aversion patente à la physique et mon indisposition chronique à l'expérimentation, tente patiemment de me convertir aux vertus salvatrices de la métrologie.
- mes collègues techniciens de l'ICA-A **Didier** et **Jean-Michel** (pour leur action au sein de mon comité de soutien) ainsi que **Fabrice** (pour les moments de détentes caféinés) et **Serge** (pour me donner un aperçu réaliste de mon caractère dans 20 ans)
- **Esther** et **Cathy** pour leur bienveillance quasi-maternelle
- tous mes collègues du LAAS : **Denis** (en particulier pour la cellule « soutient psychologique »), **Isabelle**, **Sophie**, **Dimitri**, **Frédéric** et **Christophe** qui se rappelleront longtemps de notre déplacement à Luchon et de mes talents de chauffeur, ainsi que **Patrick Danès** pour ses nombreux encouragements. **Germain**, mon compagnon de bureau, mérite un remerciement spécial pour m'avoir supporté (dans tous les sens du terme), encouragé dans les moments de doutes et rigolé de bon cœur à mes blagues

téléphoniques. Je remercie aussi tous les doctorants du groupe MAC et en particulier **Mathieu** et **Tung**, mes deux compagnons de voyage dans les contrées sauvages de la théorie de la mesure.

J'en arrive enfin à remercier mes amis qui ont suivi mon travail sans pouvoir le comprendre :

- **Laurent** pour tout le temps passé à écouter mes trop nombreuses interrogations métaphysiques et son amicale persévérance à essayer d'y répondre.
- **Denis** pour sa bonne humeur quotidienne, les concerts, les restos et sa détermination fanatique à me convertir à la sainte musique de Paul le messie et de ses apôtres John, George et Ringo
- **Aurélien**, **Laurent**, **Rémi**, **Vanessa**, **Gilles**, **Vincent**, **Didier** et **Christine** pour leur soutien constant et leur indulgence quotidienne envers mon caractère très légèrement grincheux.
- mes compagnons du co-voiturage passé et présent : **Rachel**, **Séverine**, **Jean-Jacques**, **Bruno**, **Renaud**, **Christine** et **Henri** pour leur tolérance envers mes retards et tous les bons moments passés en leur compagnie.
- **Mathieu** pour son amitié et son soutient depuis mes jeunes années d'étudiant jusqu'aux péripéties de la thèse.
- tous les rolling-stones et en particulier : **Jean-Michel**, **Pascal** et **Wilfried** que je remercie autant pour leur soutien que pour le fait de savoir que je pouvais toujours compter sur eux. Un remerciement spécial à **Jean-Pierre** pour sa bienveillante amitié, toutes nos discussions et sa patience à soigner mes maladies imaginaires.
- **Yoann**, mon premier « maitre de recherche », et **Véronique** pour leur amitié
- **Thomas** et **Lionel** pour leur si constante fraternité, il y a si longtemps que nous traînons ensemble que je leur dois plus qu'il n'y a de place dans ces remerciements.

Je voudrais que mes parents trouvent dans ce travail la récompense pour leur amour et leurs efforts d'éducation ainsi que mon immense gratitude à leur égard. Je remercie **Germain** pour son affection et pour avoir fait un si long déplacement dans ses contrées Tarnaises si éloignées de sa capitale Mauzacaise. Je remercie **Ghyslaine** et **Alain** pour leur patience et leur soutien. Quand ils sauront lire, j'espère que mes enfants trouveront dans ce petit texte la raison de ces journées passées devant un ordinateur à écrire cette chose bizarre et inconnue qui s'appelle « thèse » et qu'ainsi ils voudront bien me pardonner du temps que je n'ai pas pu passer avec eux. Je terminerai en remerciant **Géraldine** pour toutes ces petites complicités, ces attentions discrètes et ces milliers de sourires qui ont façonné ma vie.

« Impose ta chance, serre ton bonheur et va vers ton risque. A te regarder, ils s'habitueront. »

Les Matinaux (1950) – René Char

Table des matières

1 Introduction générale **1**
1.1 Vision par ordinateur . . . 1
1.2 La recherche de minimum . . . 2
1.3 Motivation . . . 3
1.4 Contribution . . . 4
1.5 Organisation du mémoire . . . 5

2 Rappels d'optimisation numérique **7**
2.1 Introduction . . . 7
2.1.1 Rappels et Définitions . . . 8
2.1.2 Classification des méthodes d'optimisation . . . 9
2.1.3 Outils de l'optimisation . . . 10
2.2 Méthodes d'optimisation locale . . . 12
2.2.1 Conditions d'optimalité . . . 12
2.2.2 Méthodes sans contraintes . . . 14
2.2.3 Méthodes avec contraintes . . . 28
2.3 Méthodes d'optimisation globale . . . 33
2.3.1 Les méthodes déterministes . . . 33
2.3.2 Les méthodes stochastiques . . . 35
2.4 Conclusion . . . 37

3 Optimisation globale via moments et polynômes positifs **39**
3.1 Introduction à l'optimisation par théorie des moments . . . 41
3.1.1 Notion de mesure. . . . 43
3.1.2 Moments et polynômes positifs. . . . 44
3.1.3 Optimisation globale polynomiale par théorie des moments. . . . 49
3.1.4 Faible couplage des variables. . . . 53
3.1.5 Bilan. . . . 57
3.2 Extension au cas d'une somme de fractions rationnelles . . . 58
3.2.1 Les épigraphes creux. . . . 59
3.2.2 Programmation rationnelle. . . . 61
3.2.3 Bilan. . . . 66
3.3 Résultats numériques . . . 67
3.3.1 Problèmes univariés . . . 68
3.3.2 Problème bivarié . . . 74
3.3.3 Problème de Kowalik . . . 75
3.3.4 Exemple multivarié creux . . . 76
3.3.5 Fonctions de Shekel . . . 80
3.3.6 Bilan . . . 82
3.4 Conclusion . . . 82

4 Application à la vision multivues **83**
4.1 Applications de l'optimisation polynomiale 84
4.1.1 Rappels de géométrie projective 84
4.1.2 Minimisation des distorsions projectives 87
4.1.3 Bilan 98
4.1.4 Estimation de la matrice fondamentale 102
4.2 Application de la programmation rationnelle à la triangulation multivues . . . 124
4.2.1 Rappels de reconstruction 3D multivues 124
4.2.2 Etat de l'art 133
4.2.3 Applications à la triangulation multivues 143
4.2.4 Conclusions 157

5 Conclusions et perspectives **161**
5.1 Bilan des contributions 161
5.1.1 Extension de l'optimisation polynomiale à l'optimisation rationnelle . 161
5.1.2 Applications à la vision multivues 162
5.2 Perspectives et potentiel de la méthode d'optimisation 163

A Images tests stéréoscopiques **165**

B Images tests de la série House **167**

C Paires d'images tests Library et Merton **169**

D Images tests de la série Dinosaure **171**

E Matrice et vecteur du deuxième problème de Shekel **175**

Bibliographie **177**

Table des figures

1.1 Présentation des différentes méthodes classiquement utilisées en vision par ordinateur pour résoudre un problème d'optimisation non-linéaire. La branche rouge correspond aux méthodes locales classiques (p.e. les méthodes de descentes). Dans cette branche, si le problème n'est pas a priori convexe, le minimum global n'est pas garanti. La Branche verte représente la famille de méthodes dites de simplification. Ces méthodes déterminent une solution approchée en résolvant un problème simplifié puis en projetant la solution obtenue sur l'espace des contraintes. Enfin la dernière branche (bleue) correspond à la famille des méthodes dites globale. Parmi ces méthodes, on peut distinguer deux ramifications (cf. [Mongeau 2007]), les méthodes exhaustives balayant au mieux l'espace de recherche et les méthodes dédiées basées sur l'utilisation de la théorie mathématique. 3

2.1 Illustration de la règle de Wolfe et de Goldstein. L'espace en vert décrit les pas acceptables pour les deux règles. 16
2.2 Diagramme représentant l'ensemble des familles de l'optimisation mono-objectif présentées dans ce chapitre. 38

3.1 Diagramme récapitulatif de l'optimisation par théorie des moments. 42
3.2 La fonction rationnelle objectif représentée sur l'ensemble $[-10, 10]$. 68
3.3 Fonction objectif du problème de maximisation de Wilkinson pour $N = 5$ . . 69
3.4 Fonction coût du problème (3.128) et, en vert, la solution obtenue par la méthode de Programmation rationnelle 70
3.5 Fonction coût du problème 1D à hauts degrés, la solution certifiée par la méthode de Programmation Rationnelle (en vert) et la solution, non certifiée, extraite de la méthode par Epigraphes Creux. 71
3.6 Fonction coût du problème à deux minimia 72
3.7 Solution calculée par la Programmation Rationnelle 73
3.8 Fonctions coûts et lignes niveau du problème 2D à degrés élevés pour $N = 5$ $N = 10$ 74
3.9 Nombre de moments en fonction du nombre de variables 78
3.10 Temps de calcul (secondes) en fonction du nombre de variables 78
3.11 Ecart sur l'optimum en fonction du nombre de variables 79
3.12 Ecart relatif à la valeur objectif pour les deux méthodes en fonction du nombre de variables 79
3.13 Fonction de Shekel et lignes de niveau pour les deux premières variables dans le domaine $[0, 10] \times [0, 10]$ 80
3.14 Fonction de Shekel et lignes de niveau pour les deux premières variables dans le domaine $[0, 10] \times [0, 10]$ 81

4.1 Relations géométriques entre deux caméras **sténopé** (**pinhole** en anglais) . . 85
4.2 Opération de rectification 86

4.3 Diagramme récapitulatif des méthodes non-symétriques de type **Mallon** . . . 90
4.4 Distorsions avant et après rectification : à gauche, les déformations dues à une modification d'orthogonalité, et à droite, les déformations dues à une modification du rapport d'aspect. 96
4.5 Schéma explicatif du processus d'évaluation de la sensibilité des critères de rectification en fonction de l'angle moyen. Les centres optiques $(\mathsf{C}_i)_{i=1,2,3,4}$ étant coplanaires, on calcule l'angle $\alpha_i = \widehat{\mathsf{C}_i\mathsf{C}_{i+1}}$ dans le plan des abscisses. Puis, on effectue une rectification entre toutes les vues i et $i+1$ (flèches noires), puis toutes les vues i et $i+2$. 97
4.6 Illustration d'appariements ayant des variances en équilibre (à droite) et d'appariements ayant un rapport de variance trop élevé (à gauche) 103
4.7 Illustration des distances épipolaires (en vert) entre un couple de points appariés (croix) et leurs lignes épipolaires correspondantes 105
4.8 Projection du cube dans la caméra en position initiale (I) et dans la caméra après application des transformations rigides $[\mathbf{R}_1\ t_1]$ (II) et $[\mathbf{R}_2\ t_2]$ (III). . . . 112
4.9 Pour les deux mouvements, $[\mathbf{R}_1\ t_1]$ (colonne de gauche) et $[\mathbf{R}_2\ t_2]$ (colonne de droite), erreurs de reprojection (e_r) et nombre d'itérations en fonction du bruit ($1^{\text{ère}}$ et $2^{\text{ème}}$ lignes) et du nombre d'appariements ($3^{\text{ème}}$ et $4^{\text{ème}}$ lignes). 113
4.10 Pour le mouvement $[\mathbf{R}_1\ t_1]$, erreurs de reprojections (e_r) et nombre d'itérations en fonction du nombre de points pour un bruit gaussien de variance égale à 1. 114
4.11 Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images A, B, C et D 117
4.12 Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images de la série **Library** 118
4.13 Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images de la série **Merton1** 118
4.14 Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images de la série **Merton2** 119
4.15 Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images de la série **Merton3** 119
4.16 Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus pour la paire d'images **Cylindre**. Les points appariés sont situés dans la boite englobante bleue. 120
4.17 Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus pour la paires d'images **Endoscope** 120
4.18 Moyenne des erreurs de reprojection initiales en fonction de l'angle moyen entre les centres optiques des caméras pour la série **Dinosaur** 121
4.19 Moyenne des erreurs de reprojection finales en fonction de l'angle moyen entre les centres optiques des caméras pour la série **Dinosaur** 121
4.20 Nombre d'itérations moyen effectués par l'ajustement de faisceaux en fonction de l'angle moyen entre les centres optiques des caméras pour la série **Dinosaur** 122
4.21 Moyenne des erreurs de reprojection initiales en fonction de l'angle moyen entre les centres optiques des caméras pour la série **House** 122

4.22 Moyenne des erreurs de reprojection finales en fonction de l'angle moyen entre les centres optiques des caméras pour la série **House** 123
4.23 Nombre d'itérations moyen effectués par l'ajustement de faisceaux en fonction de l'angle moyen entre les centres optiques des caméras pour la série **House** . 123
4.24 Les trois transformations linéaires du modèle **sténopé**. La première envoie le repère $\mathcal{R}^w$ de la scène dans celui centré sur la caméra $\mathcal{R}^c$. La seconde projette $\mathcal{R}^c$ dans le plan **rétinien**. Enfin la dernière, exprime le repère ainsi projeté dans le repère de l'image. 126
4.25 Transformations linéaires d'un banc stéréoscopique. 129
4.26 Transformations linéaires d'un système multivues. Le repère $\mathcal{R}_c^1$ se déplace vers le repère $\mathcal{R}_c^3$ soit directement par la transformation $\mathbf{T}_{1\to 3}$ soit par la composée des applications $\mathbf{T}_{2\to 3} \circ \mathbf{T}_{1\to 2}$ 132
4.27 Forme générale de la matrice J_r 136
4.28 Forme générale de la matrice H_r 136
4.29 Découpage de la matrice H_r 138
4.30 La fonction maximum d'une famille de fonctions quasi-convexe est quasi-convexe 140
4.31 Illustation de l'écart entre $\mathsf{conv}f_1 + \mathsf{conv}f_2$ et $\mathsf{conv}(f_1 + f_2)$. Le premier graphique (I) représente les deux courbes f_1 et f_2. Le deuxième (I) représente (en vert) l'enveloppe convexe de f_1, $\mathsf{conv}f_1$, et celle de f_2, $\mathsf{conv}f_2$. (III) représente, $\mathsf{conv}(f_1 + f_2)$, l'enveloppe convexe de $f_1 + f_2$ et (IV) la somme de $\mathsf{conv}f_1$ et $\mathsf{conv}f_2$. 142
4.32 Les trois configurations utilisées pour nos simulations : un mouvement latéral (**1**), une translation vers la scène (**1**), et un mouvement circulaire centré sur la scène (**3**). 144
4.33 Erreurs relatives de reconstruction 3D $\mathcal{E}_{3D}$ en fonction du bruit inséré dans les points projetés pour les trois types de configurations (de haut en bas, la configuration (**1**), puis la (**2**) et enfin la (**3**)). 146
4.34 Erreurs relatives de reconstruction 3D $\mathcal{E}_{3D}$ en fonction du bruit inséré dans les matrices de projection perspective $\mathbf{P}^j$ pour les trois types de configurations. . 148
4.35 Pour les trois types de configurations, erreurs moyennes de reconstruction 3D en fonction du nombre de vues avec un bruit Gaussien de variance $\sigma = 0,4$ inséré dans les points projetés. 152
4.36 Pour les trois types de configurations, erreurs moyennes de reconstruction 3D en fonction du nombre de vues avec un bruit Gaussien de variance $\sigma = 1$ inséré dans les points projetés. 153
4.37 Images du cylindre étalon, de 70mm de rayon interne, à partir duquel nous réalisons notre série de tests. 154
4.38 CAO du cylindre étalon utilisé pour nos tests. 154
4.39 Images du cylindre sur lequel un motif aléatoire a été projeté afin de permettre un appariement de points par des méthodes de corrélation d'images numériques. Les appariements se situent dans les boites englobantes bleues. 155
4.40 Ecart à la forme nominale du nuage 3D triangulé avec l'approche locale . . . 155
4.41 Ecart à la forme nominale du nuage 3D triangulé avec l'approche globale au sens de la norme L_∞ 156
4.42 Ecart à la forme nominale du nuage 3D triangulé avec notre approche globale 156
4.43 Images du plan à partir duquel nous réalisons notre test de reconstruction 3D avec prise en compte des distorsions. 157
4.44 Erreurs relatives de reconstruction 3D en fonction du bruit inséré dans les points projetés pour les trois types de configurations. 158

4.45 Erreurs relatives de reconstruction 3D en fonction du bruit inséré dans les matrices de rotation pour le mouvement de rotation. 159
4.46 Erreurs relatives de reconstruction 3D en fonction du bruit inséré dans le vecteur de translation lors du déplacement vers la scène. 159
4.47 Erreurs relatives de reconstruction 3D en fonction du bruit inséré dans le vecteur de translation lors du déplacement latéral à la scène. 160

5.1 Diagramme récapitulatif de l'optimisation par théorie des moments. 162

A.1 Paire d'images **Arch**. 165
A.2 Paire d'images **Drive**. 165
A.3 Paire d'images **Lab**. 166
A.4 Paire d'images **Roof**. 166
A.5 Paire d'images **Boxes**. 166
A.6 Paire d'images **Yard**. 166

C.1 Paire d'images **Library**. 169
C.2 Paire d'images **Merton1**. 169
C.3 Paire d'images **Merton2**. 170
C.4 Paire d'images **Merton3**. 170

1 Introduction générale

Sommaire

1.1 Vision par ordinateur 1
1.2 La recherche de minimum 2
1.3 Motivation 3
1.4 Contribution 4
1.5 Organisation du mémoire 5

1.1 Vision par ordinateur

De la peinture à l'image, saisir et représenter le monde qui l'entoure a toujours été une source de stimulation intellectuelle pour l'homme. Les capteurs numériques modernes capturent ainsi notre environnement tridimensionnel sur des images bidimensionnelles. Cette capture se fait donc au détriment d'une information suite à la perte d'une dimension. Il en est de même pour la perception humaine. Notre cerveau forme sur la rétine des images en deux dimensions. Cependant, à l'aide de traitements complexes, celui-ci est capable de percevoir la profondeur et les mouvements ainsi que de reconnaître des objets ou des formes. Par analogie, lorsque le capteur numérique joue le rôle des yeux et l'ordinateur celui du cerveau, le but de la vision par ordinateur consiste à retrouver de telles informations à partir d'images. La vision artificielle est donc composée de plusieurs problématiques. Dans ce mémoire, nous nous focaliserons sur une seul : la reconstruction tridimensionnelle. Dans ce problème, l'entrée visuelle est composée d'au moins deux images bidimensionnelles. Le résultat est la profondeur, c'est-à-dire la structure tridimensionnelle de la scène (3D) dont sont issues les images. Divers indices, comme la géométrie du capteur, le mouvement entre deux prises de vue, les variations d'ombre, de flou ou d'éclairage, peuvent être utilisés pour atteindre cet objectif. L'exploitation de toutes ces informations se fait au sein d'un cadre unifié : la géométrie projective. Cette théorie permet, entre autre, de modéliser mathématiquement une caméra, c'est à dire le transfert d'un point de la scène observée vers un point de l'image. Le modèle de caméra classiquement utilisé est celui dit en « trou d'épingle » ou sténopé. Dans ce modèle, l'image se forme par une projection centrale sans tenir compte des éventuelles distorsions liées aux

lentilles de l'objectif. Ces distorsions étant négligeables dans la plupart des applications, cette approximation est réaliste en pratique. La formation d'une image résulte donc d'un processus relativement élémentaire. Cependant, le processus inverse, c'est-à-dire retrouver des formes 3D (points, lignes...) à l'aide de leurs projections 2D dans les images, est beaucoup plus complexe. En effet, les problèmes mathématiques qui résultent de ce modèle général sont souvent formulés comme une recherche d'optimum (minimum ou maximum) dont la résolution s'avère souvent difficile.

1.2 La recherche de minimum

Le principal avantage du cadre général que constitue la géométrie projective réside dans la possibilité de formuler la recherche d'informations 3D comme des problèmes d'estimation formulés de la manière suivante :

$$\inf_{x \in K} f(x) \tag{1.1}$$

où f est une fonction rationnelle de $\mathbb{R}^n$ dans $\mathbb{R}$ et K un sous-ensemble de $\mathbb{R}^n$. Cependant, les objets que l'on cherche à estimer n'ont, parfois, qu'une existence purement mathématique (p. ex. matrice fondamentale) dans un espace théorique (p. ex. $\mathbb{P}^2$, l'ensemble des droites vectorielles de $\mathbb{R}^2$). Il est donc, en pratique, difficile de pouvoir exhiber ne serait-ce qu'un élément de l'ensemble des contraintes K. Bien entendu, les recherches menées en vision artificielle ont permis, dans de nombreux problèmes concrets, de pouvoir construire efficacement un élément de K. Il est alors courant de résoudre (1.1) en utilisant des méthodes d'optimisation locale. Sauf si le problème est convexe, ces approches itératives ne convergent généralement pas vers le minimum global et nécessitent de fournir une solution initiale proche de la solution exacte. Une autre démarche classique réside dans la simplification intentionnelle du problème de minimisation. La manière la plus naturelle de le simplifier consiste à ne pas tenir compte des contraintes, donc à minimiser f sur tout $\mathbb{R}^n$, puis à projeter la solution obtenue sur K. Cependant, ce processus n'est possible que si l'on dispose de l'opération de projection sur K. Or, déterminer un tel opérateur peut se révéler, selon la forme de K, aussi difficile que de résoudre le problème original. De plus, même si une projection peut être déterminée, rien n'assure a priori qu'elle soit proche d'un optimum, même local. Une autre simplification possible consiste à ajouter ou à abandonner des équations de K afin de rendre le problème linéaire. De cette manière, il devient plus simple de calculer un élément de K et ainsi d'énumérer plusieurs solutions initiales. Cependant, non seulement la solution obtenue n'est pas globale mais l'ajout ou la suppression d'équations peut créer des singularités artificielles qui n'étaient pas présentes dans le problème initial. Sauf hypothèse de convexité, les solutions que nous venons d'énumérer ne garantissent pas d'obtenir le minimum global. Or, dans certaines applications de vision artificielle, comme la métrologie dimensionnelle, il est nécessaire d'avoir un maximum de précision et donc des solutions garanties. De plus, une garantie d'optimalité est une information importante permettant de détecter des configurations géométriques dégénérées.

Théoriquement, les méthodes d'optimisation globale peuvent s'affranchir d'une estimée initiale et garantissent d'atteindre le minimum global. Il y a deux manières d'apporter un tel certificat. La première façon consiste à décrire l'espace de recherche le plus exhaustivement possible de sorte qu'un maximum de candidats soient testés. A titre d'exemple, on peut citer dans cette catégorie les méthodes de Monte-Carlo qui testent aléatoirement des éléments de K ou la recherche tabou qui permet de continuer la recherche même après avoir trouvé un minimum local. L'inconvénient majeur de ces méthodes réside principalement dans leur temps de calcul prohibitif. La seconde catégorie de méthodes apporte un certificat d'optimalité à l'aide de la théorie mathématique dont elles sont issues. Les algorithmes par séparation et évaluation (« branch and bound ») ou les méthodes d'optimisation par analyse d'intervalles

font, par exemple, partie de cette deuxième famille. Cependant, bien que ces méthodes puissent être plus rapides que celles de la catégorie précédente, leur désavantage majeur est leur manque de généralité. En effet, ces approches sont en général dédiées à un type de fonctions coûts. Elles utilisent ainsi un maximum d'informations théoriques afin d'être le plus efficace possible. L'ensemble de ces différentes approches est résumé sur la figure 1.1.

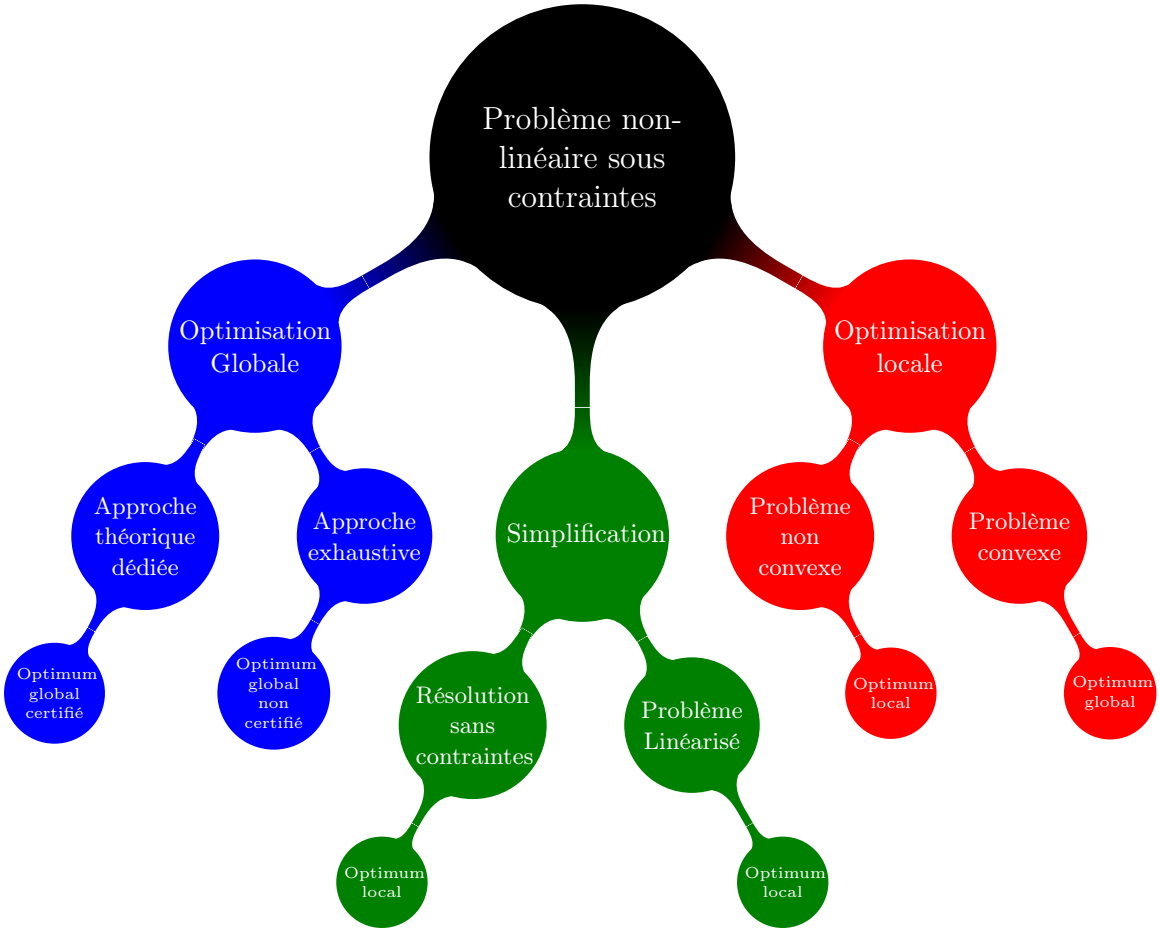


FIGURE 1.1 – Présentation des différentes méthodes classiquement utilisées en vision par ordinateur pour résoudre un problème d'optimisation non-linéaire. La branche rouge correspond aux méthodes locales classiques (p.e. les méthodes de descentes). Dans cette branche, si le problème n'est pas a priori convexe, le minimum global n'est pas garanti. La Branche verte représente la famille de méthodes dites de simplification. Ces méthodes déterminent une solution approchée en résolvant un problème simplifié puis en projetant la solution obtenue sur l'espace des contraintes. Enfin la dernière branche (bleue) correspond à la famille des méthodes dites globale. Parmi ces méthodes, on peut distinguer deux ramifications (cf. [Mongeau 2007]), les méthodes exhaustives balayant au mieux l'espace de recherche et les méthodes dédiées basées sur l'utilisation de la théorie mathématique.

1.3 Motivation

Il existe deux catégories de fonctions coûts qui sont fréquemment rencontrées en vision par ordinateur. Dans la première catégorie, il y a les fonctions coûts dites **algébriques**. Ces fonctions sont en général des polynômes. Elles sont définies dans les espaces projectifs $\mathbb{P}^2$ et $\mathbb{P}^3$ (ensemble des droites vectorielles de $\mathbb{R}^3$) et ont, souvent, pour objectif la recherche d'invariants. La minimisation de ces fonctions est théoriquement assujettie à des contraintes

de structures caractérisant ces invariants et des contraintes permettant d'éviter la solution triviale nulle. Dans les décennies précédentes, beaucoup de recherches ont été menées afin d'appliquer les méthodes d'optimisation globale pour résoudre ces problèmes algébriques. Le problème majeur de ces applications résidait dans la difficulté de prendre en compte les contraintes. Via des simplifications du problème, ces approches ont alors été principalement utilisées afin de rechercher un point de départ pour des méthodes locales.

La deuxième catégorie de fonctions coûts fréquemment rencontrées en vision artificielle sont les fonctions **métriques** ou **euclidiennes**. Elles sont, pour la plupart, issues de la projection des fonctions coûts algébriques dans les espaces euclidiens associés (c.-à-d. $\mathbb{R}^2$ et $\mathbb{R}^3$) et sont, en général, formulées comme des sommes de fractions rationnelles :

$$\min_{x \in K} \sum_{i=1}^{N} \frac{p_i(x)}{q_i(x)}, \tag{1.2}$$

où les fonctions p_i et q_i sont des polynômes. Lorsque le nombre d'inconnues et le nombre de fractions sont faibles, il est possible, par réduction au même dénominateur et en recherchant des points stationnaires, de résoudre globalement un tel problème dans un temps raisonnable. Mais, même avec un nombre d'inconnues limité, cette technique devient inutilisable lorsque N grandit. Or, les problèmes de minimisation rationnelle sont souvent utilisés pour estimer des paramètres. Lors de ces estimations, N représente le nombre de mesures expérimentales qui doit être grand (typiquement entre 50 et 100) pour que l'estimation soit précise. Ainsi, bien que cette approche naturelle puisse résoudre un large éventail de problèmes, elle n'est que peu utilisée en pratique. Cependant, les récents résultats dans les domaines de l'optimisation convexe et polynomiale ont permis l'émergence de nouvelles approches. Ces dernières ont suscité un grand intérêt dans la communauté vision par ordinateur. Ainsi, plusieurs nouvelles méthodes garantissant des solutions globales ont été étudiées et ont permis de résoudre nombre de problèmes rationnels. Les plus populaires sont les algorithmes de séparation et évaluation dans lesquels la phase d'évaluation est améliorée en faisant des hypothèses de quasi-convexité sur les numérateurs et dénominateurs (cf. [Kahl 2008a, Fangfang 2007, Kahl 2008b]). Cependant, bien que certains problèmes satisfassent à ces propriétés, ces méthodes ne peuvent résoudre le problème de minimisation rationnelle dans toute sa généralité. Afin de traiter ce problème sans faire d'hypothèses sur les fractions $\frac{p_i}{q_i}$, certaines méthodes (p. ex. [Kahl 2007]) ont proposé un cadre général de résolution à l'aide d'une hiérarchie de relaxations du problème initial. Cependant, ces méthodes ne disposent pas d'une preuve de convergence et permettent seulement de fournir une estimée intiale. Ainsi, aucune technique n'a fourni un cadre générique de résolution.

1.4 Contribution

Dans nos travaux, nous nous intéresserons premièrement à l'application de la méthode d'optimisation globale via la théorie des moments aux problèmes algébriques. Cette méthode est initialement dédiée à la minimisation de polynômes avec un ensemble de contraintes définies par des équations polynomiales. Elle a, tout d'abord, été utilisée avec succès pour résoudre un large éventail de problèmes classiques [Kahl 2007], puis employée pour des problèmes plus spécifiques comme l'estimation de la conique absolue [Chandraker 2007]. Dans ce mémoire, nos contributions consistent à :

- étudier l'application de l'optimisation polynomiale par théorie des moments à deux problèmes classiques en vision par ordinateur : la minimisation des distorsions projectives issues de l'opération de rectification et l'estimation de la matrice fondamentale. Pour le premier problème, notre contribution résidera dans la proposition de nouveaux critères

permettant l'application de la méthode polynomiale. Dans le second problème, nous chercherons à déterminer une matrice 3×3 non symétrique de rang 2. Par rapport aux solutions d'application déjà existantes, nous montrerons comment formuler le problème de la manière la plus avantageuse pour l'optimisation par théorie des moments. Puis, à l'aide d'un processus d'évaluation adéquat, nous comparerons cette méthode à la méthode d'estimation de référence.

– étendre l'optimisation polynomiale par théorie des moments à des fonctions coûts décrites par des sommes de fractions rationnelles. Nous proposerons deux méthodes permettant de telles extensions que nous comparerons sur des cas tests théoriques. Ensuite, nous appliquerons la solution retenue aux problèmes de la triangulation multivues classique (projective) et de la triangulation multivues avec prise en compte des distorsions radiales.

1.5 Organisation du mémoire

Ce mémoire est organisé de la manière suivante :

Le chapitre 2 présente un exposé exhaustif des méthodes d'optimisation numérique. L'objectif de ce chapitre est de fournir au lecteur un « arbre de décision » des méthodes d'optimisation. Le but de cet arbre est de pouvoir situer chacune des approches utilisées dans la suite du document. Dans une première partie, nous introduirons succintement les éléments théoriques nécessaires à la compréhension des approches locales et globales présentées dans le chapitre. Ensuite, nous présenterons plusieurs méthodes locales classiquement utilisées en vision par ordinateur. Puis nous exposerons brièvement différentes méthodes d'optimisation globale. Certaines ne sont que peu utilisées en vision artificielle mais elles seront introduites afin d'aider le lecteur à bien faire le distinguo entre les méthodes aléatoires et les méthodes à certificat. Enfin, nous ferons un bilan permettant de situer au mieux la branche de l'optimisation dans laquelle se situent nos travaux et les méthodes qui nous serviront de comparaison dans la suite du document.

Le chapitre 3 a pour but d'introduire la technique d'optimisation globale polynomiale sur laquelle sont basés nos travaux, ainsi que son extension au cas d'une somme de fractions rationnelles. Dans un premier temps, nous ferons de brefs rappels sur les mesures et leurs moments. Ensuite, nous introduirons la méthode proprement dite. Puis nous développerons les deux extensions que nous proposons pour résoudre le problème rationnel. Enfin, nous testerons ces deux approches sur plusieurs cas tests afin, d'une part de vérifier leur efficacité, et d'autre part d'identifier laquelle est la plus adaptée à notre problématique.

Le chapitre 4 a pour but de présenter les applications des techniques d'optimisation polynomiale et rationnelle vues dans le chapitre précédent. Nous étudierons tout d'abord les applications de l'optimisation polynomiale. Nous commencerons par une application simple : la minimisation des distorsions projectives. Pour cet exemple, nous mettrons en avant l'existence de nouveaux critères polynomiaux. Parmi ces critères, à l'aide de l'optimisation polynomiale, nous montrerons lequel est le plus performant. Ensuite, nous nous focaliserons sur un problème plus complexe : l'estimation de la matrice fondamentale. En effet, ce problème, dans la forme où nous le traitons, comporte neuf variables d'optimisation et deux contraintes polynomiales. Enfin, nous appliquerons notre méthode d'optimisation rationnelle au problème de triangulation multivues. Ce problème constitue en lui-même un problème difficile mais qui peut être résolu à l'aide de sa structure particulière. Pour finir, nous appliquerons notre méthode sur le problème de triangulation multivues avec prise en compte des distorsions radiales, pour lequel il n'est pas possible d'utiliser d'hypothèses simplificatrices.

2

Rappels d'optimisation numérique

Sommaire

2.1 Introduction 7
2.1.1 Rappels et Définitions 8
2.1.2 Classification des méthodes d'optimisation 9
2.1.3 Outils de l'optimisation 10
2.2 Méthodes d'optimisation locale 12
2.2.1 Conditions d'optimalité 12
2.2.2 Méthodes sans contraintes 14
2.2.3 Méthodes avec contraintes 28
2.3 Méthodes d'optimisation globale 33
2.3.1 Les méthodes déterministes 33
2.3.2 Les méthodes stochastiques 35
2.4 Conclusion 37

2.1 Introduction

Le but de ce chapitre est de fournir un exposé exhaustif des méthodes d'optimisation numérique. En effet en optimisation numérique, le point crucial consiste à identifier correctement le type de problème considéré afin de savoir quelle famille d'algorithmes il convient d'utiliser. Ainsi l'objectif de ce chapitre est de fournir au lecteur un « arbre de décision » des méthodes d'optimisation et de pouvoir situer chacune des approches utilisées dans la suite du document. Dans une première partie, nous rappellerons succinctement les éléments théoriques nécessaires à la compréhension des approches locales et globales présentées dans les sections suivantes. Enfin, nous ferons un bilan permettant de situer au mieux la branche de l'optimisation dans laquelle se situent nos travaux et les méthodes qui nous serviront de comparaison dans la suite du document.

2.1.1 Rappels et Définitions

Soit K un ensemble non vide de $\mathbb{R}^n$, considérons un problème de la forme suivante :

$$\min_{x \in K} f(x) \tag{2.1}$$

On adopte la nomenclature suivante :
- l'ensemble $K \subset \mathbb{R}^n$ est appelé domaine ou ensemble des contraintes
- la fonction $f : \mathbb{R}^n \to \mathbb{R}$ est appelée fonction coût, objectif ou critère
- tout point $x \in \mathbb{R}^n$ vérifiant : $x \in K$ est appelé point admissible ou réalisable de l'ensemble des contraintes du problème (2.1).

Définition 1 *On dit que $x^\star$ est un minimiseur local de (2.1) s'il existe un voisinage $\mathcal{V}^\star$ de $x^\star$ tel que :*

$$f(x^\star) \leqslant f(x) \quad \forall x \in \mathcal{V}^\star \cap K. \tag{2.2}$$

On dit que $x^\star$ est un minimiseur global de (2.1) si :

$$f(x^\star) \leqslant f(x) \quad \forall x \in K. \tag{2.3}$$

Les notions de maximum local et global sont définies de façon similaire. En effet, on peut facilement démontrer que les problèmes (avec ou sans contraintes) $\min_{x \in K} f(x)$ et $\max_{x \in K} -f(x)$ sont équivalents puisqu'ils ont le même ensemble de solutions et que

$$\min_{x \in K} f(x) = -\max_{x \in K} -f(x). \tag{2.4}$$

Comme la recherche d'un maximum peut donc se ramener à la recherche d'un minimum, nous nous limiterons dans ce chapitre aux problèmes de minimisation. Par ailleurs, si $K = \mathbb{R}^n$ ou si $K \subsetneq \mathbb{R}^n$ est un ouvert de $\mathbb{R}^n$, on dira que le problème est **sans contraintes**. Si K est un fermé de $\mathcal{R}^n$ alors on dira que le problème est **sous contraintes**. Dans le reste de ce chapitre nous considèrerons que K est un ensemble de contraintes **fonctionnelles**, c.-à-d. défini par des contraintes égalités et/ou inégalités :

$$K \quad = \quad \{x \in \mathbb{R}^n : h(x) = 0,\ g(x) \leqslant 0\} \tag{2.5}$$

où $h : \mathbb{R}^n \to \mathbb{R}^p$ et $g : \mathbb{R}^n \to \mathbb{R}^q$. La fonction h représente les contraintes égalités et g les contraintes inégalités. Ensuite, pour une fonction deux fois continuement différentiable de $\mathbb{R}^n$ dans $\mathbb{R}$:

Définition 2 *(**Gradient et hessienne**) On appelle **gradient** de f et **hessienne** de f les vecteurs et matrices définis par :*

$$\nabla f(x) \quad \triangleq \quad \left[\frac{\partial f(x)}{\partial x_i}\right] \in \mathbb{R}^n, \tag{2.6}$$

$$\nabla^2 f(x) \quad \triangleq \quad \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right] \in \mathbb{R}^{n \times n}. \tag{2.7}$$

Pour une matrice $A \in \mathbb{R}^{n \times n}$, on rappelle les définitions de **positivité** (noté $\succ 0$) et de **semi-définie positivité** (noté $\succeq 0$) :

Définition 3 *(**Positivité et semi-définie positivité**) On dit que $A \succeq 0$ si et seulement si :*

$$v^\top A v \geqslant 0, \quad \forall v \in \mathbb{R}^n \tag{2.8}$$

et $A \succ 0$ si et seulement si :

$$v^\top A v > 0, \quad \forall v \in (\mathbb{R}^n) \backslash \{0_{\mathbb{R}^n}\} \tag{2.9}$$

On peut aussi montrer que, dans le cas de matrices diagonalisables, ces définitions sont équivalentes à :

$$A \succeq 0 \quad \Leftrightarrow \quad Sp(A) \subset \mathbb{R}_+. \tag{2.10}$$

$$A \succ 0 \quad \Leftrightarrow \quad Sp(A) \subset \mathbb{R}_+^*. \tag{2.11}$$

où $Sp(A)$ désigne l'ensemble des valeurs propres de A. Enfin on dira qu'une fonction de $\mathbb{R}^n$ dans $\mathbb{R}$ est convexe sur $K \subset \mathbb{R}^n$ si, pour tout couple de points distincts $(x_1, x_2) \in K$, le graphe de la fonction est sous la corde construite à partir des points $(x_1, f(x_1))$ et $(x_2, f(x_2))$:

Définition 4 *(**convexité**) On dit que f est **convexe** sur $K \subset \mathbb{R}^n$ si et seulement si, pour tout couple $(x_1, x_2) \in K^2$ tel que $x_1 \neq x_2$ et pour tout $\lambda \in [0,1]$, l'inégalité suivante est vérifiée :*

$$f(\lambda\, x_1 + (1-\lambda)\, x_2) \leqslant \lambda\, f(x_1) + (1-\lambda)\, f(x_2). \tag{2.12}$$

Si l'inégalité est stricte, on parlera de stricte convexité. De plus, si f est deux fois continuement différentiable sur K, alors la convexité est équivalente à :

$$\nabla^2 f(x) \succeq 0, \quad \forall x \in K, \tag{2.13}$$

et si $\nabla^2 f(x)$ est définie positive sur K, alors f est strictement convexe sur K. Par suite, on adoptera la définition de la quasi-convexité suivante :

Définition 5 *f est **quasi-convexe** sur $K \subset \mathbb{R}^n$ si et seulement si, pour tout réel α, l'ensemble*

$$S_\alpha \quad = \quad \{x \in K \mid f(x) \leqslant \alpha\} \tag{2.14}$$

est convexe ou, de manière équivalente, si, pour tout couple $(x_1, x_2) \in K^2$ tel que $x_1 \neq x_2$ et pour tout $\lambda \in [0,1]$, l'inégalité suivante est vérifiée :

$$f(\lambda\, x_1 + (1-\lambda)\, x_2) \leqslant \max(f(x_1), f(x_2)). \tag{2.15}$$

Enfin, on définit **l'enveloppe convexe**, notée $\mathsf{conv}\, f$, de f par :

Définition 6 *(**Enveloppe convexe**) **L'enveloppe convexe** de f, notée $\mathsf{conv}\, f$ est une fonction convexe sur $K \subset \mathbb{R}^n$ vérifiant :*

(i) $\forall x \in K, \quad \mathsf{conv}\, f(x) \leqslant f(x)$

(ii) Pour toute fonction convexe g telle que $g(x) \leqslant f(x)$, alors $g(x) \leqslant \mathsf{conv}\, f(x)$

2.1.2 Classification des méthodes d'optimisation

La résolution d'un problème d'optimisation est étroitement liée à la granularité avec laquelle on peut le caractériser. Ainsi les problèmes d'optimisation sont souvent classés en plusieurs grandes familles permettant une identification rapide, par exemple :

- L'optimisation discrète où $K \subset \mathbb{Z}^n$,
- L'optimisation numérique où $K \subset \mathbb{R}^n$,
- La commande optimale où K est un espace de fonctions (donc infini-dimensionnel).

Notons qu'il existe classiquement une autre branche qui peut aussi se décliner selon son domaine de contraintes : l'optimisation multi-critères où l'on cherche à minimiser plusieurs fonctions objectifs simultanément. Dans ce document, nous nous concentrons uniquement sur l'optimisation en dimension finie et mono-objectif. Au sein de cette catégorie, il existe plusieurs problèmes classiques auxquels nous ferons souvent référence, leur description est donnée ci-dessous :

Les problèmes linéaires. On dit qu'un problème est linéaire si la fonction objectif et les fonctions qui définissent les contraintes sont linéaires. Tout problème linéaire peut alors être mis sous la forme (dite standard) suivante :

$$\begin{array}{ll} \min & c^\top x \\ \text{s.l.c} & \left\{ \begin{array}{l} Ax = b \\ x \geqslant 0 \end{array} \right. \end{array} \tag{2.16}$$

où $c \in \mathbb{R}^n,\ A \in \mathbb{R}^{p \times n},\ b \in \mathbb{R}^p$.

Les problèmes quadratiques. On dit qu'un problème est quadratique si la fonction objectif est quadratique et les fonctions qui définissent les contraintes sont affines :

$$\begin{array}{ll} \min & \frac{1}{2} x^\top H x + g^\top x + d \\ \text{s.l.c} & \left\{ \begin{array}{l} Ax \leqslant b \\ x \geqslant 0 \end{array} \right. \end{array} \tag{2.17}$$

où $g \in \mathbb{R}^n,\ H \in \mathbb{R}^{n \times n},\ A \in \mathbb{R}^{q \times n},\ b \in \mathbb{R}^m,\ d \in \mathbb{R}$.

Les problèmes de moindres carrés sont les problèmes du type :

$$\min_{x \in \mathbb{R}^n} \ \frac{1}{2} \|r(x)\|_2^2 \tag{2.18}$$

où $r : \mathbb{R}^n \to \mathbb{R}^m$ est une fonction deux fois différentiable. Lorsque r est de la forme $r(x) = Ax - b$ avec $A \in \mathbb{R}^{m \times n}$ et $b \in \mathbb{R}^m$, on parle de problème de moindres carrés linéaires.

Les problèmes convexes sont les problèmes de la forme (2.1) dans lesquels f est une fonction convexe et le domaine des contraintes K est un ensemble convexe. Un **problème quasi-convexe** est défini de manière identique, à l'exception de f qui est, dans ce cas, **quasi-convexe**.

2.1.3 Outils de l'optimisation

2.1.3.1 Convergence et vitesse de convergence

Lorsque l'on parle de l'étude d'une méthode d'optimisation, on fait implicitement référence à l'étude de la suite des itérés générés par celle-ci. Pour le problème (2.1), on introduit la notion de convergence d'un algorithme :

Définition 7 *Soit un algorithme itératif $\mathcal{A}$ générant une suite $(x_k)_{k \in \mathbb{N}}$ dans $\mathbb{R}^n$ afin de résoudre (2.1). $\mathcal{A}$ est dit convergent si la suite $(x_k)_{k \in \mathbb{N}}$ converge vers un point limite $x^\star$. La convergence est dite **locale** si la convergence n'a lieu que pour des points initiaux x_0 situés dans un voisinage de $x^\star$. Si la convergence a lieu quel que soit le point initial x_0, la convergence est dite globale.*

Cependant, l'objectif pratique d'un algorithme d'optimisation **sans contraintes** est de converger vers un point critique de f, c'est-à-dire, dans le cas où f est de classe $\mathcal{C}^1$, de converger vers une solution $x^\star$ telle que $\nabla f(x^\star) = 0$. Ainsi, on introduit la définition de convergence suivante :

Définition 8 *Si f est de classe $\mathcal{C}^1$, on dira que la convergence d'un algorithme itératif $\mathcal{A}$ est globale si et seulement si la suite des itérés $(x_k)_{k \in \mathbb{N}}$ générés par $\mathcal{A}$ est telle que :*

$$\forall x_0 \in \mathbb{R}^n,\ \lim_{k \to +\infty} \|\nabla f(x_k)\| = 0. \tag{2.19}$$

Cette définition garantit que le critère d'arrêt $\|\nabla f(x)\| \leqslant \varepsilon$ sera toujours satisfait à partir d'un certain rang quelle que soit la précision $\varepsilon > 0$ demandée. En pratique, on ne considère que des algorithmes dont la convergence est globale au sens de la définition 7. Cependant, une fois la convergence d'un algorithme assurée, la vitesse à laquelle il converge est également un facteur important à prendre en compte lors de son utilisation. Dans le but de quantifier ce paramètre, on introduit la notion de vitesse de convergence dont l'objectif est de mesurer l'évolution de l'erreur $\|x_k - x^*\|$ commise par l'algorithme à l'itéré k :

Définition 9 *Soit $\mathcal{A}$ un algorithme globalement convergent, soit $(x_k)_{k\in\mathbb{N}}$ la suite des itérés générée par $\mathcal{A}$, on note $x^\star = \lim_{k\to+\infty} x_k$. On suppose de plus que l'algorithme ne converge pas en un nombre fini d'itérations, c.-à-d. $\forall k \in \mathbb{N}$: $x_k \neq x^\star$. On dit que la convergence est :*

- *linéaire si l'erreur $\varepsilon_k \triangleq \|x^\star - x_k\|$ décroît linéairement, c.-à-d. s'il existe $\tau \in]0,1[$ tel que :*
$$\lim_{k\to+\infty} \frac{\varepsilon_{k+1}}{\varepsilon_k} = \tau. \tag{2.20}$$
- *super-linéaire si :*
$$\lim_{k\to+\infty} \frac{\varepsilon_{k+1}}{\varepsilon_k} = 0. \tag{2.21}$$
- *d'ordre p, s'il existe $\tau \geqslant 0$ tel que :*
$$\lim_{k\to+\infty} \frac{\varepsilon_{k+1}}{\varepsilon_k^p} = \tau. \tag{2.22}$$

  En particulier, si $p = 2$ la convergence est dite quadratique.

Théoriquement, on a bien entendu intérêt à ce que la convergence d'un algorithme soit la plus élevée possible afin de converger vers la solution en un minimum d'itérations pour une précision donnée. Cependant, en pratique, une convergence rapide se paye souvent par des hypothèses plus fortes sur la fonction objectif f.

2.1.3.2 Critères d'arrêt

En pratique les solutions $x^\star$ ne sont pas connues, il faut donc choisir un test d'arrêt afin de garantir que l'algorithme s'arrête toujours après un nombre fini d'itérations et que le dernier itéré calculé soit suffisamment proche d'une solution $x^\star$. Par exemple dans le cas de la minimisation sans contraintes d'une fonction différentiable, il est naturel d'utiliser un critère issu de la définition 8. Ainsi, pour ε fixé, on testera à chaque itération si :

$$\|\nabla f(x_k)\| \leqslant \varepsilon, \tag{2.23}$$

auquel cas l'algorithme s'arrête et renvoit l'itéré x_k comme solution. Néanmoins, f peut ne pas être différentiable ou le test peut ne pas être satisfait après un grand nombre d'itérations alors que les itérés sont proches d'un point critique. On doit donc utiliser d'autres critères fondés sur des observations pratiques :

la stagnation de la solution : $\|x_{k+1} - x_k\| \leqslant \varepsilon(1 + \|x_k\|)$

la stagnation de la valeur courante : $|f(x_{k+1}) - f(x_k)| \leqslant \varepsilon(1 + |f(x_k)|)$

le dépassement d'un nombre d'itérations $nbIter$ préalablement fixé : $k \leqslant nbIter$.

En pratique, on utilise plutôt une combinaison de ces critères. Enfin, il est important de retenir qu'il vaut mieux utiliser des critères basés sur les erreurs relatives plutôt que sur les erreurs absolues, trop liées aux échelles de grandeur.

Les sections suivantes sont dévolues à la présentation des éléments théoriques nécessaires à la compréhension des méthodes d'optimisation locale (§2.2) et globale (§2.3).

2.2 Méthodes d'optimisation locale

Dans cette section, nous nous intéressons aux méthodes numériques pour la résolution des problèmes de la forme :

$$\min_{x \in K} f(x) \tag{2.24}$$

où $K \subset \mathbb{R}^n$ définit l'ensemble des contraintes appliquées aux variables d'optimisation x. La fonction coût $f : \mathbb{R}^n \to \mathbb{R}$ est supposée non linéaire.

2.2.1 Conditions d'optimalité

L'objectif de ce paragraphe est d'introduire succinctement les conditions théoriques qui permettent de sélectionner des points candidats à l'optimalité pour des problèmes avec et sans contraintes. Ces conditions servent en pratique à construire des algorithmes, prouver des convergences et définir des tests d'arrêts. Nous présenterons dans un premier temps les conditions d'optimalité pour des problèmes sans contraintes (K est un ouvert de $\mathbb{R}^n$) puis nous introduirons leurs généralisations à ceux avec contraintes (K défini par (2.4)).

Conditions d'optimalité du premier ordre. Nous nous focalisons dans un premier temps sur l'étude des problèmes d'optimisation sans contraintes. Autrement dit, le domaine K des contraintes est un ouvert de $\mathbb{R}^n$. Sans perte de généralité, nous supposerons dans la suite que $K = \mathbb{R}^n$. Ainsi, considérons le problème suivant :

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2.25}$$

Si f est une fonction continuement différentiable de $\mathbb{R}^n$, une condition nécessaire pour qu'un point $x^\star$ de $\mathbb{R}^n$ soit un minimum **local** de f est :

$$\nabla f(x^\star) = 0. \tag{2.26}$$

On dit alors que $x^\star$ est un point **stationnaire** ou **critique** de f. Si on suppose que f est **convexe** alors cette condition devient une condition nécessaire et suffisante d'optimalité globale : $x^\star$ est un minimum global de f dans $\mathbb{R}^n$. Si de plus, f est strictement convexe, alors $x^\star$ est **l'unique** minimum global de f.

Conditions d'optimalité du second ordre. Si f est une fonction deux fois continuement différentiable de $\mathbb{R}^n$, une condition nécessaire pour qu'un point $x^\star$ de $\mathbb{R}^n$ soit un minimum **local** de f est :

$$\nabla f(x^\star) = 0 \quad \text{et} \quad \nabla^2 f(x^\star) \succeq 0. \tag{2.27}$$

Une condition suffisante pour que $x^\star$ soit un minimum **local** de f est :

$$\nabla f(x^\star) = 0 \quad \text{et} \quad \nabla^2 f(x^\star) \succ 0. \tag{2.28}$$

On s'intéresse ensuite à des problèmes d'optimisation où le domaine des contraintes K est défini par :

$$K = \{x \in \mathbb{R}^n \mid h_i(x) = 0 \ \ \forall i \in \{1, \dots, p\}\,, \text{ et } g_j(x) \leqslant 0 \ \ \forall j \in \{1, \dots, q\}\}\,. \tag{2.29}$$

avec $h : \mathbb{R}^n \to \mathbb{R}^p$ et $g : \mathbb{R}^n \to \mathbb{R}^q$ supposées au moins différentiables une fois. Avec de telles conditions, les éléments théoriques que nous venons de présenter ne sont bien évidemment plus valables. Cependant, l'idée directrice selon laquelle les points $x^\star$ tels que $\nabla f(x^\star) = 0$ sont

des candidats légitimes à la minimisation peut être reprise. Ainsi, les conditions d'optimalité présentées ci-après sont basées sur la recherche des points critiques d'une nouvelle fonction $\mathcal{L}$ appelée **le Lagrangien** associé au problème (2.1). Il est défini par :

Définition 10 *(**Lagrangien**) Le Lagrangien $\mathcal{L}$ du problème (2.1) avec des contraintes déterminées par (2.29) est défini par :*

$$\begin{aligned} \mathcal{L} : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q_+ &\rightarrow \mathbb{R} \\ \mathcal{L}(x, \lambda, \mu) &\triangleq f(x) + \sum_{i=1}^{p} \lambda_i h_i(x) + \sum_{j=1}^{q} \mu_j g_j(x). \end{aligned} \tag{2.30}$$

*Les vecteurs $\lambda \in \mathbb{R}^p$ et $\mu \in \mathbb{R}^q_+$ sont appelés **les multiplicateurs de Lagrange**.*

Définissons ensuite la notion de **saturation** ou **d'activation** pour des contraintes de type inégalité par :

Définition 11 *(**Contrainte active**) Une contrainte d'inégalité g_j est dite **active** ou **saturée** en un point x si l'égalité est vérifiée en ce point, i.e :*

$$g_j(x) = 0 \tag{2.31}$$

et inactive en x si $g_j(x) < 0$. L'ensemble des indices des contraintes actives au point x, noté $\mathcal{I}_0(x)$, est ainsi défini par :

$$\mathcal{I}_0(x) \triangleq \{j \mid j \in \{1, \ldots, q\} \;\; et \;\; g_j(x) = 0\}. \tag{2.32}$$

Théorème 1 *(**Conditions d'optimalité de Karush-Kuhn-Tucker (KKT)**) Soit $x^\star$ un minimum local de f sur K. Si la famille de vecteurs $\mathcal{F}(x^\star)$ définie par :*

$$\mathcal{F}(x^\star) \triangleq \{\nabla h_i(x^\star),\ i \in \{1, \ldots, p\}\} \cup \{\nabla g_j(x^\star),\ j \in \mathcal{I}_0(x^\star)\} \tag{2.33}$$

est libre, alors il existe un vecteur unique $\lambda^\star \in \mathbb{R}^p$ et un vecteur unique $\mu^\star \in \mathbb{R}^q_+$ vérifiant les trois propriétés $(KKT)_1, (KKT)_2$ et $(KKT)_3$ définies par :

$$\begin{cases} (KKT)_1 \triangleq \nabla_x \mathcal{L}(x^\star, \lambda^\star, \mu^\star) = \nabla f(x^\star) + \sum_{i=1}^{p} \lambda_i^\star \nabla h_i(x^\star) + \sum_{j=1}^{q} \mu_j^\star \nabla g_j(x^\star) = 0. \\ (KKT)_2 \triangleq \mu_j^\star \geqslant 0, \quad \forall j \in \{1, \ldots, q\}. \\ (KKT)_3 \triangleq \mu_j^\star g_j(x^\star) = 0, \quad \forall j \in \{1, \ldots, q\}. \end{cases} \tag{2.34}$$

De plus, si f, h et g sont deux fois différentiables alors :

$$y^\top \nabla^2_{xx} \mathcal{L}(x^\star, \lambda^\star, \mu^\star) y \geqslant 0, \quad \forall y \in \mathcal{Y}(x^\star) \tag{2.35}$$

*où $\mathcal{Y}(x^\star)$, **le cône tangent à K en $x^\star$**, est l'ensemble des vecteurs non nuls de $\mathbb{R}^n$ tels que :*

$$y^\top \nabla h_i(x^\star) = 0 \quad \forall i \in \{1, \ldots, p\} \tag{2.36}$$

$$y^\top \nabla g_j(x^\star) \leqslant 0 \quad \forall j \in \mathcal{I}_0(x^\star). \tag{2.37}$$

L'objectif de l'introduction de la notion de « contraintes actives » réside dans le fait qu'à l'optimum local $x^\star$, grâce à $(KKT)_2$ et $(KKT)_3$, les contraintes actives peuvent être remplacées par des contraintes égalités, tandis que les contraintes inactives peuvent être ignorées. Cependant, puisque l'on ne connaît pas a priori $x^\star$ et quelles sont les contraintes actives, le passage d'un problème à contraintes inégalités à un problème égalité n'est que purement théorique. En outre, cette notion de contraintes actives permet aussi une interprétation géométrique

des conditions de Karush-Kuhn-Tucker. En effet, on peut remarquer que, si ces conditions sont vérifiées, alors $-\nabla f(x^\star)$ est une combinaison linéaire des vecteurs $(\nabla h_i(x^\star))_{i=1..p}$ et $(\nabla g_j(x^\star))_{j\in\mathcal{I}_0(x^\star)}$. Ces vecteurs forment, dans le cadre du Théorème 1, une base du cône polaire de $\mathcal{Y}(x^\star)$. Cependant, sans hypothèses de **qualification des contraintes**, $\mathcal{Y}(x^\star)$ n'est pas, en général, égal au cône des directions tangentes aux contraintes. Dans l'énoncé du théorème (1), la notion de liberté assure cette qualification. Il est cependant possible d'affaiblir cette hypothèse. Pour plus de détails sur cette théorie, le lecteur pourra se référer aux ouvrages suivants [Bonnans 2006, Hiriart-Urruty 2007, Hiriart-Urruty 1996].

2.2.2 Méthodes sans contraintes

Nous abordons dans cette partie la description des algorithmes utilisés pour résoudre les problèmes d'optimisation sans contraintes. Sur la base des conditions d'optimalité décrites précédemment, nous exposerons dans premier temps (§2.2.2.1) une classe de méthodes appelées **méthodes de descente**. S'incrivent dans cette classe de méthodes, les méthodes dites du **premier ordre** comme les algorithmes **de gradient** ou des **directions conjuguées**, ainsi que celles du **deuxième ordre** comme les algorithmes de **Newton** ou de **quasi-Newton**. Nous présenterons ensuite (§2.2.2.4) les algorithmes de **région de confiance** qui constituent une alternative aux méthodes de descente. Enfin (§2.2.2.5), nous détaillerons un problème d'optimisation particulier, le problème dit des **moindres carrés**, que l'on rencontre très fréquemment en vision par ordinateur.

2.2.2.1 Méthodes de descente

Considérons une fonction f continue, à dérivées premières continues et supposons que l'on sait évaluer, soit directement, soit par différences finies, le gradient ∇f de f en tout point de l'espace de recherche. Les méthodes de descente regroupent une famille d'algorithmes itératifs qui, à partir d'un point initial x_0, construisent une suite d'itérés $(x_k)_{k\in\mathbb{N}}$ définie par :

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.38}$$

où $d_k \in \mathbb{R}^n$ est appelé **la direction de descente** de f en x_k vérifiant nécessairement l'inégalité :

$$\nabla f(x_k)^\top d_k < 0, \tag{2.39}$$

et α_k la longueur du déplacement ou **pas** effectué dans cette direction. Un algorithme de descente est donc complètement caractérisé par la façon de construire les directions de descentes d_k et la longueur des pas α_k que l'on fait dans ces directions.

Les méthodes de Gradient. Les méthodes de gradient sont basées sur l'idée suivante : puisque le gradient $\nabla f(x_k)$ est la direction de la plus grande augmentation de f au point x_k, il est naturel de se déplacer dans la direction opposée. Ainsi, pour l'ensemble de ces méthodes, les directions de descente sont définies par $d_k = -\nabla f(x^k)$, générant une suite d'itérés définis par :

$$x_{k+1} = x_k - \alpha_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \text{ avec } \alpha_k > 0. \tag{2.40}$$

Dès lors, le distinguo entre toutes les approches se fera dans le choix de la longueur du pas α_k. Pour la classe de méthodes, dites **à Gradient à pas prédéterminé**, la suite des $(\alpha_k)_{k\in\mathbb{N}}$ est fixée à priori. Mais afin d'assurer la convergence de l'algorithme, cette suite doit vérifier certaines hypothèses. Ainsi, on montre que si

$$\lim_{k\to\infty} \alpha_k = 0 \quad \text{et} \quad \sum_{k=0}^{\infty} \alpha_k = +\infty \tag{2.41}$$

alors la suite des itérés x_k converge vers $x^\star$. Mais en pratique, la convergence de cette méthode est très lente. C'est pourquoi la recherche du pas à chaque itéré peut être vue comme un problème d'optimisation univarié.

Ainsi, dans une deuxième classe de méthodes dites **de Gradient à pas optimal**, α_k est choisi à chaque itération comme étant égal à la solution du problème d'optimisation :

$$\min_{\alpha>0} f(x_k - \alpha \nabla f(x_k)). \tag{2.42}$$

Cette étape est souvent appelée **recherche linéaire exacte**. Cependant, ce problème d'optimisation unidimensionnel peut être difficile à priori. Mais, dans le cas particulier où f est deux fois continuement différentiable, on peut remplacer dans (2.42) f par :

$$m_k(x) \triangleq f(x_k) + (x - x_k)^\top \nabla f(x_k) + \frac{1}{2}(x - x_k)^\top \nabla^2 f(x_k)(x - x_k), \tag{2.43}$$

son modèle quadratique d'ordre 2 au point x_k. On obtient ainsi une direction qui approxime la solution exacte de (2.42). Cette direction particulière, est appelée **direction ou point de Cauchy**. Dans le cas convexe, ce point est donné par :

$$d_C \triangleq -\frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^\top \nabla^2 f(x_k) \nabla f(x_k)} \nabla f(x_k). \tag{2.44}$$

Mais, cette approximation peut ne pas s'avérer suffisamment fiable pour assurer un résultat précis. Ainsi, dans une majorité de cas, les algorithmes de Gradient à pas optimal ne figurent pas parmi les plus performants.

La dernière catégorie de méthodes, dites **à Gradient accéléré**, se fonde sur l'idée qu'un pas de descente est acceptable s'il fait « raisonnablement » décroître la fonction objectif. On sous-entend dans l'adverbe sibyllin « raisonnablement » que α_k doit être ni trop grand pour assurer la convergence, ni trop petit pour éviter une convergence prématurée. Les deux règles les plus couramment utilisées pour le choix de α_k sont les règles de Wolfe et de Goldstein. Elles spécifient qu'un pas α_k est acceptable s'il satisfait les conditions :

La règle de Goldstein :

$$\begin{array}{rcl} \phi_k(\alpha_k) & \leqslant & \phi_k(0) + \varepsilon \phi_k'(0)\alpha_k \\ \phi_k(\alpha_k) & > & \phi_k(0) + (1-\varepsilon)\phi_k'(0)\alpha_k. \end{array} \tag{2.45}$$

La règle de Wolfe :

$$\begin{array}{rcl} \phi_k(\alpha_k) & \leqslant & \phi_k(0) + \varepsilon \phi_k'(0)\alpha_k \\ \phi_k'(\alpha_k) & \geqslant & (1-\varepsilon)\phi_k'(0). \end{array} \tag{2.46}$$

où $\phi_k(\alpha) \triangleq f(x^k + \alpha d^k)$ et ε fixé dans l'intervalle $]0, \frac{1}{2}[$. La figure 2.1 illustre graphiquement la mise en œuvre de ces deux règles. Plus de détails sur ces techniques sont disponibles dans [Bonnans 2006, Nocedal 2006]. Pour ces deux dernières méthodes, on peut montrer, cf. [Luenberger 2008], que si $f \in \mathcal{C}^2(\mathbb{R}^n)$ et si la suite des itérés $(x_k)_{k\in\mathbb{R}^n}$ converge vers un minimum local $x^\star$, alors la suite $(f(x_k))_{k\in\mathbb{R}^n}$ converge **linéairement** vers $f(x^\star)$. En outre, la constante d'erreur asymptotique est bornée par une constante dépendant uniquement du conditionnement numérique de $\nabla^2 f(x^\star)$.

Tout en restant dans la catégorie des méthodes de descente d'ordre 1, les alternatives aux méthodes de gradient sont les méthodes dites de **directions conjuguées** ou de **gradients conjugués**.

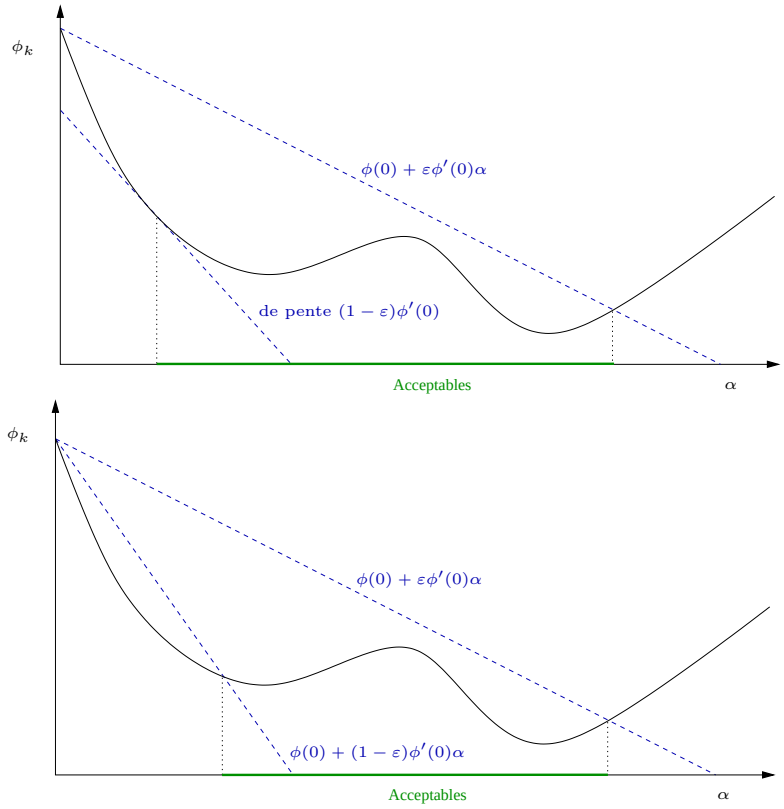


FIGURE 2.1 – Illustration de la règle de Wolfe et de Goldstein. L'espace en vert décrit les pas acceptables pour les deux règles.

Méthode des directions conjuguées. La méthode des directions conjuguées a initialement été développée pour résoudre de grands systèmes linéaires. Elle a, par la suite, été adaptée afin de résoudre des problèmes d'optimisation non-linéaires. Introduite dans [Hestenes 1952], elle a pour but la résolution de deux problèmes équivalents :

$$Ax = b \tag{2.47}$$

et

$$\min_{x\in\mathbb{R}^n} \Phi(x) \triangleq \frac{1}{2}x^\top Ax - b^\top x + c, \tag{2.48}$$

où A est supposée symétrique et définie positive. Elle repose sur la création économique d'une suite de directions $(d_k)_{k\in\mathbb{R}^n}$ orthogonales pour le produit scalaire induit par la matrice A. Plus précisément, on dit que les n directions $d_1, \ldots, d_n$ sont A-conjuguées si :

$$\forall i, j \in [\![1, n]\!], \quad i \neq j, \quad \langle d_i, d_j\rangle_A \triangleq \langle d_i, Ad_j\rangle = d_j^\top Ad_i = 0. \tag{2.49}$$

Des vecteurs non nuls et A-conjugués forment une base de $\mathbb{R}^n$. Il y a donc au plus n directions A-conjuguées dans $\mathbb{R}^n$. Par conséquent, un algorithme initialisé à partir de $x_0 \in \mathbb{R}^n$, disposant à chaque itération k de $k-1$ directions conjuguées $d_1, \ldots, d_{k-1}$ et minimisant $\Phi(x)$ sur la variété linaire $\{x_0\} + vect(d_1, \ldots, d_k)$, convergera nécessairement en, au plus, n itérations. De manière plus succincte, un tel algorithme aura parcouru tout l'espace de recherche en, au plus, n itérations. La méthode des directions conjuguées définit les directions d_k par la formule :

$$d_{k+1} \quad = \quad -(Ax_{k+1} - b) + \frac{d_k^\top A(Ax_{k+1} - b)}{d_k^\top Ad_k} d_k, \tag{2.50}$$

où x_{k+1} est préalablement calculé à partir de (2.38). Remarquons tout d'abord qu'aucune opération (p. ex. inversion ou transposition) n'est effectuée sur A durant l'algorithme. Une

première façon d'améliorer l'algorithme consiste donc à ne pas stocker directement la matrice A, mais à simplement implanter les produits lignes-colonnes nécessaires. Le taux de convergence de la méthode étant directement lié à la répartition des valeurs propres de A, une deuxième façon d'améliorer l'algorithme consiste à transformer la matrice A de sorte que ses valeurs propres soient mieux distribuées. Ce processus s'appelle un **pré-conditionnement**. Il consiste à changer la variable x en une variable $\hat{x}$ au moyen d'une matrice C non-singulière : $\hat{x} = Cx$. Ceci conduit à minimiser non plus directement Φ, mais :

$$\hat{\Phi}(\hat{x}) \quad := \quad \frac{1}{2}\hat{x}^\top (C^{-\top} A C^{-1})\hat{x} + (b^\top C^{-1})\hat{x} + c. \tag{2.51}$$

Le taux de convergence de la méthode dépend alors des valeurs propres de la matrice $C^{-\top} A C^{-1}$ et non plus de celles de A. C peut alors être choisie de manière à accélérer la méthode.

Remarquons ensuite que $\nabla\Phi(x) = (Ax - b)$, on constate alors que la $k^{\text{ème}}$ direction d_k est obtenue par combinaison linéaire du gradient $\nabla\Phi$ au point x_{k-1} et des directions précédentes. Cette remarque permet la généralisation de cette méthode aux fonctions $\mathcal{C}^2$ non quadratiques en remplaçant Φ par le développement de Taylor de f à l'ordre 2. C'est la méthode dite de Fletcher et Reeves [Fletcher 1964], pour laquelle les directions conjuguées d_k sont définies par :

$$d_{k+1} \quad = \quad -\nabla f(x_{k+1}) + \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2} d_k, \tag{2.52}$$

avec x_{k+1} précédemment calculé par (2.38). De manière pratique, la convergence globale des méthodes de directions conjuguées de Fletcher-Reeves est assurée par une étape dite de « redémarrage » : après un cycle de n itérations sans que le test d'arrêt soit satisfait, on redémarre la méthode avec une méthode de gradient, par exemple $d_k = -\nabla f(x_k)$. Ceci est illustré dans l'algorithme 1. La convergence de cet algorithme est alors super-linéaire. Notons

Algorithme 1 Directions Conjuguées

Précondition : $x_0 \in \mathbb{R}^n$

1: $d_0 \leftarrow -\nabla f(x_0)$
2: **tant que** Test d'arrêt en x_k non vérifié **faire**
3: Calcul de α_k
4: $x_{k+1} \leftarrow x_k + \alpha_k d_k$
5: **si** $k+1$ est un multiple de n **alors**
6: $d_k \leftarrow -\nabla f(x_k)$ (redémarrage)
7: **sinon**
8: $d_k \leftarrow -\nabla f(x_{k-1}) + \beta_k d_{k-1}$, où $\beta_k := \dfrac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$
9: $k \leftarrow k+1$
10: **retour** x_k

qu'il existe une variante, dite de Polack-Ribière [Polak 1969], qui consiste à remplacer β_k par :

$$\beta_k \quad = \quad \frac{\nabla f(x_{k+1})^\top (\nabla f(x_{k+1}) - \nabla f(x_k)))}{\|\nabla f(x_k)\|^2}. \tag{2.53}$$

Cependant la convergence de la variante de Polak-Ribière n'est pas assurée. Par contre, on peut montrer que cette dernière, quand elle converge, le fait plus rapidement. Enfin, notons que la notion de conjugaison n'a pas de sens dans le cas non-quadratique (sauf près de l'optimum, mais il est inconnu a priori). Il faut donc tester, au cours des itérations, si l'hypothèse d'approximation quadratique est vérifiée. Pour ce faire, on peut par exemple surveiller les

indicateurs suivants :

$$|\nabla f(x_{k+1})^\top \nabla f(x_k)| \leqslant \varepsilon, \quad 0 < \varepsilon \ll 0 \tag{2.54}$$

$$\frac{\nabla f(x_{k+1})^\top d_{k+1}}{\|\nabla f(x_{k+1})\| \, \|d_{k+1}\|} \leqslant -\eta, \qquad 0 \ll \eta. \tag{2.55}$$

Les méthodes de descente du premier ordre n'utilisent que les données issues du gradient de f. Afin d'accélérer la convergence et améliorer la précision de la solution, les méthodes du second ordre exploitent les informations fournies par les dérivées secondes ou leurs approximations. Supposons donc que f est deux fois continuement différentiable sur $\mathbb{R}^n$ et que, outre le gradient, on sait évaluer soit directement, soit par différences finies, la hessienne $\nabla^2 f(x)$ de f en tout point de l'espace de recherche.

2.2.2.2 Méthode de Newton

Le principe général de cette méthode est de minimiser, au voisinage de x_k, l'approximation quadratique de f :

$$m_k(x) \quad = \quad f(x_k) + (x - x_k)^\top \nabla f(x_k) + \frac{1}{2}(x - x_k)^\top \nabla^2 f(x_k)(x - x_k). \tag{2.56}$$

Le minimum d_N de m_k, appelé **direction de Newton** au point x_k, vérifie $\nabla q(d_N) = 0$, soit :

$$d_N \quad \triangleq \quad -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k). \tag{2.57}$$

Une itération de l'algorithme de Newton s'écrit alors :

$$x_{k+1} \quad = \quad x_k + d_N. \tag{2.58}$$

En pratique, la méthode n'est pas mise en œuvre avec une inversion directe de $\nabla^2 f(x_k)$ qui peut être mal conditionnée, mais utilise plutôt la résolution du système :

$$\nabla^2 f(x_k) d_N \quad = \quad -\nabla f(x_k). \tag{2.59}$$

La méthode de Newton revêt un intérêt particulier car sa convergence est quadratique au voisinage de la solution, c'est-à-dire qu'il existe $\tau \geqslant 0$ tel que :

$$\frac{\|x^\star - x_{k+1}\|}{\|x^\star - x_k\|^2} \quad \leqslant \quad \tau. \tag{2.60}$$

Cependant, la convergence n'est assurée que si le point initial x_0 est suffisamment proche de $x^\star$. En outre, à chaque itération k, d_N existe et vérifie (2.39) si et seulement si $\nabla^2 f(x_k) > 0$. Ces deux restrictions font que l'algorithme de Newton, décrit par l'algorithme 2, a un intérêt limité.

Algorithme 2 Newton

Précondition : $x_0 \in \mathbb{R}^n$

1: **tant que** Test d'arrêt en x_k non vérifié **faire**
2: Calcul de la direction de Newton d_N solution de $\nabla^2 f(x_k) d_N = -\nabla f(x_k)$
3: $x_{k+1} \leftarrow x_k + d_N$
4: $k \leftarrow k + 1$
5: **retour** x_k

Néanmoins, en minimisant itérativement un **modèle quadratique perturbé**, il est possible de remédier à ces deux inconvénients. Minimiser un tel modèle revient à modifier itérativement

la matrice hessienne $\nabla^2 f(x_k)$, afin d'assurer d'une part l'existence de d_k et d'autre part que celle-ci soit une direction de descente. De manière pratique, à chaque itération, on ajoute à $\nabla^2 f(x_k)$ une matrice diagonale de manière à forcer sa positivité :

$$\left(\nabla^2 f(x_k) + \mu_k I_n\right) d_k \quad = \quad -\nabla f(x_k) \tag{2.61}$$

où I_n est la matrice identité de $\mathbb{R}^{n\times n}$ et μ_k choisi de telle sorte que $\min_{\lambda \in Sp(\nabla^2 f(x_k))} \lambda + \mu_k$ soit strictement positif. Compte tenu qu'il faut résoudre (2.61) à chaque itération, il est en outre souhaitable de choisir μ_k de manière à améliorer le conditionnement de $\nabla^2 f(x_k) + \mu_k I_n$. Enfin, tout algorithme mettant en œuvre cette méthode utilise en pratique une factorisation de $\nabla^2 f(x_k)$. Cette factorisation sert à accélérer les étapes suivantes :

1. établir si $\nabla^2 f(x_k) \succ 0$
2. calculer une matrice hessienne **augmentée**, $\nabla^2 f(x_k) + \mu_k I_n \succ 0$
3. résoudre le système $\left(\nabla^2 f(x_k) + \mu_k I_n\right) d^k = -\nabla f(x_k)$.

Enfin, bien que l'existence des directions de descente soit acquise, il est nécessaire de rendre l'algorithme globalement convergent (i.e. convergent quel que soit le point de départ). Ceci est réalisé via l'introduction d'un pas α_k calculé soit de manière exacte (2.42), soit par recherche linéaire (2.45, 2.46). L'algorithme 3 peut alors être vu comme un compromis entre les méthodes de gradient (convergence globale mais linéaire) qui correspondent à μ_k très grand et la méthode de Newton (convergence locale autour de $x^\star$ mais quadratique) correspondant à μ_k nul.

Algorithme 3 Newton perturbé

Précondition : $x_0 \in \mathbb{R}^n$

tant que Test d'arrêt en x_k non vérifié **faire**
  Calcul de α_k
  Factorisation de $\nabla^2 f(x_k)$
  si $\nabla^2 f(x_k) \succ 0$ **alors**
    résoudre $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$
  sinon
    Calcul de μ_k
    résoudre $\left(\nabla^2 f(x_k) + \mu_k I_n\right) d_k = -\nabla f(x_k)$
  $x_{k+1} \leftarrow x_k + \alpha_k d_k$
  $k \leftarrow k + 1$
retour x_k

Les inconvénients de ces méthodes résident dans le fait qu'elles obligent à utiliser la matrice hessienne de f à chaque itération. Or, le calcul de cette matrice pose des problèmes pratiques. En effet, que les dérivées secondes soient calculées de manière exacte ou par différences finies, leur implémentation est souvent fastidieuse et entachée d'erreurs numériques. Ensuite, la résolution à chaque itération des systèmes (2.59) ou (2.61) peut consommer beaucoup de temps de calcul et donc s'avérer très pénalisante. Dès lors, les méthodes suivantes s'attachent à maintenir la structure des algorithmes de Newton sans toutefois utiliser la matrice hessienne.

2.2.2.3 Méthodes de quasi-Newton (ou méthodes à métrique variable)

Ces méthodes sont des généralisations de la formule itérative de Newton. L'idée est de construire à chaque itération des matrices H_k approximant la hessienne $\nabla^2 f(x_k)$, ou des matrices B_k approchant son inverse $(\nabla^2 f(x_k))^{-1}$, en n'utilisant que $f(x_k)$, $\nabla f(x_k)$ et des informations de courbures relatives aux itérés précédents. Une méthode de quasi-Newton est

une méthode du type :

$$\left\{ \begin{array}{lcl} d_k & = & -\mathsf{H}_k^{-1}\nabla f(x_k) \\ x_{k+1} & = & x_k + \alpha_k d_k \end{array} \right. \tag{2.62}$$

ou du type :

$$\left\{ \begin{array}{lcl} d_k & = & -\mathsf{B}_k\nabla f(x_k) \\ x_{k+1} & = & x_k + \alpha_k d_k \end{array} \right. \tag{2.63}$$

où H_k (respectivement B_k) est une matrice destinée à approcher la hessienne de f (respectivement l'inverse de la hessienne de f) en x_k. Il se pose alors deux problèmes : quelle stratégie adopter pour faire ces approximations et comment ensuite les mettre à jour au cours des itérations ?

Tout d'abord, écrivons le développement limité de ∇f au voisinage de x_{k+1} appliqué en x_k :

$$\nabla f(x_{k+1}) \quad = \quad \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) + o(\|x_{k+1} - x_k\|). \tag{2.64}$$

Si on fait l'hypothèse que l'approximation quadratique est bonne, on peut négliger le reste et considérer que l'on a :

$$\nabla f(x_{k+1}) - \nabla f(x_k) \quad = \quad \nabla^2 f(x_k)(x_{k+1} - x_k). \tag{2.65}$$

Ceci conduit à la relation de définition de la relation de **quasi-Newton** :

Définition 12 ***(Relations de Quasi-Newton)*** *On dit que les matrices H_{k+1} et B_{k+1} vérifient une relation de quasi-Newton si on a :*

$$\nabla f(x_{k+1}) - \nabla f(x_k) \quad = \quad \mathsf{H}_{k+1}(x_{k+1} - x_k), \tag{2.66}$$

ou

$$\mathsf{B}_{k+1}\left(\nabla f(x_{k+1}) - \nabla f(x_k)\right) \quad = \quad x_{k+1} - x_k. \tag{2.67}$$

Les approximations de $\nabla^2 f(x_k)$ et de son inverse $(\nabla^2 f(x_k))^{-1}$ sont alors construites suivant ces deux dernières relations. Cependant, connaissant H_k et B_k, comment mettre à jour les matrices H_{k+1} et B_{k+1} tout en assurant leur positivité ? Le principe de la mise à jour consiste, pour une itération donnée de l'algorithme :

$$\left\{ \begin{array}{lcl} d_k & = & -\mathsf{H}_k^{-1}\nabla f(x_k) \\ x_{k+1} & = & x_k + \alpha_k d_k, \end{array} \right.$$

à ajouter à H_k l'information nouvellement acquise sur la courbure de f :

$$\mathsf{H}_{k+1} \quad = \quad \mathsf{H}_k + \Delta_k \tag{2.68}$$

avec Δ_k symétrique vérifiant les relations :

$$\nabla f(x_{k+1}) - \nabla f(x_k) \quad = \quad \mathsf{H}_{k+1}(x_{k+1} - x_k) \tag{2.69}$$

$$\mathsf{H}_k \succ 0 \quad \Rightarrow \quad \mathsf{H}_{k+1} \succ 0. \tag{2.70}$$

Suivant le rang de Δ_k on parlera de **correction** de rang 1 ou de rang 2. Une méthode de quasi-Newton est donc complètement définie par le type d'équation de quasi-Newton qu'elle vérifie et la nature des matrices de mise à jour Δ_k qu'elle utilise. L'algorithme 4 résume, sous forme de pseudo-code, les éléments que nous venons de détailler. Plusieurs formules sont possibles pour réaliser la mise à jour, chacune donnant lieu à une méthode particulière. Les deux méthodes les plus utilisées sont détaillées ci-après.

Algorithme 4 Quasi-Newton

Précondition : $x_0 \in \mathbb{R}^n$, $\mathsf{H}_0 = I$ ou $\mathsf{B}_0 = I$.

1: **tant que** Test d'arrêt en x_k non vérifié **faire**
2: $d_k \leftarrow \mathsf{H}_k^{-1}\nabla f(x_k)$ ou $d_k \leftarrow \mathsf{B}_k\nabla f(x_k)$
3: Calcul de α_k
4: $x_{k+1} \leftarrow x_k + \alpha_k d_k$
5: Mise à jour de H_k ou B_k selon la méthode choisie
6: $k \leftarrow k+1$
7: **retour** x_k

Algorithme de Davidon, Fletcher, Powell (DFP) La formule de mise à jour de Davidon, Fletcher et Powell est une formule de correction de rang 2 donnée par :

$$\mathsf{H}_{k+1} = \mathsf{H}_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{\mathsf{H}_k s_k s_k^\top \mathsf{H}_k}{s_k^\top \mathsf{H}_k s_k} \tag{2.71}$$

avec

$$s_k \triangleq x_{k+1} - x_k \tag{2.72}$$
$$y_k \triangleq \nabla f(x_{k+1}) - \nabla f(x_k). \tag{2.73}$$

Si $\mathsf{H}_k \succ 0$ et si $s_k^\top y_k > 0$ alors $\mathsf{H}_{k+1} \succ 0$, ce qui garantit que les directions d_k successives sont des directions de descente. La condition $s_k^\top y_k > 0$ est vérifiée si x_{k+1} est obtenu par une minimisation unidimensionnelle dans la direction $d_k = -\mathsf{H}_k\nabla f(x_k)$ ou au moyen de la recherche linéaire (2.45, 2.46). En outre, il est intéressant de remarquer que dans le cas d'une fonction quadratique, les directions sont conjuguées, assurant ainsi une convergence en au plus n itérations.

Algorithme de Broyden, Fletcher, Goldfarb, Shanno (BFGS) La formule de mise à jour de Broyden, Fletcher, Goldfarb et Shanno est une formule de correction de rang 2 qui s'obtient à partir de la formule DFP en intervertissant les rôles de s_k et y_k. La formule obtenue permet de mettre à jour une approximation B_k de l'inverse du hessien possédant les mêmes propriétés, à savoir $\mathsf{B}_{k+1} \succ 0$ si $\mathsf{B}_k \succ 0$ et vérifiant la relation de quasi-Newton (2.67). La formule de mise à jour est donnée par :

$$\mathsf{B}_{k+1} = \left(I - \frac{s_k y_k^\top}{y_k^\top s_k}\right)\mathsf{B}_k\left(I - \frac{s_k y_k^\top}{y_k^\top s_k}\right) + \frac{s_k s_k^\top}{y_k^\top s_k}. \tag{2.74}$$

La méthode BFGS possède les mêmes propriétés que la méthode DFP : dans le cas quadratique les directions engendrées sont conjuguées. De plus, la vitesse de convergence ce cette méthode est moins sensible que la méthode DFP aux imprécisions dans la recherche du pas de descente.

2.2.2.4 Méthode des régions de confiance

Nous venons de détailler une classe de méthodes d'optimisation qui permet notamment de pallier les défauts de la méthode de Newton locale tout en essayant de conserver ses propriétés de convergence. En utilisant une approche différente, les méthodes dites des **régions de confiance** visent le même objectif. Comme nous l'avons vu, une itération de la méthode de Newton consiste à minimiser un modèle quadratique m_k de f. Suivant les propriétés du développement de Taylor, m_k est une bonne approximation de f lorsque l'on est proche de l'itéré x_k. Le principe des régions de confiance réside ainsi dans le fait de définir une région autour de l'itéré x_k dans laquelle on peut faire confiance à m_k. En pratique, cette région est

une boule de centre x_k et de rayon r_k. Les deux points clés des algorithmes de régions de confiance sont donc le calcul et la mise à jour du rayon de la région de confiance r_k et la résolution du sous-problème de région de confiance.

Définition 13 *(**Sous-problème des régions de confiance**) Soit $x_k \in \mathbb{R}^n$ et $r_k > 0$, le modèle quadratique m_k est défini par :*

$$m_k(x_k + d) \quad = \quad f(x_k) + d^\top \nabla f(x_k) + \frac{1}{2} d^\top \nabla^2 f(x_k) d. \tag{2.75}$$

Le sous-problème de région de confiance est défini par :

$$\min_{\|d\| \leqslant r_k} m_k(x_k + d). \tag{2.76}$$

Intéressons-nous au conditions d'optimalité de ce problème. Pour cela, sans perte de généralité, on redéfinit les contraintes par :

$$\frac{1}{2}(\|d\|^2 - r_k^2) \quad \leqslant \quad 0. \tag{2.77}$$

Le lagrangien de ce problème s'écrit alors :

$$\mathcal{L}(x, \mu) \quad = \quad f(x_k) + d^\top \nabla f(x_k) + \frac{1}{2} d^\top \nabla^2 f(x_k) d + \frac{\mu}{2}(\|d\|^2 - r_k^2) \tag{2.78}$$

Si $d_k^\star$ est solution du sous-problème de région de confiance, alors les conditions de (KKT) assurent qu'il existe $\mu_k^\star \in \mathbb{R}$ tel que :

$$\nabla f(x_k) + \nabla^2 f(x_k) d_k^\star + \mu_k^\star d_k^\star \quad = \quad 0 \tag{2.79}$$

$$\mu_k^\star \quad \geqslant \quad 0 \tag{2.80}$$

$$\mu_k^\star (\|d_k^\star\|^2 - r_k^2) \quad = \quad 0. \tag{2.81}$$

Si $d_k^\star$ se situe à l'intérieur de la région de confiance alors $\mu_k^\star = 0$. Par conséquent (2.79) est équivalent au système (2.59) ramenant la méthode à celle de Newton locale. Si la solution se trouve sur la frontière de la région de confiance alors (2.79) devient :

$$(\nabla^2 f(x_k) + \mu_k^\star I) d_k^\star \quad = \quad -\nabla f(x_k). \tag{2.82}$$

Dans ce cas, on peut montrer (cf. [Conn 2000]) que $\nabla^2 f(x_k)$ est semi-définie positive. Ceci souligne le lien entre le problème de région de confiance et la minimisation du problème quadratique perturbé (2.61). En effet, dans les deux cas, un multiple de l'identité est ajouté à la matrice hessienne afin de corriger les défauts de la méthode de Newton locale. Cependant, soulignons que (2.81) fournit un lien explicite entre le paramètre $\mu_k^\star$ et le rayon de la région de confiance associée $d_k^\star$. Néanmoins, à l'instar de (2.59), il est trop coûteux de résoudre exactement (2.79). Il est plus avantageux de déterminer une solution approximative. Il existe plusieurs méthodes d'approximation, les deux plus utilisées sont la méthode **Dogleg** et celle de **Seihaug-Toint**.

La première est basée sur les informations relatives à la taille de la région de confiance et n'est valable que si $\nabla^2 f(x_k)$ est définie positive. En effet si cette dernière est petite, alors il est légitime de penser que le terme quadratique $\nabla^2 f(x_k)$ joue un rôle mineur dans m_k. Dès lors, on choisit $d_k^\star$ comme étant égal à la direction de **Cauchy** (2.44). A contrario, si r_k est très grand, alors $\nabla^2 f(x_k)$ joue un rôle prépondérant dans m_k. On choisira alors $d_k^\star$ comme étant égal à la direction de Newton (2.57). Si r_k n'est pas suffisamment significatif pour induire un comportement particulier de m_k, alors $d_k^\star$ est choisi comme une combinaison entre le point de Cauchy et celui de Newton.

La seconde, qui ne nécessite pas d'hypothèses sur $\nabla^2 f(x_k)$, est basée sur une adaptation de la méthode des directions conjuguées pour résoudre le problème (2.79). Plus de détails sur ces deux méthodes sont disponibles dans l'ouvrage de référence [Conn 2000].

Le rayon de la région de confiance est déterminé par essais-erreurs. A la première itération, r_0 est fixé arbitrairement. Ensuite, à chaque itération, on évalue la qualité de la solution $d_k^\star$ du sous-problème de région de confiance et on adapte r_k en fonction de cette évaluation. L'évaluation de la qualité de la solution est réalisée à l'aide de la quantité :

$$\rho \triangleq \frac{f(x_k + d_k^\star) - f(x_k)}{m_k(x_k + d_k^\star) - m_k(x_k)} \tag{2.83}$$

qui correspond au rapport de l'accroissement (positif ou négatif) de f sur celui du modèle m_k. Si m_k est fiable, cette quantité devrait être proche, voire plus grande, que 1. Inversement si elle est proche de 0, m_k n'est pas fiable. Pratiquement, afin de caractériser ces cas et de mettre à jour r_k, introduisons les constantes η_1 et η_2 telles que $0 < \eta_1 \leqslant \eta_2 < 1$. Distinguons trois cas :

- si $\rho \geqslant \eta_2$ alors m_k est très fiable et r_k est doublé
- si $\eta_1 \leqslant \rho < \eta_2$ alors l'adéquation entre m_k est f n'est pas parfaite, mais si elle a permis de réduire la valeur de f, alors on peut la supposer fiable et r_k n'est pas modifié
- si $\rho < \eta_1$ alors m_k n'est pas fiable et r_k est réduit à $\frac{1}{2}\|d_k^\star\|$

Pour donner un ordre d'idée, on fixe en pratique $\eta_1 = 0.01$ et $\eta_2 = 0.9$. L'ensemble de tous ces éléments est résumé dans l'algorithme 5.

Algorithme 5 Régions de confiance

Précondition : $x_0 \in \mathbb{R}^n$, η_1, η_2, r_0

tant que Test d'arrêt en x_k non vérifié **faire**
  Calcul de $d_k^\star$ par résolution approximative du sous-problème de région de confiance (13) à l'aide de la méthode **Dogleg** ou **Steihaug-Toint**
  Calcul de $\rho = \dfrac{f(x_k + d_k^\star) - f(x_k)}{m_k(x_k + d_k^\star) - m_k(x_k)}$
  si $\rho < \eta_1$ **alors**
    $x_{k+1} \leftarrow x_k$
    $r_k \leftarrow \frac{1}{2}\|d_k^\star\|$
  sinon
    $x_{k+1} \leftarrow x_k + d_k$
    si $\rho \geqslant \eta_2$ **alors**
      $r_{k+1} \leftarrow 2r_k$
    sinon
      $r_{k+1} \leftarrow r_k$
  $k \leftarrow k + 1$
retour x_k

2.2.2.5 Problème des moindres carrés

Les problèmes de moindres carrés sont des problèmes d'optimisation de la forme (2.18). Ces problèmes sont très fréquents en vision par ordinateur, car ils apparaissent dès que l'on désire ajuster un modèle mathématique sur des données expérimentales. La première idée consiste à appliquer directement la méthode de Newton locale. Pour cela, calculons le gradient et la

hessienne du problème. Si f est égal à la fonction coût $\frac{1}{2}r^\top(x)r(x)$, alors :

$$\nabla f(x) = J_r(x)r(x) \tag{2.84}$$
$$\nabla^2 f(x) = J_r(x)^\top J_r(x) + H_r(x)r(x) \tag{2.85}$$

où $J_r(x)$ et $H_r(x)$ sont respectivement la jacobienne et la hessienne de r. Cette hessienne est en général très coûteuse à calculer. De plus, au voisinage de l'optimum (domaine de validité théorique de la méthode de Newton) r sera petit, rendant la contribution du terme $H_r(x)r(x)$ négligeable. Il est donc judicieux de vouloir l'ignorer et ainsi de considérer que :

$$\nabla^2 f(x) \simeq J_r(x)^\top J_r(x). \tag{2.86}$$

Cette approximation est connue sous le nom **d'hypothèse de Gauss-Newton**. Elle permet d'une part, d'approcher $\nabla^2 f(x)$ par des dérivées premières et d'assurer, d'autre part, que cette dernière est toujours semi-définie positive. Cette hypothèse conduit à l'algorithme 6 dit de **Gauss-Newton**.

Algorithme 6 Gauss-Newton

Précondition : $x_0 \in \mathbb{R}^n$

tant que Test d'arrêt en x_k non vérifié **faire**
  Calcul de d_N solution de $J_r(x_k)^\top J_r(x_k)d_N = -J_r(x_k)r(x_k)$
  $x_{k+1} \leftarrow x_k + d_N$
  $k \leftarrow k + 1$
retour x_k

Cependant la convergence de cette méthode n'est pas garantie. En effet, si x_0 est trop loin d'un optimum local, alors $J_r(x_k)^\top J_r(x_k)$ peut être singulière ou mal conditionnée. Mais ces inconvénients sont liés à la méthode de Newton et non à l'hypothèse de Gauss-Newton elle-même. Afin de pallier ces inconvénients, on peut donc envisager de minimiser un modèle quadratique perturbé. Comme nous l'avons déjà vu, cela revient à augmenter itérativement la diagonale de la hessienne afin de la rendre positive et mieux conditionnée. Cette approche, connue sous le nom **d'algorithme de Levenberg-Marquardt**, est détaillée dans l'algorithme 7.

Algorithme 7 Levenberg-Marquardt

Précondition : $x_0 \in \mathbb{R}^n$, μ_0

tant que Test d'arrêt en x_k non vérifié **faire**
  Calcul de α_k
  Factorisation de $J_r(x_k)^\top J_r(x_k)$
  si $J_r(x_k)^\top J_r(x_k) \succ 0$ **alors**
    résoudre $J_r(x_k)^\top J_r(x_k)d_k = -J_r(x_k)r(x_k)$
  sinon
    Calcul de μ_k de sorte que $\left(J_r(x_k)^\top J_r(x_k) + \mu_k I_n\right) \succ 0$
    résoudre $\left(J_r(x_k)^\top J_r(x_k) + \mu_k I_n\right) d_k = -J_r(x_k)r(x_k)$
  $x_{k+1} \leftarrow x_k + \alpha_k d_k$
  $k \leftarrow k + 1$
retour x_k

Notons qu'en appliquant un algorithme de région de confiance à (2.18) et en écrivant les conditions de (KKT) sous l'hypothèse de Gauss-Newton (2.86), il est possible d'écrire un lien direct entre μ_k et le rayon de la région de confiance d_k.

Du fait que les problèmes rencontrés en vison par ordinateur ont la propriété de rendre creuse la matrice $J_r(x_k)$, cet algorithme est actuellement, sous plusieurs déclinaisons que nous détaillerons par la suite, le plus employé dans ce domaine.

2.2.2.6 Méthodes sans dérivées

Nous avons vu précédemment comment minimiser une fonction f à l'aide, soit des dérivées premières, soit des dérivées premières et secondes (ou de leurs approximations). L'objet des deux paragraphes suivants est d'introduire des notions sur les techniques de minimisation qui n'utilisent aucune des informations relatives aux dérivées. Le premier paragraphe détaille des méthodes dédiées à la minimisation de fonctions univariées qui sont souvent utilisées pour la recherche exacte de pas de descente. Le deuxième paragraphe se veut plus introductif sur des méthodes plus générales qui sont utilisées notamment quand les dérivées ne sont pas fournies ou trop coûteuses à calculer.

Méthodes sans dérivées pour fonctions univariées et unimodales. Nous détaillons dans ce paragraphe certaines méthodes directes dédiées à des fonctions coût univariées et unimodales. Nous cherchons alors à résoudre :

$$\min_{x\in\mathbb{R}} f(x). \tag{2.87}$$

On dira qu'une fonction univariée f est unimodale sur un intervalle $[a, b]$, s'il n'existe qu'un seul minimum de f dans cet intervalle. Ces méthodes consistent alors à réduire l'intervalle de recherche de manière itérative jusqu'à obtenir un encadrement satisfaisant du minimum. Elles présentent l'avantage d'être très simples à implémenter et peuvent être utilisées, par exemple, dans des étapes de recherche de pas optimal.

La méthode la plus classique est la méthode dite **à section symétrique**. Considérons un intervalle de recherche $[a, b]$. Comme f est unimodale, si la valeur de f est connue en a, b et en deux autres points $a_0 < b_0$ (correspondant aux premiers itérés de la méthode), alors il existe toujours un sous-intervalle parmi $[a, a_0]$ et $[b_0, b]$ qui ne contient pas le minimum $x^\star$. On obtient ainsi un intervalle réduit sur lequel on répète l'opération. L'algorithme 8 résume cette méthode.

Algorithme 8 Section Symétrique

Précondition : a, b

$a_0 \leftarrow a$ et $b_0 \leftarrow b$
Choix d'un paramètre $w_0 \in [a_0, b_0]$ qui caractérise la méthode.
tant que Test d'arrêt en x_k non vérifié **faire**
  $c_k \leftarrow w_k$
  si $w_k < a_k + b_k - w_k$ **alors**
    $d_k \leftarrow a_k + b_k - w_k$
  sinon
    $c_k \leftarrow a_k + b_k - w_k$
    $d_k \leftarrow w_k$
  si $f(c_k) \leqslant f(d_k)$ (i.e $x^\star \in [a_k, d_k]$) **alors**
    $a_{k+1} \leftarrow a_k$
    $b_{k+1} \leftarrow d_k$
    $w_{k+1} \leftarrow c_k$
  sinon
    $x^\star \in [c_k, b_k]$
    $a_{k+1} \leftarrow c_k$
    $b_{k+1} \leftarrow b_k$
    $w_{k+1} \leftarrow d_k$
  $k \leftarrow k + 1$
retour x_k

On obtient ainsi une famille de méthodes complètement définie par le choix de w_0. Deux déclinaisons particulières sont parmi les plus utilisées. La première, dite **à section dorée**, définit w_0 comme étant égal à $\tau a_0 + (1-\tau)b_0$ avec $\tau = \frac{1+\sqrt{5}}{2}$ (le nombre d'or). Avec w_0 ainsi défini, on peut montrer que :

$$\frac{b_{k+1} - a_{k+1}}{b_k - a_k} = \frac{1}{\tau}, \tag{2.88}$$

de sorte que

$$b_{k+1} - a_{k+1} = \frac{1}{\tau^k}(b-a). \tag{2.89}$$

En prenant $x_k = \frac{b_{k+1}+a_{k+1}}{2}$ comme approximation de $x^\star \in [a_k, b_k]$, on obtient :

$$|x_k - x^\star| = \frac{1}{\tau^k}\frac{b-a}{2}. \tag{2.90}$$

La méthode s'arrête lorsque $\frac{b_k+a_k}{2}$ correspond à la précision souhaitée. La seconde méthode, dite de **Fibonacci**, choisit w_0 égal à $t_0 a_0 + (1-t_0)b_0$ avec $t_0 = \frac{F_{n+2}}{F_{n+3}}$, $(F_k)_{k\in\mathbb{N}}$ étant la suite de Fibonacci (définie par $F_0 = 0, F_1 = 1$ et $F_{k+2} = F_{k+1} + F_k$) et n le nombre d'étapes (déterminé a priori) que l'on souhaite effectuer. On peut alors montrer que :

$$\frac{b_{k+1} - a_{k+1}}{b_k - a_k} = \frac{F_{n+2-k}}{F_{n+3-k}}. \tag{2.91}$$

En prenant $x_k = \frac{b_{k+1}+a_{k+1}}{2}$ comme approximation de $x^\star \in [a_k, b_k]$, on obtient :

$$|x_k - x^\star| = \frac{b-a}{F_{n+3}}. \tag{2.92}$$

La méthode de Fibonacci est optimale dans le sens où, pour un nombre prédéterminé d'itérations n, elle obtient un intervalle $[a_k, b_k]$ contenant $x^\star$ le plus petit possible.

Méthodes sans dérivées Ces méthodes, dites **directes**, ne nécessitent pas le calcul de dérivées et sont parfois plus faciles à mettre en œuvre que les méthodes vues précédemment, dites **indirectes**. Cependant, leurs performances peuvent souvent s'avérer limitées dès que la complexité du problème à traiter devient importante. La méthode la plus répandue est la méthode dite de **Nelder-Mead** [Nelder 1965]. Le principe de la méthode, appelée aussi méthode du **simplexe** (à ne pas confondre avec l'algorithme du Simplexe pour la résolution de problèmes de programmation linéaire), consiste à définir une figure géométrique dans l'espace de dimension n, appelée simplex, possédant $n+1$ sommets. La fonction objectif est ensuite évaluée en chacun des sommets. Une fois trouvé le sommet avec les moins bonnes performances (sommet présentant la valeur maximale de la fonction objectif dans le cas d'un problème de minimisation), un nouveau simplex est construit : le plus mauvais sommet est éliminé et remplacé par un point symétrique par rapport au centre de gravité des sommets non éliminés. La recherche du minimum se poursuit donc en s'éloignant du plus mauvais sommet. Cette procédure de construction d'un nouveau simplex est répétée jusqu'à ce qu'une solution soit trouvée à la précision recherchée. Plusieurs résultats sur la convergence de cette méthode sont disponibles dans [McKinnon 1999]. L'algorithme 9 détaille cette procédure. Il existe une autre catégorie de méthodes sans dérivée. Celle-ci est basée sur l'utilisation d'évaluation de f en plusieurs points pour construire un modèle approximant f et pouvant être minimisé (cf. [Conn 2009]). Bien sûr, plus le nombre de points est élevé, plus l'approximation sera proche de f. Mais, comme f est souvent coûteuse à évaluer, on ne dispose en pratique que de peu

Algorithme 9 Nelder-Mead

Précondition : $n+1$ point $x_0, \ldots, x_n$

tant que Test d'arrêt en x_n non vérifié **faire**
  Evaluer f en chacun de ces points les réordonner de telle sorte que : $f(x_0) \leqslant f(x_1) \leqslant \cdots \leqslant f(x_n)$
  Calculer le centre de gravité x_g des points $x_0, \ldots, x_n$
  Construire le point symmétrique $x_s = x_g + (x_g - x_n)$ de x_n par rapport au centre de gravité x_g.
  si $f(x_s) < f(x_0)$ **alors**
    Agrandir le simplexe dans la direction de x_s :
    $x_a \leftarrow x_g + 2(x_g - x_n)$
    si $f(x_a) < f(x_s)$ **alors**
      $x_n \leftarrow x_a$
    sinon
      $x_n \leftarrow x_s$
  si $f(x_{n-1}) < f(x_s)$ **alors**
    Réduire le simplexe :
    $x_r \leftarrow x_k + \frac{1}{2}(x_g - x_n)$
    si $f(x_r) < f(x_s)$ **alors**
      $x_n \leftarrow x_r$
    sinon
      $x_n \leftarrow x_s$
retour x_n

de points pour construire un modèle fiable. Il faudra donc accorder un soin particulier à leur répartition dans l'espace de recherche. De plus, la façon de construire le modèle joue aussi un rôle prépondérant et plusieurs approches sont disponibles dans la littérature : les polynômes, le krigeage ou les réseaux de neurones. Dès lors, on comprend que les deux points clés de ce type de méthode sont le choix des points du modèle et le modèle lui-même. L'algorithme 10 schématise les principes de base de cette catégorie de méthode.

Algorithme 10 Modèles d'interpolation

Précondition : x_0 point initial, n points de modèles $\bar{x}_1, \ldots, \bar{x}_n$

tant que Test d'arrêt en x_k non vérifié **faire**
  Construire un modèle $\bar{f}$ de f à partir de $\bar{x}_1, \ldots, \bar{x}_n$.
  Calculer x_{k+1} minimum de $\bar{f}$ à l'aide de méthodes d'optimisation locales (régions de confiance, quasi-Newton...)
  Mettre à jour les points de modèles en incluant x_k
retour $k \leftarrow k+1$

2.2.3 Méthodes avec contraintes

Le but des paragraphes suivants est de décrire succinctement les méthodes de minimisation locales avec contraintes. En effet, nous détaillerons moins ces méthodes que les précédentes, car elles sont moins utilisées en vision par ordinateur, et ce essentiellement pour deux raisons. La première est que tout simplement certains problèmes, comme l'ajustement de faisceaux, ne sont pas contraints. La seconde est que la nature complexe des contraintes empêche de disposer (même si cela n'est pas toujours nécessaire) d'un point initial les vérifiant. On préfère alors résoudre le problème sans contraintes puis le projeter sur ces dernières.

2.2.3.1 Programmation linéaire

Les problèmes de programmation linéaire (PL) sont des problèmes d'optimisation convexes où la fonction objectif et les contraintes sont toutes linéaires. On cherche donc à résoudre le problème (2.93) :

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ &\text{s.l.c. } g(x) \leqslant 0 \\ &\qquad h(x) = 0, \end{aligned} \tag{2.93}$$

où f, g et h sont des fonctionnelles multivariées linéaires. On rappelle qu'un problème de programmation linéaire est défini sous sa forme standard par (2.17). La programmation linéaire est un domaine central de l'optimisation, car les problèmes de PL sont les problèmes d'optimisation les plus faciles - toutes les contraintes y étant linéaires. Beaucoup de problèmes réels de recherche opérationnelle peuvent être exprimés comme un problème de PL. Pour cette raison, un grand nombre d'algorithmes pour la résolution d'autres problèmes d'optimisation sont fondés sur la résolution de problèmes linéaires. Le terme programmation linéaire suppose que les solutions à trouver doivent être représentées en variables réelles. Cependant, il peut être nécessaire d'utiliser des variables discrètes dans la modélisation du problème (contraintes dites d'intégrité), on parle alors de programmation linéaire en nombres entiers (PLNE). Il est important de savoir que ces derniers sont nettement plus complexes à résoudre que les PL en variables continues. Dans cette section, après avoir présenté les différentes formulations d'un problème de PL, nous faisons une courte introduction à l'algorithme du simplexe.

Méthode du Simplexe. L'algorithme du simplexe - ou algorithme de Dantzig - est une technique à la fois fondamentale et très populaire pour les problèmes de programmation linéaire car très efficace en pratique. Il est implanté dans tous les solveurs de programmes linéaires. Les autres algorithmes de résolution de programmes linéaires sont basés soit sur la méthode de l'ellipsoïde soit sur la méthode des points intérieurs. La stratégie fondamentale de la méthode du simplexe repose sur l'amélioration successive des solutions réalisables du problème (2.17). En effet, chaque itération consiste à passer d'une solution réalisable $x_1, \ldots, x_{n+m}$ à une autre solution $\bar{x}_1, \ldots, \bar{x}_{n+m}$ réalisable telle que :

$$z = \sum_{j=1}^{n} c_j x_j \quad < \quad \bar{z} = \sum_{j=1}^{n} c_j \bar{x}_j.$$

Ce passage d'une solution réalisable à une autre se fait en construisant à chaque itération un système linéaire, appelé **dictionnaire**, vérifiant :

- un dictionnaire exprime z et m variables (parmi $x_1, \ldots, x_{n+m}$) en terme de n autres variables ;
- toute solution d'un dictionnaire doit être aussi solution de (2.17) (qui représente le dictionnaire initial) et réciproquement. Une solution d'un dictionnaire est appelée solution **réalisable de base**.

Ainsi tout dictionnaire décrit une solution réalisable. Cependant, au cours des itérations, il est possible d'obtenir des dictionnaires non bornés, ou des solutions réalisables **dégénérées** (c.-à-d. une solution avec une ou plusieurs variables nulles). La dégénérescence est un phénomène courant en pratique. L'algorithme fait alors du sur-place durant plusieurs itérations avant de poursuivre avec une itération vers une solution non dégénérée. En outre, l'algorithme du simplexe peut aussi cycler, c'est-à-dire obtenir un dictionnaire identique dans deux itérations différentes entre lesquelles les itérations sont dégénérées. Cependant, en tirant parti de la structure particulière du problème, ces difficultés peuvent être évitées. Finalement, l'algorithme du simplexe permet d'affirmer que tout problème de PL a les propriétés suivantes :

- s'il n'a pas de solution optimale, alors il est soit non-réalisable, soit non-borné
- s'il a une solution optimale, alors il a une solution optimale de base.

Ainsi, dans les cas pratiques, l'algorithme du simplexe converge en un nombre fini d'itérations vers une solution. Comme le problème est convexe, cette solution est globale.

2.2.3.2 Programmation Quadratique Séquentielle

Cette méthode, aussi appelée **SQP** pour « **Sequential Quadratic Programming** », consiste à remplacer le problème initial par une suite de problèmes quadratiques pouvant être efficacement résolus (par la méthode des directions conjuguées par exemple).

Problèmes avec contraintes d'égalité. Dans un premiers temps, on s'intéresse à la résolution de

$$\begin{array}{l} \min\ f(x) \\ \text{s.l.c}\ h_i(x) = 0 \quad \forall i \in \{1, \dots, p\}\,. \end{array} \tag{2.94}$$

Le Lagrangien $\mathcal{L}$ de ce problème au point $(x, \lambda) \in \mathbb{R}^{n \times p}$ s'écrit :

$$\mathcal{L}(x, \lambda) \quad = \quad f(x) + \sum_{i=1}^{p} \lambda_i h_i(x). \tag{2.95}$$

Nous avons vu (c.f. 2.34) que si $x^\star$ est une solution de ce problème, alors il existe $\lambda^\star \in \mathbb{R}^p$ tel que :

$$\begin{cases} \nabla f(x^\star) + \sum_{i=1}^{p} \lambda_i^\star \nabla h_i(x^\star) = 0 \\ h_i(x^\star) = 0 \quad \forall i \in \{1, \dots, p\}\,. \end{cases} \tag{2.96}$$

Un première approche consiste à résoudre ce système d'équations non linéaires par la méthode de Newton pour équations (cf. 2.2.2.2). Cette méthode génère alors une suite d'itérés définis par :

$$x_{k+1} \quad = \quad x_k + \delta_x \tag{2.97}$$
$$\lambda^{k+1} \quad = \quad \lambda^k + \delta_\lambda. \tag{2.98}$$

avec δ_x et δ_λ solutions du système linéaire :

$$\begin{pmatrix} \mathcal{H}^k & (\nabla h^k)^\top \\ \nabla h^k & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = \begin{pmatrix} -\nabla f(x_k) - (\nabla h^k)^\top \lambda^k \\ -h(x_k) \end{pmatrix}, \tag{2.99}$$

où

- $\mathcal{H}^k \triangleq \nabla_x^2 \mathcal{L}(x_k, \lambda_k) = \nabla^2 f(x_k) + \sum_{i=1}^{p} \lambda_i^k \nabla^2 h_i(x_k)$ est la matrice hessienne du Lagrangien par rapport à x évaluée en x_k $\mathcal{L}$ au point (x^k, λ^k),
- $\nabla h^k \triangleq [\nabla h_1(x_k)\ \nabla h_2(x_k) \cdots \nabla h_p(x_k)]^\top$ est le Jacobien de h au point x_k.

Au voisinage de l'optimum $(x^\star, \lambda^\star)$, la convergence de la méthode est quadratique, mais comme nous l'avons déjà vu, elle n'est pas assurée pour un point de départ éloigné de l'optimum. Notons qu'il est possible d'utiliser une méthode de quasi-Newton pour éviter le calcul exact de la matrice hessienne du Lagrangien à chaque étape (cf. 2.2.2.3). Mais l'inconvénient majeur de cette approche réside dans la formulation de (2.99). En effet, tous les points critiques (minimum, maximum et point-cols) sont solutions de ce système. Afin de pallier ce défaut, la méthode S.Q.P. reformule (2.99) comme un problème de minimisation quadratique sous contraintes dont seuls les minimums sont solutions. Pour cela, on peut montrer que les équations du système linéaire (2.99) correspondent aux conditions d'optimalité du problème :

$$(\mathcal{Q}_k) : \begin{cases} \min\limits_{\delta_x \in \mathbb{R}^n} \ \nabla_x \mathcal{L}(x_k, \lambda_k)^\top \delta_x + \dfrac{1}{2} \delta_x^\top \mathcal{H}^k \delta_x \\ \text{s.l.c} \ \ h(x_k) + \nabla h^k \delta_x = 0. \end{cases} \tag{2.100}$$

si δ_λ est vu comme le multiplicateur de Lagrange associé à la contrainte de $(\mathcal{Q}_k)$. De la même manière, les équations du système linéaire (2.99) correspondent aux conditions d'optimalité du problème :

$$(\mathcal{Q}'_k) : \begin{cases} \min\limits_{\delta_x \in \mathbb{R}^n} \ \nabla f(x_k)^\top \delta_x + \dfrac{1}{2} \delta_x^\top \mathcal{H}^k \delta_x \\ \text{s.l.c} \ \ h(x_k) + \nabla h^k \delta_x = 0. \end{cases} \tag{2.101}$$

La résolution du problème quadratique $(\mathcal{Q}'_k)$ à chaque itération permet d'éviter les points qui ne sont pas des minima (contrairement à la méthode de Newton). La convergence est quadratique au voisinage de l'optimum (si H^k est définie positive). L'algorithme 11 résume cette première version de la méthode S.Q.P. pour les problèmes à contraintes égalités. Notons

Algorithme 11 S.Q.P. à contraintes égalités

Précondition : $x_0 \in \mathbb{R}^n$, $\lambda^0 \in \mathbb{R}^p$

1: **tant que** Test d'arrêt en x_k non vérifié **faire**
2: Résolution du sous-problème $(\mathcal{Q}'_k)$ pour obtenir une solution primale δ_x et un multiplicateur de Lagrange λ associé à contraintes égalité.
3: Factorisation de $J_r(x_k)^\top J_r(x_k)$
4: $x_{k+1} \leftarrow x_k + \delta_x$
5: $\lambda^{k+1} \leftarrow \lambda^k + \lambda$
6: $k \leftarrow k + 1$
7: **retour** x_k

enfin que cette méthode n'introduit pas de longueur de pas α_k comme les méthodes sans contraintes et qu'il n'a donc pas de comparaison directe entre x_k et $x_k + \alpha_k \delta_x$. Cette étape est réalisée en pratique au moyen de **fonctions de mérite** dont nous ne détaillons pas la théorie ici.

Problèmes avec contraintes égalités et inégalités. On s'intéresse dans ce cas à la résolution de :

$$\begin{array}{ll} \min & f(x) \\ \text{s.l.c} & h_i(x) = 0 \quad \forall i \in \{1, \dots, p\} \\ & g_j(x) \leqslant 0 \quad \forall j \in \{1, \dots, q\}. \end{array} \tag{2.102}$$

Le Lagrangien $\mathcal{L}$ de ce problème au point $(x, \lambda, \mu) \in \mathbb{R}^{n \times p \times q}$ s'écrit :

$$\mathcal{L}(x, \lambda, \mu) \quad = \quad f(x) + \sum_{i=1}^{p} \lambda_i h_i(x) + \sum_{j=1}^{q} \mu_j g_j(x). \tag{2.103}$$

Par analogie au problème précédent, on doit résoudre à chaque étape un sous problème quadratique composé d'une approximation quadratique du Lagrangien $\mathcal{L}$ contraint par une linéarisation de g et h :

$$(\mathcal{Q}_k'') : \begin{cases} \min\limits_{\delta_x \in \mathbb{R}^n} \nabla f(x_k)^\top \delta_x + \dfrac{1}{2} \delta_x^\top \mathcal{H}^k \delta_x \\ \nabla h(x_k) + \nabla h^k \delta_x = 0 \\ \nabla g(x_k) + \nabla g^k \delta_x \leqslant 0. \end{cases} \tag{2.104}$$

Cette démarche conduit à l'algorithme 12 dédié à la minimisation de problème sous contraintes égalités et inégalités.

Algorithme 12 S.Q.P. à contraintes égalité et inégalité

Précondition : $x_0 \in \mathbb{R}^n$, $\lambda_0 \in \mathbb{R}^p$, $\mu_0 \in \mathbb{R}^q$

1: **tant que** Test d'arrêt en x_k non vérifié **faire**
2: Résolution du sous-problème $(\mathcal{Q}_k'')$ pour obtenir une solution primale δ_x, un multiplicateur de Lagrange λ associé aux contraintes égalité et un multiplicateur de Lagrange μ associé aux contraintes inégalité.
3: Factorisation de $J_r(x_k)^\top J_r(x_k)$
4: $x_{k+1} \leftarrow x_k + \delta_x$
5: $\lambda^{k+1} \leftarrow \lambda^k + \lambda$
6: $\mu^{k+1} \leftarrow \mu^k + \mu$
7: $k \leftarrow k + 1$
8: **retour** x_k

2.2.3.3 Les méthodes par pénalisation

Cette autre famille de méthodes adopte une position radicalement différente de celle que nous venons de détailler. L'idée directrice de ces méthodes consiste à remplacer un problème sous contraintes

$$\min_{x \in K} f(x),$$

par une suite de problèmes sans contraintes :

$$\min_{x \in \mathbb{R}^n} f(x) + r\eta(x), \tag{2.105}$$

où η est une fonction qui pénalise la violation des contraintes et $r > 0$ un paramètre de pénalisation. Ainsi, un avantage essentiel de cette famille de méthode réside dans le fait qu'il n'est plus nécessaire de disposer d'un itéré initial admissible. init. Chaque méthode est alors caractérisée par le type de pénalisation qu'elle définit. Deux grandes catégories sont communément utilisées. Elles sont définies par :

Définition 14 *(**Pénalisation**) Soit η une fonction de $\mathbb{R}^n$ dans $\mathbb{R}$. On dira que η définit une **pénalisation extérieure** si elle vérifie :*

- *η est continue sur $\mathbb{R}^n$*
- *$\forall x \in \mathbb{R}^n, \eta(x) \geqslant 0$*
- *$\eta(x) = 0$ si et seulement si $x \in K$.*

*On dira que η définit une **pénalisation intérieure** si elle vérifie :*

- *η est continue sur l'intérieur de K ;*
- *$\forall x \in K, \eta(x) \geqslant 0$;*
- *$\eta(x) \underset{x \to \partial K}{\longrightarrow} \infty$.*

Par exemple, dans le cas où $K = \{x \in \mathbb{R}^n : g_j(x) \leqslant 0,\ i = 1, \ldots, q\}$, la fonction

$$\eta(x) = \sum_{j=1}^{q} \left(\max\{0, g_j(x)\}\right)^2, \tag{2.106}$$

définit une pénalisation extérieure et la fonction

$$\eta(x) = -\sum_{i=1}^{m} \frac{1}{g_i(x)},$$

une pénalisation intérieure. Quelle que soit la pénalisation retenue, la résolution de (2.105) peut alors être effectuée à l'aide de méthodes sans contraintes que nous avons détaillées plus haut (directions conjuguées, quasi-Newton...). Cependant, le coefficient de pénalisation r doit être suffisamment grand pour que le minimum obtenu soit proche de l'ensemble réalisable K, mais suffisamment faible pour éviter un mauvais conditionnement du problème. En pratique, la résolution est réalisée en suivant un chemin de solutions : en partant d'un r_0 faible et en générant la suite de solutions $x^\star(r_k)$ (avec par exemple $r_{k+1} = \alpha r_n,\ \alpha > 1$) jusqu'à ce que $\eta(x^\star(r_k))$ soit faible (c.-à-d. $x^\star(r_k)$ proche de l'ensemble K). La théorie montre que lorsque $r_k \to \infty$, la suite $x^\star(r_k)$ admet au moins un point d'accumulation et que ce dernier est solution optimale du problème.

Pour les fonctions générales, les méthodes par pénalisation ont une précision faible mais elles peuvent être utilisées pour déterminer un point de départ pour des méthodes plus précises et à convergence plus rapide au voisinage de l'optimum (méthode de Newton). Cependant une déclinaison particulière, appelée méthode **des points intérieurs**, se révèle très efficace pour résoudre les problèmes dits de programmation semi-définie que nous verrons dans le chapitre suivant.

2.3 Méthodes d'optimisation globale

Les méthodes d'optimisation globale n'ont de sens que lorsque le problème considéré n'est pas convexe. En effet, si le problème traité est convexe, alors les méthodes locales que nous venons de détailler fournissent un minimum global à partir de l'initialisation utilisée. De manière générale, on classe les algorithmes d'optimisation globale en deux grandes catégories : les méthodes déterministes et les méthodes stochastiques. Le choix d'un algorithme d'optimisation globale est un compromis entre l'information disponible sur la fonction à minimiser, le temps disponible pour résoudre le problème et le degré de précision que l'on souhaite obtenir. Le dernier point revêt une importance essentielle. En effet, certaines méthodes fournissent un encadrement ou une approximation du minimum global alors que d'autres le fournissent avec un certificat d'optimalité. D'un point de vue purement mathématique, on privilégiera une méthode garantissant la globalité de la solution. Cependant, de telles méthodes n'existent que pour certains types de problèmes. D'un point de vue pratique on préfèrera une méthode rapide et fournissant un bon minimum local. En général, ce type de minimum est fourni par des méthodes qui décrivent au mieux (c.-à-d. le plus rapidement possible) l'espace de solutions. Notons que pour les méthodes globales, la notion de vitesse de convergence peut-être délaissée, car elle ne peut s'appliquer à l'ensemble des méthodes.

Dans cette section, nous allons présenter de façon la plus pédagogique les principes des algorithmes d'optimisation globale. Pour une présentation plus exhaustive des méthodes d'optimisation globale, le lecteur pourra se référer à [Mongeau 2007, Weise 2007] qui présentent clairement l'ensemble des méthodes récentes.

2.3.1 Les méthodes déterministes

Les méthodes déterministes regroupent un certain type de méthodes que l'on pourrait définir ainsi : avec les mêmes données, la méthode exécutera toujours la même suite d'opérations. Ces méthodes ne comportent donc aucun aspect aléatoire ou stochastique. Si l'on ne fait aucune hypothèse sur la structure particulière du problème (c.-à-d. programmation linéaire ou quadratique, fonctions convexes...), on peut faire encore une séparation entre les méthodes basées sur des heuristiques et celles disposant d'un « certificat d'optimalité » ou au moins d'un encadrement du minimum global.

2.3.1.1 Conditions d'optimalité du premier ordre

Si un problème sans contraintes a un minimum global et qu'il est possible de résoudre, dans un temps raisonnable, les conditions d'optimalité du premier ordre, alors on dispose de tous les candidats à la minimisation. Ensuite, par une simple comparaison, on détermine celui qui minimise globalement la fonction objectif. Il est évident que la résolution des conditions d'optimalité du premier ordre ne peut se faire que pour des fonctions objectifs particulières. Dans le cas où il s'agit de fonctions objectifs polynomiales ou rationnelles multivariées, les conditions d'optimalité sont alors représentées par un système d'équations polynomiales qui peuvent être résolues par différentes méthodes numériques, citons de manière non exhaustive : les bases de Groebner [Cox 2007], les méthodes d'homotopie [Sommese 2005, Verschelde 2008], le calcul des valeurs propres [Elkadi 2007].

2.3.1.2 L'optimisation par analyse d'intervalles.

L'idée principale de cette méthode consiste à optimiser non plus sur des points de $\mathbb{R}^n$, mais sur des intervalles de $\mathbb{R}^n$. Cela implique que les calculs ne sont plus effectués sur des points mais sur des intervalles. Une analyse d'intervalle a donc été définie (cf. [Hansen 2003, Moore 2009])

et ses propriétés d'encadrement ont été utilisées par plusieurs méthodes d'optimisation globale (voir par exemple [Kearfott 2009]). Les principaux défauts de ces méthodes résident dans la difficulté de construction des fonctions d'inclusions et dans des temps de calcul importants. Cependant ces méthodes ont été utilisées, avec succès, sur des problèmes d'autocalibrage en vision artificielle [Bocquillon 2007].

2.3.1.3 La séparation et évaluation (branch and bound)

Le principe de cette méthode est basé sur l'énumération implicite (cf. [Land 1960]) : toutes les solutions possibles peuvent être énumérées (branching), mais l'analyse des propriétés du problème permet d'éviter l'énumération de larges classes de mauvaises solutions (bounding). Ainsi, dans un bon algorithme de séparation et évaluation, seules les solutions potentiellement bonnes sont énumérées. L'algorithme 13 schématise la structure générale des méthodes de séparation et évaluation. Cependant, toute la difficulté réside dans la phase d'évaluation. En effet, le rejet d'un ensemble E de mauvaises solutions se fait, en général, à l'aide du calcul d'une borne inférieure de la fonction objectif sur un ensemble E' contenant E. L'applicabilité de cette classe d'algorithmes est donc fortement liée au calcul d'une bonne minoration de la fonction coût. En pratique, ce calcul n'est possible qu'en exploitant la structure particulière de certains problèmes comme les problèmes de programmation linéaire en variables mixtes ou de programmation non-linéaire en variables mixtes (MILP et MINLP en anglais). Toutefois, ces algorithmes sont les plus utilisés. Parmi eux, le logiciel BARON [Sahinidis 2010] fait partie des plus populaires.

Algorithme 13 Branch And Bound

Précondition : Structure de donnée L initiale

1: **tant que** Test d'arrêt non vérifié **faire**
2: Extraire un élément E de L
3: Division de E en plusieurs sous-domaines E_i
4: **pour** chaque sous-domaine E_i **faire**
5: Calcul d'un minorant de f sur E_i
6: **si** Le minorant de f sur E_i est supérieur au minimum courant **alors**
7: Supprimer E_i
8: **sinon**
9: Insérer E_i dans L

2.3.1.4 Les méthodes avec heuristique

Pour un grand nombre de variables, ou si l'on ne prend pas en compte certaines propriétés de la fonction coût, les méthodes précédentes requièrent souvent des temps de calcul excessifs. C'est pourquoi, lorsqu'on est confronté aux problèmes pratiques, on se contente souvent de solutions réalisables ayant une bonne valeur de fonction objectif. On a, dans ce cas, recours à des heuristiques. On entend par heuristique [Rayward-Smith 1996, Michalewicz 2004, Pearl 1984], une technique de résolution de problèmes qui tient compte à chaque étape des résultats précédents et en déduit la stratégie à adopter par la suite. Contrairement aux méthodes précédentes, les méthodes heuristiques n'assurent pas que l'on arrivera à un résultat en un nombre fini d'itérations. Nous donnons ci-dessous des exemples d'heuristiques.

L'algorithme Glouton, que l'on pourrait qualifier d'algorithme « myope », recherche à chaque pas, à améliorer le plus possible (en fonction des pas considérés comme acceptables) la fonction objectif. De manière équivalente, on peut dire qu'il est basé sur le postulat de faire, étape par étape, un choix d'optimum local, dans l'espoir d'obtenir un optimum global. **L'algo-**

rithme par améliorations locales se contente d'améliorer la valeur de la fonction objectif en se restreignant à un certain type de déplacements acceptables, de sorte à obtenir à terme un minimum local (relatif au nombre de déplacements acceptables). **L'agorithme par Initialisations multiples** applique une heuristique d'amélioration locale à partir de différentes solutions réalisables [Las 2008]. Enfin **La recherche Tabou** [Glover 1986, Hansen 1990] consiste à poursuivre la recherche d'optima même si la fonction objectif se détériore, afin de s'échapper d'optima locaux, à l'aide de mécanisme évitant le cyclage en interdisant certains retours (rendus « tabous »). Ce dernier exemple est résumé sous la forme d'un pseudo-code dans l'algorithme 14

Algorithme 14 Recherche Tabou

```
tant que Test d'arrêt non vérifié faire
    Choix aléatoire d'un point initial
    Résolution du problème via un algorithme d'optimisation locale
    Ajout d'un tabou rendant non réalisable le voisinage de la solution trouvée
    Gestion de la liste des tabous
```

2.3.2 Les méthodes stochastiques

Cette deuxième classe de méthodes d'optimisation globale regroupe les méthodes comportant une forme de recherche aléatoire, généralement un échantillonnage de la fonction objectif en des points choisis aléatoirement dans le domaine réalisable. De manière schématique, ces méthodes font appel à des tirages de nombres aléatoires pour explorer l'espace des solutions. Ces méthodes sont souvent très faciles à mettre en œuvre puisqu'elles ne nécessitent que des calculs directs de la fonction objectif sans avoir besoin des informations sur les dérivées. On distinguera les méthodes aléatoires (Monte-Carlo et recuit simulé) qui sont basées sur la génération de points successifs de manière aléatoire et les méthodes évolutionnaires (par exemple les algorithmes génétiques) qui font évoluer une population initiale selon des règles établies.

2.3.2.1 Méthodes aléatoires

Une première classe de méthodes, dites *à deux phases*, distingue deux étapes : une phase globale qui évalue la fonction objectif en un nombre de points engendrés aléatoirement puis une phase locale qui manipule ces points, par exemple à l'aide d'améliorations locales. A titre d'exemple, on peut citer la méthode MLSL (multi level single linkage, [Rinnooy Kan 1987]) qui combine propriétés théoriques solides et bonnes performances numériques. La recherche Tabou, que nous avons déjà détaillée, peut par exemple se situer aussi dans cette catégorie.

2.3.2.2 Les heuristiques stochastiques

Ces méthodes, appelées aussi plus généralement méta-heuristiques, utilisent essentiellement la théorie des probabilités pour leur permettre de s'échapper d'optima locaux. Les plus connues sont brièvement décrites ci-dessous.

Méthode du recuit simulé. Cette méthode a été développée par analogie à la physique (en particulier aux procédés de recuit des matériaux) [Kirkpatrick 1983, Dreo 2003]. La probabilité pour un système physique à l'équilibre à la température T de posséder une énergie donnée E est proportionnelle à $e^{\frac{-E}{kT}}$ où k est la constante de Boltzmann. Pour simuler l'évolution d'un système vers son équilibre, on effectue un déplacement élémentaire. Si ce déplacement provoque une diminution de l'énergie, il est accepté. Sinon, il est accepté avec une probabilité de $e^{\frac{-\Delta E}{kT}}$, où ΔE est l'augmentation d'énergie provoquée. Plus la température T est élevée, plus

la probabilité que le déplacement soit accepté est grande. Lorsque l'équilibre est atteint pour une température, celle-ci est diminuée pour recommencer l'opération. L'application à l'optimisation est directe et permet une bonne exploration de l'espace lors des premières itérations (lorsque la température T est élevée) ce qui augmente les chances d'obtenir un optimum global. La convergence de la méthode dépend toutefois du choix de la température initiale, du critère de changement de température et de la loi d'évolution de la température. La température T est donc un paramètre de contrôle de l'optimisation qui conditionne le nombre d'états accessibles. Si elle est diminuée rapidement, elle conduit à un optimum local (équivalent à une trempe). Si elle est diminuée lentement, il est possible d'atteindre l'équilibre à l'optimum. L'algorithme 15 résume cette approche.

Algorithme 15 Recuit Simulé

Précondition : Fixer une température T élevée

tant que la température T n'est pas suffisamment basse **faire**
  Générer un point initial x.
  Générer un point x' par une opération aléatoire sur x.
  si $f(x')$ est meilleur que $f(x)$ **alors**
    $x = x'$
    Réduire la température T
  sinon
    x est remplacé par x' avec probabilité égale à $e^{\frac{-\Delta f}{T}}$

Méthode de Monte-Carlo. Grâce à la simplicité de son principe, cette méthode figure parmi les plus populaires. L'algorithme commence par choisir un point du domaine de façon aléatoire. Puis, les déplacements depuis ce point sont effectués aléatoirement et seuls les déplacements permettant de diminuer la fonction objectif sont acceptés. Pour obtenir des résultats statistiquement fiables, il est nécessaire de réaliser un grand nombre d'évaluations de la fonction (à partir de nombreux points de départ) et ce nombre doit être d'autant plus grand que le nombre de paramètres est élevé. Une telle méthode est donc avantageuse par sa simplicité mais nécessite un temps de calcul pénalisant. L'Algorithme 16 synthétise cette approche.

Algorithme 16 Monte-Carlo

Générer un point initial x
Générer un point x' par une opération aléatoire sur x
tant que Un critère d'arrêt non satisfait **faire**
  si $f(x')$ est meilleur que $f(x)$ **alors**
    $x = x'$
    Générer un point x' par une opération aléatoire sur x
  sinon
    Générer un point initial x

Algorithmes génétiques. Dues à Holland [Holland 1992], ils reproduisent les phénomènes biologiques de l'évolution : ils appliquent la notion de sélection naturelle à une population de solutions réalisables afin de ne conserver que les meilleurs candidats. Ainsi, les algorithmes génétiques font évoluer une population d'individus grâce aux effets combinés de la sélection, de la recombinaison avec croisement et de la mutation. Des exemples de mise en œuvre et d'applications sont présentées par Renders [Renders 1995]. Ces méthodes permettent de

rechercher les solutions dans un espace complexe en ayant un bon compromis entre l'exploitation des meilleures solutions disponibles à un moment donné (comme en programmation mathématique) et l'exploration de l'espace des solutions possibles (comme en recherche aléatoire). L'algorithme permet l'optimisation globale d'une fonction (fonction d'adéquation) à partir d'une population initiale couvrant l'espace des solutions possibles. Chaque individu est composé d'un ensemble de caractéristiques pouvant prendre plusieurs valeurs et possède une valeur de fonction d'adéquation (avec l'environnement) indépendante des autres individus. On cherche alors la meilleure combinaison de ces valeurs pour obtenir le maximum d'adéquation. A chaque génération, une nouvelle population du même nombre d'individus est créée par :

1. **évaluation** des individus (calcul de la valeur de la fonction d'adéquation de chaque individu),
2. **sélection** des individus les plus forts (dont la valeur de la fonction d'adéquation est la plus grande),
3. **reproduction** par croisement entre les individus les plus forts et **mutation** (modification aléatoire de certains paramètres).

Les difficultés d'utilisation en optimisation sont liées au choix d'un certain nombre de paramètres comme l'implantation des variables (la représentation des variables par un ensemble de gènes), le choix de la fonction d'adéquation, le choix des mécanismes de reproduction et de mutation, les probabilités de sélection et de mutation. Pour plus de détails sur ces aspects, le lecteur pourra, par exemple, se référer à [Renders 1995]. Notons enfin qu'il existe d'autres méthodes dans cette catégorie, comme l'optimisation globale par colonie de fourmis [Dorigo 2005] ou l'optimisation globale par essaims particulaires [Vaz 2007].

2.4 Conclusion

Dans ce chapitre, nous avons effectué une présentation exhaustive de l'ensemble des familles de l'optimisation mono-objectif (cf. Figure 2.2). Nous les avons classées en deux grandes catégories : l'optimisation locale et l'optimisation globale. Les méthodes de la première catégorie ne convergent généralement pas vers un minimum global et sont fortement dépendantes de l'estimée initiale choisie. De plus, lorsque le problème présente des contraintes complexes, il peut s'avérer difficile de déterminer un tel point de départ. Les méthodes de la deuxième catégorie ne présentent pas de tels défauts. Mais elles présentent, elles aussi, certains inconvénients. Nous avons vu qu'au sein de cette deuxième famille, il existait deux classes de méthodes. Les méthodes globales de la première classe décrivent le plus efficacement possible l'espace de recherche. Cette efficacité est étroitement liée à la connaissance qu'elles ont, a priori, du problème et des informations qu'elles peuvent en tirer. Cependant, bien que l'espace de recherche soit balayé de manière exhaustive, les solutions proposées ne peuvent être qualifiées que de candidats « très probables » à l'optimum global. Pour obtenir une telle certification, il faut se tourner vers les méthodes globales de la deuxième classe. Celles-ci garantissent l'optimalité globale à partir de la théorie mathématique dont elles sont issues. Cependant, ces théories ne sont, en général, valables que pour des types très particuliers de fonctions coûts. Dans le chapitre suivant, nous allons exposer une telle méthode dédiée à l'optimisation polynomiale.

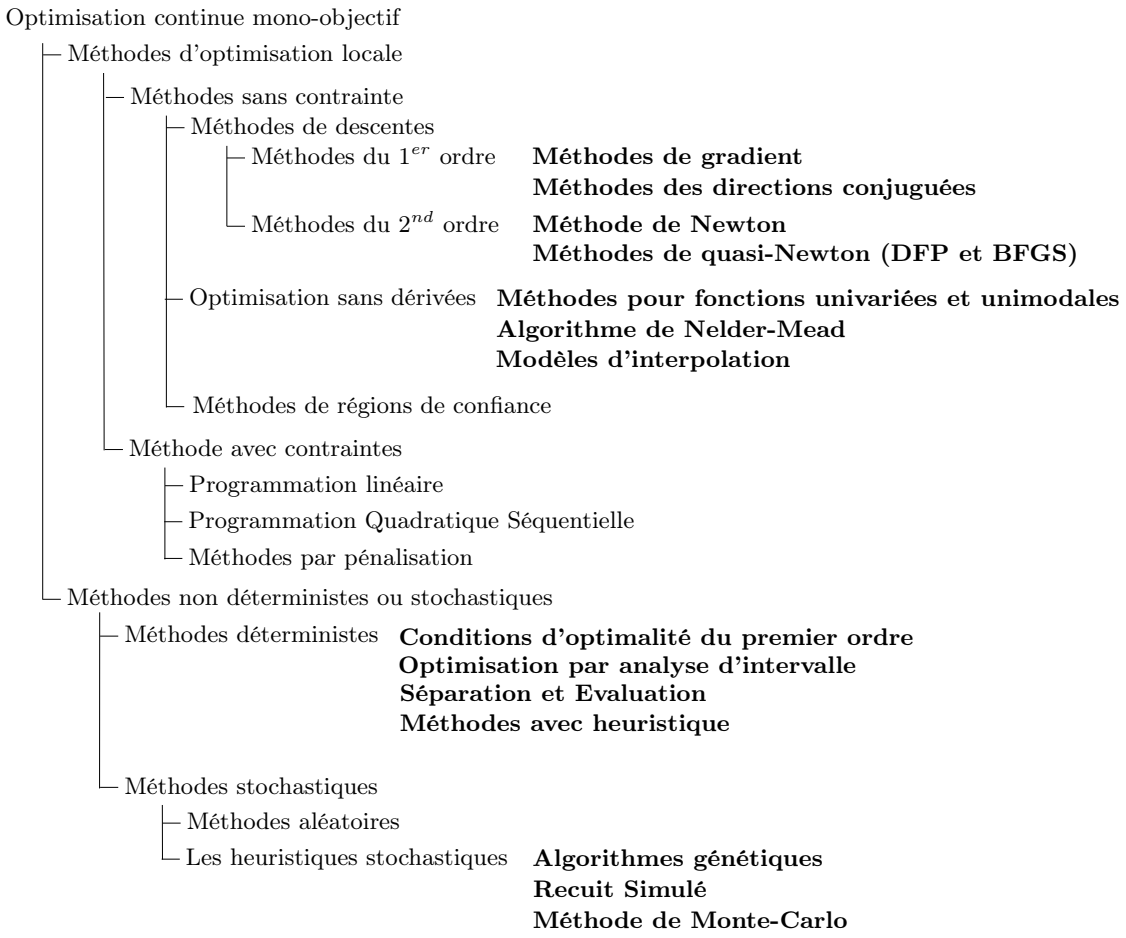


FIGURE 2.2 – Diagramme représentant l'ensemble des familles de l'optimisation mono-objectif présentées dans ce chapitre.

3

Optimisation globale via moments et polynômes positifs

Sommaire

3.1 Introduction à l'optimisation par théorie des moments 41
3.1.1 Notion de mesure. 43
3.1.2 Moments et polynômes positifs. 44
3.1.3 Optimisation globale polynomiale par théorie des moments. 49
3.1.4 Faible couplage des variables. 53
3.1.5 Bilan. 57
3.2 Extension au cas d'une somme de fractions rationnelles 58
3.2.1 Les épigraphes creux. 59
3.2.2 Programmation rationnelle. 61
3.2.3 Bilan. 66
3.3 Résultats numériques 67
3.3.1 Problèmes univariés 68
3.3.2 Problème bivarié 74
3.3.3 Problème de Kowalik 75
3.3.4 Exemple multivarié creux 76
3.3.5 Fonctions de Shekel 80
3.3.6 Bilan 82
3.4 Conclusion 82

Le but de ce chapitre est d'introduire la technique d'optimisation globale polynomiale sur laquelle est basée la contribution théorique de nos travaux. Rappelons que, par rapport à la classification proposée au chapitre précédent, cette technique appartient aux méthodes d'optimisation globale déterministes. La motivation conduisant à appliquer et développer cette méthode est qu'elle permet de résoudre globalement des problèmes de minimisation de fonctions polynomiales sous contraintes polynomiales :

$$\inf_{x \in K} p(x) \tag{3.1}$$

où :

$$K \triangleq \{x \in \mathbb{R}^n \mid g_1(x) \geqslant 0, \ldots, g_m(x) \geqslant 0\} \tag{3.2}$$

avec $p, g_1, \ldots, g_m : \mathbb{R}^n \mapsto \mathbb{R}$ des polynômes à coefficients réels. Or, il est courant en vision artificielle de minimiser des fonctions coûts dites **algébriques**. Ces fonctions sont en général polynomiales. Elles sont définies dans les espaces projectifs $\mathbb{P}^2$ (ensemble des droites vectorielles de $\mathbb{R}^2$) et $\mathbb{P}^3$ (ensemble des droites vectorielles de $\mathbb{R}^3$) et ont, souvent, pour objectif la recherche d'invariants. Les minimisations de ces fonctionnelles sont théoriquement assujetties à des contraintes de structures caractérisant ces invariants (par ex. des contraintes de rang sur des matrices) et des contraintes permettant d'éviter la solution triviale nulle, conduisant à formuler des problèmes du type (3.1). Or, lorsque les fonctions $g_1, \ldots, g_m$ sont fortement non linéaires, il est aussi difficile de trouver un point dans K que de résoudre le problème $\inf_{x \in K} f(x)$ lui-même. Les méthodes utilisées classiquement dans la communauté de vision artificielle préfèrent ainsi minimiser la fonction coût sans contrainte, puis projeter, lorsque cela est possible, la solution obtenue sur K. Cependant, ce processus n'est concevable en pratique que si la projection de $x^\star$ sur K peut être réalisée par une suite de manipulations algébriques simples et rapides (par ex. diagonalisation, décomposition en valeurs singulières...). L'archétype de ce type de problème est l'estimation de la matrice fondamentale que nous détaillerons dans le chapitre suivant. Cependant, l'opération de projection peut ne pas se décomposer en une suite d'opérations algébriques simples, rendant ainsi difficile le calcul d'un point dans K. De plus, même si cette projection peut être déterminée, rien n'assure a priori qu'elle soit proche d'un optimum, même local. Etant donné que la méthode d'optimisation par théorie des moments permet de s'affranchir d'une estimée initiale pour des problèmes avec contraintes, on comprend tout l'avantage de son utilisation pour résoudre ce type de problèmes.

Le deuxième type de fonctions coûts fréquemment rencontrées en vision artificielle sont les fonctions coût dites **métriques** ou **euclidiennes**. Ces fonctions sont, en général, rationnelles et, pour la plupart, issues de la projection des fonctions coûts algébriques dans les espaces euclidiens associés (c.-à-d. $\mathbb{R}^2$ et $\mathbb{R}^3$). Cette projection est réalisée au moyen de l'opérateur de projection canonique de $\mathbb{P}^n$ dans $\mathbb{R}^n$ défini par :

$$\begin{array}{rccc} \Psi_n : & \mathbb{P}^n & \rightarrow & \mathbb{R}^n \\ & (x_1, \ldots, x_n, s)^\top & \rightarrow & \left(\frac{x_1}{s}, \ldots, \frac{x_n}{s}\right)^\top . \end{array} \tag{3.3}$$

Notons que seules les fonctions coûts sont projetées, les contraintes restent, pour leur part, inchangées. Les problèmes de minimisation de telles fonctionnelles se formulent alors, de manière générique, de la manière suivante :

$$\inf_{x \in K} \sum_{i=1}^{N} \frac{p_i(x)}{q_i(x)}. \tag{3.4}$$

où les p_i et les q_i sont des polynômes multivariés à coefficients réels. Ainsi, notre apport consiste à étendre la méthode d'optimisation polynomiale à des fonctions coûts décrites par des sommes de fractions rationnelles. Comme nous le verrons dans le chapitre suivant, il existe déjà en vision par ordinateur des méthodes d'optimisation globale dédiées à ce type de problème. Cependant, elles ne considèrent que des contraintes simples (par ex. linéaires, affines...) et ne peuvent être mises en œuvre qu'au prix d'hypothèses théoriques faites sur les numérateurs et dénominateurs. On verra ainsi que leurs implantations pratiques sont réalisées au détriment de leurs généralités. Au contraire, l'avantage des méthodes que nous proposerons sera de pouvoir traiter, sans hypothèses a priori sur les fractions $\frac{p_i}{q_i}$, n'importe quel problème de la forme (3.4).

Ce chapitre s'articule de la façon suivante : nous ferons tout d'abord un rappel de différents éléments théoriques, ensuite nous présenterons le principe de l'optimisation polynomiale et

enfin les différentes extensions possibles au cas d'une somme de fractions rationnelles. Une dernière section sera consacrée aux résultats obtenus par les approches proposées sur des cas tests théoriques issus de la communauté d'optimisation globale.

3.1 Introduction à l'optimisation par théorie des moments

Le but de cette section est de présenter les concepts fondamentaux nécessaires à la compréhension de l'optimisation par théorie des moments. Essayons, tout d'abord, d'appréhender le principe de cette approche. Soit $K \subseteq \mathbb{R}^n$ et f une fonction de K dans $\mathbb{R}$, de manière générale on cherche à minimiser globalement f sur K ou, autrement dit, à résoudre le problème :

$$f^\star = \inf_{x \in K} f(x). \tag{3.5}$$

Pour plus de simplicité, nous supposerons que K est compact, de sorte que (3.5) ait, au moins, une solution (c.-à-d. inf = min). Il est facile de montrer que résoudre ce problème est équivalent à résoudre le problème :

$$\begin{array}{l} \max \ \ \lambda \\ \text{s.l.c } f(x) - \lambda \geqslant 0, \ \forall x \in K. \end{array} \tag{3.6}$$

Cependant, à cause de la contrainte $f(x) - \lambda \geqslant 0 \ \forall x \in K$ (c.-à-d. une infinité non dénombrable de contraintes scalaires), ce problème est aussi complexe que l'original. Une autre formulation du problème initial (3.5) est la suivante :

$$\widehat{f} = \min_{\mu \in \mathcal{P}(K)} \int_K f(x) d\mu \tag{3.7}$$

où $\mathcal{P}(K)$ est l'espace des mesures de probabilité sur K. En effet, puisque :

$$f^\star \leqslant f(x), \ \forall x \in K, \tag{3.8}$$

on a nécessairement que :

$$f^\star = \int_K f^\star \, d\mu \leqslant \int_K f(x) \, d\mu, \qquad \forall \mu \in \mathcal{P}(K). \tag{3.9}$$

Donc $f^\star \leqslant \widehat{f}$. De plus, si $x^\star$ est une solution de (3.5), alors $\delta_{x^\star}$ la mesure de **Dirac** au point $x^\star$ définie, pour tout sous-ensemble A de $\mathbb{R}$, par :

$$\delta_{x^\star}(A) \triangleq \begin{cases} 1 \text{ si } x^\star \in A, \\ 0 \text{ sinon} \end{cases} \tag{3.10}$$

est admissible pour le problème (3.7) et donc :

$$f^\star = \int_K f(x) \, \delta_{x^\star} \geqslant \widehat{f}. \tag{3.11}$$

De même que pour (3.6), ce problème infini-dimensionnel n'apparaît pas, non plus, comme une simplification de (3.5). En effet, nous avons remplacé les variables d'optimisation classiques par des objets abstraits (les mesures sur K) difficiles à manipuler en pratique. Toutefois, quand f est un polynôme, le problème (3.7) peut, généralement, se ramener à un problème en dimension finie. Par exemple, lorsque $n = 1$, si $f : \mathbb{R} \to \mathbb{R}$ est un polynôme de degré r, alors il s'écrit sous la forme $f(x) = \sum_{i=0}^{r} f_i x^i$ où certains f_i peuvent être nuls. Ainsi, son intégrale se décompose de la manière suivante :

$$\int_K f(x) d\mu = \sum_{i=0}^{r} f_i \underbrace{\int_K x^i \, d\mu}_{\triangleq y_i}. \tag{3.12}$$

Par conséquent, s'il est possible de remplacer chaque intégrale $\int_K x^i d\mu$ par y_i, les coordonnées d'un vecteur $\mathbf{y}$, on se ramène au problème :

$$\begin{aligned} &\min \sum_{i=1}^{r} f_i y_i \\ &\text{s.l.c } \mathbf{y} \in \left\{ \mathbf{y} \in \mathbb{R}^r \mid \exists \mu \in \mathcal{P}(K) \text{ telle que : } y_i = \textstyle\int_K x^i d\mu \ \ \forall i = 1, \ldots, r \right\}. \end{aligned} \tag{3.13}$$

Cependant, bien que le problème soit de dimension finie, on ne sait pas caractériser le cône convexe $\left\{ \mathbf{y} \in \mathbb{R}^r \mid \exists \mu \in \mathcal{P}(K) \text{ telle que : } y_i = \int_K x^i d\mu \ \ \forall i = 1, \ldots, r \right\}$ en termes de contraintes simples sur $\mathbf{y}$. Mais, pour $n \geqslant 1$, le problème d'appartenance à ce cône, couramment appelé le **problème des moments sur** K, est résolu dans le cas où K est un ensemble **semi-algébrique de base**. Cette catégorie particulière d'ensembles est définie par :

$$K \ \triangleq \ \{x \in \mathbb{R}^n \ \mid \ g_1(x) \geqslant 0, \ldots, g_m(x) \geqslant 0\} \tag{3.14}$$

avec $g_1, \ldots, g_m : \mathbb{R}^n \mapsto \mathbb{R}$ des polynômes à coefficients réels.

Nous verrons ensuite que cette caractérisation permet de définir une hiérarchie de problèmes convexes de dimensions finies dont la suite monotone des valeurs optimales converge vers $f^\star$. De plus, cette convergence est souvent finie en pratique. L'ensemble des étapes que nous allons détailler est résumé sur la Figure 3.1.

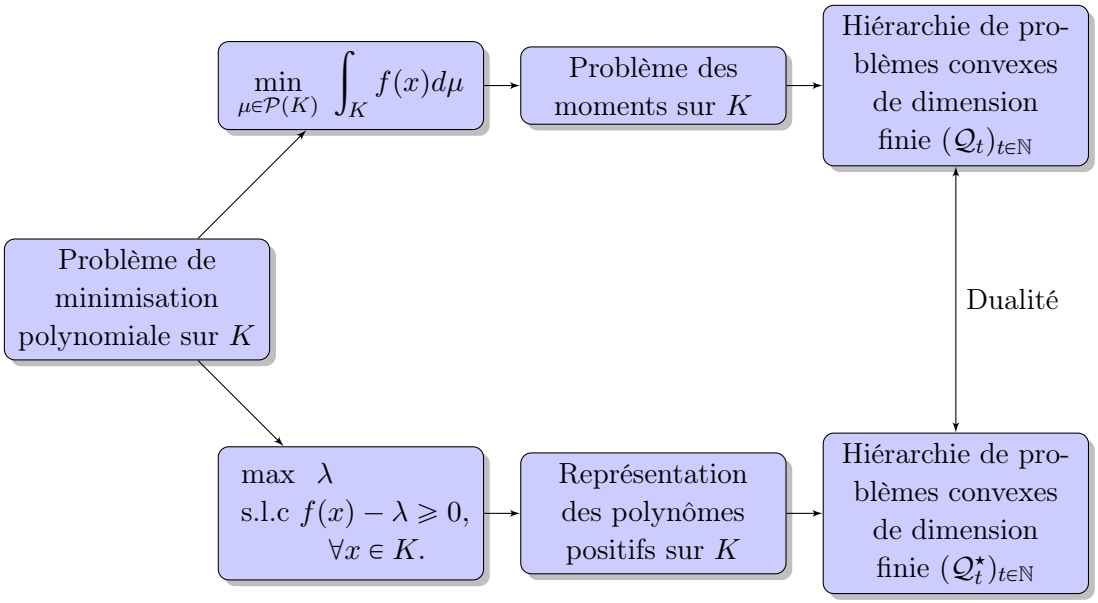


FIGURE 3.1 – Diagramme récapitulatif de l'optimisation par théorie des moments.

Cette section s'organise comme suit : après une brève présentation du concept de **mesure**, nous détaillerons le problème de la représentation des polynômes positifs sur ensemble semi-algébrique de base K, ainsi que la dualité avec le problème des moments sur K. Enfin nous terminerons en exposant la méthode d'optimisation basée sur cette théorie.

3.1.1 Notion de mesure.

Avant d'aborder le concept de mesure proprement dit, il est utile de rappeler quelques principes basiques de cette théorie. Une mesure μ est une application qui associe à tout ensemble A (dans une certaine classe $\mathcal{T}$) un nombre positif $\mu(A)$, appelé mesure de A, vérifiant certaines propriétés (monotonie, additivité...). Par exemple, la mesure de Lebesgue correspond à la longueur en dimension 1, à l'aire en dimension 2 et au volume en dimension 3. Cependant, dans le cas général, une mesure ne peut pas être définie pour tout sous-ensemble de $\mathbb{R}^n$. Il est nécessaire de se restreindre à une sous-famille d'ensembles avec des propriétés particulières. Cette sous-famille est appelée une **tribu** ou **σ-algèbre** :

Définition 15 *(**Tribu**) Un ensemble $\mathcal{T}$ de parties de $\mathbb{R}^n$ est appelé une **tribu** ou une **σ-algèbre** s'il vérifie les propriétés suivantes :*

1. $\mathbb{R}^n \in \mathcal{T}$
2. *Stabilité par passage au complémentaire : si $A \in \mathcal{T}$ alors $A^{\complement} \in \mathcal{T}$ (où $A^{\complement} \triangleq \mathbb{R}^n \backslash A$ est le complémentaire de A dans $\mathbb{R}^n$)*
3. *Stabilité par réunion dénombrable : si $(A_n)_{n\in\mathbb{N}}$ est une suite de $\mathcal{T}$ alors $\bigcup_{n=1}^{\infty} A_n \in \mathcal{T}$.*

Il découle de cette définition que l'intersection d'une famille quelconque de tribus est encore une tribu. Ceci permet de définir la notion de tribu engendrée par une famille de parties de $\mathbb{R}^n$:

Définition 16 *(**Tribu engendrée**) Soit F une famille de parties de $\mathbb{R}^n$ et $\sigma(F)$ la tribu définie par :*

$$\sigma(F) = \bigcap_{\{\mathcal{T} \text{ tribu de } \mathbb{R}^n \mid F\subset\mathcal{T}\}} \mathcal{T}. \tag{3.15}$$

$\sigma(F)$ est appelée la tribu engendrée par F. C'est la plus petite tribu de $\mathbb{R}^n$ contenant F.

Ainsi, on appelle **tribu de Borel** la tribu $\sigma(\mathcal{O})$ engendrée par $\mathcal{O}$ l'ensemble des ouverts de $\mathbb{R}^n$. Cette tribu est notée $\mathcal{B}(\mathbb{R}^n) \triangleq \sigma(\mathcal{O})$. En toute généralité, il est possible de définir une mesure sur n'importe quelle tribu, cependant nous nous limiterons dans ce mémoire à l'utilisation des mesures dites **boréliennes**, c'est-à-dire définies sur $\mathcal{B}(\mathbb{R}^n)$. Plus formellement, ces mesures sont définies par :

Définition 17 *(**Mesure borélienne**) Une mesure **borélienne** est une application de $\mathcal{B}(\mathbb{R}^n)$ dans $\mathbb{R}_+ \cup \{+\infty\}$ vérifiant les propriétés suivantes :*

1. $\mu(\varnothing) = 0$
2. *l'additivité dénombrable ou σ-additivité : si $(A_i)_{i\in\mathbb{N}}$ est une suite d'ensembles de $\mathbb{R}^n$ tels que $A_i \cap A_j = \varnothing$ dès que $i \neq j$, alors :*

$$\mu\Big(\bigcup_{i=1}^{\infty} A_i\Big) = \sum_{i=1}^{\infty} \mu(A_i). \tag{3.16}$$

*On dit qu'une mesure est finie si $\mu(\mathbb{R}^n) < \infty$ et on note $\mathcal{M}_+(\mathbb{R}^n)$ l'ensemble des mesures **boréliennes finies** sur $\mathbb{R}^n$. Enfin, les mesures boréliennes finies telles que $\mu(\mathbb{R}^n) = 1$ sont appelées des mesures de **probabilités** et on désigne par $\mathcal{P}(\mathbb{R}^n)$ l'ensemble de ces mesures.*

Enfin, terminons ces rappels par la notion de **support** d'une mesure :

Définition 18 *(**Support d'une mesure**) Soit μ une mesure borélienne, son **support** noté $\mathrm{supp}(\mu)$, est défini comme le plus petit ensemble fermé de $\mathbb{R}^n$ tel que $\mu(\mathbb{R}^n \backslash \mathrm{supp}(\mu)) = 0$. Par suite, si μ est une mesure borélienne et $K \subset \mathbb{R}^n$ un ensemble semi-algébrique de base, on dira que μ est une **mesure supportée** sur K si $\mathrm{supp}(\mu) \subseteq K$. On notera alors $\mathcal{M}_+(K)$ (resp. $\mathcal{P}(K)$) l'ensemble des mesures boréliennes finies supportées sur K (resp. l'ensemble des mesures de probabilité supportées sur K).*

3.1.2 Moments et polynômes positifs.

Le but de ce paragraphe est de présenter la dualité entre la représentation des polynômes positifs sur un ensemble semi-algébrique de base K et le problème des moments sur K. Rappelons que ce problème se formule de la manière suivante : quelles sont les conditions nécessaires pour qu'une séquence de nombres représente les moments d'une mesure supportée sur K. Ainsi nous commencerons par définir l'ensemble des notations nécessaires, puis nous introduirons la notion de polynôme **somme de carrés** et comment ceux-ci peuvent servir à représenter les polynômes positifs sur K. Enfin, nous présenterons les concepts de **moment**, de **matrice des moments** et de **matrice de localisation** d'une mesure de $\mathcal{M}_+(K)$ ainsi que le lien entre ces notions et la représentation des polynômes positifs.

3.1.2.1 Notation

Soit $t \in \mathbb{N}$, introduisons tout d'abord la notation de l'espace de dimension n des vecteurs d'entiers positifs de norme 1 inférieure ou égale à t. Noté $\mathbb{N}^n_t$, il est défini par :

$$\mathbb{N}^n_t \triangleq \left\{ \alpha \in \mathbb{N}^n \mid \|\alpha\|_1 = \sum_{i=1}^{n} \alpha_i \leqslant t \right\}. \tag{3.17}$$

Cet ensemble, de dimension $s(t) \triangleq \binom{n+t}{n}$, nous servira essentiellement à définir les degrés des monômes et des polynômes multivariés. Ainsi, pour $\alpha \in \mathbb{N}^n_t$ et $x \in \mathbb{R}^n$, on définit le **monôme** x^α par :

$$x^\alpha \triangleq x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}, \text{ pour tout } \alpha \in \mathbb{N}^n. \tag{3.18}$$

Son degré, noté $\deg(x^\alpha)$, est par définition égal à la norme 1 du vecteur α :

$$\deg(x^\alpha) \triangleq \|\alpha\|_1. \tag{3.19}$$

On note ensuite $\mathbb{R}[x] = \mathbb{R}[x_1, x_2, \ldots, x_n]$, l'anneau des polynômes réels à n variables et $\mathbb{R}_t[x]$ celui des polynômes réels à n variables de degré au plus t. Soit :

$$\left\{1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_1x_n, x_2x_3, \ldots, x_n^2, \ldots, x_1^t, \ldots, x_n^t\right\} \tag{3.20}$$

la base canonique de $\mathbb{R}_t[x]$ de dimension $s(t)$. Tout polynôme p de cet espace se décompose de la manière suivante :

$$p(x) = \sum_{\alpha \in \mathbb{N}^n_t} p_\alpha x^\alpha. \tag{3.21}$$

Ainsi p peut être identifié au vecteur de coefficients $\mathbf{p} \triangleq (p_\alpha)_{\|\alpha\|_1 \leqslant t}$. Comme chaque indice α appartient à $\mathbb{N}^n_t$, le vecteur $\mathbf{p}$ appartient à $\mathbb{R}^{\mathbb{N}^n_t} \simeq \mathbb{R}^{s(t)}$. Les matrices de $\mathbb{R}^{\mathbb{N}^n_t} \times \mathbb{R}^{\mathbb{N}^n_t}$ (c.-à-d. les applications linéaires de $\mathbb{R}^{\mathbb{N}^n_t}$ dans $\mathbb{R}^{\mathbb{N}^n_t}$) sont, elles aussi, indexées à l'aide de $\mathbb{N}^n_t$. Pour une telle matrice M, le produit matrice-vecteur s'écrit de manière similaire à celui des indices classiques :

$$(M\mathbf{p})_\beta = \sum_{\alpha \in \mathbb{N}^n_t} M_{\beta,\alpha} p_\alpha. \tag{3.22}$$

Ensuite, le degré du polynôme p, noté $\deg(p)$, est défini par :

$$\deg(p) \triangleq \max_{\{\alpha \mid p_\alpha \neq 0\}} \|\alpha\|_1, \tag{3.23}$$

et on définit le « demi-degré » d_p par :

$$\mathrm{d}_p \triangleq \lceil \deg(p)/2 \rceil, \tag{3.24}$$

où $\lceil\ \rceil$ désigne la partie entière par excès. Notons que ces définitions sont indépendantes de la base de l'espace vectoriel des polynômes choisie. Par suite, pour une famille de polynômes $g_1, \ldots, g_m$ on définit :

$$\mathrm{d}_g \triangleq \max(\mathrm{d}_{g_1}, \ldots, \mathrm{d}_{g_m}). \tag{3.25}$$

Cette dernière définition peut être indifféremment utilisée aussi bien pour une famille de polynômes que pour un ensemble semi-algébrique de base K. Dans ce dernier cas, on posera $\mathrm{d}_K \triangleq \mathrm{d}_g$.

3.1.2.2 Polynômes positifs

En toute généralité, caractériser les polynômes positifs sur tout l'espace $\mathbb{R}^n$ est un problème difficile. Cependant, sur un ensemble semi-algébrique de base K, il existe plusieurs résultats théoriques. Afin d'introduire ces résultats, on rappelle d'abord la notion de polynôme **somme de carrés** ou polynôme **SOS** (comme Sum Of Squares en anglais) :

Définition 19 *(**Polynôme SOS**) Un polynôme p réel est dit **SOS**, s'il peut s'écrire comme une somme finie de carrés de polynômes réels :*

$$p(x) \quad = \quad \sum_{j=1}^{M} u_j(x)^2 \ \textit{où les } u_j \textit{ sont des polynômes réels.} \tag{3.26}$$

De plus (cf [Laurent 2009]), pour toute décomposition $(u_j)_j$ de p, on a $\deg(u_j) \leqslant \frac{\deg(p)}{2}$ pour tout j. On notera $\Sigma[x]$, l'ensemble de tous les polynômes SOS de $\mathbb{R}^n$ dans $\mathbb{R}$.

Il existe une condition permettant de détecter si un polynôme de degré fini est une somme de carrés :

Théorème 2 *(**Détection de polynômes SOS**) Si $p = \sum_{\alpha \in \mathbb{N}^n_{2t}} p_\alpha x^\alpha$ est un polynôme de $\mathbb{R}_{2t}[x]$, alors p est un polynôme SOS si et seulement si le problème suivant :*

$$\begin{cases} X \succeq 0 \\ \displaystyle\sum_{\{\gamma,\beta \in \mathbb{N}^n_t \mid \gamma+\beta=\alpha\}} X_{\beta,\gamma} = p_\alpha, \quad \forall \alpha \ \mid \ \|\alpha\|_1 \leqslant 2t \end{cases} \tag{3.27}$$

possède une solution.

Le problème (3.27) a pour inconnue une matrice symétrique $X = (X)_{\beta,\gamma \in \mathbb{N}^n_t}$ de taille $\binom{n+t}{n}$ à $\binom{n+2t}{n}$ variables. La complexité de résolution de (3.27) croît donc rapidement avec n et t. Cependant, le point important réside dans la forme du problème (3.27). En effet, il peut être assimilé à un problème de **Programmation Semi-Définie-Positive** (SDP). Un problème de cette catégorie se présente sous la forme :

$$\begin{array}{ll} \min\limits_{X \in \mathbb{S}^n} & \mathrm{trace}\left(C^\top X\right) \\ \text{s.l.c.} & \mathrm{trace}\left(A_k^\top X\right) = b_k, \quad k = 1, \ldots, m, \\ & X \succeq 0 \end{array} \tag{3.28}$$

où $A_k \in \mathbb{R}^{n \times n}$ pour tout k dans $\{1, \ldots, m\}$ et $\mathbb{S}^n$ est l'espace des matrices symétriques de taille n. Comme ces problèmes sont convexes, ils peuvent être efficacement résolus par des méthodes de points intérieurs [Vandenberghe 1996]. Beaucoup de ces solveurs sont disponibles via l'interface YALMIP([Löfberg 2004]). On peut en citer plusieurs comme : CSDP [Brian 1999], DSDP [Benson 2005], SDPA [Fujisawa 2003], SDPT3 [Toh 2003], SeDuMi [Sturm 1999]. Il est important de souligner l'importance de cette classe de problèmes, car chaque élément de la hiérarchie que nous cherchons à construire est un problème SDP.

Définissons ensuite la notion de **module quadratique** engendrés par une famille de polynômes $g_1, \ldots, g_m$:

Définition 20 *(**Module quadratique**) Soit $g_1, \ldots, g_m$ une famille de polynômes de $\mathbb{R}^n$ dans $\mathbb{R}$, l'ensemble*

$$\mathbf{M}(g_1, \ldots, g_m) \triangleq \left\{ u_0 + \sum_{j=1}^{m} u_j g_j, \quad \text{où les } u_j \text{ sont dans } \Sigma[x] \right\} \tag{3.29}$$

*est appelé **le module quadratique** engendré par la famille $g_1, \ldots, g_m$. Le module quadratique sera dit **archimédien** si :*

$$\exists N \in \mathbb{N} \text{ tel que } N - \sum_{i=1}^{n} x_i^2 \in \mathbf{M}(g_1, \ldots, g_m). \tag{3.30}$$

Il est important de souligner que cette propriété implique la compacité de K.

Maintenant que nous avons introduit les notions et notations nécessaires, énonçons le résultat théorique essentiel permettant de caractériser les polynômes positifs sur un ensemble semi-algébrique de base :

Théorème 3 *(**Positivstellensatz de Putinar**) Si le module quadratique $\mathbf{M}(g_1, \ldots, g_m)$ est archimédien et si $p \in \mathbb{R}[x]$ est strictement positif sur K, alors il existe une famille $(u_j)_{j=0,\ldots,m}$ de $m+1$ polynômes SOS telle que :*

$$p = u_0 + \sum_{j=1}^{m} u_j g_j. \tag{3.31}$$

Autrement dit, tout polynôme strictement positif sur K appartient à $M(g_1, \ldots, g_m)$, le module quadratique engendré par la famille de polynômes $(g_j)_{j=1\ldots m}$.

Remarquons qu'en pratique la condition « $\mathbf{M}(g_1, \ldots, g_m)$ est archimédien » peut être facilement réalisée. En effet, comme K doit être compact, il suffit de connaître N tel que :

$$K \subseteq \left\{ x \in \mathbb{R}^n \mid N - \|x\|^2 \geqslant 0 \right\}, \tag{3.32}$$

et d'inclure la contrainte redondante $g_{m+1}(x) = N - \|x\|_2^2$ à la famille $(g_j)_{j=1,\ldots,m}$. Notons qu'il existe un résultat plus général de caractérisation de polynômes positifs sur un semi-algébrique de base. Ce résultat, appelé **Positivstellensatz de Schmüdgen**, ne nécessite qu'une hypothèse de compacité sur K. Mais, bien que remarquable d'un point de vue théorique, il se révèle peu utilisable en pratique, car le nombre de polynômes SOS utilisés dans la décomposition de p croît exponentiellement avec le nombre de polynômes g_j.

A présent que nous savons caractériser les polynômes positifs sur un ensemble semi-algébrique de base K, nous exposons dans le paragraphe suivant la dualité entre la caractérisation des polynômes positifs et le problème des moments sur K.

3.1.2.3 Moments d'une mesure

La représentation des polynômes positifs sur K a une facette duale qui est le problème des moments sur K, c'est-à-dire la caractérisation des séquences de nombres qui sont les moments d'une mesure supportée sur K. Pour caractériser ces séquences, nous introduisons la **fonctionnelle de Riesz**. Celle-ci formalise et généralise à $\mathbb{R}[x]$ l'application linéaire sur les polynômes univariés que nous avons évoquée dans l'introduction (cf. (3.13)).

Définition 21 *(**Fonctionnelle de Riesz**) Soit* $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}$*, la forme linéaire* $L_{\mathbf{y}}$ *de* $\mathbb{R}[x]$ *dans* $\mathbb{R}$ *définie par :*

$$\begin{array}{rccl} L_{\mathbf{y}}: & \mathbb{R}[x] & \to & \mathbb{R} \\ & p = \sum\limits_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha & \to & \mathbf{y}^\top \mathbf{p} = \sum\limits_{\alpha \in \mathbb{N}^n} p_\alpha y_\alpha \quad \text{où } \mathbf{p} = (p_\alpha)_{\alpha \in \mathbb{N}^n}, \end{array} \tag{3.33}$$

est appelée la ***fonctionnelle de Riesz****.*

Si $p \in \mathbb{R}[x]$, remarquons alors que :

$$\int_K p \, d\mu = \int_K \sum_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha d\mu = \sum_{\alpha \in \mathbb{N}^n} p_\alpha \int_K x^\alpha d\mu = L_{\mathbf{y}}(p). \tag{3.34}$$

où $\mathbf{y} = (y_\alpha)_{\alpha \in \mathbb{N}^n}$ avec $y_\alpha = \int_K x^\alpha d\mu$, $\forall \alpha \in \mathbb{N}^n$. Pour un α donné, l'intégrale sur K du monôme x^α est appelée **moment d'ordre** α de la mesure μ. Réciproquement, le problème des moments considère la question : étant donné $\mathbf{y} = (y_\alpha)_{\alpha \in \mathbb{N}^n}$ sous quelles conditions existe-t-il $\mu \in \mathcal{M}_+(K)$ telle que $y_\alpha = \int_K x^\alpha d\mu$, $\forall \alpha \in \mathbb{N}^n$. La réponse est donnée par le résultat suivant :

Théorème 4 *(**Putinar**) Soit* $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}$ *une séquence infinie de réels et* K *un ensemble semi-algébrique de base défini par la famille de polynômes* $(g_j)_{j=1\ldots m}$*. Si le module quadratique* $\mathbf{M}(g_1, \ldots, g_m)$ *est archimédien on a alors l'équivalence suivante :*

$$\text{il existe une mesure } \mu \in \mathcal{M}_+(K) \text{ telle que : } y_\alpha = \int_K x^\alpha d\mu \quad \forall \alpha \in \mathbb{N}^n \tag{3.35}$$

$$\Leftrightarrow \quad L_{\mathbf{y}}(p) \geqslant 0 \quad \forall p \in \mathbf{M}(g_1, \ldots, g_m) \tag{3.36}$$

$$\Leftrightarrow \quad L_{\mathbf{y}}(p^2 g_j) \geqslant 0 \quad \forall j \in \{0, \ldots, m\} \text{ avec } g_0 \equiv 1 \ \ \forall p \in \mathbb{R}[x]. \tag{3.37}$$

Notons que ce théorème peut être vu comme une spécialisation du théorème plus général de **Riesz-Haviland** lorsque K est un ensemble semi-algébrique de base et que le module quadratique associé est archimédien. En supposant simplement que K est fermé, celui-ci établit que $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}$ représente le vecteur des moments d'une mesure de $\mathcal{M}_+(K)$ si et seulement si $L_{\mathbf{y}}(p) \geqslant 0$ pour tout polynôme p non-négatif sur K. Cependant ce théorème se révèle peu utilisable en pratique, car contrairement aux ensembles semi-algébriques de base, il n'existe pas de caractérisation explicite des polynômes non-négatifs sur les ensembles fermés de $\mathbb{R}^n$.

A présent, nous disposons d'une dualité explicite entre la représentation par une mesure et les polynômes positifs sur K. Ce lien est réalisé au moyen d'une condition de positivité de la fonctionnelle de Riesz $L_{\mathbf{y}}$ sur le module quadratique $\mathbf{M}(g_1, \ldots, g_m)$. Nous allons voir maintenant que cette condition se traduit par une condition sur le vecteur $\mathbf{y}$. Pour cela, il est nécessaire d'introduire deux nouveaux outils : la **matrice des moments** et la **matrice de localisation** d'un vecteur $\mathbf{y}$ de $\mathbb{R}^{\mathbb{N}^n_{2t}}$.

Définition 22 *(**Matrice des moments**) Soit* $t \in \mathbb{N}$ *et* $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n_{2t}}$*. La* ***matrice des moments*** *de* $\mathbf{y}$*, notée* $M_t(\mathbf{y})$*, est définie par éléments à l'aide de la formule suivante :*

$$(M_t(\mathbf{y}))_{\alpha,\beta} = L_{\mathbf{y}}(x^\alpha x^\beta) = y_{\alpha+\beta} \quad \forall \alpha, \beta \in \mathbb{N}^n_t \tag{3.38}$$

Indicée selon $\mathbb{N}^n_t$*, cette matrice est symétrique, linéaire en* $\mathbf{y}$*, de taille* $\binom{n+t}{n}$ *et comporte* $\binom{n+2t}{n}$ *variables.*

Par exemple, dans le cas où $n = t = 2$, la matrice $M_2(\mathbf{y})$ s'écrit :

$$M_2(\mathbf{y}) \quad = \quad \left[\begin{array}{c|cc|ccc} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ \hline y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ \hline y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{array}\right]. \tag{3.39}$$

Définition 23 *(**Matrice de localisation**) Soit $y \in \mathbb{R}^{\mathbb{N}^n_{2t}}$, $g \in \mathbb{R}_t[x]$ et $\mathbf{g} \in \mathbb{R}^{\mathbb{N}^n_t}$ son vecteur de coefficients. Le vecteur de $\mathbb{R}^{\mathbb{N}^n_t}$ défini par :*

$$g\mathbf{y} \quad \triangleq \quad M(\mathbf{y})\mathbf{g} \tag{3.40}$$

*est appelé **le déplacé de $\mathbf{y}$ par** g. Ainsi, pour tout $\alpha \in \mathbb{N}^n_t$, chaque coefficient de $g\mathbf{y}$ est donné par la formule :*

$$(g\mathbf{y})_\alpha = \sum_{\beta \in \mathbb{N}^n_t} g_\beta y_{\alpha+\beta}. \tag{3.41}$$

*La **matrice de localisation** de $\mathbf{y}$ par rapport à g, noté $M_t(g\mathbf{y})$, est définie comme la matrice des moments du déplacé de $\mathbf{y}$ par g. Elle est donc définie par éléments à l'aide de la formule suivante :*

$$(M(g\mathbf{y}))_{\alpha,\beta} = L_{\mathbf{y}}(g(x)x^\alpha x^\beta) = \sum_{\gamma \in \mathbb{N}^n_t} g_\gamma y_{\alpha+\beta+\gamma} \quad \forall \alpha, \beta \in \mathbb{N}^n. \tag{3.42}$$

Indicée selon $\mathbb{N}^n_t$, cette matrice est symétrique, linéaire en $\mathbf{y}$, de taille $\binom{n+t}{n}$ et comporte $\binom{n+2t}{n}$ variables.

Cette matrice est introduite dans le but de permettre la « localisation » du support d'une mesure représentée par un vecteur $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}$. Par exemple, pour $n = 2$ avec :

$$M_1(\mathbf{y}) = \left[\begin{array}{ccc} y_{00} & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{array}\right] \text{ et } g(x) = a - x_1^2 - x_2^2, \tag{3.43}$$

on obtient :

$$M_1(g\mathbf{y}) = \left[\begin{array}{ccc} ay_{00} - y_{20} - y_{02} & ay_{10} - y_{30} - y_{12} & ay_{01} - y_{21} - y_{03} \\ ay_{10} - y_{30} - y_{12} & ay_{20} - y_{40} - y_{22} & ay_{11} - y_{31} - y_{13} \\ ay_{01} - y_{21} - y_{03} & ay_{11} - y_{31} - y_{13} & ay_{02} - y_{22} - y_{04} \end{array}\right]. \tag{3.44}$$

Ensuite, par construction, on observe les égalités suivantes :

$$\begin{aligned} L_{\mathbf{y}}(pq) &= \mathbf{p}^\top M_t(\mathbf{y})\mathbf{q} \quad \forall p, q \in \mathbb{R}_t[x] \quad \forall t \in \mathbb{N} \\ L_{\mathbf{y}}(g\,pq) &= \mathbf{p}^\top M_t(g\mathbf{y})\mathbf{q} \quad \forall p, q \in \mathbb{R}_t[x] \quad \forall t \in \mathbb{N}, \end{aligned} \tag{3.45}$$

avec $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n_{2t}}$. En particulier :

$$\begin{aligned} L_{\mathbf{y}}(p^2) &= \mathbf{p}^\top M_t(\mathbf{y})\mathbf{p} \quad \forall p \in \mathbb{R}_t[x] \quad \forall t \in \mathbb{N} \\ L_{\mathbf{y}}(g\,p^2) &= \mathbf{p}^\top M_t(g\mathbf{y})\mathbf{p} \quad \forall p \in \mathbb{R}_t[x] \quad \forall t \in \mathbb{N}, \end{aligned} \tag{3.46}$$

On obtient donc les équivalences suivantes :

$$\begin{aligned} M_t(\mathbf{y}) \succcurlyeq 0 \quad &\Leftrightarrow \quad L_{\mathbf{y}}(p^2) \geqslant 0 \quad \forall p \in \mathbb{R}_t[x] \quad \forall t \in \mathbb{N} \\ M_t(g\mathbf{y}) \succcurlyeq 0 \quad &\Leftrightarrow \quad L_{\mathbf{y}}(gp^2) \geqslant 0 \quad \forall p \in \mathbb{R}_t[x] \quad \forall t \in \mathbb{N}. \end{aligned} \tag{3.47}$$

Ainsi, $\mathbf{y}$ est représenté par une mesure $\mu \in \mathcal{M}_+(K)$ dès lors que toutes les matrices $M_t(\mathbf{y})$ et $M_t(g\mathbf{y})$ sont semi-définies positives. Comme annoncé, il est à présent possible de remplacer les conditions de positivité de la fonctionnelle de Riesz $L_{\mathbf{y}}$ dans le Théorème 4 par des conditions basées sur les matrices des moments et les matrices de localisation. En effet, grâce aux relations (3.45) et (3.46), on peut montrer (cf. [Laurent 2009]) l'équivalence suivante :

$$\begin{aligned} & L_{\mathbf{y}}(p) \geqslant 0 \ \ \forall p \in M(g_1, \ldots, g_m) \\ \Leftrightarrow \quad & L_{\mathbf{y}}(p^2 g_j) \geqslant 0 \quad \forall j \in \{0, 1, \ldots, m\} \quad \forall p \in \mathbb{R}[x] \ \text{ avec } g_0 \equiv 1 \\ \Leftrightarrow \quad & M_t(\mathbf{y}) \succcurlyeq 0, \ M_t(g_j\mathbf{y}) \succcurlyeq 0 \ \ \forall j \in \{0, 1, \ldots, m\} \quad \forall t \in \mathbb{N} \end{aligned} \tag{3.48}$$

Notons que, dans cette équivalence, nous avons par abus de notation utilisé la même terminologie pour le vecteur infini-dimensionnel et ses troncatures. En effet, dans la première partie de l'équivalence, $\mathbf{y}$ désigne un vecteur de $\mathbb{R}^{\mathbb{N}^n}$ et dans la deuxième partie ses troncatures dans $\mathbb{R}^{\mathbb{N}^n_{2t}}$.

Pour faire un bilan de ce paragraphe, nous pouvons synthétiser les résultats présentés en les reformulant en terme d'ensembles. Pour cela, définissons :

$$\mathcal{M}_K \ \triangleq \ \left\{ \mathbf{y} \in \mathbb{R}^{\mathbb{N}^n} \ \mid \ \exists \mu \in \mathcal{M}_+(K) \text{ telle que : } y_\alpha = \int_K x^\alpha d\mu \ \ \forall \alpha \in \mathbb{N}^n \right\}. \tag{3.49}$$

et :

$$\mathcal{M}^{put}_{\geq}(g_1, \ldots, g_m) \ \triangleq \ \left\{ \mathbf{y} \in \mathbb{R}^{\mathbb{N}^n} \mid M_t(\mathbf{y}) \succcurlyeq 0, \ M_t(g_j\mathbf{y}) \succcurlyeq 0 \ \ \forall j = 1, .., m \ \ \forall t \in \mathbb{N} \right\}. \tag{3.50}$$

Les définitions précédentes nous permettent d'observer que, par construction, on a :

$$\mathcal{M}_K \ \subseteq \ \mathcal{M}^{put}_{\geq}(g_1, \ldots, g_m). \tag{3.51}$$

Mais, le Théorème 4 permet d'affirmer que, si le module quadratique $\mathrm{M}(g_1, \ldots, g_m)$ est archimédien, alors :

$$\mathcal{M}_K \ = \ \mathcal{M}^{put}_{\geq}(g_1, \ldots, g_m). \tag{3.52}$$

Ce résultat, très puissant d'un point de vue théorique, est à la base de l'optimisation par théorie des moments présentée dans le paragraphe suivant.

3.1.3 Optimisation globale polynomiale par théorie des moments.

Comme nous l'avons précisé dans l'introduction, la résolution du problème (3.5) s'avère trop complexe lorsque celle-ci est traitée dans toute sa généralité. Considérons alors ce problème dans le cas particulier où la fonction coût est un polynôme $p \in \mathbb{R}[x]$ et K un ensemble semi-algébrique de base défini à partir d'une famille de m polynômes $g_1, \ldots, g_m \in \mathbb{R}[x]$:

$$p^\star \ \triangleq \ \min_{x \in K} p(x) \tag{3.53}$$

Enfin, on suppose dans la suite de cette section que le module quadratique $\mathbf{M}(g_1, \ldots, g_m)$ engendré par les polynômes $g_1, \ldots, g_m$ est archimédien.

3.1.3.1 Une famille de relaxations.

Ainsi que nous l'avons vu précédemment, le problème (3.53) est équivalent au problème suivant :

$$\widehat{p} \triangleq \min_{\mu \in \mathcal{P}(K)} \int_K p(x) d\mu (= p^\star). \tag{3.54}$$

Remarquons que, comme le suggère la preuve de cette équivalence dans le cas général, si $\mu^\star$ est solution de (3.54) et que (3.53) possède un nombre fini, L, de solutions, alors $\mu^\star$ est une mesure atomique s'exprimant comme une combinaison linéaire des L solutions $\{x_1, \ldots, x_L\}$ de (3.53) :

$$\mu^\star = \sum_{i=1}^{L} \lambda_i \delta_{x_i}. \tag{3.55}$$

Ensuite, à l'aide des reformulations vues précédemment, on observe que (3.53) est équivalent au problème :

$$\begin{array}{rcll} \widehat{p} & = & \min & L_{\mathbf{y}}(p) \\ & & \text{s.l.c} & y_0 = 1 \\ & & & \mathbf{y} \in \mathcal{M}_K. \end{array} \tag{3.56}$$

Remarquons que la condition $y_0 = 1$ est nécessaire pour imposer que μ soit une mesure de probabilité sur K. Or, puisque nous avons supposé que le module quadratique $\mathbf{M}(g_1, \ldots, g_m)$ était archimédien, par (3.52), on obtient que (3.56) est équivalent au problème :

$$\begin{array}{rll} \widehat{p} = & \inf & L_{\mathbf{y}}(p) \\ & \text{s.l.c} & y_0 = 1 \\ & & M_t(\mathbf{y}) \succeq 0,\ M_t(g_j \mathbf{y}) \succeq 0 \ \ j = 1, \ldots, m \ \ \forall t \in \mathbb{N}. \end{array} \tag{3.57}$$

Nous avons donc transformé le problème (3.53) en un problème SDP où l'inconnue est un vecteur $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}$. Ce problème est donc de dimension infinie. Le principe fondamental de l'optimisation par théorie des moments consiste alors à tronquer ce vecteur, et par conséquent les matrices des moments et les matrices de localisations associées, de manière à fabriquer une hiérarchie de problèmes SDP de dimensions finies. Ainsi, pour $t \geqslant t_0 = \max(\mathrm{d}_p, \mathrm{d}_K)$, on définit la famille de relaxations semi-définies positives :

$$\mathcal{Q}_t \left\{ \begin{array}{rll} p_t^{mom} \triangleq & \min\limits_{\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n_{2t}}} & L_{\mathbf{y}}(p) \\ & \text{s.l.c} & y_0 = 1 \\ & & M_t(\mathbf{y}) \succeq 0,\ M_{t-\mathrm{d}_{g_j}}(g_j \mathbf{y}) \succeq 0 \ \ \forall j \in \{1, \ldots, m\}. \end{array} \right. \tag{3.58}$$

Notons que ces relaxations utilisent des matrices de tailles $\binom{n+t}{n}$ comportant chacune $\binom{n+2t}{n}$ variables. La complexité de résolution de ces relaxations SDP augmente donc rapidement avec l'ordre de relaxation t. En outre, le nombre de contraintes augmentant avec t, on observe :

$$p_t^{mom} \leqslant p_{t+1}^{mom} \leqslant \widehat{p}. \tag{3.59}$$

Ensuite, chaque problème (3.58) a un dual dont on peut montrer (cf. [Laurent 2009]) qu'il s'écrit comme le problème SDP suivant :

$$\mathcal{Q}_t^\star \left\{ \begin{array}{rll} p_t^{sos} \triangleq & \sup\limits_{\lambda,\, u_j} \lambda & \\ & \text{s.l.c} \quad p - \lambda = u_0 + \sum\limits_{j=1}^{m} u_j g_j & \text{avec } u_0, \ldots, u_m \in \Sigma \\ & & \text{et } \deg(u_0), \deg(u_j g_j) \leqslant 2t. \end{array} \right. \tag{3.60}$$

On observe alors l'inégalité suivante :

$$p_t^{sos} \leqslant p_t^{mom}. \tag{3.61}$$

L'autre point clé de l'optimisation par théorie des moments réside dans l'étude des suites de solutions $(p_t^{mom})_{t\geqslant t_0}$ et $(p_t^{sos})_{t\geqslant t_0}$. Les résultats qu'elle établie sont donnés (cf [Lasserre 2001]) par le théorème suivant :

Théorème 5 *(**Convergence des solutions**) Sous les hypothèses et définitions précédentes, les suites de solutions $(p_t^{mom})_{t\geqslant t_0}$ et $(p_t^{sos})_{t\geqslant t_0}$ convergent asymptotiquement vers $\widehat{p}$:*

$$\lim_{t\to+\infty} p_t^{sos} = \lim_{t\to+\infty} p_t^{mom} = \widehat{p}. \tag{3.62}$$

De plus, cette convergence peut être finie. En effet, si $\mathbf{y}^\star \in \mathbb{R}^{\mathbb{N}^n_{2t}}$ est une solution de (3.58) et si :

$$\exists s \in \mathbb{N} \;\mid\; \max(\mathrm{d}_p, \mathrm{d}_K) \leqslant s \leqslant t \;\; et \;\; \mathrm{rang}(M_s(\mathbf{y}^\star)) = \mathrm{rang}(M_{s-\mathrm{d}_K}(\mathbf{y}^\star)) = L \tag{3.63}$$

alors $p_t^{mom} = \widehat{p}$ et il y a L minima globaux. En particulier, si $\mathrm{rang}(M_t(\mathbf{y}^\star)) = 1$ alors la condition est vérifiée.

Ainsi, si est (3.63) vérifiée, la solution d'un problème de minimisation d'un polynôme sous contraintes polynomiales est atteinte pour une solution d'une relaxation SDP particulière. Notons que la solution de cette relaxation représente une mesure de Dirac, ou une combinaison linéaire de mesures de Dirac s'il y a plusieurs minima globaux. Chaque relaxation peut être résolue numériquement à l'aide de méthodes de points intérieurs [Vandenberghe 1996].

3.1.3.2 Algorithme général.

L'algorithme 17 synthétise la méthode d'optimisation globale par théorie de moments. Il est implanté dans le logiciel GloptiPoly [Henrion 2002]. Cependant, la précision finale de la résolution de (3.54) dépendra fortement de la précision avec laquelle on pourra extraire numériquement les solutions. Nous n'avons pas détaillé ici les algorithmes d'extraction des solutions, pour plus de détails le lecteur pourra se référer à [Lasserre 2010, Algorithme 4.2].

Algorithme 17 Optimisation globale par théorie des moments.

Précondition : $p, g_1, \ldots, g_m \in \mathbb{R}[x]$ et $k > \max(\mathrm{d}_p, \mathrm{d}_K)$ l'index représentant l'ordre de la plus grande relaxation.

$i \leftarrow \max(\mathrm{d}_p, \mathrm{d}_K)$
test d'arrêt $\leftarrow$ FAUX
tant que $(i \leqslant k)$ ou (test d'arrêt=FAUX) **faire**
  Résolution de (R_i), la i$^{\text{ème}}$ relaxation SDP (3.58), à l'aide d'une méthode de points intérieurs.
  si (R_i) possède une solution optimale $\mathrm{y}_i^\star$ **alors**
    test de la condition : $[\ \exists s \in \mathbb{N} \ \mid \ i \leqslant s \leqslant k$ et $\mathrm{rang}(M_s(\mathrm{y}_i^\star)) = \mathrm{rang}(M_{s-\mathrm{d}_K}(\mathrm{y}_i^\star))\ ]$
    si cette condition est vérifiée **alors**
      test d'arrêt $\leftarrow$ VRAI
    sinon
      $i \leftarrow i + 1$
  sinon
    $i \leftarrow i + 1$
si test d'arrêt=FAUX **alors**
  retour de $p_i^{mom} = L_{\mathbf{y}_i^\star}(p)$ qui est une borne inférieure de $\widehat{p}$.
sinon
  retour de $p_i^{mom} = \widehat{p}$ et, si l'extraction est réussie, un ensemble de $l = \mathrm{rang}(M_s(\mathbf{y}^\star))$ minimiseurs globaux.

3.1.4 Faible couplage des variables.

Comme nous l'avons déjà mentionné, la taille des matrices $M_t(\mathbf{y})$ et $M_t(g\mathbf{y})$ utilisées dans les familles de relaxations SDP (3.58) augmente rapidement avec n, le nombre de variables d'optimisations considérées. Ceci fait que le champ applicatif de l'algorithme précédent, dit **dense**, se limite aux problèmes de moyenne ou petite taille. Cependant, pour une certaine classe de problèmes de grande taille, il est possible de prendre en compte **le faible couplage** ou **parcimonie** des variables afin d'utiliser plus efficacement l'Algorithme 17. Le but de ce paragraphe est de présenter cette classe de problèmes ainsi que la famille de relaxations SDP qui lui est associée. Nous introduirons dans un premier temps les notations nécessaires, puis la notion de **schéma de parcimonie structurée** des variables et enfin la famille de relaxations proprement dites ainsi que ses propriétés de convergence.

Introduisons tout d'abord l'ensemble des notations nécessaires. Ces notations portent essentiellement sur les ensembles d'indices de variables et visent à définir de façon mathématique les concepts de **faible couplage** ou **parcimonie** de variables. Par **faible couplage** ou **parcimonie** entre deux ensembles de variables d'un polynôme p, nous entendons qu'il n'y a pas de monômes de p impliquant simultanément certaines variables des deux ensembles. Par exemple, si on considère le polynôme :

$$p(x_1, x_2, x_3, x_4, x_5) = x_1x_2x_4 + x_3x_4x_5, \tag{3.64}$$

de $\mathbb{R}[x_1, x_2, x_3, x_4, x_5]$, alors il existe un **faible couplage** ou une **parcimonie** entre les ensembles de variables $\{x_1, x_2, x_4\}$ et $\{x_3, x_4, x_5\}$. En effet, aucun des deux monômes de p ne fait intervenir simultanément les variables x_1, x_2 et x_3, x_5. Afin de présenter plus précisément des définitions de parcimonie particulières, introduisons de nouvelles notations. Par la notation $\mathbb{R}[x] = \mathbb{R}[x_1, x_2, \ldots, x_n]$ nous avons précédemment défini l'anneau des polynômes réels sur les variables $x_1, \ldots, x_n$. Soit $I_0 \triangleq \{1, 2, \ldots, n\}$ l'ensemble regroupant la totalité des indices des variables et $I \subseteq I_0$, on définit par $x(I)$ l'ensemble de variables indicées par I :

$$x(I) \triangleq \{x_i,\ i \in I\}. \tag{3.65}$$

Alors, l'ensemble $\mathbb{R}[x(I)]$ désigne l'anneau des polynômes sur les variables $x(I)$. Remarquons qu'avec ces notations $\mathbb{R}[x] = \mathbb{R}[x(I_0)]$. Ensuite pour $\alpha \in \mathbb{N}^n_t$, à l'instar d'une mesure, on définit le support de α par :

$$\mathrm{supp}(\alpha) \triangleq \{i \in I_0 \mid \alpha_i \geqslant 1\}. \tag{3.66}$$

Par exemple, pour $n = 6$ et $\alpha = (010509)$, $\mathrm{supp}(\alpha) = \{2, 4, 6\}$. Ensuite, pour $I \subseteq I_0$, on pose :

$$\Lambda^I \triangleq \{\alpha \in \mathbb{N}^n \mid \mathrm{supp}(\alpha) \subseteq I\} \tag{3.67}$$

et

$$\Lambda^I_t \triangleq \{\alpha \in \mathbb{N}^n_t \mid \mathrm{supp}(\alpha) \subseteq I\}, \tag{3.68}$$

dont la dimension est $\binom{n_I+t}{n_I}$, avec $n_I \triangleq \mathrm{Card}(I)$. Ainsi, un polynôme de $p \in \mathbb{R}_t[x(I)]$ peut être vu comme un polynôme de $\mathbb{R}_t[x]$ en écrivant :

$$p(x) = p(x(I)) = \sum_{\alpha \in \Lambda^I_t} p_\alpha x^\alpha. \tag{3.69}$$

Ensuite, comme pour les polynômes, on peut définir les matrices des moments et les matrices de localisation pour un sous-ensemble d'indices. Ainsi, pour $I \subseteq I_0$, ces matrices seront définies

pour des vecteurs $\mathbf{y} \in \mathbb{R}^{\Lambda^I_{2t}}$. Plus précisément, **la matrice des moments par rapport à l'ensemble d'indices** I, notée $M_t(\mathbf{y}, I)$, est construite en conservant les lignes et les colonnes de $M_t(\mathbf{y})$ dont l'indice α est tel que $\mathrm{supp}(\alpha) \subseteq I$. De même, pour $g \in \mathbb{R}_t[x(I)]$, **la matrice de localisation par rapport à l'ensemble d'indices** I, notée $M_t(g\mathbf{y}, I)$, est construite en conservant les lignes et les colonnes de $M_t(g\mathbf{y})$ dont l'indice α est tel que $\mathrm{supp}(\alpha) \subseteq I$. Ces deux matrices sont de taille $\binom{n_I+t}{n_I} \times \binom{n_I+t}{n_I}$. On observe alors des égalités similaires aux égalités (3.45) et (3.46), conduisant aux équivalences :

$$M_t(\mathbf{y}, I) \succeq 0 \quad \Leftrightarrow \quad L_{\mathbf{y}}(p^2) \geqslant 0 \quad \forall p \in \mathbb{R}_t[x(I)] \quad \forall t \in \mathbb{N} \tag{3.70}$$

$$M_t(g\,\mathbf{y}, I) \succeq 0 \quad \Leftrightarrow \quad L_{\mathbf{y}}(p^2) \geqslant 0 \quad \forall p \in \mathbb{R}_t[x(I)] \quad \forall t \in \mathbb{N}, \tag{3.71}$$

avec $\mathbf{y} \in \mathbb{R}^{\Lambda^I_{2t}}$.

Ensuite, soit $(I_k)_{k=1,\ldots,r}$ une famille de sous-ensembles d'indices recouvrant I_0, c'est-à-dire tels que $I_0 = \bigcup_{k=1}^r I_k$, et $(J_k)_{k=1,\ldots,r}$ une famille de sous-ensembles d'indices recouvrant $J_0 \triangleq \{0, 1, \ldots, m\}$, l'ensemble des indices des contraintes (g_0 étant la contrainte additionnelle assurant que le module quadratique est archimédien), vérifiant les propriétes suivantes :

(i) Pour tout $j \in J_k$, g_j est un polynôme ne comportant que des variables indicées par I_k, c'est-à-dire :

$$\forall k \in \{1, \ldots, r\}, \ \forall j \in J_k, \ \ g_j \in \mathbb{R}[x(I_k)]. \tag{3.72}$$

(ii) Le polynôme objectif $p \in \mathbb{R}[x]$ se décompose sous la forme :

$$p = \sum_{k=1}^{r} p_k \ \text{ avec } \ p_k \in \mathbb{R}[x(I_k)], \ \forall k = 1, \ldots, r. \tag{3.73}$$

(iii) Pour tout k dans $\{1, \ldots, r-1\}$, il existe $s \in \mathbb{N}^*$ tel que :

$$s \leqslant k \quad \text{ et } \quad \left(I_{k+1} \cap \bigcup_{j=1}^{k} I_j \right) \subseteq I_s. \tag{3.74}$$

Cette dernière propriété est appelée la **Propriété d'Intersection Courante**. Si $(I_k)_{k=1,\ldots,r}$ vérifie *(i)* et *(ii)* alors $(I_k)_{k=1,\ldots,r}$ est appelé **un schéma de parcimonie structurée** de x. Il est important de souligner que la famille $(I_k)_k$ peut indifféremment représenter des variables ou des indices de variables. Dans le cas de faible couplage des variables, il est possible d'établir [Laurent 2009] un corollaire du théorème 3 :

Théorème 6 ***(Positivstellensatz de Putinar avec parcimonie)***
Soit $(I_k)_{k=1,\ldots,r}$ un schéma de parcimonie structurée vérifiant la propriété d'intersection courante (3.74). Si les modules quadratiques

$$\mathrm{M}_k \triangleq \mathrm{M}(\{g_j \ \mid \ j \in J_k\}) \subseteq \mathbb{R}[x(I_k)]$$

engendrés par les familles de polynômes $(g_j)_{j \in J_k}$ sont archimédiens et si p est strictement positif sur K, alors il existe une famille $(u_k, u_{j_k})_k$ de polynômes SOS telle que :

$$p \ = \ \sum_{k=1}^{r} \left(u_k + \sum_{j \in J_k} u_{j_k} g_j \right), \quad \textit{avec } u_k, u_{j_k} \in \Sigma[x(I_k)]. \tag{3.75}$$

Autrement dit, si p est strictement positif sur K, alors :

$$p \in \mathrm{M}_1 + \ldots + \mathrm{M}_r. \tag{3.76}$$

Par conséquent, nous pouvons appliquer un raisonnement analogue à celui de la section précédente : nous transformons le problème d'optimisation initial en un problème de minimisation de la fonctionnelle de Riesz sous contraintes que le vecteur recherché est représenté par une mesure prenant en compte la parcimonie des variables.

Dans un premier temps, définissons le problème sur l'espace des mesures. Pour cela, pour chaque I_k, on définit $n_k \triangleq \text{Card}(I_k)$. Notons que l'intersection des I_k entre eux peut être non-vide, c'est-à-dire qu'il peut y avoir des recouvrements entre les I_k. Ainsi, on adopte la notation suivante :

$$I_{jk} \triangleq I_j \bigcap I_k, \qquad \forall j,k \in \{1,\dots,r\}, \tag{3.77}$$

et, pour $I_{jk} \neq \varnothing$, on définit $n_{jk} \triangleq \text{Card}(I_{jk})$. Enfin, on pose :

$$\Gamma_j \triangleq \left\{ h \in \mathbb{N} \mid j < h \leqslant r \text{ et } I_j \bigcap I_h \neq \varnothing \right\}. \tag{3.78}$$

A l'aide de la famille d'indices $(J_k)_{l=1,\dots,r}$, on définit :

$$K_k \triangleq \{x \in \mathbb{R}^{n_k} \mid g_j(x) \geqslant 0 \ \ \forall j \in J_k\}, \tag{3.79}$$

de sorte que :

$$K = \left\{ x \in \mathbb{R}^n \mid g_j(x) \geqslant 0 \ \ \forall j \in \bigcup_{k=1}^{r} J_k \right\} = \{x \in \mathbb{R}^n \mid x(I_k) \in K_k, \ \ k = 1,\dots,r\}. \tag{3.80}$$

Enfin, pour toute mesure $\mu \in \mathcal{M}_+(K)$, soit $\pi_k\mu \in \mathcal{B}(K_k)$, la projection de μ sur les variables $x(I_k)$ définie par :

$$\pi_k\mu(A) \triangleq \mu(\{x \in K \mid (x_h)_{h\in I_k} \in A\}), \quad \forall A \in \mathcal{T}_B(K_k). \tag{3.81}$$

Considérons alors le problème suivant :

$$\begin{aligned} \hat{p} \triangleq \ & \min_{\mu_k \in \mathcal{B}(K_k)} \sum_{k=1}^{r} \int_{K_k} p_k \, d\mu_k \\ & \text{s.l.c} \ \ \mu_k(K_k) = 1, \ \ k = 1,\dots,r \\ & \qquad \pi_{jk}\mu_j = \pi_{kj}\mu_k \ \ \forall k \in \Gamma_j, \ \forall j \in \{1\dots,r\} \end{aligned} \tag{3.82}$$

Par rapport au problème original (3.54), on ne cherche pas une seule mesure définie pour n variables, mais r mesures dont chacune est définie pour $n_k = \text{Card}(I_k)$ variables. Les contraintes :

$$\pi_{jk}\mu_j = \pi_{kj}\mu_k \qquad \forall k \in \Gamma_j, \ \forall j \in \{1\dots,r\} \tag{3.83}$$

très complexes de prime abord, signifient simplement que ces mesures sont égales où les ensembles I_k se chevauchent. On peut d'ailleurs les réécrire à l'aide des moments :

$$\int_{K_j} x(I_{jk})^\alpha \, d\mu_j = \int_{K_k} x(I_{kj})^\alpha \, d\mu_k \qquad \forall \alpha \in \mathbb{N}^{n_{jk}}. \tag{3.84}$$

Alors, sous quelles conditions sur les $(I_k)_{k=1,\dots,r}$ les problèmes (3.82) et (3.54) sont-ils équivalents ? Par le théorème [Lasserre 2010, théorème 4.6], si $(I_k)_{k=1,\dots,r}$ est un schéma de parcimonie structurée vérifiant la propriété d'intersection courante (3.74), alors les mesures $\mu_k \in \mathcal{P}(K_k)$ sont les projections d'une mesure de $\mathcal{P}(K)$ sur les variables $x(I_k)$, autrement dit les problèmes (3.82) et (3.54) sont équivalents.

Ensuite, par analogie à la section précédente, l'objectif consiste à reformuler (3.82) à l'aide des matrices des moments $M_t(\mathbf{y}, I)$ et des matrices localisation $M_t(g\mathbf{y}, I)$. Appliquons donc

un raisonnement similaire : un vecteur $\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}$ représente une mesure borélienne vérifiant les containtes du problème (3.82) si et seulement si $L_{\mathbf{y}}$ est positive sur chacun des M_k. Cette dernière condition est equivalente à la condition :

$$M_t(\mathbf{y}, I_k) \succeq 0 \quad k = 1, \ldots, r \quad \forall t \in \mathbb{N} \tag{3.85}$$

$$M_t(g_j \mathbf{y}, I_k) \succeq 0,\ \forall j \in J_k,\ k = 1, \ldots, r \quad \forall t \in \mathbb{N}. \tag{3.86}$$

où $\mathbf{y}$ appartient à $\mathbb{R}^{\mathbb{N}^n_{2t}}$. Bien que $\mathbf{y}$ soit le vecteur total de $\mathbb{R}^{\mathbb{N}^n_{2t}}$, chaque matrice ne porte que sur les variables I_k, autrement dit $\mathbf{y}$ est vu comme un vecteur de $\mathbb{R}^{\Lambda_{2t}^{I_k}}$. Comparé aux matrices de tailles $\binom{n+t}{n}$ du problème dense, nous obtenons donc des matrices de taille $\binom{n_k+t}{n_k}$. Par conséquent, on a la reformulation de (3.82) suivante :

$$\begin{aligned} \widehat{p} \ \triangleq \ & \min_{\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n}} \sum_{k=1}^{r} L_{\mathbf{y}}(p_k) \\ & \text{s.l.c} \quad y_0 = 1 \\ & \quad M_t(\mathbf{y}, I_k) \succeq 0, \quad k = 1, \ldots, r \quad \forall t \in \mathbb{N} \\ & \quad M_{t - \mathrm{d}_{g_j}}(g_j \mathbf{y}, I_k) \succeq 0,\ \forall j \in J_k,\ k = 1, \ldots, r \quad \forall t \in \mathbb{N}. \end{aligned} \tag{3.87}$$

Ainsi, pour $t \geqslant t_0 = \max(\mathrm{d}_p, \mathrm{d}_{K_1}, \ldots, \mathrm{d}_{K_r})$, on définit les relaxations SDP de (3.87) par :

$$\mathcal{Q}_t \begin{cases} p_t^{mom} \ \triangleq \ \min\limits_{\mathbf{y} \in \mathbb{R}^{\mathbb{N}^n_{2t}}} \sum\limits_{k=1}^{r} L_{\mathbf{y}}(p_k) \\ \quad \text{s.l.c} \quad y_0 = 1 \\ \qquad M_t(\mathbf{y}, I_k) \succeq 0, \quad k = 1, \ldots, r \\ \qquad M_{t - \mathrm{d}_{g_j}}(g_j \mathbf{y}, I_k) \succeq 0,\ \forall j \in J_k,\ k = 1, \ldots, r. \end{cases} \tag{3.88}$$

Ce problème a un dual dont on peut montrer qu'il s'écrit comme le problème SDP suivant :

$$\mathcal{Q}_t^{\star} \begin{cases} p_t^{sos} \ \triangleq \ \sup\limits_{\lambda,\, u_{kj}} \lambda \\ \quad \text{s.l.c} \quad p - \lambda = \sum_{k=1}^{r} \left(u_{k0} + \sum\limits_{j \in J_k} u_{kj} g_j \right) \\ \qquad \text{avec } u_{k0}, u_{kj} \in \Sigma\left[x(I_k)\right], \quad k = 1, \ldots, r \\ \qquad \text{et } \deg(u_{k0}), \deg(u_{kj} g_j) \leqslant 2t,\ k = 1, \ldots, r. \end{cases} \tag{3.89}$$

De la même manière que les solutions des relaxations SDP denses, on observe (cf. [Lasserre 2010, théorème 5.9]) le résultat de convergence suivant :

Théorème 7 *Sous les hypothèses et définitions précédentes, les suites de solutions $(p_t^{mom})_{t \geqslant t_0}$ et $(p_t^{sos})_{t \geqslant t_0}$ convergent asymptotiquement vers p^{mom} :*

$$\lim_{t \to +\infty} p_t^{sos} = \lim_{t \to +\infty} p_t^{mom} = \widehat{p} = p^{\star}. \tag{3.90}$$

De plus, si $\mathbf{y}^{\star} \in \mathbb{R}^{\Lambda_{2t}^{I_0}}$ est une solution de (3.88) et s'il existe $(s_k)_{k=1\ldots r} \in \mathbb{N}^r$ tels que :

$$\begin{aligned} & t_0 \leqslant s_k \leqslant t \\ & \mathrm{rang}(M_{s_k}(\mathbf{y}_k^{\star}), I_k) = \mathrm{rang}(M_{s_k - \mathrm{d}_{K_k}}(\mathbf{y}_k^{\star}, I_k)), \quad \forall k = 1, \ldots r \\ & \mathrm{rang}(M_{s_k}(\mathbf{y}_k^{\star}), I_k \bigcap I_j) = 1 \quad \forall j \in \Gamma_k \quad \forall k = 1, \ldots r. \end{aligned} \tag{3.91}$$

alors $p_t^{mom} = p^{mom}$ et il y a $l_k = \mathrm{rang}(M_{s_k}(\mathbf{y}_k^{\star}, I_k))$ minima globaux dans K_k.

Ainsi, on obtient une famille de problèmes SDP dont les contraintes portent sur des matrices de taille $\binom{n_k+t}{n_k}$ comportant chacune $\binom{n_k+2t}{n_k}$ variables. Il est dès lors possible, si les n_k restent raisonnables, de pouvoir résoudre des problèmes de grande taille. Afin d'apprécier le gain réalisé, dans [Kojima 2010, Mituhiro , Kim 2009], des problèmes à 1000 variables sont résolus à l'aide d'une formulation prenant en compte la parcimonie alors que la formulation dense ne peut traiter la première relaxation.

3.1.5 Bilan.

Dans cette section, nous avons montré comment le problème de minimisation polynomiale :

$$p^\star \triangleq \min_{x \in K} p(x) \tag{3.92}$$

pouvait être approché par une hiérarchie de problèmes SDP dont la suite de solutions converge vers $p^\star$. Cependant, pour un nombre d'inconnues trop important, la taille des matrices utilisées dans cette hiérarchie se révèle trop grande pour une utilisation pratique. Mais, nous avons introduit comment, dans certain cas, ces matrices peuvent être fractionnées pour réduire très significativement les temps de calcul.

3.2 Extension au cas d'une somme de fractions rationnelles

Le but de cette section est de présenter notre contribution. Elle consiste à étendre l'optimisation polynomiale de manière à pourvoir traiter des fonctions coût définies comme des sommes de fractions rationnelles. De manière générique, nous appelerons ce problème : **l'optimisation rationnelle**. Il est formulé de la manière suivante :

$$r^\star \triangleq \min_{x\in K} r(x) \triangleq \sum_{i=1}^{N} r_i(x) \tag{3.93}$$

où chaque terme $r_i : \mathbb{R}^n \to \mathbb{R}$ est une fonction rationnelle définie par :

$$r_i(x) \triangleq \frac{p_i(x)}{q_i(x)}, \tag{3.94}$$

avec $p_i, q_i \in \mathbb{R}[x]$ et $q_i > 0$ sur K pour tout $i = 1, \ldots, N$. Comme dans la section précedente, K est l'ensemble semi-algébrique défini par :

$$K \triangleq \{x \in \mathbb{R}^n \,:\, g_j(x) \geqslant 0,\, j = 1, \ldots, m\}, \tag{3.95}$$

où $g_j \in \mathbb{R}[x]$ pour tout $j = 1, \ldots, m$. Quitte à ajouter une contrainte supplémentaire du type $g_{m+1} = N - \|x\|_2^2$, nous supposons dans toute cette section que le module quadratique $\mathbf{M}(g_1, \ldots, g_m)$ est archimédien et par conséquent que K est compact. Ainsi, puisque r est continue sur K compact, (3.93) possède au moins une solution $x^\star$. En outre, nous supposons que les degrés de chaque g_j, p_i et q_i sont relativement faibles (de l'ordre de 10) mais que le nombre de termes N peut être très grand (de 10 à 100 environ). Dans le cas d'un problème dense, le nombre de variables n est supposé assez faible (jusqu'à 10). Toutefois, sous les hypothèses (3.74), nous verrons que ce nombre peut être largement augmenté (de 10 à 100). Rappelons enfin que, même si le problème (3.93) possède un intérêt théorique, notre motivation première reste certaines applications en vision par ordinateur, où de tels problèmes sont fréquement rencontrés. Ces problèmes sont souvent des problèmes d'estimation dans lesquels N est le nombre de mesures physiques utilisées. Dès lors, on comprend que N doit être grand (typiquement entre 50 et 100) pour que l'estimation soit précise. Nous décrirons ces applications plus formellement dans le chapitre 4.

En toute généralité, c'est-à dire sans faire d'hypothèses sur les polynômes p_i et q_i, le problème d'optimisation rationnelle (3.93) est assez difficile. En effet, même avec un nombre relativement faible de fractions et en supposant que les p_i (resp. q_i) sont convexes (resp. concaves), l'étude réalisée dans [Schaible 2003] montre que (3.93) reste un problème difficile car la somme de fractions à numérateurs convexes et dénominateurs concaves n'est généralement pas quasiconvexe (un minimum local n'est donc généralement pas global). Théoriquement, une première approche consiste à réduire toutes les fractions $\frac{p_i}{q_i}$ au même dénominateur afin obtenir une seule fraction rationnelle à minimiser. Une fois cette fraction calculée, on peut utiliser une hiérarchie de relaxations SDP comme dans [de Klerk 2006] ou comme conséquence de [Lasserre 2010, prop 5.20]. Mais une telle stratégie est difficilement utilisable en pratique. En effet, même si n est petit, le degré du dénominateur commun peut s'avérer important, de sorte qu'il soit impossible de résoudre la première relaxation de la hiérarchie, principalement en raison des limitations des solveurs de points intérieurs. De plus, cette stratégie conduit généralement à la destruction de la parcimonie potentiellement présente dans la formulation d'origine (3.93), interdisant ainsi toute réduction de la complexité de la hiérarchie SDP.

Dans cette section, nous montrerons qu'il existe deux solutions théoriques à ce problème : les **Epigraphes Creux** et la **Programmation Rationnelle**. Puis nous montrerons que la condition de parcimonie vue dans la section précédente peut être appliquée aux deux approches, permettant ainsi de traiter un grand nombre de variables.

3.2.1 Les épigraphes creux.

Cette première solution peut être vue comme une application directe des resultats du paragraphe (3.1.4). La première étape de cette application consiste à reformuler le problème initial à l'aide de variables additionnelles γ_i, appelées variables de **relèvement** (**lifting** an anglais). On réécrit alors (3.93) sous une forme polynomiale de la manière suivante :

$$\begin{array}{rcll} r^\star_{EC} & \triangleq & \min\limits_{(x,\gamma)\in K\times\mathbb{R}^N} & \gamma_1+\ldots+\gamma_N \\ & & \text{s.l.c} & p_i(x)\leqslant\gamma_i q_i(x), \quad \forall i\in\{1,\ldots,N\}\,. \end{array} \tag{3.96}$$

Nous appelerons cette formulation le problème des **épigrahes**. En effet, pour une fonction $f:\mathbb{R}^n\to\mathbb{R}$, l'épigraphe de f, noté $\mathbf{epi}(f)$, est défini comme l'ensemble :

$$\mathbf{epi}(f) \triangleq \{(x,\gamma)\in\mathbb{R}^n\times\mathbb{R} \mid f(x)\leqslant\gamma\}\,. \tag{3.97}$$

Ainsi, la dénomination de (3.96) vient du fait que, pour un vecteur solution (x,γ) de (3.96), chaque sous-vecteur (x,γ_i) est dans $\mathbf{epi}(\frac{p_i}{q_i})$.
Preuve: Démontrons cette équivalence, si $(\gamma^\star,x^\star)$ est solution de (3.96), alors, pour tout $(x,\gamma)\in K\times\mathbb{R}^N$ tel que $p_i(x)\leqslant\gamma_i q_i(x),\ \forall i\in\{1,\ldots,N\}$, on a :

$$\sum_{i=1}^N\gamma_i^\star \leqslant \sum_{i=1}^N\gamma_i. \tag{3.98}$$

Or, pour tout $x\in K$, $(x,\frac{p_1(x)}{q_1(x)},\ldots,\frac{p_N(x)}{q_N(x)})$ est un point réalisable de (3.96). Ainsi, on observe les inégalités suivantes :

$$\sum_{i=1}^N\frac{p_i(x^\star)}{q_i(x^\star)}\leqslant\sum_{i=1}^N\gamma_i^\star\leqslant\sum_{i=1}^N\frac{p_i(x)}{q_i(x)}, \quad \forall x\in K, \tag{3.99}$$

prouvant ainsi que $x^\star$ est solution de (3.93). Réciproquement, si $x^\star$ est solution de (3.93), alors en prenant $\gamma_i^\star=\dfrac{p_i(x^\star)}{q_i(x^\star)},\ \forall i\in\{1,\ldots,N\}$, on a bien que :

$$\sum_{i=1}^N\gamma_i^\star\leqslant\sum_{i=1}^N\gamma_i, \tag{3.100}$$

pour tout vecteur $(\gamma_i)_{i=1\ldots,N}$ tel que : $p_i(x)\leqslant\gamma_i q_i(x),\ \forall i\in\{1,\ldots,N\}$, impliquant ainsi que $(x^\star,\gamma^\star)$ est solution de (3.96). □
Ensuite, le problème dans l'espace des mesures équivalent à (3.96) s'écrit :

$$\begin{array}{rcll} \widehat{r}_{EC} & \triangleq & \min\limits_{\mu\in\mathcal{B}(K\times\mathbb{R}^N)} & \sum\limits_{i=1}^N\int_{K\times\mathbb{R}^N}\gamma_i d\mu \\ & & \text{s.l.c} & p_i(x)\leqslant\gamma_i q_i(x)\ \forall i=1,\ldots,N. \end{array} \tag{3.101}$$

Sans perte de généralité, on peut supposer que les γ_i sont bornés car la fonction objectif r est supposée, comme dans le cas polynomial, bornée. Ainsi le module quadratique engendré par les $(g_j)_{j=1,\ldots,m}$ et les N contraintes $\gamma_i q_i(x)-p_i(x)$ est archimédien (cf. 3.30). Par conséquent, puisque (3.101) est sous forme polynomiale, la suite des solutions des relaxations SDP associées à (3.101) convergent asymptotiquement vers $r^\star=r^\star_{EC}$. Cependant, l'espace de recherche du problème (3.101) est celui des mesures boréliennes dont le support est contenu dans $K\times\mathbb{R}^N$. L'espace de recherche des relaxations SDP associées sera donc $\mathbb{R}^{\mathbb{N}^{n+N}_{2t}}$ avec des contraintes sur des matrices de taille $\binom{n+N+t}{n+N}$ à $\binom{n+N+2t}{n+N}$ variables. Dès lors, on comprend que si N (le nombre de fractions) est grand, alors même les premières relaxations deviennent impossibles

à résoudre. Ainsi, il est légitime d'étudier si les variables de (3.96) possèdent une parcimonie exploitable pour la méthode présentée dans la section précédente. Ainsi, soit $(I_k)_{k=1,\ldots,N}$ un schéma de parcimonie structurée sur les variables $\{x, \gamma_1, \ldots, \gamma_N\}$ défini par $I_k \triangleq \{x, \gamma_k\}$. Pour tout $k \in \{1, \ldots, N\}$, on a :

$$\left(I_{k+1} \cap \bigcup_{j=1}^{k} I_j \right) = \{x, \gamma_{k+1}\} \cap \{x, \gamma_1, \ldots, \gamma_k\} = \{x\} \subset I_1. \tag{3.102}$$

Le schéma de parcimonie structurée $(I_k)_{k=1,\ldots,N}$ vérifie donc la propriété d'intersection courante (3.74). Ainsi, en remarquant que $I_{jk} = \{x\}$ pour tout $j, k \in \{1, \ldots, N\}$, le problème (3.96) est équivalent au problème dans l'espace des mesures suivant :

$$\begin{aligned} \widehat{r}_{EC} \quad \triangleq \quad & \min_{\mu_i \in \mathcal{B}(K \times \mathbb{R})} \quad \sum_{i=1}^{N} \int_{K \times \mathbb{R}^N} \gamma_i d\mu_i \\ & \text{s.l.c} \quad p_i(x) \leqslant \gamma_i q_i(x) \; \forall i = 1, \ldots, N. \\ & \qquad \int_K x^\alpha \, d\mu_i = \int_K x^\alpha \, d\mu_j \qquad \forall (i,j) \in \{1, \ldots, N\}, \, i \neq j \\ & \qquad \qquad \forall \alpha \in \mathbb{N}^n. \end{aligned} \tag{3.103}$$

On obtient alors un problème dans lequel on recherche N mesures $\mu_i \in \mathcal{B}(K \times \mathbb{R})$. Puisque ce problème est sous forme polynomiale et qu'il vérifie toutes les propriétés nécessaires, la suite de solutions issue de la hiérarchie de relaxations SDP associées définies par (3.88), converge vers $r^\star_{EC}$. Cependant, l'espace de recherche de ces relexations SDP reste toujours $\mathbb{R}^{\mathbb{N}^{n+N}_{2t}}$, mais les matrices définissant les contraintes sont de taille $\binom{n+1+t}{n+1}$, diminuant ainsi la complexité de $O((n+N)^{2t})$ à $O((n+1)^{2t})$. Notons que le schéma de parcimonie structurée utilisé est toujours valable quelque soit les polynômes p_i et q_i. On peut donc le qualifier de schéma de parcimonie structurée **propre à la structure du problème des épigraphes**. Par conséquent, nous disposons d'une première approche générique permettant de résoudre des problèmes d'optimisation rationnelle (3.93) où N est supposé grand.

Pour conclure, il est important de remarquer que nous n'avons pas tenu compte d'une éventuelle parcimonie des inconnues $x \in K$. Ainsi, soit $(I_k)_{k=1,\ldots,N}$ un schéma de parcimonie structurée des variables x vérifiant la propriété d'intersection courante (3.74). Définissons le schéma de parcimonie structurée $(\{I_k, \gamma_k\})_{k=1,\ldots,N}$ sur les variables $\{x, \gamma_1, \ldots, \gamma_N\}$. Puisque pour tout $j \neq k$, on a :

$$\{I_j, \gamma_j\} \cap \{I_k, \gamma_k\} = I_j \cap I_k,$$

il est facile de montrer que le schéma de parcimonie structurée $(\{I_k, \gamma_k\})_{k=1,\ldots,N}$ vérifie encore la propriété d'intersection courante (3.74). Donc, en utilisant les notations du paragraphe (3.1.4), le problème (3.103) est équivalent au problème sur l'espace des mesures suivant :

$$\begin{aligned} \widehat{r}_{EC} \quad \triangleq \quad & \min_{\mu_i \in \mathcal{B}(K_i \times \mathbb{R})} \quad \sum_{i=1}^{N} \int_{K_i \times \mathbb{R}} \gamma_i d\mu_i \\ & \text{s.l.c} \quad p_i(x) \leqslant \gamma_i q_i(x) \; \forall i = 1, \ldots, N \\ & \qquad \int_{K_k} x(J_{jk})^\alpha \, d\mu_k = \int_{K_j} x(J_{jk})^\alpha \, d\mu_j \qquad \forall k \in \mathit{\Gamma}_j, \; \forall j \in \{1 \ldots, r\} \\ & \qquad \qquad \forall \alpha \in \mathbb{N}^{n_{jk}}. \end{aligned} \tag{3.104}$$

Puisque ce problème est aussi sous forme polynomiale, la suite de solutions issue de la hiérarchie de relaxation SDP définie par (3.88), converge vers la solution de (3.101). Ceci permet donc de diminuer encore à $\binom{n_k+1+t}{n_k+1}$ la taille des matrices utilisées pour les contraintes et à $O((n_k+1)^{2t})$ la complexité. Cependant, même en tenant compte de toutes les parcimonies possibles, nous ne pouvons éviter l'ajout de N variables de relèvement. De plus, comme nous

l'avons mentionné, il est nécessaire de borner ces variables. Or, nous verrons dans la section dédiée aux tests numériques que la précision de cette approche dépend fortement de la précision avec laquelle cette borne est fixée. Pour toutes ces raisons, nous avons cherché une autre approche que nous présentons dans le paragraphe suivant.

3.2.2 Programmation rationnelle.

Le but de ce paragraphe est de présenter notre deuxième approche. Comme dans le paragraphe précédent, nous introduirons tout d'abord le problème sur l'espace des mesures puis nous montrerons son équivalence avec le problème général. Ensuite, nous présenterons la hiérarchie de relaxations SDP associées et prouverons sa convergence. Enfin, nous reproduirons le même processus mais avec une formulation qui présente un schéma de parcimonie structurée.

Le principe de cette seconde approche consiste à supprimer les dénominateurs q_i en les introduisant dans des nouvelles mesures ν_i. Pour mieux comprendre cette méthode, plaçons nous dans le cas où $N = 1$ et considérons le problème :

$$r^\star \triangleq \min_{x \in K} \frac{p(x)}{q(x)}. \tag{3.105}$$

Le problème dans l'espace des mesures s'écrit :

$$\hat{r}_{PR} \triangleq \min_{\mu \in \mathcal{P}(K)} \int_K \frac{p(x)}{q(x)}\, d\mu. \tag{3.106}$$

Afin d'éliminer le dénominateur, puisque $q(x) > 0,\ \forall x \in K$, on pose $\nu \triangleq q^{-1}\,\mu$, c'est-à-dire :

$$\nu(A) \triangleq \int_{K \cap A} q(x)^{-1} d\mu(x),\ \forall A \in \mathcal{B}(\mathbb{R}^n). \tag{3.107}$$

Remarquons que, par construction, $\mathrm{supp}(\nu) = K$. Mais afin d'avoir une équivalence avec (3.106), comme μ est une mesure de probabilité sur K, il est nécessaire que $\int_K d\mu = \int_K q\, d\nu = 1$. On obtient ainsi le problème suivant :

$$\begin{aligned} \hat{r}_{PR} \triangleq \min_{\nu \in \mathcal{M}_+(K)} & \int_K p\, d\nu \\ \text{s.l.c} & \int_{\mathbf{K}} q\, d\nu = 1. \end{aligned} \tag{3.108}$$

Supposons maintenant que $N > 1$. Définissons alors le problème dans l'espace des mesures suivant :

$$\begin{aligned} \hat{r}_{PR} \triangleq \min_{\nu_i \in \mathcal{M}_+(K)} & \sum_{i=1}^{N} \int_K p_i\, d\nu_i \\ \text{s.l.c.} & \int_K q_1 d\nu_1 = 1 \\ & \int_K x^\alpha q_i d\nu_i = \int_K x^\alpha q_1 d\nu_1, \quad \forall \alpha \in \mathbb{N}^n \\ & \qquad\qquad\qquad\quad \forall i \in \{2, \ldots, N\}. \end{aligned} \tag{3.109}$$

Nous démontrons alors le résultat suivant :

Théorème 8 *Si $q_i > 0$ sur K pour tout $i \in \{1, \ldots, N\}$, alors $\hat{r}_{PR} = r^\star$.*

Preuve: Prouvons tout d'abord que $r^\star \geqslant \hat{r}_{PR}$. Soit $x^\star \in K$ une solution de (3.93) telle que $r(x^\star) = r^\star$. Définissons $\nu_i \triangleq q_i(x^\star)^{-1}\delta_{x^\star}$ pour tout $i \in \{1,\ldots,N\}$. Par construction, les mesures $(\nu_i)_{i=1,\ldots,N}$ vérifient les contraintes de (3.109) avec pour valeur objectif :

$$\sum_{i=1}^{N}\int_K p_i\, d\nu_i \;=\; \sum_{i=1}^{N}\frac{p_i(x^\star)}{q_i(x^\star)} = r(x^\star) = r^\star. \tag{3.110}$$

Par conséquent, on a nécessairement $r^\star \geqslant \hat{r}_{PR}$. Réciproquement, soit $(\nu_i^\star)_{i=1,\ldots,N}$ une solution de (3.109). Pour chaque $i \in \{1,\ldots,N\}$, définissons les mesures $\mu_i^\star \triangleq q_i\nu_i^\star$. Comme la famille $(\nu_i^\star)_{i=1,\ldots,N}$ vérifie les contraintes de (3.109), on a :

$$\int_K x^\alpha q_i\, d\nu_i^\star = \int_K x^\alpha q_1\, d\nu_1^\star \quad \forall i \in \{2,\ldots,N\} \;\Rightarrow\; \int_K d\mu_i^\star = \int_K d\mu_1^\star \quad \forall i \in \{2,\ldots,N\}, \tag{3.111}$$

c'est-à-dire : $\mu_i = \mu_1$, pour tout $i \in \{2,\ldots,N\}$. De plus, puisque $\int_K q_1 d\nu_1 = 1$, μ_1 est nécessairement une mesure de probabilité sur K. Ainsi, on observe :

$$\sum_{i=1}^{N}\int_K p_i d\nu_i^\star \;=\; \sum_{i=1}^{N}\int_K \frac{p_i}{q_i}q_i d\nu_i^\star = \sum_{i=1}^{N}\int_K \frac{p_i}{q_i}d\mu_1^\star \tag{3.112}$$

$$= \int_K\left(\sum_{i=1}^{N}\frac{p_i}{q_i}\right)d\mu_1^\star \;=\; \int_K r d\mu_1^\star \geqslant \int_K r^\star d\mu_1^\star \;=\; r^\star, \tag{3.113}$$

ce qui prouve que $\hat{r}_{PR} \geqslant r^\star$ et conclut la preuve. □

Maintenant que nous avons montré l'équivalence du problème dans l'espace des mesures (3.109) avec le problème (3.93), il faut démontrer que la suite de solutions issue de la hiérarchie SDP associée à (3.109) converge vers la solution de (3.93), et ce avec des propriétés identiques au cas polynomial. Ainsi, pour $t \geqslant t_0 = \max(d_p, d_q, d_K)$, on définit la hiérarchie de relaxations SDP de (3.109) par :

$$\mathcal{Q}_t\begin{cases} r_t^{mom} \;\triangleq & \displaystyle\min_{(\mathbf{y}_i)_{i=1\ldots N}\in\left(\mathbb{R}^{\mathbb{N}^n_{2t}}\right)^N} \sum_{i=1}^{N} L_{\mathbf{y}_i}(p_i) \\ & \text{s.l.c} \quad M_t(\mathbf{y}_i) \succeq 0,\ \forall i \in \{1,\ldots,N\} \\ & \quad M_{t-d_{g_j}}(g_j\mathbf{y}_i) \succeq 0,\ \forall j \in \{1,\ldots,m\}\ \ \forall i \in \{1,\ldots,N\} \\ & \quad L_{\mathbf{y}_1}(q_1) = 1 \\ & \quad L_{\mathbf{y}_1}(x^\alpha q_1) = L_{\mathbf{y}_i}(x^\alpha q_i)\ \ \forall \alpha \in \mathbb{N}^n_{t-\mathrm{d}_{q_i}}\ \forall i \in \{2,\ldots,N\}. \end{cases} \tag{3.114}$$

Pour cette hiérarchie de relaxations SDP, nous démontrons le théorème de convergence suivant :

Théorème 9 *Sous l'hypothèse que* $\mathbf{M}(g_1,\ldots,g_m)$ *soit archimédien, on considère la hiérarchie de relaxations SDP* (3.114)*. On a alors les résultats suivants :*

(i) $\lim_{t\to+\infty} r_t^{mom} = \hat{r}_{PR} = r^\star$.

(ii) De plus, si $(\mathbf{y}_i)_{i=1\ldots N} \subset \mathbb{R}^{\mathbb{N}^n_{2t}}$ *est une solution optimale de* (3.114) *et si :*

$$\mathrm{rang}(M_t(\mathbf{y}_i)) \;=\; \mathrm{rang}(M_{t-\mathrm{d}_{q_i}}(\mathbf{y}_i)) \triangleq R, \quad \forall i \in \{1,\ldots,N\} \tag{3.115}$$

alors $r_t^{mom} = \hat{r}_{PR}$ *et on peut extraire* R *solutions globales.*

Preuve: Cette preuve est similaire à celle du théorème [Lasserre 2010, Théorème 4.7]. Ainsi, nous renvoyons le lecteur vers sa preuve pour le détail des topologies utilisées pour les convergences. Considérons la suite $(\mathbf{y}_i^t)_{t\geqslant t_0}$ de points proches des solutions optimales de la hiérarchie (3.114) (en toute rigueur, rien n'assure que chaque relaxation possède une solution), c'est-à-dire :

$$r_k^\star \leqslant \sum_{i=1}^N L_{\mathbf{y}_i^k}(p_i) \leqslant r_k^\star + \frac{1}{k},$$

Alors (cf.preuve de [Lasserre 2010, Théorème 4.7]), il existe une sous-suite convergente $(\mathbf{y}_i^{t_\ell})_\ell$ telle que :

$$\lim_{\ell\to\infty} \mathbf{y}_{i\alpha}^{t_\ell} = \mathbf{y}_{i\alpha}, \quad \forall \alpha \in \mathbb{N}^n \;\; \forall i = 1,\ldots,N.$$

Or, puisque les $(\mathbf{y}_i^t)_t$ sont solution de la hiérarchie (3.114), pour t fixé, on a :

$$M_t(\mathbf{y}_i^t) \succeq 0,\; M_t(g_j\mathbf{y}_i^t) \succeq 0, \quad \forall j = 1,\ldots,m \;\; \forall i = 1,\ldots,N$$

Par passage à la limite lorsque $\ell \to \infty$, on obtient :

$$M_t(\mathbf{y}_i) \succeq 0,\; M_t(g_j\mathbf{y}_i) \succeq 0, \quad \forall j = 1,\ldots,m\; \forall i = 1,\ldots,N\; \forall t \geqslant t_0.$$

Donc, par le théorème 4 de Putinar, pour chaque $\mathbf{y}_i$, il existe une mesure $\mu_i \in \mathcal{M}_+(K)$ telle que $\mathbf{y}_i$ représente le vecteur de ses moments, c'est-à-dire :

$$L_{\mathbf{y}_i}(p) = \int_K p d\mu_i, \qquad \forall\, p \in \mathbb{R}[x].$$

De plus, toujours par passage à la limite :

$$L_{\mathbf{y}_i}(q_i x^\alpha) = \int_K x^\alpha q_i(x) d\mu_i = L_{\mathbf{y}_1}(q_1 x^\alpha) = \int_K x^\alpha q_1(x) d\mu_1, \quad \forall \alpha \in \mathbb{N}^n. \tag{3.116}$$

Ensuite, pour tout $i = 1,\ldots,N$, on définit la mesure $\nu_i \in \mathcal{P}(K)$ par : $\nu_i \triangleq q_i\mu_i$. Par (3.116), on observe que : $\nu_i = \nu_1$ pour tout $i = 1,\ldots,N$. Enfin, puisque $r_t^{mom} \leqslant \hat{r}_{PR} = r^\star$ pour tout $t \geqslant t_0$, toujours par passage à la limite, il vient :

$$\begin{aligned} r^\star \geqslant \lim_{\ell\to\infty} r_{t_\ell}^{mom} &= \lim_{\ell\to\infty} \sum_{i=1}^N L_{\mathbf{y}_i^{t_\ell}}(p_i) = \sum_{i=1}^N \int_K p_i d\mu_i \\ &= \sum_{i=1}^N \int_K \frac{p_i}{q_i} q_i d\mu_i = \sum_{i=1}^N \int_K \frac{p_i}{q_i} d\nu_1 \\ &= \int_K \left(\sum_{i=1}^N \frac{p_i}{q_i}\right) d\nu_1 \geqslant r^\star. \end{aligned}$$

Comme le résultat est vrai pour n'importe quelle sous-suite convergente et que $r_{t_\ell}^{mom}$ est monotone et croissante, nous obtenons finalement que $\lim_{t\to+\infty} r_t^{mom} = r^\star$. De plus, ν_1 est une solution optimale de (3.93) avec pour valeur objectif $r^\star = \hat{r}_{PR}$. Le point (ii) est une conséquence directe du théorème de **Curto and Fialkow** [Lasserre 2010, Théorème 3.7] et chaque $\mathbf{y}_i$ représente une mesure atomique supportée par les R minima globaux dans K. □

Analysons la hiérarchie de relaxations SDP (3.114) en regard de celle issue du problème des moments généralisé (3.103). Dans le cas de (3.114), on cherche N vecteurs dans $\mathbb{R}^{\mathbb{N}^n_{2t}}$. Il résulte que l'on doit traiter, dans les contraintes, des matrices de moments et de localisation chacune de taille $\binom{n+t}{n}$ à $\binom{n+2t}{n}$ variables. Comparativement, dans les relaxations issues de (3.103), on

doit traiter des matrices de taille $\binom{n+1+t}{n+1}$ à $\binom{n+1+2t}{n+1}$ variables. On comprend dès lors qu'il est plus favorable, dès que l'ordre relaxation t augmente, d'utiliser la hiérarchie de relaxations SDP (3.114) plutôt que celle associée au problème des moments généralisés (3.103).

Par analogie au problème polynomial, il est possible de prendre en compte la parcimonie éventuellement présente dans les inconnues x afin de diminuer la taille des matrices de contraintes et ainsi de pouvoir traiter des problèmes plus importants. Le but de ce paragraphe est donc de présenter la hiérarchie de relaxations SDP prenant en compte un schéma de parcimonie structurée ainsi que les théorèmes de convergence qui lui sont associés. Nous utiliserons dans ce paragraphe les notations introduites dans le paragraphe (3.1.4). Ainsi, soit $(I_k)_{i=1,\ldots,N}$ un schéma de parcimonie structurée sur les variables x. Considérons le problème dans l'espace des mesures suivant :

$$\begin{aligned} \hat{r}_{PR} \triangleq \min_{\nu_i \in \mathcal{B}(K_i)} & \sum_{i=1}^{N} \int_K p_i \, d\nu_i \\ \text{s.l.c.} & \int_{K_i} q_1 d\nu_1 = 1, \quad \forall i \in \{1,\ldots,N\} \\ & \pi_{ij}(q_i\nu_i) = \pi_{ji}(q_j\nu_j), \quad \forall j \in \Gamma_i, \ \ \forall i \in \{1,\ldots,N-1\}. \end{aligned} \tag{3.117}$$

Nous démontrons alors le résultat suivant :

Théorème 10 *Si le schéma de parcimonie structurée $(I_k)_{k=1,\ldots,N}$ vérifie la propriété d'intersection courante (3.74) et si, pour tout $i \in \{1,\ldots,N\}$, $q_i > 0$ sur K alors $\hat{r}_{PR} = r^\star$.*

Preuve: Soit $x^\star$ une solution de (3.93) et μ la mesure de Dirac $\delta_{x^\star}$ au point $x^\star$. On pose $\mu_i \triangleq \pi_i\mu$ la projection de μ sur K_i, c'est-à-dire : $\mu_i = \delta_{\{x_j^\star,\, j\in I_i\}}$, la mesure de Dirac au point $\left\{x_j^\star,\, j \in I_i\right\}$ de K_i. Ensuite, pour tout $i = 1,\ldots,N$, on définit la mesure $\nu_i \triangleq q_i(x^\star)^{-1}\mu_i \in \mathcal{B}(K_i)$. Ainsi, $(\nu_i)_{i=1,\ldots,N}$ appartient à l'ensemble des contraintes de (3.117). En effet, $\int_{K_i} q_i d\nu_i = 1$ pour tout $i = 1,\ldots,N$, et :

$$\{x_k^\star,\, k \in I_i \cap I_j\} \;=\; \pi_{ij}\nu_i = \pi_{ji}\nu_j, \quad \forall j \neq i \text{ tels que : } I_j \cap I_i \neq \varnothing.$$

Finalement, on obtient :

$$\sum_{i=1}^{N} \int_{K_i} p_i d\nu_i \;=\; \sum_{i=1}^{N} \frac{p_i(x^\star)}{q_i(x^\star)} \;=\; r^\star,$$

et donc $\hat{r}_{PR} \leqslant r^\star$. Réciproquement montrons que : $\hat{r}_{PR} \geqslant r^\star$. Soit $(\nu_i^\star)$ une solution (3.117). Pour chaque $i = 1,\ldots,N$, on définit μ_i la mesure de densité q_i par rapport à ν_i, c'est-à-dire :

$$\mu_i(A) \triangleq \int_{K_i \cap A} q_i(x)\, d\nu_i^\star(x), \qquad \forall A \in \mathcal{B}(K_i).$$

Par construction, $\mu_i \in \mathcal{P}(K_i)$ et, puisque $(\nu_i^\star)$ est de solution (3.117), on a : $\pi_{ij}\mu_i = \pi_{ji}\mu_j$ pour tout couple $j \neq i$ tel que $I_j \cap I_i \neq \varnothing$. Ainsi, par [Lasserre 2010, Lemme B.13], il existe une mesure de probabilité μ sur K telle que $\pi_i\mu = \mu_i$ pour tout $i = 1,\ldots,N$. Enfin, on observe que :

$$\begin{aligned} \sum_{i=1}^{N} \int_{K_i} p_i \, d\nu_i^\star = \sum_{i=1}^{N} \int_{K_i} \frac{p_i}{q_i}\, d\mu_i \;&=\; \sum_{i=1}^{N} \int_{K_i} \frac{p_i}{q_i}\, d\mu \\ &=\; \int_K \left(\sum_{i=1}^{N} \frac{p_i}{q_i}\right) d\mu \geqslant r^\star, \end{aligned}$$

et donc que : $\hat{r}_{PR} \geqslant r^\star$. □

Comme précédemment, il faut ensuite démontrer que la suite de solutions issues de la hiérarchie de relaxations SDP associée à (3.117) converge vers $r^\star$. Tout d'abord définissons $\rho_i \triangleq t - \max(\mathrm{d}_{p_i}, \mathrm{d}_{q_i})$ pour tout $i \in \{1, \ldots, N\}$. Ensuite, pour $t \geqslant t_0 = \max(d_p, d_q, d_K)$, on définit la hiérarchie de relaxations SDP de (3.117) suivante :

$$\mathcal{Q}_t \left\{ \begin{array}{rll} r_t^{mom} & \triangleq & \min\limits_{(\mathbf{y}_i)_{i=1\ldots N} \in \left(\mathbb{R}^{\mathbb{N}^n_{2t}}\right)^N} \sum\limits_{i=1}^N L_{\mathbf{y}}(p_i) \\ & \text{s.l.c} & M_t(\mathbf{y}_i, I_i) \succeq 0, \\ & & M_{t-\mathrm{d}_{g_j}}(g_j \mathbf{y}_i, I_i) \succeq 0, \quad \forall j \in J_i,\ \forall i \in \{1, \ldots, N\} \\ & & L_{\mathbf{y}_i}(q_i) = 1, \quad \forall i \in \{1, \ldots, N\} \\ & & L_{\mathbf{y}_i}(x^\alpha q_i) = L_{\mathbf{y}_j}(x^\alpha q_j) \ \ \forall \alpha \in \Lambda_{\rho_i}^{I_i \cap I_j} \\ & & \qquad\qquad\qquad\qquad\quad \forall j \in \Gamma_i,\ \forall i \in \{1, \ldots, N-1\}. \end{array} \right. \tag{3.118}$$

Notons que, pour chaque i dans $\{1, \ldots, N\}$, $\mathbf{y}_i = (y_{i\alpha})_{\alpha \in \mathbb{N}^n_{2t}}$ est un vecteur de $\mathbb{R}^{\mathbb{N}^n_{2t}}$ dont seul les coefficients $(y_{i\alpha})_{\alpha \in \Lambda^{I_i}}$ sont pertinents (tous les autres coefficients pouvent être mis à zéro). Pour cette hiérarchie de relaxations SDP, nous démontrons le résultat de convergence suivant :

Théorème 11 *Si le schéma de parcimonie structurée $(I_k)_{k=1,\ldots,N}$ vérifie la propriété d'intersection courante (3.74), alors, pour la hiérarchie SDP définie par (3.118), on a les résultats suivants :*

(i) $\lim\limits_{t \to +\infty} r_t^{mom} = r^\star$.

(ii) Pour tout i, posons $v_i \triangleq \max_{j \in J_i}(\mathrm{d}_{g_j})$. Avec cette notation, si on a de plus $\mathbf{y}^\star = (\mathbf{y}_1^\star, \ldots, \mathbf{y}_N^\star) \in \left(\mathbb{R}^{\mathbb{N}^n_{2t}}\right)^N$ une solution optimale de (3.118) telle que :

$$\mathrm{rang}(M_t(\mathbf{y}_i^\star, I_i)) \ = \ \mathrm{rang}(M_{t-v_i}(\mathbf{y}_i^\star)) \triangleq R_i, \quad \forall i \in \{1, \ldots, N\} \tag{3.119}$$

et

$$\mathrm{rang}(M_t(\mathbf{y}_i^\star, I_i \bigcap I_j)) \ = \ 1 \quad \forall j \in \Gamma_i \ \ \forall i \in \{1, \ldots, N\} \tag{3.120}$$

alors $r_t^{mom} = r^\star$ et on peut extraire un nombre fini de solutions globales.

Preuve: La preuve est similaire à celle du théorème (9). Considérons la suite $(\mathbf{y}^t)_{t \geqslant t_0} = (\mathbf{y}_1^t, \ldots, \mathbf{y}_N^t)_{t \geqslant t_0}$ de points proches des solutions optimales de la hierarchie (3.118), c'est-à-dire :

$$r_k^\star \leqslant \sum_{i=1}^N L_{\mathbf{y}_i^k}(p_i) \leqslant r_k^\star + \frac{1}{k},$$

Alors, il existe une sous-suite convergente $(\mathbf{y}^{t_\ell})_\ell$ telle que :

$$\lim_{\ell \to \infty} \mathbf{y}_{i\alpha}^{t_\ell} \ = \ \mathbf{y}_{i\alpha}, \quad \forall \alpha \in \Lambda^{I_i} \ \forall i = 1, \ldots, N.$$

Par passage à la limite lorsque $\ell \to \infty$, on obtient :

$$M_t(\mathbf{y}_i, I_i) \succeq 0, \ M_t(g_j \mathbf{y}_i, I_i) \succeq 0, \quad \forall j \in J_i \ \forall i = 1, \ldots, N \ \forall t \geqslant t_0.$$

Comme chaque M_k est archimédien, par le théorème de Putinar [Lasserre 2010, Théorème 2.14], le vecteur $\mathbf{y}^i = (y_\alpha)_{\alpha \in \Lambda^{I_i}}$ (c.-à-d. le sous-vecteur de $\mathbf{y}_i$ obtenu en ne retenant que

les coefficients $y_{i\alpha}$, $\alpha \in \Lambda^{I_i}$) représente une mesure μ_i supportée dans K_i. Alors, encore par passage à la limite, $L_{\mathbf{y}^i}(q_i)(= L_{\mathbf{y}_i}(q_i)) = 1$, $i = 1, \ldots, N$, et :

$$L_{\mathbf{y}^i}(q_i x^\alpha) \;=\; \int_{K_i} x^\alpha \underbrace{q_i(x) d\mu_i}_{d\nu_i(x)} \;=\; \int_{K_j} x^\alpha \underbrace{q_j(x) d\mu_j}_{d\nu_j(x)}, \quad \forall \alpha \in \Lambda^{I_i \cap I_j},\ \forall j \in \Gamma_i. \tag{3.121}$$

Ainsi, pout tout $i = 1, \ldots, N$, la mesure $\nu_i \triangleq q_i \mu_i$ est mesure de probabilité dont le support est inclus dans K_i. De plus, par (3.121), on a :

$$\pi_{ij} \nu_i \;=\; \pi_{ji} \nu_j, \qquad \forall j \in \Gamma_i \;\; \forall i = 1 \ldots N.$$

Donc, par [Lasserre 2010, Lemme B.13], il existe une mesure de probabilité ν à support dans K telle que $\pi_i \nu = \nu_i$ pour tout $i = 1, \ldots, N$. En outre, on observe que

$$\begin{aligned} r^\star \geqslant \lim_{\ell \to \infty} r_{t_\ell}^{mom} \;&=\; \lim_{\ell \to \infty} \sum_{i=1}^N L_{\mathbf{y}_i^{t_\ell}}(p_i) = \sum_{i=1}^N \int_{K_i} p_i d\mu_i \\ &=\; \sum_{i=1}^N \int_{K_i} \frac{p_i}{q_i} q_i d\mu_i = \sum_{i=1}^N \int_{K_i} \frac{p_i}{q_i} d\nu_i \\ &=\; \int_K \left(\sum_{i=1}^N \frac{p_i}{q_i} \right) d\nu \geqslant r^\star. \end{aligned}$$

Comme le résultat est vrai pour n'importe quelle sous-suite convergente et que $r_{t_\ell}^{mom}$ est monotone et croissante, nous obtenons finalement que $\lim_{t \to +\infty} r_t^{mom} = r^\star$. De plus, ν est une solution optimale de (3.109) avec pour valeur à l'objectif $r^\star = \hat{r}_{PR}$. Comme pour le théorème 9, le point (ii) est une conséquence directe du théorème de **Curto and Fialkow** [Lasserre 2010, Theorem 3.7] et, pour tout $i = 1, \ldots, N$, chaque $\mathbf{y}_i^\star = (y_\alpha^\star)_{\alpha \in \Lambda^{I_i}}$ représente une mesure atomique supportée par les R_i minima globaux contenus dans K_i. □

3.2.3 Bilan.

Dans cette section, nous avons montré comment le problème de minimisation rationnelle :

$$f^\star \;\triangleq\; \min_{x \in K} \sum_{i=1}^N \frac{p_i(x)}{q_i(x)} \tag{3.122}$$

pouvait être approché par deux hiérarchies de problèmes SDP dont les suites de solutions convergent vers $f^\star$. Pour ces deux hiérarchies, nous avons en outre montré que, lorsque ces convergences étaient finies, il était alors possible d'extraire les solutions globales. Enfin, nous avons montré qu'il était possible de tenir compte d'une certaine forme de parcimonie des inconnues afin de diminuer significativement la complexité des relaxations SDP et ainsi de pouvoir traiter des problèmes de plus grande taille. Dans la section suivante, nous étudierons, sur plusieurs cas tests, quelle méthode offre les meilleures performances numériques en termes de précision et temps de calcul.

3.3 Résultats numériques

Le but de cette section est de présenter les résultats obtenus par les deux méthodes proposées sur un ensemble de cas tests. Cette comparaison nous permettra de mettre en avant, en termes de précision et de rapidité, les performances des deux approches et, le cas échéant, de les départager. Nous avons utilisé, pour cette comparaison, deux types de cas tests. La première catégorie regroupe les exemples de la communauté pour lesquels nous disposons des minima globaux. Le but évident de cette famille de tests est de nous permettre de vérifier la précision des deux méthodes. Cependant, comme nous ne traitons que des problèmes rationnels, cette catégorie est relativement pauvre car elle ne regroupe que quatre exemples. De plus, ces problèmes sont notoirement difficiles car ils sont faits pour éprouver la précision des solveurs d'optimisation globale. Ceux-ci ne permettront donc pas nécessairement de mettre en avant les différences entre les deux approches. Ainsi, nous avons construit une série d'exemples pour lesquels nous ne disposons pas des minima globaux mais qui possèdent certaines particularités permettant de distinguer des comportements propres à chacune des solutions proposées.

Avant de présenter l'étude détaillée, il convient de préciser un élément propre à l'implantation de nos deux solutions dans le logiciel libre GloptiPoly [Henrion 2002]. Supposons que l'on cherche à minimiser un polynôme p à l'aide de ce logiciel. Bien que les conditions de rang soient satisfaites pour un ordre de relaxation t, il arrive parfois que ce dernier ne soit pas assez grand pour permettre d'extraire des solutions $(x_i^\star)_{i=1,\ldots,r}$ du vecteur des moments $y_t^\star$. En effet, lorsque les conditions de rang sont satisfaites, le logiciel Glopitpoly procède à l'extraction des solutions $(x_i^\star)_{i=1,\ldots,r}$. Puis celui-ci calcule les erreurs suivantes :

$$\varepsilon_i \triangleq |p(x_i^\star) - L_{y_t^\star}(p)|, \quad \forall i = 1, \ldots, r. \tag{3.123}$$

Si toutes les erreurs ε_i sont inférieures à un certain seuil ζ (fixé arbitrairement petit), alors l'extraction est validée. Sinon le logiciel renvoie un code de sortie mentionnant cette violation. Dans ce cas, il est tout de même possible de dégager des solutions, soit en forçant l'extraction automatique en augmentant ζ, soit en extrayant manuellement une solution de la matrice des moments $M_t(y_t^\star)$. Cependant, si on ne souhaite pas forcer l'extraction et conserver le seuil d'origine, il suffit généralement de résoudre la relaxation suivante. Ainsi, lorsque nous parlerons de l'ordre de relaxation nécessaire pour obtenir un certificat d'optimalité, nous ferons référence à l'ordre de relaxation nécessaire pour satisfaire les contraintes de rang **et** celui nécessaire à l'extraction. Nous parlerons alors de **solution certifiée**. Enfin, lorsque les contraintes de rang ne sont pas satisfaites, mais que nous souhaitons disposer d'une solution, nous extrayons manuellement une solution de la matrice des moments $M_t(y_t^\star)$. Nous parlerons alors de **solution non certifiée**.

Dans le but de progresser des problèmes les plus simples vers les plus complexes, nous commencerons par étudier les résultats obtenus sur des exemples univariés, puis sur un exemple bivarié avec des degrés importants et enfin sur des problèmes académiques. Enfin, dans le but de se référer à un solveur existant et reconnu, nous comparons tous nos résultats à ceux obtenus avec le logiciel commercial BARON [Tawarmalani 2005, Sahinidis 2010] que nous avons utilisé via le serveur NEOS, disponible gratuitement à l'adresse suivante : `http://www.neos-server.org/neos/solvers/go:BARON/GAMS.html`. Notons que via ce serveur, le temps de calcul effectif est limité à 1000 secondes. Outre la solution et la valeur de la fonction objectif, nous avons choisi, pour ce logiciel, de ne donner que le nombre total d'itérations. En effet, comme nous ne disposons pas des spécifications de la machine sur lequel le solveur est exécuté et que nous employons ce logiciel sans faire appel à des options particulières qui permettraient de l'utiliser le plus finement possible, le temps de calcul n'est pas une indication satisfaisante. Enfin, le logiciel BARON possède une étape de pré-traitements

qui lui permet de déterminer une première estimation de la solution. Si la solution globale proposée par ce logiciel est atteinte durant cette étape, nous le spécifierons clairement.

3.3.1 Problèmes univariés

Le but de ce paragraphe consiste plus à appréhender le comportement des deux solutions proposées sur des exemples simples que de réellement tester leurs performances respectives. Ainsi, ce paragraphe ne regroupe que des exemples que nous avons construits et pour lesquels nous ne disposons pas de minima globaux certifiés. Les comparaisons avec le logiciel BARON sont alors fournies pour s'assurer que nos méthodes restent dans des précisions acceptables.

Exemple n°1. Dans cet exemple on cherche à résoudre le problème suivant :

$$\min_{x\in[-10,10]} f(x) \triangleq \frac{1+x+x^2}{1+x^2} + \frac{1+x^2}{1+2x^2}. \tag{3.124}$$

Sur l'intervalle $[-10, 10]$, la fonction f ne comporte pas de difficultés particulières. Les numérateurs et dénominateurs sont strictements positifs et de degrés faibles. Elle est représentée sur la Figure 3.2.

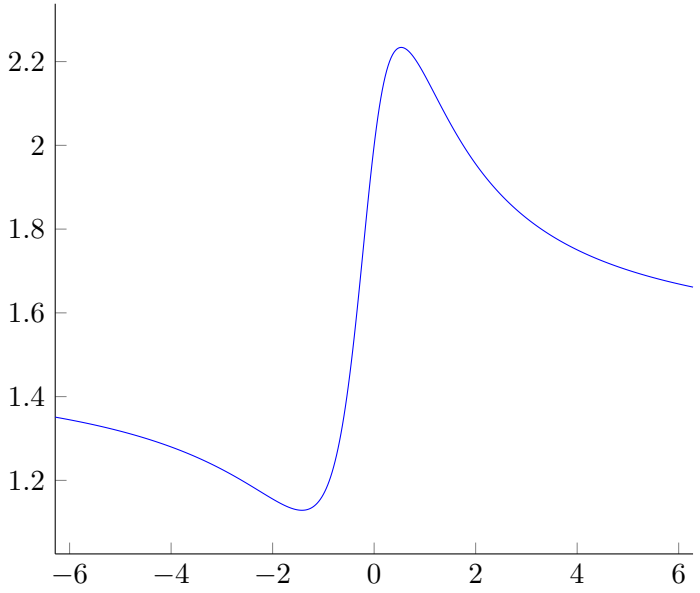


FIGURE 3.2 – La fonction rationnelle objectif représentée sur l'ensemble $[-10, 10]$.

Les deux méthodes renvoient un minimum global certifié en $x^\star = -1.421$ avec pour valeur objectif $f^\star = 1.128$. Les deux méthodes nécessitent de résoudre la cinquième relaxation pour obtenir un certificat d'optimalité. Le temps de calcul est de 0.58 secondes pour la Programmation Rationnelle contre 0.74 secondes avec les Epigraphes Creux. A titre de comparaison le logiciel BARON fournit la même solution en 9 itérations. La très légère différence de temps de calcul n'est pas assez significative pour conclure en faveur de l'une ou l'autre méthode. Cependant, l'ordre de relaxation des Epigraphes Creux est fortement lié aux bornes imposées sur les variables x et γ, alors qu'il n'est pas nécessaire de borner la variable x dans la Programmation Rationnelle. Par exemple, pour obtenir un certificat d'optimalité pour la solution calculée par la méthode des Epigraphes Creux, il est nécessaire de fixer ces deux bornes à 20. Si cette borne diminue alors l'ordre de relaxation diminue et inversement. Par exemple, il passe à 4 pour une borne égale à 10. Nous vérifierons si cette tendance s'observe sur les exemples suivants.

Exemple n°2 : Les polynômes de Wilkinson. On cherche à résoudre le problème d'optimisation suivant :

$$\max_x \sum_{k=1}^{N} f_k(x) \tag{3.125}$$

où les f_k sont des fonctions rationnelles, inspirées du polynôme de Wilkinson, définies par :

$$f_k(x) \triangleq \frac{1}{k+x^2}, \quad \forall k = 1, \ldots, N. \tag{3.126}$$

La Figure 3.3 représente la fonction coût pour $N = 5$.

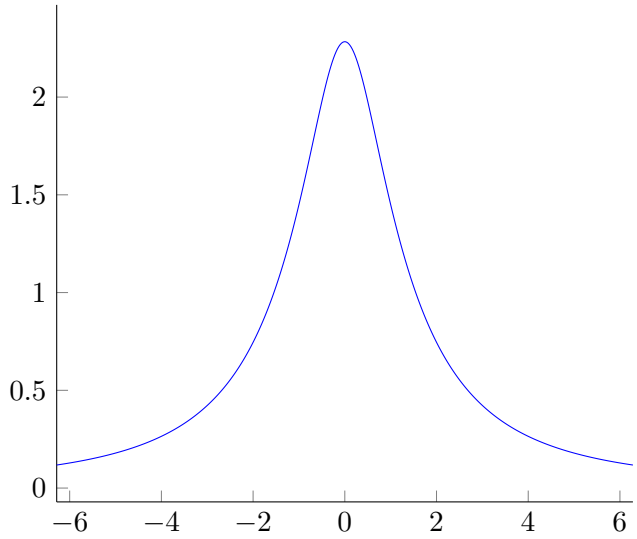


FIGURE 3.3 – Fonction objectif du problème de maximisation de Wilkinson pour $N = 5$

Puisque :

$$\frac{1}{k} \leqslant \frac{1}{k+x^2} \quad \forall x \in \mathbb{R},\ \forall 1 \leqslant k \leqslant N,\ \forall N > 1, \tag{3.127}$$

il est facile de vérifier que, pour tout $N \geqslant 1$, l'unique solution théorique est atteinte en $x^\star = 0$ et vaut $f^\star = \sum_{k=1}^{N} \frac{1}{k}$. Afin de tester un cas où le degré résultant de la réduction au même dénominateur serait grand, nous fixons $N = 20$. Sans borner la variable x, la solution $x^\star_{PR} = 0$ donnée par la Programmation Rationnelle, en 0.62 secondes, est certifiée dès la première relaxation. Pour le cas de la solution par Epigraphes Creux, avec une borne sur x et γ égale à 10, il faut résoudre la quatrième relaxation pour avoir un résultat certifié. La solution obtenue est $x^\star_{EC} = 0$ en 2.2 secondes. A titre de comparaison, le logiciel BARON fournit le résultat dès l'étape de pré-traitements. Cependant, si on remplace les inégalités par des égalités dans les contraintes liées aux épigraphes :

$$p_k(x) \leqslant \gamma_k q_k(x) \rightarrow p_k(x) = \gamma_k q_k(x)$$

le problème reste équivalent au problème initial. Ce remplacement permet de se ramener à un comportement proche de la Programmation Rationnelle. Dans ce cas, pour une borne sur x et γ égale à 10, le résultat est certifié dès la première relaxation. On obtient alors la solution $x^\star_{EC} = 0$ en 1.11 secondes. Outre la nécessité de borner le problème, on retiendra de cet exemple que la méthode des Epigraphes Creux se comporte mieux si on remplace les inégalités par des égalités dans les contraintes liées aux épigraphes. Ainsi dans toute la suite de cette étude, lorsque nous nous référerons à la résolution par Epigraphes Creux, nous sous-entendrons que c'est celle (plus favorable a priori) avec des contraintes égalités.

Exemple n°3 : fonction univariée à plusieurs minima locaux. En examinant attentivement la formulation par Epigraphes Creux, on observe que nous avons dû ajouter N nouvelles contraintes égalités dans K. Comme nous l'avons remarqué dans l'exemple précédent, ces contraintes peuvent, dans certains cas, désavantager la résolution par Epigraphes Creux. Par ailleurs, la Programmation Rationnelle fait intervenir les polynômes dénominateurs q_i dans ses contraintes. Elle possède donc des contraintes sur les moments plus complexes que celles des Epigraphes Creux. Les deux font ainsi intervenir les polynômes q_i dans leurs contraintes, mais seule la formulation par Epigraphes Creux fait intervenir les polynômes numérateurs p_i. Ainsi, dans cet exemple nous cherchons à vérifier que si les polynômes p_i sont de degrés nuls, alors les deux approches sont équivalentes. C'était déjà le cas dans l'exemple précédent, mais la fonction coût ne possédait qu'un seul maximum local qui était aussi global. Dans cet exemple, la fonction coût possède plusieurs minima locaux. Le problème de minimisation est défini de la manière suivante :

$$\min_{x\in[0,1]} \sum_{k=1}^{10} \frac{-1}{(10x-a_k)^2+c_k} \tag{3.128}$$

avec :

$$a = (4,1,8,6,3,2,3,8,6,7)^\top \tag{3.129}$$
$$c = (0.1,0.2,0.2,0.4,0.4,0.6,0.3,0.7,0.5,0.5)^\top . \tag{3.130}$$

La fonction coût est représentée sur la Figure 3.4. La solution $x^\star_{PR} = 0.390$ donnée par la

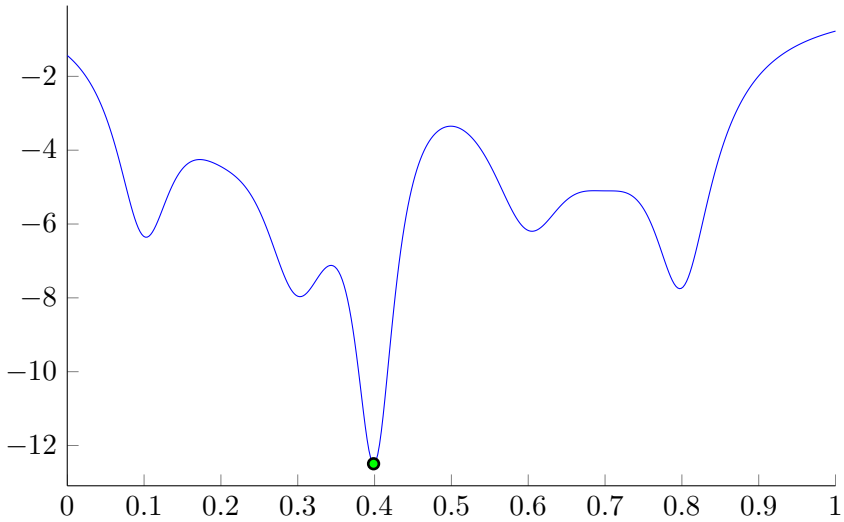


FIGURE 3.4 – Fonction coût du problème (3.128) et, en vert, la solution obtenue par la méthode de Programmation rationnelle

Programmation Rationnelle est certifiée à partir de la $5^{\text{ème}}$ relaxation. Elle est calculée en 2.98 secondes et vaut à l'objectif $f^\star = -12.49$. Pour le cas des Epigraphes Creux, avec une borne sur les variables de relèvement égale à 10, la solution n'est jamais certifiée pour des ordres de relaxation raisonnables (c.-à-d. inférieurs à 10). Cependant, dès la première relaxation, et bien que les conditions de rang ne soient pas satisfaites, une solution $x_{EC} = 0.400$ valant à l'objectif $f^\star = -12.48$ peut être extraite. Le temps de calcul correspondant est de 2.58 secondes. A titre d'exemple, le logiciel BARON fournit la solution $x^\star_B = 0.39$ en 7 itérations. Nous pouvons donc conclure que, même dans un cas qui semble l'avantager, la solution par Epigraphes Creux ne certifie pas la solution globale. Dans un même temps, la Programmation Rationnelle garantit le minimum global.

Exemple n°4 : Exemple avec des degrés élevés Avec cet exemple, nous avons souhaité augmenter plus significativement la complexité de la fonction coût à minimiser. Pour cela nous considérons le problème suivant :

$$\min_{x\in[-6,6]} \sum_{k=1}^{30} \frac{(1+x)(k^2+1+x^2+x^4)-(1+kx^2)(1+x^2+2k)}{(1+x^2+2k)(1+k^2+x^2+x^4)}. \tag{3.131}$$

La fonction est donc composée de 30 fractions rationnelles ayant chacune un numérateur de degré 5 et un dénominateur de degré 6. On comprend, dès lors, que la réduction au même dénominateur va impliquer une explosion des degrés. La fonction coût de ce problème est représentée sur la Figure 3.5. Avec la Programmation Rationnelle, il faut calculer la $5^{\text{ième}}$ relaxation pour certifier la solution $x^\star_{PR} = -4.06$ valant $f^\star_{PR} = -14.1$ à l'objectif. Le calcul s'effectue en 3.85 secondes. Avec les Epigraphes Creux, une solution peut être extraite à partir de la $4^{\text{ième}}$ relaxation, mais ne peut être certifiée (c.-à-d. conditions de rang non satisfaites) pour des ordres raisonnables. Cette solution $x_{EC} = -2.66$, calculée en 7.48 secondes vaut $f_{EC} = -11.8$ à l'objectif. Elle est donc nettement moins précise que celle obtenue via la Programmation Rationnelle (cf 3.5). On peut donc affirmer que la résolution par Epigraphes Creux échoue sur cet exemple. Comme nous l'avons souligné, ceci peut être dû aux degrés élevés des polynômes p_i qui, dans ce cas, pénalisent la résolution par Epigraphes Creux. Par ailleurs, on notera que le logiciel BARON fournit une solution identique à la programmation rationnelle mais en 73074 itérations et dépasse le maximum de temps de calcul autorisé (1000 secondes). Ce problème peut donc être qualifié de difficile, soulignant ainsi la performance de la Programmation Rationnelle.

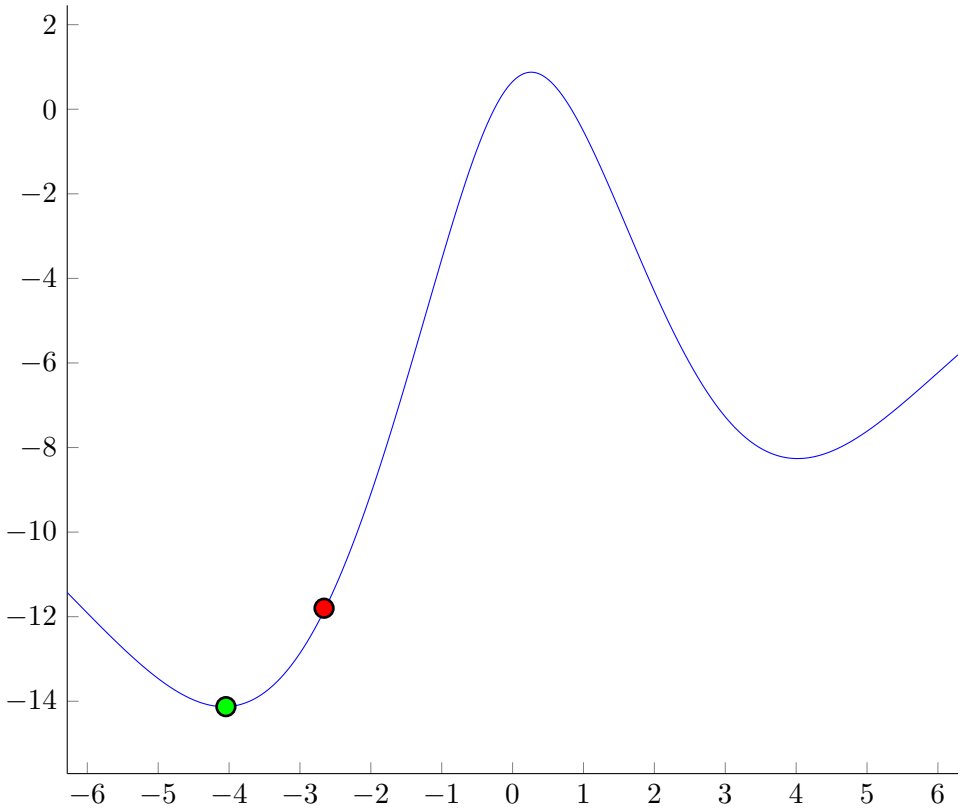


FIGURE 3.5 – Fonction coût du problème 1D à hauts degrés, la solution certifiée par la méthode de Programmation Rationnelle (en vert) et la solution, non certifiée, extraite de la méthode par Epigraphes Creux.

Exemple n°5 : problèmes à deux minima globaux Dans cet exemple nous avons voulu tester le comportement des deux méthodes face à un problème comportant deux minima globaux. Le problème testé est issu d'un problème de triangulation à partir de deux vues (cf [Hartley 2003, exemple 12.4(a)]). Il est formulé de la manière suivante :

$$\min_{x\in[-4,6]} \frac{x^2}{1+x^2} + \frac{(3x+4)^2}{(2x+3)^2+(3x+4)^2}. \tag{3.132}$$

Cette fonction rationnelle, repésentée sur la Figure 3.6, possède 3 minima : 1 local et 2 globaux, dont un est relativement mal conditionné (c.-à-d. f varie de manière importante dans un petit voisinage autour de ce point). Avec la Programmation Rationnelle (PR), dès

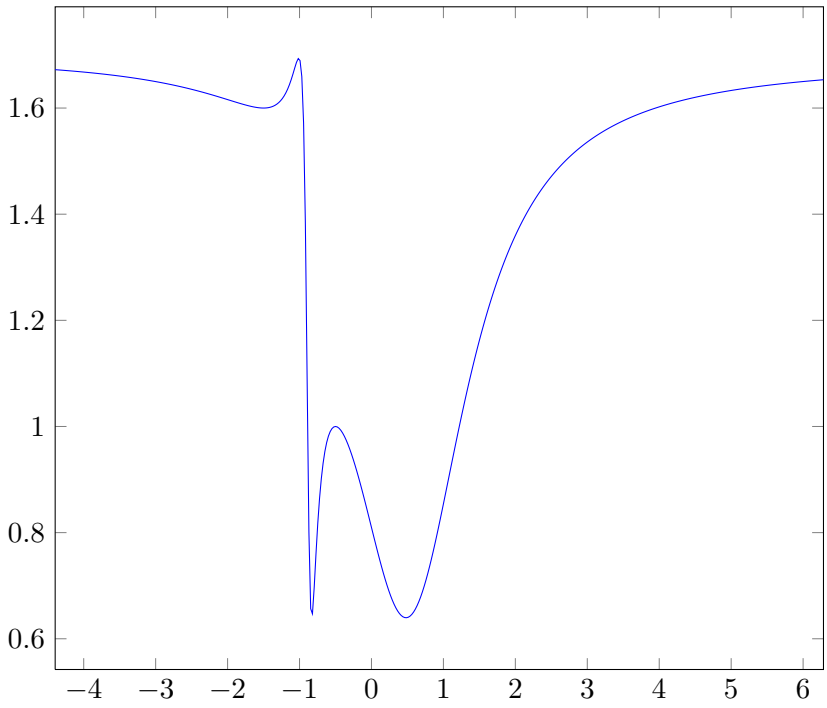


FIGURE 3.6 – Fonction coût du problème à deux minimia

la 1ère relaxation, les solutions $(x^\star_{PR,1}, x^\star_{PR,2}) = (-0.830, 0.500)$ sont certifiées. Elles valent, à l'objectif, $f^\star_{PR} = 0.639$ et sont calulées en 0.68 secondes. Avec les Epigraphes Creux (EC), on obtient une solution certifiée à la 2ème relaxation. Les solutions calculées sont $(x^\star_{EC,1}, x^\star_{EC,2}) = (-0.830, 0.480)$ et valent $f^\star_{EC} = 0.640$ à l'objectif. Le temps de calcul correspondant est de 1.1 secondes. Bien que la solution par Epigraphes Creux résolve aussi le problème, il semble que la Programmation Rationnelle soit donc plus avantageuse sur cet exemple, car elle trouve la solution à un ordre de relaxation moins élevé. Le logiciel BARON détecte les deux minima au bout d'une itération. Afin de faciliter la lecture de ces résultats, nous les avons résumés dans le tableau 3.1.

Méthode	$x_1^\star$	$x_2^\star$	$f^\star$	Temps (s)	Relaxation	Nb itérations
PR	-0.830	0.500	0.639	0.68	1	N.A. [1]
EC	-0.830	0.480	0.640	1.1	2	N.A.
BARON	-0.830	0.500	0.639	N.A	N.A.	1

TABLE 3.1 – Résultats obtenus pour le problème à deux minima globaux

1. Non Applicable

Exemple n°6 : problème avec discontinuités Le but de cet exemple est d'étudier le comportement des deux approches sur une somme de fractions rationnelles avec des discontinuités. Cet exemple est inspiré par la fonction coût de triangulation car chaque fraction amène une discontinuité différente des autres. Le problème est donné par :

$$\min_{x\in[0,1]} \sum_{k=1}^{N} \frac{(k(N+1)x)^2 + k(N+1)x - k^4}{100((N+1)x-k)^2}. \tag{3.133}$$

Pour l'ensemble de l'expérience, on suppose que $N = 5$. Dans le cas de la Programmation Rationnelle, la solution $x_{PR} = 0.800$ peut être extraite à la $3^{\text{ème}}$ relaxation. Elle vaut à l'objectif $f^\star_{PR} = -3.63$. Le temps de calcul correspondant est de 0.8 secondes. Bien que les conditions de rang ne permettent pas de certifier cette solution, on remarquera (cf. Figure 3.7) qu'elle est très proche du minimum global. Cependant, augmenter l'ordre de relaxation ne permet pas d'améliorer la qualité de la solution. En effet, il semble que, plus on contraint le problème, plus il devient complexe à résoudre. Dans le cas de la formulation par Epigraphes Creux, on peut extraire la solution $x^\star_{EC} = 0.560$ à la $3^{\text{ème}}$ relaxation en 2.55 secondes. Mais cette solution n'est pas dans le bon bassin d'attraction (cf. Figure 3.7). Cependant, selon tous les tests que nous avons réalisés, augmenter l'ordre de relaxation ne permet pas de passer dans le bon bassin d'attraction. Le logiciel BARON renvoie la solution $x^\star_B = 0.451$ qui vaut $f^\star_{Bar} = -1.43$ en 6419 itérations. La solution $x^\star_B$ n'est donc pas, elle non plus, dans le bon bassin d'attraction. Ainsi, on constate que, sur cet exemple difficile, la Programmation Rationnelle arrive à trouver une solution de bonne qualité.

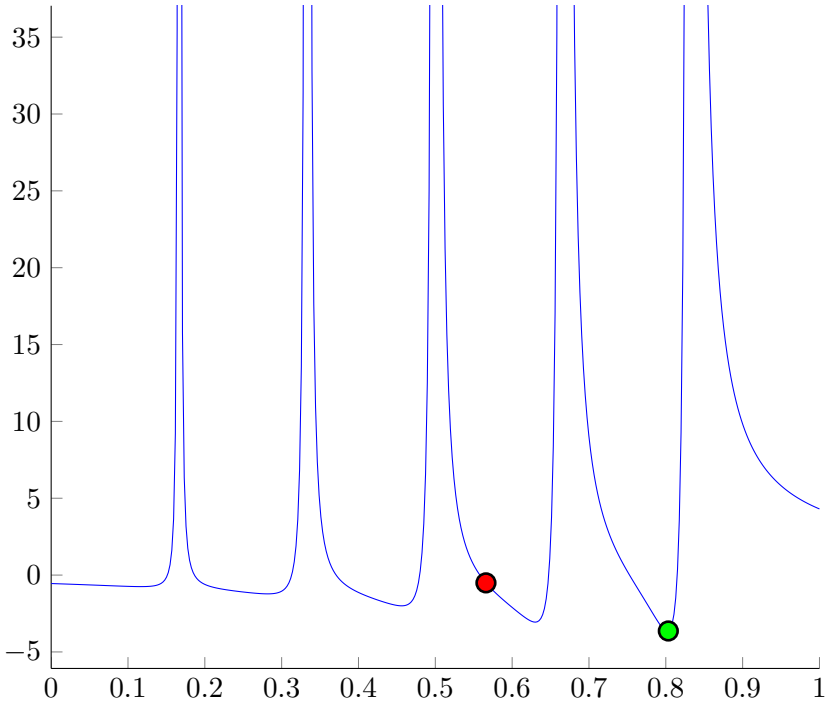


FIGURE 3.7 – Solution calculée par la Programmation Rationnelle

3.3.2 Problème bivarié

Cet exemple constitue le pendant de l'exemple univarié à degrés élevés, mais en deux dimensions. Ainsi, le but de cet exemple est de vérifier que la méthode par Epigraphes Creux est défavorisée lorsque les degrés des numérateurs sont élevés. Rappelons en effet que le numérateur n'intervient que dans la fonction coût de la Programmation Rationnelle, et pas dans ses contraintes comme c'est le cas pour les Epigraphes Creux. De plus, cet exemple possède plusieurs minima locaux et il est difficile de le résoudre par réduction au même dénominateur. Le problème étudié dans ce paragraphe est donc le suivant :

$$\min_{(x,y)\in[-10,10]^2} \sum_{k=1}^{N} \frac{(x+y)(k^2+x^2+x^2y^2+y^4)-(1+ky^2)(x^4+y^2+2k)}{(x^4+y^2+2k)(k^2+x^2+x^2y^2+y^4)}. \tag{3.134}$$

La fonction est représentée graphiquement pour N valant respectivement 5 et 10 sur la Figure 3.8. Nous traitons ce problème pour $N = 10$. Dans le cas de la Programmation Ration-

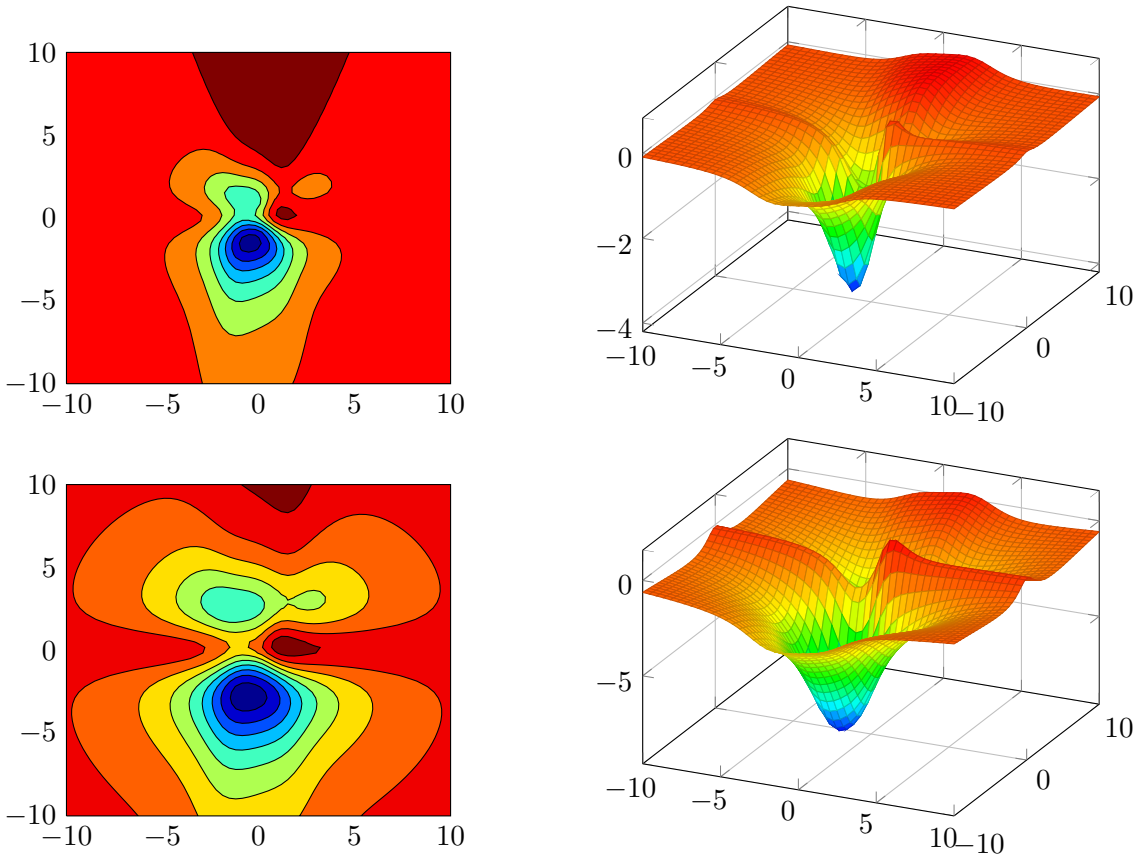


FIGURE 3.8 – Fonctions coûts et lignes niveau du problème 2D à degrés élevés pour $N = 5$ $N = 10$

nelle, la solution $x^\star_{PR} = (-0.600, -2.20)$ est certifiée à la $6^{\text{ème}}$ relaxation en 3.19 secondes. Elle vaut à l'objectif $f^\star_{PR} = -6.28$. Pour le cas de la solution via les Epigraphes Creux, la solution n'est jamais certifiée pour des ordres de relaxation raisonnables. La meilleure solution en termes de précision est obtenue à la $6^{\text{ème}}$ relaxation. Après extraction, elle vaut $x^\star_{EC} = (-0.580, -2.09)$ et a pour valeur objectif $f^\star_{EC} = -6.27$. Ce résultat est donc très proche du précédent, cependant il est obtenu en 71.79 secondes. Ceci confirme donc que le degré des numérateurs désavantage la résolution par Epigraphes Creux. Enfin, dans le but de souligner que ce problème est difficile, voici les résultats obtenus par le logiciel BARON : la solution proposée est $x^\star_B = (-0.88, -3.03)$, valant à l'objectif $f^\star_B = -5.72$ et calculée en 37147 itérations.

Méthode	$x_1^\star$	$x_2^\star$	$x_3^\star$	$x_4^\star$	$f^\star$	tps(s)	nb Mom.	nb Iter.
PR	0.192	0.190	0.123	0.135	3.10×10^{-4}	9.37	2310	N.A.
EC	0.197	0.207	0.163	0.059	0.010	4.12	1316	N.A.
BARON	0.193	0.190	0.123	0.136	3.07×10^{-4}	N.A.	N.A.	2377
Théorique	0.192	0.190	0.123	0.135	3.10×10^{-4}	N.A.	N.A.	N.A.

TABLE 3.2 – Résultats obtenus pour le problème de Kowalik.

3.3.3 Problème de Kowalik

Dans cet exemple, nous avons voulu tester un problème de moindres carrés. Le problème considéré est défini par :

$$\min_{x\in\left[0,\frac{42}{100}\right]^4} \sum_{k=1}^{11} \left(a_k - \frac{x_1(1+x_2b_k)}{(1+x_3b_k+x_4b_k^2)} \right)^2 \tag{3.135}$$

avec :

$$\begin{aligned} a &= (0.195\ \ 0.194\ \ 0.173\ \ 0.160\ \ 0.084\ \ 0.062\ \ 0.045\ \ 0.034\ \ 0.032\ \ 0.023\ \ 0.024)^\top \\ b &= \left(\ 1/4\ \ 1/2\ \ 1\ \ 2\ \ 4\ \ 6\ \ 8\ \ 10\ \ 12\ \ 14\ \ 16\ \right)^\top. \end{aligned}$$

Cet exemple est un problème académique pour lequel nous disposons du minimum global. Il semble relativement simple puisqu'il n'a que 4 variables d'optimisation et les polynômes p_i et q_i sont de degré 2. La solution certifiée par la Programmation Rationnelle est obtenue à la $3^{\text{ème}}$ relaxation. La meilleure solution par Epigraphes Creux est obtenue à la $2^{\text{ème}}$ relaxation, d'où des temps de calcul moindres. Elle n'est cependant pas certifiée. Notons que pour les Epigraphes Creux, la borne sur les variables de relèvement est fixée à 1, de sorte que : $\gamma_i^2 \leqslant 1,\ \forall i = 1,\ldots,11$. En effet, en faisant varier cette borne de 0.021 à 2 par pas de 0.1, nous avons constaté que les solutions obtenues ne gagnaient pas en précision. Le tableau (3.2) récapitule les résultats obtenus par les deux méthodes ainsi que ceux proposés par le logiciel BARON. On constate donc que la Programmation Rationnelle fournit exactement le minimum global dans un temps raisonnable. Modulo de très légers écarts qui pourraient être corrigés par une utilisation plus fine du logiciel, la solution proposée par BARON est, elle aussi, à l'optimum global. On notera cependant un nombre d'itérations relativement élevé soulignant la complexité du problème. La solution fournie par les Epigraphes Creux semble éloignée du minimum global. Cependant, après raffinage à l'aide d'une méthode de Newton locale, on obtient le minimum global attendu en 30 itérations. Donc cette solution semble, tout de même, dans le bon bassin d'attraction. Pour cet exemple, il nous a semblé intéressant de faire apparaître, dans le tableau (3.2), le nombre de moments traités dans chaque relaxation.

Méthode	nb contraintes de support	nb contraintes sur les moments	nb substitutions
(EC)	33	701	700
(PR)	11	351	0

De manière plus précise, on a :

- Dans le cas des Epigraphes Creux, il y a 11 contraintes de localisation sur les x_i, 11 sur les γ_i et 11 pour former l'épigraphe. Les 701 contraintes sur les moments sont imposées par la prise en compte de la parcimonie de structure de l'épigraphe. Les 700 substitutions (c.-à-d. les remplacements de variables entre elles) sont fournies par les égalités des sous-mesures entre elles (elles sont par exemple toutes égales à la première).
- Dans le cas de la Programmation Rationnelle, il y a seulement 11 contraintes de localisation sur les x_i. Les contraintes sur les moments représentent l'égalité des mesures ν_i entre elles.

On constate donc que le nombre de moments est beaucoup plus faible pour la Programmation Rationnelle que pour les Epigraphes Creux. Cependant, cette dernière méthode permet un nombre très important de substitutions et ainsi une diminution effective du nombre de la taille des matrices de contraintes. Cette exemple souligne ainsi que les contraintes de support liées à la formation des épigraphes jouent un rôle défavorable dans les Epigraphes Creux qui n'est pas compensé par un nombre très important de substitutions. Ceci confirme donc la tendance observée précédemment, à savoir que la formulation par Epigraphes Creux est moins précise que la Programmation Rationnelle.

3.3.4 Exemple multivarié creux

Le but de cet exemple est de fournir un problème contenant une parcimonie évidente et dont on connaît théoriquement le minimum global. Considérons la fonction définie par :

$$\begin{aligned} f : \mathbb{R}^n &\to \mathbb{R} && (3.136)\\ x &\mapsto \sum_{j=1}^{m} p_j(x)^2 && (3.137) \end{aligned}$$

où $p_j : \mathbb{R}^n \to \mathbb{R}$. Par conséquent, si R_f, l'ensemble des zéros de f donné par :

$$R_f = \{x \in \mathbb{R}^n \mid q_j(x) = 0 \ \ \forall j = 1 \dots m\},$$

est non vide et si $c \in \mathbb{R}^n_{+*}$ alors :

$$\arg\min_{x \in \mathbb{R}^n} f(x) = \arg\max_{x \in \mathbb{R}^n} \sum_{j=1}^{m} \frac{1}{p_j(x)^2 + c_j} = R_f. \tag{3.138}$$

En effet, puisque $p_j(x)^2 + c_j \geqslant c_j(x)^2 > 0 \ \ \forall j = 1 \dots n$, on a :

$$f(x) \leqslant \sum_{j=1}^{m} c_j = f(x^\star), \forall (x, x^\star) \in \mathbb{R}^n \times \mathbb{R}^n$$

. Ceci nous permet donc de fabriquer des sommes de fonctions rationnelles à partir de polynômes dont on connaît les zéros. Considérons alors la fonction de Rosenbrock définie par :

$$\begin{aligned} f : \mathbb{R}^n &\to \mathbb{R} && (3.139)\\ x &\mapsto \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2. && (3.140) \end{aligned}$$

On a donc que :

$$\arg\min_{x \in \mathbb{R}^n} f = \arg\max_{x \in \mathbb{R}^n} \sum_{i=1}^{n-1} \frac{1}{100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 + 1} = \{1, \dots, 1\}. \tag{3.141}$$

Ainsi, la valeur optimale du problème est :

$$f^\star = f(1, \dots, 1) = n - 1. \tag{3.142}$$

Ce problème possède un schéma de parcimonie structurée évident car les variables x_i sont décalées d'une fraction à l'autre. De plus ce schéma vérifie la propriété **d'intersection courante**. En effet, en choisissant la famille $(I_i)_{i=1,\dots,n-1}$ de variables définie par :

$$I_i = \{x_i, x_{i+1}\}, \quad \forall i = 1, \dots, n-1, \tag{3.143}$$

on a, pour tout k dans $\{1, \ldots, n-1\}$, la relation suivante :

$$\left(I_{k+1} \cap \bigcup_{j=1}^{k} I_j \right) = \{x_k, x_{k+1}\} \cap \{x_1, \ldots, x_k\} = \{x_k\} \subset I_{k-1}. \tag{3.144}$$

Ainsi, comparons sur ce problème la Programmation Rationnelle et les Epigraphes Creux en tenant compte de cette parcimonie. Etudions tout d'abord le nombre total de moments et le temps de calcul en fonction du nombre de variables. Pour cela, on effectue 50 résolutions successives obtenues en faisant varier le nombre d'inconnues de 20 à 1000 variables par pas de 20. Le nombre de moments est représenté sur la figure 3.9 et les temps de calcul sur la Figure 3.10. Une constatation surprenante est que les temps de résolution semblent évoluer linéairement avec le nombre de variables et donc le nombre de moments. Les temps de calcul moindres pour la Programmation Rationnelle peuvent s'expliquer par le fait que les p_i sont de degré nul. Analysons maintenant les précisions des solutions obtenues. Tout d'abord, notons que nous ne comparons que les résultats obtenus à la $1^{\text{ère}}$ relaxation car, celle-ci suffit pour un grand nombre de cas à obtenir un certificat d'optimalité global de la solutions. De plus, il devient très couteux de résoudre les relaxations d'ordre supérieur en temps des temps de calcul raisonnables car, même en prenant en compte le schéma de parcimonie, les relaxations augmentent très vite en taille. la Programmation Rationnelle a toujours certifié les solutions, sauf pour $N = 760$ et $N = 780$. Mais pour ces valeurs, les valeurs de la fonction objectif sont tout de même égales à la valeur théorique $f^\star$. Bien que les valeurs à l'objectif soient très proches de $f^\star$, les Epigraphes Creux n'ont pas certifié les solutions pour $N = 340, 420, 550, 600, 620, 680, 700, 800, 820, 880, 920, 1000$.

Ensuite, nous avons tracé les erreurs $\|x^\star - x^\star_{PR}\|$ et $\|x^\star - x^\star_{EC}\|$ en fonction de N sur la figure 3.11, puis les erreurs $\dfrac{|f^\star - f(x^\star_{PR})|}{f^\star}$ et $\dfrac{|f^\star - f(x^\star_{EC})|}{f^\star}$ en fonction de N sur la Figure 3.12. On remarque que les résultats obtenus par les Epigraphes Creux sont plus précis que ceux obtenus par la Programmation Rationnelle. Ceci provient de la borne imposée sur les γ_i qui, si elle est correctement fixée (ce qui est souvent difficile), apporte une information supplémentaire aux Epigraphes Creux. Nous avons aussi traité cet exemple avec le logiciel BARON. Cependant, celui-ci est clairement désavantagé car il ne tient pas compte de la parcimonie des variables. Les résultats qu'obtient cet algorithme ne nous permettent donc pas de pouvoir effectuer une comparaison. A titre informatif, on mentionnera seulement que BARON obtient la solution théorique jusqu'à $N = 120$ en 7426 itérations et atteint son temps de calcul maximal alloué. Pour des valeurs supérieures, BARON ne trouve pas de solutions. Pour résumer, en présence d'un schéma de parcimonie, les deux formulations donnent des résultats satisfaisants, mais la Programmation Rationnelle est plus robuste et plus rapide que les Epigraphes Creux. Cette dernière méthode s'avère cependant plus précise lorsque les bornes sur les variables de relèvement sont correctement fixées. On conclura donc que la prise en compte de la parcimonie permet aux deux méthodes de résoudre des problèmes de grande taille, avec un léger avantage cependant pour la Programmation Rationnelle.

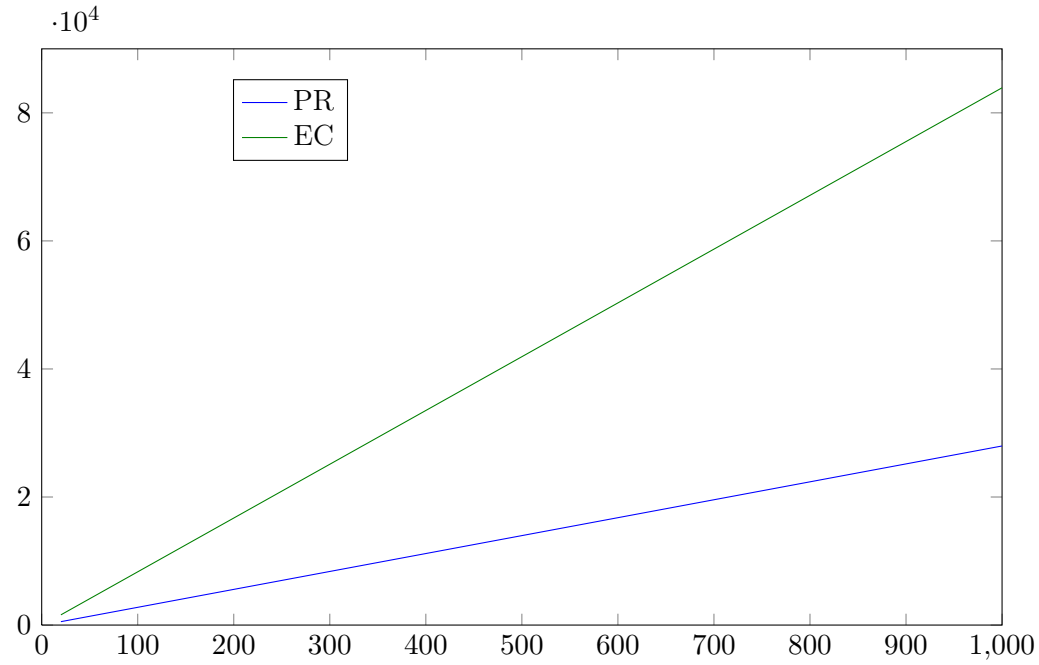


FIGURE 3.9 – Nombre de moments en fonction du nombre de variables

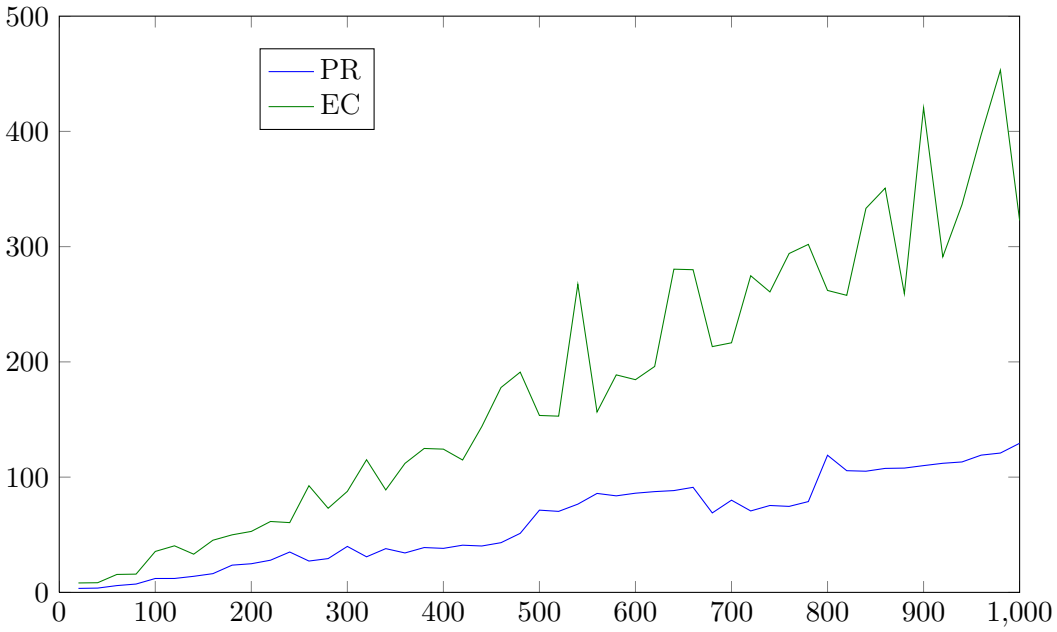


FIGURE 3.10 – Temps de calcul (secondes) en fonction du nombre de variables

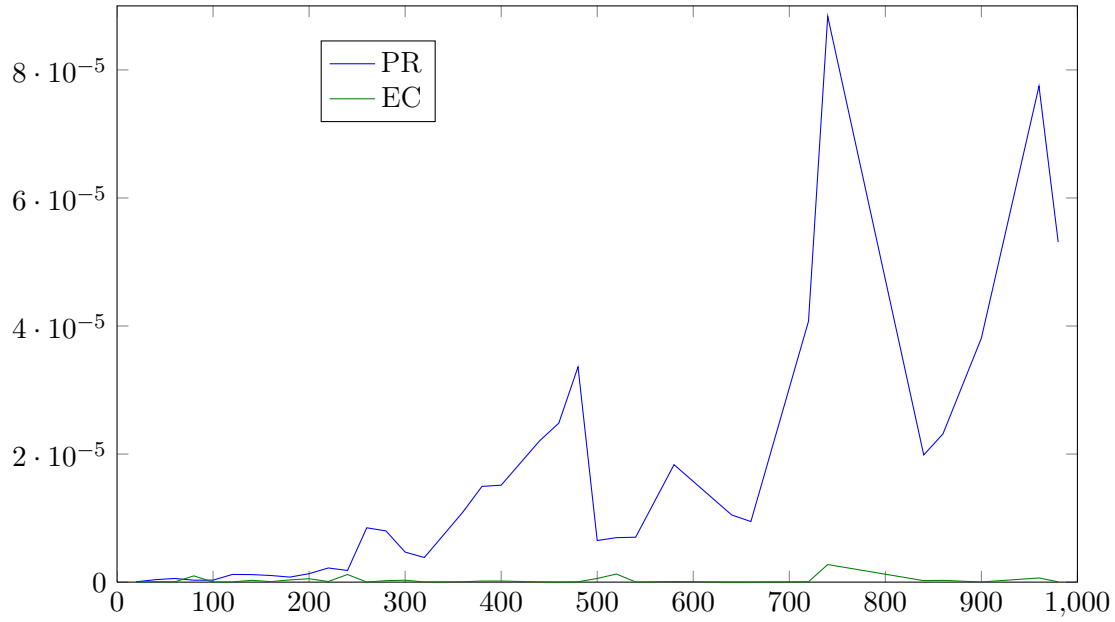


FIGURE 3.11 – Ecart sur l'optimum en fonction du nombre de variables

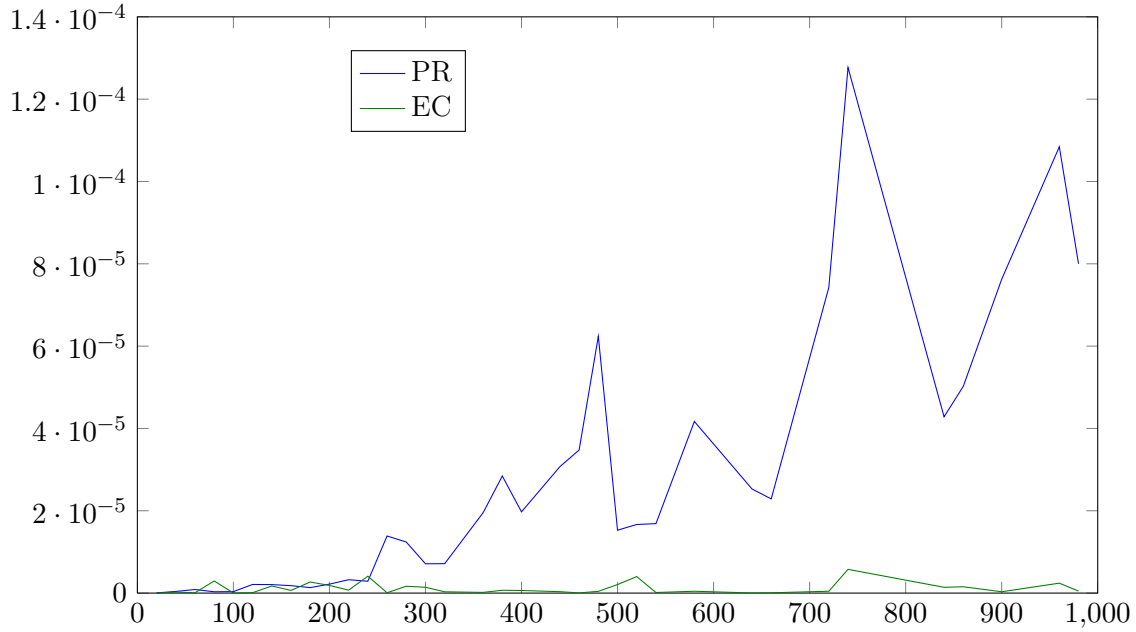


FIGURE 3.12 – Ecart relatif à la valeur objectif pour les deux méthodes en fonction du nombre de variables

3.3.5 Fonctions de Shekel

La fonction de Shekel est une fonction coût rationnelle [Bersini 1996] définie théoriquement par :

$$\min_{x\in[0,10]^m} \; -\sum_{i=1}^{m} \frac{1}{c_i + \sum_{j=1}^{n}(x_j - a_{ij})^2}, \tag{3.145}$$

où $A = (a_{i,j})_{i,j} \in \mathbb{R}^{n\times m}$ et $c = (c_i) \in \mathbb{R}^n$. Dans ce paragraphe, nous allons étudier deux types de fonctions de Shekel pour lesquelles le minimum global est connu.

Test n°1. Pour cet exemple, on utilise :

$$A \;=\; \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{pmatrix} \tag{3.146}$$

et

$$c \;=\; \begin{pmatrix} 0.1 & 0.2 & 0.2 & 0.4 & 0.4 & 0.6 & 0.3 & 0.7 & 0.5 & 0.5 \end{pmatrix}. \tag{3.147}$$

Ce problème a donc 4 variables d'optimisation et 10 fractions rationnelles. Les résultats théoriques pour cette fonction sont disponibles dans [Ali 2005]. Ainsi, pour $n = 5, 7, 10$, le minimum global est $x^\star = (4, 4, 4, 4)^\top$. En traçant la fonction pour les deux premières variables, on remarque que la fonction possède plusieurs minima locaux et un minimum global au voisinage duquel la fonction coût subit de rapides changements de valeurs. Entre ces minima, le gradient de le fonction coût a des variations presque nulles. La figure 3.13 représente cette fonction de Shekel et ses lignes de niveau. La solution par Epigraphes Creux n'a pas donné, sur cet

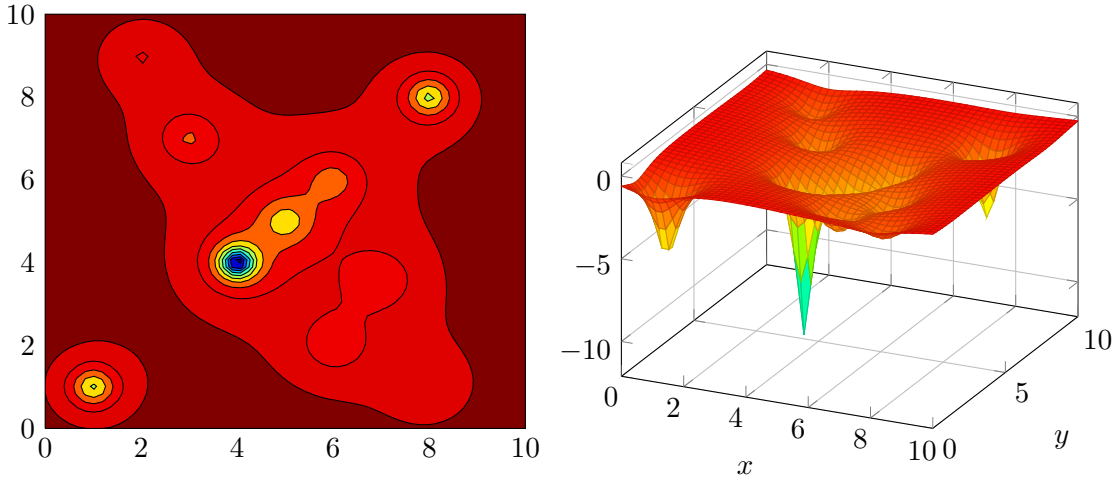


FIGURE 3.13 – Fonction de Shekel et lignes de niveau pour les deux premières variables dans le domaine $[0, 10] \times [0, 10]$

M	Méthode	$x_1^\star$	$x_2^\star$	$x_3^\star$	$x_4^\star$	$f(x^\star)$	Tps (s)	Relaxations	nb iter.
5	PR	4.00	4.00	4.00	4.00	-10.1	20.6	4	N.A.
	BARON	4.00	4.00	4.00	4.00	-10.1	N.A.	N.A.	7
	Théorique	4.00	4.00	4.00	4.00	-10.1	N.A.	N.A.	N.A.
7	PR	4.00	4.00	3.99	4.00	-10.4	256.8	4	N.A.
	BARON	4.00	4.00	4.00	3.99	-10.4	N.A.	N.A.	11
	Théorique	4.00	4.00	4.00	4.00	-10.4	N.A.	N.A.	N.A.
10	PR	4.00	4.00	3.99	4.00	-10.5	2506.6	5	N.A.
	BARON	4.00	4.00	4.00	3.99	-10.5	N.A.	N.A.	81
	Théorique	4.00	4.00	4.00	4.00	-10.5	N.A.	N.A.	N.A.

TABLE 3.3 – Résultats obtenus pour la première fonction de Shekel.

exemple, de résultats exploitables car les premières relaxations n'ont pas permis d'extraire des solutions exploitables. Dans le cas de le Programmation Rationnelle, toutes les solutions sont certifiées. L'ensemble des résultats obtenus est détaillé dans le tableau 3.3. On remarque que la Programmation Rationnelle fournit l'optimum global théorique. Cependant, ses temps de calcul explosent avec l'augmentation de l'ordre de relaxation. Un phénomène similaire est observé avec le logiciel BARON pour lequel le nombre d'itérations augmente significativement. Nous concluons donc que, malgré des temps de calcul importants, nous avons mis en avant un exemple où la Programmation Rationnelle apporte des résultats satisfaisants tandis que les Epigraphes Creux s'avèrent inopérants.

Test n°2. Dans ce deuxième exemple, on considère la matrice A et le vecteur c donnés dans l'annexe E. Ce problème possède 10 variables d'optimisation et 30 fractions rationnelles. Les résultats théoriques pour ce cas sont disponibles dans [Ali 2005]. En traçant le cas $n = 2$, on remarque que la fonction possède plusieurs minima locaux et un minimum global autour duquel la fonction objectif varie de manière importante. La figure 3.14 représente cette fonction de Shekel et ses lignes de niveau.

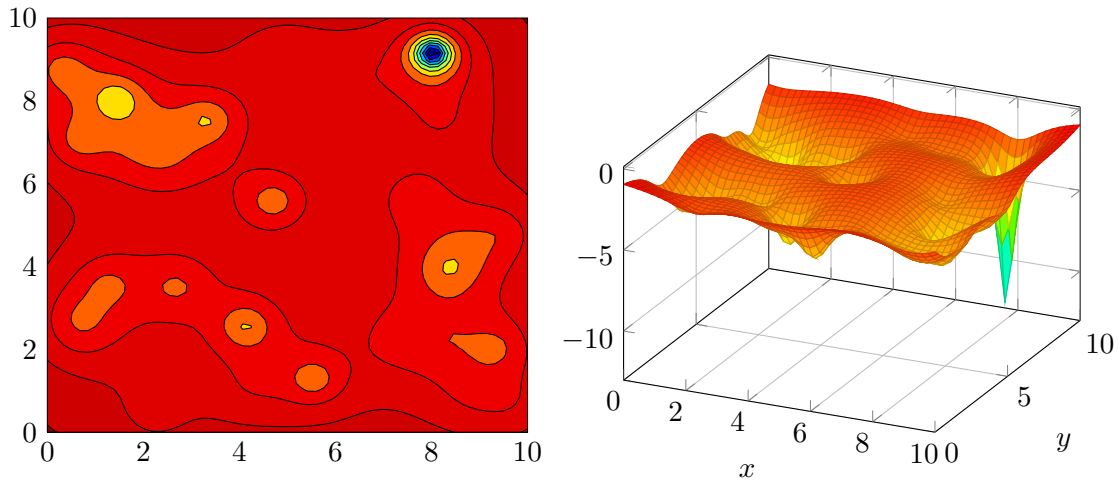


FIGURE 3.14 – Fonction de Shekel et lignes de niveau pour les deux premières variables dans le domaine $[0, 10] \times [0, 10]$

Seule la Programmation Rationnelle a fourni des résultats pour ce problème, car le trop grand nombre de variables et de fractions ne permet pas aux Epigraphes Creux de calculer une solution en des temps raisonnables. En effet, les premières relaxations n'ont pu renvoyer

de solutions exploitables et les autres se sont avérées trop gourmandes en temps de calcul. Pour la Programmation Rationnelle, la solution :

$$x^\star = (8.02 \;\; 9.15 \;\; 5.11 \;\; 7.62 \;\; 4.56 \;\; 4.77 \;\; 2.99 \;\; 6.12 \;\; 0.734 \;\; 4.98)^\top$$

valant $f(x^\star) = -10.2$ à l'objectif, est certifiée à la $2^{\text{ème}}$ relaxation en 239 secondes. Le logiciel BARON fournit la même solution en 165187 itérations et atteint le maximum de temps de calcul alloué. Ces deux méthodes atteignent l'optimum global donné par [Ali 2005]. On constate donc que la Programmation Rationnelle fonctionne avec succès sur ce cas complexe tandis que les Epigraphes Creux se sont avérés, comme pour l'exemple précédent, inutilisables.

3.3.6 Bilan

A l'aide d'exemples simples, nous avons montré que la méthode de Programmation Rationnelle pouvait être appliquée avec succès sur plusieurs problèmes. Ces exemples ont aussi montré que la méthode des Epigraphes Creux était plus complexe à mettre en œuvre. Cette tendance s'est confirmée sur les exemples issus de la communauté d'optimisation globale. Enfin, bien que ce point mérite d'être approfondi, on notera que la méthode de Programmation Rationnelle peut favorablement se comparer avec une méthode reconnue du domaine.

3.4 Conclusion

Après avoir rappelé quelques principes de la théorie de la mesure, nous avons commencé ce chapitre par une introduction à l'optimisation polynomiale via la théorie des moments. Nous avons ensuite exposé notre contribution qui consiste à étendre la théorie précédente au cas de la minimisation d'une somme de fractions rationnelles. Cette extension a pu être réalisée au moyen de deux méthodes : les Epigraphes Creux et la Programmation Rationnelle. Pour chacune des deux méthodes, nous avons prouvé l'équivalence avec le problème original ainsi que la convergence de la suite de solutions issue de la hiérarchie SDP associée vers la solution du problème initial. Enfin, nous avons testé les deux approches sur plusieurs problèmes tests. Ces tests ont souligné le net avantage de la Programmation Rationnelle au détriment des Epigraphe Creux. Nous concluons donc que la Programmation Rationnelle est la meilleure des deux méthodes pour résoudre globalement la minimisation, sous contraintes, d'une somme de fractions rationnelles. C'est cette méthode qui sera donc utilisée pour nos applications en vision artificielle.

4

Application à la vision multivues

Sommaire

4.1 Applications de l'optimisation polynomiale 84
4.1.1 Rappels de géométrie projective 84
4.1.2 Minimisation des distorsions projectives 87
4.1.3 Bilan 98
4.1.4 Estimation de la matrice fondamentale 102
4.2 Application de la programmation rationnelle à la triangulation multivues 124
4.2.1 Rappels de reconstruction 3D multivues 124
4.2.2 Etat de l'art 133
4.2.3 Applications à la triangulation multivues 143
4.2.4 Conclusions 157

Le but de ce chapitre est de présenter les applications des techniques d'optimisation polynomiale et rationnelle vues dans le chapitre précédent. Rappelons que ces techniques permettent de certifier numériquement un ou plusieurs minima globaux. Dans la première application nous appliquerons la technique de minimisation polynomiale sur le problème de minimisation des distorsions projectives induites par des homographies de rectification. Pour mettre en œuvre cette méthode, il est nécessaire de disposer d'un critère polynomial ou rationnel dont on peut extraire les racines avec une précision numérique suffisante. Ainsi, après une discussion sur les critères de minimisation existants, nous proposerons une série de nouvelles fonctions objectifs s'exprimant comme une somme de polynômes ou de fractions rationnelles. Puis, sur plusieurs séries de tests, nous départagerons ces critères à l'aide de l'optimisation polynomiale.

L'autre avantage de la méthode de minimisation polynomiale est de pouvoir s'affranchir d'une estimée initiale. Ce point revêt un intérêt particulier car, lorsque les contraintes sont fortement non linéaires, il est aussi difficile de trouver un point dans le domaine des contraintes K que de résoudre le problème $\min_{x \in K} f(x)$ lui-même. Cette limitation fait que, pour un grand nombre de problèmes rencontrés en vision par ordinateur, on préfère minimiser la fonction coût sans contrainte, puis projeter, lorsque cela est possible, sur K. Cependant, ce processus

n'est concevable en pratique que si on peut projeter $x^\star$ sur K par une suite de manipulations algébriques simples et rapides (p. ex. diagonalisation, décomposition en valeurs singulières...). L'archétype de ce type de problème est l'estimation de la matrice fondamentale que nous détaillerons dans le §4.1.4. Bien que plusieurs applications de l'optimisation polynomiale sur ce problème ont déjà été réalisées, nous montrerons comment réaliser une mise en œuvre plus adéquate de cette technique. Enfin, nous comparerons les résultats obtenus sur des cas tests réels et simulés. Afin de réaliser cette comparaison, nous proposerons une procédure d'évaluation basée sur la qualité de la reconstruction 3D.

L'objet de la dernière application concerne la reconstruction 3D multivues. Le but de l'étape de reconstruction 3D est de déterminer les coordonnées, dans un repère donné, de chaque point d'un nuage 3D. Mathématiquement, ce problème est un problème d'estimation au sens des moindres carrés à trois inconnues. Chaque résidu est une fraction rationnelle dont le degré est lié à la modélisation utilisée. Pour notre application, nous utiliserons deux modélisations différentes. La première ne tient pas compte des distorsions liées à l'objectif de la caméra. Il résulte que les degrés des numérateurs et des dénominateurs de ses résidus sont égaux à 2. Dans la deuxième application, qui prend en compte de telles distorsions, les résidus ont des numérateurs et des dénominateurs de degrés égaux à 6. Pour ces deux problèmes, nous testerons la méthode de Programmation Rationnelle sur des cas réels et simulés.

4.1 Applications de l'optimisation polynomiale

4.1.1 Rappels de géométrie projective

L'objectif de ce paragraphe est de rappeler les fondements de la géométrie épipolaire ainsi que le principe de calcul de la matrice fondamentale et de la rectification d'images.

4.1.1.1 Géométrie épipolaire

La géométrie épipolaire définit la géométrie entre deux caméras issues d'un banc stéréoscopique ou entre deux positions d'une caméra mobile. Nous définissons ainsi, sans perte de généralité, la géométrie épipolaire comme la géométrie entre deux caméras de centre optique C et C' fournissant une paire d'images notée $\mathcal{I}$ et $\mathcal{I}'$. Notons que dans le reste de ce document, $\mathcal{I}$ et $\mathcal{I}'$ feront aussi bien référence aux images elles-mêmes qu'au plan de l'espace projectif qui les supporte. Comme le montre la figure 4.1, lorsqu'un point Q de la scène est visible simultanément par les deux caméras, il se projette dans les images $\mathcal{I}$ et $\mathcal{I}'$ en deux points notés respectivement q et q'. En outre, les points Q, C et C' définissent un plan dans l'espace 3D appelé **plan épipolaire** et noté $\mathcal{P}$. l'intersection entre $\mathcal{P}$ et l'image $\mathcal{I}$ (resp. $\mathcal{I}'$) définit une ligne $\mathsf{l}_{\mathsf{q}'}$ (resp. l'_{q}) dans $\mathcal{I}$ (resp $.\mathcal{I}'$), appelée **ligne épipolaire gauche** (resp. **ligne épipolaire droite**). Ainsi, par construction, on constate que q doit appartenir à la ligne épipolaire gauche $\mathsf{l}_{\mathsf{q}'}$ et q' à la ligne épipolaire droite l'_{q}. Cette contrainte particulière est appelée **contrainte épipolaire** : pour tout point q dans l'image $\mathcal{I}$, son stéréo-correspondant q' dans $\mathcal{I}'$ doit appartenir à l'_{q}, la ligne épipolaire associée à q. Inversement, q doit appartenir à $\mathsf{l}_{\mathsf{q}'}$, la ligne épipolaire associée à q'. Ces relations sont décrites algébriquement par les équations :

$$\mathsf{q}' \in \mathsf{l}'_{\mathsf{q}} \quad \Rightarrow \quad \mathsf{q}'^{\top}\mathsf{l}'_{\mathsf{q}} = 0 \tag{4.1}$$

$$\mathsf{q} \in \mathsf{l}_{\mathsf{q}'} \quad \Rightarrow \quad \mathsf{q}^{\top}\mathsf{l}_{\mathsf{q}'} = 0. \tag{4.2}$$

Le point clé de la géométrie épipolaire repose sur le fait que la relation entre les coordonnées rétiniennes d'un point q de $\mathcal{I}$ et sa ligne épipolaire correspondante l'_{q} dans $\mathcal{I}'$ est linéaire dans l'espace projectif $\mathbb{P}^2$. Par conséquent, cette correspondance peut être représentée par

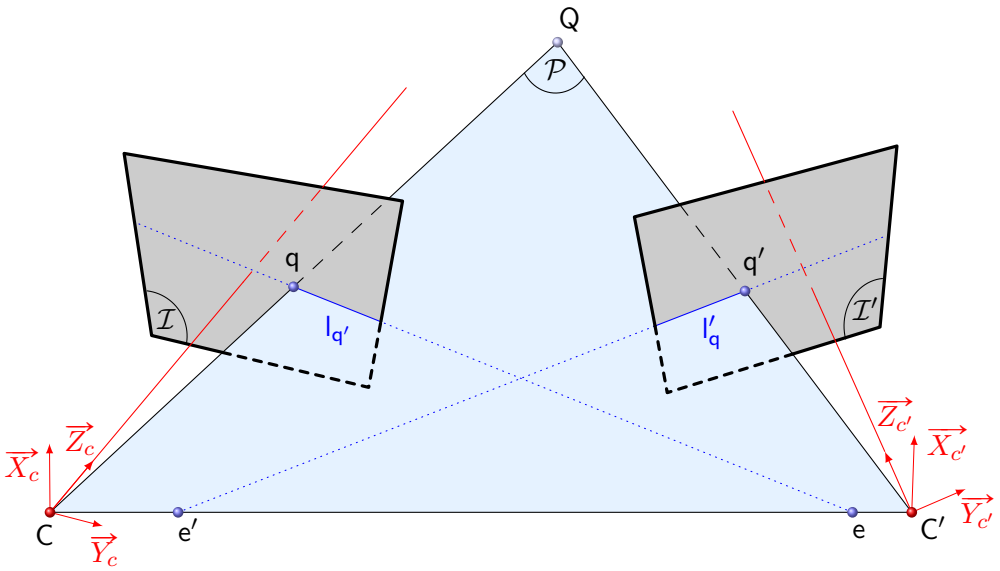


FIGURE 4.1 – Relations géométriques entre deux caméras **sténopé** (**pinhole** en anglais)

une application linéaire de $\mathbb{P}^2$ dans $\mathbb{P}^2$, c'est-à-dire une matrice 3×3 appelée **matrice fondamentale**, notée $\mathbf{F}$, et telle que :

$$l'_q = \mathbf{F}q. \tag{4.3}$$

De façon symétrique, $\mathbf{F}^\top$ décrit la correspondance entre un point q' de $\mathcal{I}'$ et la ligne épipolaire associée l'_q dans $\mathcal{I}'$, de sorte que :

$$l_{q'} = \mathbf{F}^\top q'. \tag{4.4}$$

Ainsi, en substituant (4.3) dans les contraintes épipolaires (4.1), on obtient la formulation matricielle de la contrainte épipolaire :

$$q'^\top \mathbf{F} q = 0. \tag{4.5}$$

Les épipoles e (**épipole droit**) et e' (**épipole gauche**) sont des points particuliers de $\mathcal{I}$ et $\mathcal{I}'$ vérifiant les équations suivantes :

$$\mathbf{F}e = \mathbf{F}^\top e' = 0_{\mathbb{P}^2}. \tag{4.6}$$

Géométriquement, e (resp. e') est le point d'intersection du faisceau de droites formé par les lignes épipolaires $l_{q'}$ (resp. l'_q) lorsque q' (resp. q) décrit $\mathcal{I}'$ (resp. $\mathcal{I}$). Ils impliquent donc que le rang de $\mathbf{F}$ est inférieur ou égal à 2. Il est cependant considéré, en pratique, comme étant égal à 2. En outre, puisque $\mathbf{F}$ est définie comme une application linéaire de $\mathbb{P}^2$ dans $\mathbb{P}^2$, elle est déterminée à un facteur près. En additionnant la contrainte de rang à cette dernière remarque, on conclut que $\mathbf{F}$ ne dépend, sous son paramétrage minimal, que de sept coefficients. Pour plus de détails sur la géométrie épipolaire et la matrice fondamentale, le lecteur pourra se référer à [Hartley 2003].

4.1.1.2 Rectification d'images

La rectification d'image est une étape importante en stéréovision car elle autorise un gain de rapidité lors de la recherche de stéréo-correspondants. En effet, après rectification, la recherche de points appariés se fait le long de droites épipolaires horizontales, communes aux

deux images. Les méthodes de rectification que nous étudions dans ce document sont fondées sur la connaissance a priori de la géométrie épipolaire. Sous cette hypothèse, la rectification d'image peut être vue comme un processus visant à transformer la géométrie épipolaire standard (4.1) en une géométrie particulière. Dans cette géométrie les lignes épipolaires sont parallèles avec l'axe horizontal de l'image (cf. Figure 4.2). La matrice fondamentale $\mathbf{F}_0$ corres-

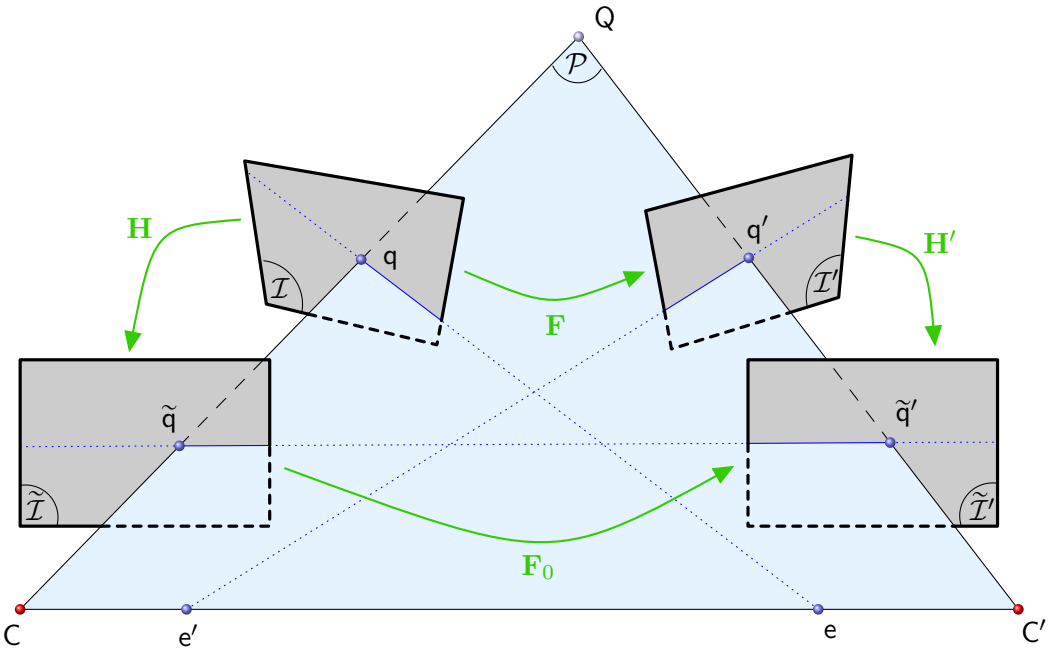


FIGURE 4.2 – Opération de rectification

pondant à cette géométrie, appelée **matrice fondamentale rectifiée**, est alors de la forme :

$$\mathbf{F}_0 \triangleq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.7}$$

En pratique, ce passage d'une géométrie épipolaire à une autre est accompli en appliquant aux images $\mathcal{I}$ et $\mathcal{I}'$ des homographies $\mathbf{H}$ et $\mathbf{H}'$ dont le rôle consiste à projeter les épipoles e et e' à l'infini. Plus précisément, $\mathbf{H}$ et $\mathbf{H}'$ transforment les points originaux q et q' de $\mathcal{I}$ et $\mathcal{I}'$ en deux points $\tilde{\mathsf{q}}$ et $\tilde{\mathsf{q}}'$ des images rectifiées $\tilde{\mathcal{I}}$ et $\tilde{\mathcal{I}}'$ par :

$$\tilde{\mathsf{q}} = \mathbf{H}\mathsf{q} \tag{4.8}$$
$$\tilde{\mathsf{q}}' = \mathbf{H}'\mathsf{q}'. \tag{4.9}$$

Ainsi, en exprimant (4.5) dans la géométrie rectifiée, il vient :

$$\tilde{\mathsf{q}}'^{\top}\mathbf{F}_0\tilde{\mathsf{q}} = \mathsf{q}'^{\top}\underbrace{\mathbf{H}'^{\top}\mathbf{F}_0\mathbf{H}}_{=\mathbf{F}}\mathsf{q} = 0, \tag{4.10}$$

ce qui conduit à la factorisation :

$$\mathbf{F} = \mathbf{H}'^{\top}\mathbf{F}_0\mathbf{H}. \tag{4.11}$$

$\mathbf{F}$ étant supposée connue, les homographies satisfaisant (4.11) sont appelées des **homographies compatibles**. Le but de la rectification consiste donc à déterminer les coefficients de ces homographies de sorte que les déformations entre images originales et rectifiées soient les plus petites possible.

4.1.2 Minimisation des distorsions projectives

4.1.2.1 Etat de l'art

Comme nous l'avons vu, l'objectif de la rectification est de calculer une paire de transformations projectives, $\mathbf{H}$ et $\mathbf{H}'$ dont les coefficients, calculés à partir de la connaissance de la géométrie épipolaire, maximisent les similitudes entre images originales et images rectifiées. Cependant, ces homographies ne sont pas uniques et plusieurs approches existent dans la littérature. Les solutions actuelles s'articulent autour de deux étapes : l'extraction d'homographies $\mathbf{H}$ et $\mathbf{H}'$ particulières à partir de la matrice fondamentale $\mathbf{F}$ et la minimisation des déformations projectives induites par celles-ci. Les approches proposées [Devernay 1997, Loop 1999, Hartley 1999, Gluckman 2001, Mallon 2005] diffèrent alors en fonction de la manière dont sont extraites les homographies ainsi que des manipulations qui leurs sont appliquées afin de minimiser les distorsions. L'étape d'extraction joue un rôle clé car c'est elle qui va déterminer la structure de $\mathbf{H}$ et $\mathbf{H}'$, et, par conséquent, le type de manipulations qu'il est possible de leur appliquer. Cependant cette étape est assujettie à deux contraintes. D'une part, $\mathbf{H}$ et $\mathbf{H}'$ doivent nécessairement envoyer les épipoles e et e' à l'infini. C'est la contrainte dite de **projection**. D'autre part, les homographies doivent vérifier l'équation (4.11). C'est la contrainte dite de **compatibilité**. Notons que, en raison de la forme de $\mathbf{F}_0$, cette dernière équation ne porte que sur les deux dernières lignes de $\mathbf{H}$ et $\mathbf{H}'$ et laisse donc la première indéterminée. On voit alors apparaître la dichotomie qui va séparer les deux grandes catégories de méthodes. Dans la première classe de méthodes dites **symétriques**, $\mathbf{H}$ et $\mathbf{H}'$ ont la même forme, assurant à la fois la contrainte de projection et celle de compatibilité. Dans la deuxième classe de méthodes, dites **non symétriques**, la forme de $\mathbf{H}$ (resp. $\mathbf{H}'$) est fixée de manière à garantir la contrainte de projection et celle de $\mathbf{H}'$ (resp. $\mathbf{H}$) est déduite de $\mathbf{H}$ (resp. $\mathbf{H}'$) via la contrainte de compatibilité. Cependant, ces deux familles ont un point commun : l'utilisation des coefficients de la première ligne laissés libres par (4.11) comme des paramètres d'optimisation afin de minimiser les déformations induites par $\mathbf{H}$ et $\mathbf{H}'$.

Les méthodes symétriques. Les deux approches les plus classiques rencontrées dans cette catégorie sont la méthode de **Loop et Zang** [Loop 1999] et celle de **Devernay** [Devernay 1997]. La première résulte d'une démarche géométrique : $\mathbf{H}$ et $\mathbf{H}'$ sont fixées comme étant des compositions de transformations jouant un rôle précis. Ainsi, pour cette technique, $\mathbf{H}$ s'écrit :

$$\mathbf{H} = \underbrace{\begin{pmatrix} h^s_{11} & h^s_{12} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\triangleq \mathbf{H}_s} \underbrace{\begin{pmatrix} h^r_{11} & h^r_{12} & 0 \\ h^r_{21} & h^r_{22} & h^r_{23} \\ 0 & 0 & 1 \end{pmatrix}}_{\triangleq \mathbf{H}_r} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h^p_{31} & h^p_{32} & 1 \end{pmatrix}}_{\triangleq \mathbf{H}_p} \tag{4.12}$$

où

- $\mathbf{H}_p$ est une matrice, dite **de projection**, qui déplace l'épipole e à l'infini
- $\mathbf{H}_r$ est une matrice, dite **de similarité**, dont le rôle consiste à aligner les droites épipolaires des deux images pour les rendre colinéaires.
- $\mathbf{H}_s$ est une matrice dite **de cisaillement** (**shearing** en anglais) qui permet la minimisation des déformations (situées principalement sur l'axe des abscisses) générées par $\mathbf{H}_r$.

A l'exception de $\mathbf{H}_p$, tous les coefficients de chaque matrice sont déterminés par des formules algébriques où interviennent les paramètres de $\mathbf{F}$ ainsi que la taille de l'image rectifiée. Les paramètres h^s_{11} et h^s_{12} sont, quant à eux, déterminés à l'aide de la méthode des régions de confiance (cf. Algorithme 5). La méthode étant symétrique, $\mathbf{H}'$ est calculée de manière analogue à l'aide des matrices $\mathbf{H}'_p$, $\mathbf{H}'_r$ et $\mathbf{H}'_s$ de forme identique à $\mathbf{H}_p$, $\mathbf{H}_r$ et $\mathbf{H}_s$. Notons que la

contrainte de compatibilité est garantie par les matrices $\mathbf{H}_r$ et $\mathbf{H}'_r$ qui, bien qu'ayant la même forme, ont des coefficients liés par (4.11).

La deuxième méthode [Devernay 1997] est basée sur une approche plus mathématique puisqu'elle découle directement d'une décomposition en valeurs singulières de $\mathbf{F}$. Cette dernière matrice étant de rang 2, une de ses valeurs singulières est nécessairement nulle. Après normalisation par la plus grande des deux valeurs singulières restantes, on peut montrer que :

$$\mathbf{F} = \begin{pmatrix} \mathsf{e}' & \mathbf{u}_1 & \mathbf{u}_2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sigma \end{pmatrix} \begin{pmatrix} \mathsf{e}^\top \\ \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \end{pmatrix} \tag{4.13}$$

$$\tag{4.14}$$

$$\Leftrightarrow \quad \mathbf{F} = \underbrace{\begin{pmatrix} \mathsf{e}' & \mathbf{u}_1 & \sqrt{\sigma}\mathbf{u}_2 \end{pmatrix}}_{\triangleq \mathbf{H}_0'^\top} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \underbrace{\begin{pmatrix} \mathsf{e}^\top \\ \sqrt{\sigma}\mathbf{v}_2^\top \\ -\mathbf{v}_1^\top \end{pmatrix}}_{\triangleq \mathbf{H}_0} \tag{4.15}$$

où $\sigma \in \mathbb{R}^{+*}$ et e et e' sont les épipoles de $\mathcal{I}$ et $\mathcal{I}'$. Par construction, $\mathbf{H}_0$ et $\mathbf{H}'_0$ vérifient donc les contraintes de projection et de compatibilité. Il est ensuite possible de composer $\mathbf{H}_0$ et $\mathbf{H}'_0$ avec des matrices

$$\mathbf{T} = \begin{pmatrix} a_1 & b_1 & c_1 \\ 0 & e & f \\ 0 & h & i \end{pmatrix} \quad \text{et} \quad \mathbf{T}' = \begin{pmatrix} a_2 & b_2 & c_2 \\ 0 & e & f \\ 0 & h & i \end{pmatrix} \tag{4.16}$$

de sorte que $\mathbf{T}\mathbf{H}_0$ et $\mathbf{T}'\mathbf{H}'_0$ vérifient toujours la contrainte de compatibilité. Les coefficients de $\mathbf{T}$ et $\mathbf{T}'$ sont alors minimisés afin de réduire les déformations des images rectifiées.

Les méthodes non symétriques. Cette catégorie est essentiellement composée des différentes déclinaisons de la méthode proposée dans [Hartley 1999]. Comme la méthode de **Loop et Zang**, elle est basée sur une décomposition géométrique de $\mathbf{H}$ visant d'abord à satisfaire la contrainte de projection :

$$\mathbf{H} = \mathbf{G}\mathbf{R}\left[\mathbf{t}\right]_\times \tag{4.17}$$

où

- $\mathbf{G}$ est une matrice qui déplace un épipole $\bar{\mathsf{e}}$ se trouvant sur l'axe horizontal (c.-à-d. de coordonnées $(\mathsf{e}_u, 0, 1)^\top$) à l'infini (i.e en $(\mathsf{e}_u, 0, 0)^\top$)
- $\mathbf{R}$ est une matrice de rotation et $\left[\mathbf{t}\right]_\times$ une matrice de translation dont la composée amène l'épipole e sur l'axe des abscisses : $\mathbf{R}\left[\mathbf{t}\right]_\times \mathsf{e} = \bar{\mathsf{e}}$.

$\mathbf{H}'$ est ensuite déduite de $\mathbf{H}$ via la minimisation d'un critère non-linéaire ayant pour but de garantir la contrainte de compatibilité. Une contribution importante de cette méthode réside dans l'introduction d'un outil théorique permettant de contrôler localement la création et la destruction de pixels. Cet outil est le jacobien de la transformation induite par $\mathbf{H}$. Cette transformation, notée $\mathrm{f}_\mathbf{H}$, associe un pixel de l'image initiale q à un pixel $\tilde{\mathsf{q}}$ de l'image rectifiée. Elle est définie par :

$$\begin{aligned} \mathrm{f}_\mathbf{H} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2 \\ \mathsf{q} &\rightarrow \tilde{\mathsf{q}} = \Psi_2\left(\mathbf{H}\begin{pmatrix} \mathsf{q} \\ 1 \end{pmatrix}\right) \end{aligned} \tag{4.18}$$

où Ψ_n est l'opérateur de projection canonique de $\mathbb{P}^n$ dans $\mathbb{R}^n$:

$$\begin{aligned} \Psi_n : \mathbb{P}^n &\rightarrow \mathbb{R}^n \\ (x_1, \ldots, x_n, s)^\top &\rightarrow \left(\frac{x_1}{s}, \ldots, \frac{x_n}{s}\right)^\top. \end{aligned} \tag{4.19}$$

Par conséquent, dans un voisinage $\mathcal{V}_{\mathsf{q}}$ de q, on a :

$$\|\mathsf{f}_{\mathbf{H}}(\mathsf{q}+\mathsf{h}) - \mathsf{f}_{\mathbf{H}}(\mathsf{q})\| \leqslant |||\mathrm{J}_{\mathbf{H}}(\mathsf{q})|||.\|\mathsf{h}\| \quad \forall\, \mathsf{h} \in \mathcal{V}_{\mathsf{q}} \tag{4.20}$$

où $|||\ |||$ est la norme d'opérateur matriciel. Ainsi, on constate que contrôler $|||\mathrm{J}_{\mathbf{H}}(\mathsf{q})|||$ permet de contrôler l'accroissement de $\mathsf{f}_{\mathbf{H}}$ et donc localement la création et la destruction de pixels. En outre, remarquons qu'une transformation **rigide**, c'est-à-dire la composée d'une rotation et d'une translation, ne crée ou ne supprime localement aucun pixel. En combinant ces deux arguments, on comprend que le but de ces méthodes consiste à faire en sorte que $\mathbf{H}$ et $\mathbf{H}'$ soient les plus proches possibles d'une transformation rigide en imposant des contraintes numériques sur la norme de leurs jacobiens. Ce contrôle, comme nous le détaillerons dans la suite, se fait au moyen des paramètres laissés libres par (4.11). Ainsi, dans [Hartley 1999], seul $\mathrm{J}_{\mathbf{G}}(\mathsf{q})$ doit être contraint afin d'éviter des distorsions dans l'image rectifiée. Le développement de cette méthode, proposée dans [Gluckman 2001] et reprise dans [Mallon 2005], consiste à directement fixer $\mathbf{H}$ comme une projection de e (et non de $\bar{\mathsf{e}}$) à l'infini :

$$\mathbf{H} \triangleq \begin{bmatrix} 1 & 0 & 0 \\ h_{21} & 1 & 0 \\ h_{31} & 0 & 1 \end{bmatrix} \text{ avec } \begin{cases} h_{21} \triangleq -\dfrac{\mathsf{e}_v}{\mathsf{e}_u} \\ h_{31} \triangleq -\dfrac{1}{\mathsf{e}_u} \end{cases}. \tag{4.21}$$

Puisque (4.11) laisse libre la première ligne de $\mathbf{H}'$, on a :

$$\mathbf{H}' \triangleq \begin{bmatrix} 1 & 0 & 0 \\ h'_{21} & h'_{22} & h'_{23} \\ h'_{31} & h'_{32} & h'_{33} \end{bmatrix}. \tag{4.22}$$

Ensuite, en substituant (4.21) et (4.22) dans la contrainte de compatibilité (4.11), on obtient :

$$\begin{bmatrix} h_{21}h'_{31} - h_{31}h'_{21} & h'_{31} & -h'_{21} \\ h_{21}h'_{32} - h_{31}h'_{22} & h'_{32} & -h'_{22} \\ h_{21}h'_{33} - h_{31}h'_{23} & h'_{33} & -h'_{23} \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \tag{4.23}$$

où les f_{ij} sont les coefficients de la matrice fondamentale $\mathbf{F}$. On remarque alors que les paramètres h'_{ij} s'expriment directement en fonction des f_{ij}. Mais, dans l'hypothèse fortement probable où les valeurs de $\mathbf{F}$ sont perturbées (en raison d'appariements entachés d'erreurs), il est plus judicieux de calculer une solution de (4.23) au sens des moindres carrés. Ainsi, (4.23) est transformée en un système linéaire $AX = 0$ où le vecteur d'inconnues X est formé par les paramètres h'_{ij}. On obtient ensuite, par un algorithme classique de résolution numérique (p.e. **D**écomposition en **V**aleurs **S**ingulières ou **D.V.S.**), une matrice $\mathbf{H}'$ vérifiant exactement la contrainte de compatibilité. Reprenant l'idée vue dans la méthode de **Loop et Zang**, $\mathbf{H}$ et $\mathbf{H}'$ sont composées avec des matrices de **cisaillement** :

$$\mathbf{A} \triangleq \begin{bmatrix} a_{11} & a_{12} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{A}' \triangleq \begin{bmatrix} a'_{11} & a'_{12} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.24}$$

afin de minimiser les déformations projectives qu'elles induisent. La méthode s'achève alors en imposant, via les paramètres a_{ij} et a'_{ij}, des contraintes numériques sur les jacobiens $\mathrm{J}_{\mathbf{H}_a}$ et $\mathrm{J}_{\mathbf{H}'_{a'}}$ des matrices $\mathbf{H}_a \triangleq \mathbf{AH}$ et $\mathbf{H}'_{a'} \triangleq \mathbf{A}'\mathbf{H}'$. Ces contraintes numériques sont, en pratique, imposées au moyen d'un problème de minimisation. Selon la zone à rectifier, un ensemble de $2n$ points, dits de **contrôle** et noté $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1..n}$, est choisi dans $\mathcal{I} \times \mathcal{I}'$. Puis, on maîtrise

simultanément la création et la perte de pixels dans les voisinages de ces points en résolvant les problèmes :

$$\min_{a} \sum_{i=1}^{n} \mathcal{C}(\mathrm{J}_{\mathbf{H}_a}(\mathsf{q}_i)) \tag{4.25}$$

$$\min_{a'} \sum_{i=1}^{n} \mathcal{C}(\mathrm{J}_{\mathbf{H}'_{a'}}(\mathsf{q}'_i)) \tag{4.26}$$

avec $\mathcal{C}$ un critère permettant de contrôler la norme d'opérateur des jacobiens. Il est important de remarquer que ces deux problèmes n'ont pas la même complexité car $\mathbf{H}'_{a'}$ est plus dense que $\mathbf{H}_a$. Cette technique est résumée sous forme de diagramme sur la figure 4.3.

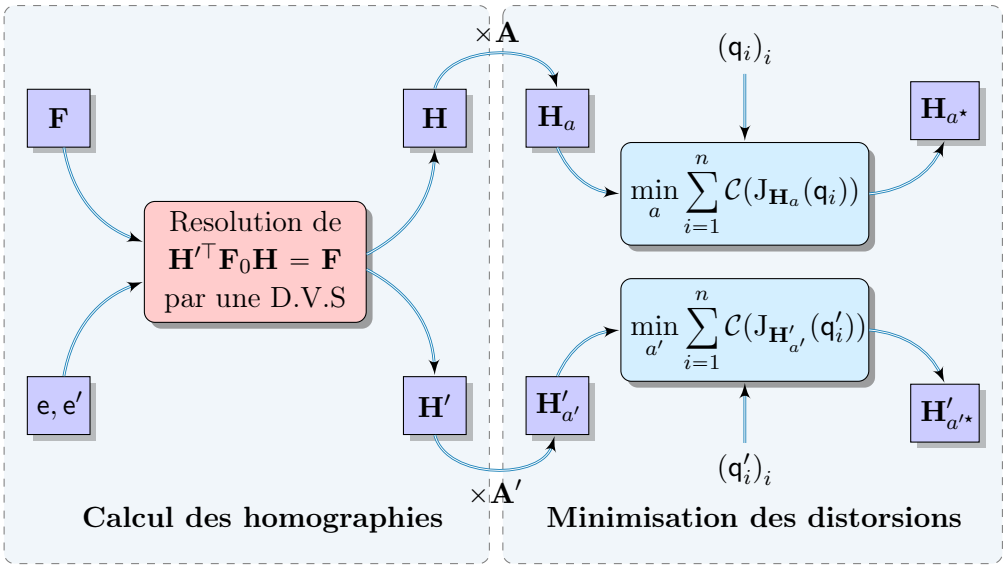


FIGURE 4.3 – Diagramme récapitulatif des méthodes non-symétriques de type **Mallon**

4.1.2.2 Critères de minimisation

Après une présentation du critère étudié dans [Mallon 2005], nous proposons dans les paragraphes suivants plusieurs critères polynomiaux et rationnels permettant de contrôler la rigidité des déformations induites par les homographies de rectification. Par souci de lisibilité, nous adopterons pour la suite du document les terminologies suivantes : les jacobiens $\mathrm{J}_{\mathbf{H}_a}(\mathsf{q}_i)$ et $\mathrm{J}_{\mathbf{H}'_{a'}}(\mathsf{q}'_i)$ seront notés :

$$\mathrm{J}_i \triangleq \begin{bmatrix} \mathrm{j}_{11} & \mathrm{j}_{12} \\ \mathrm{j}_{21} & \mathrm{j}_{22} \end{bmatrix} \quad \text{et} \quad \mathrm{J}'_i \triangleq \begin{bmatrix} \mathrm{j}'_{11} & \mathrm{j}'_{12} \\ \mathrm{j}'_{21} & \mathrm{j}'_{22} \end{bmatrix}. \tag{4.27}$$

Enfin, $\mathcal{C}$ étant le même dans les problèmes (4.25) et (4.26), nous ne définirons nos nouveaux critères que pour J_i.

Critères existants. Le premier critère, initialement introduit dans [Hartley 1999] puis repris dans [Gluckman 2001], est défini comme étant égal au déterminant de J_i et s'écrit : $\det(\mathrm{J}_i) = \mathrm{j}_{11}\mathrm{j}_{22} - \mathrm{j}_{12}\mathrm{j}_{21}$. Cependant, en prenant par exemple $\mathrm{j}_{11} = 4, \mathrm{j}_{22} = \frac{1}{2}, \mathrm{j}_{12} = 3$ et $\mathrm{j}_{21} = \frac{1}{3}$, on peut aisément constater qu'il est possible que $\det(\mathrm{J}_i) = 1\ \forall i = 1 \dots n$, alors même que la transformation projective $\mathbf{H}$ crée ou détruit localement des pixels. Le déterminant seul

n'impose donc pas suffisamment de contraintes pour obliger $\mathbf{H}$ à être proche d'une transformation rigide. Puisque toute transformation rigide a ses valeurs singulières égales à 1, il est proposé dans [Mallon 2005] d'imposer à la place du déterminant que :

$$\sigma_1(\mathrm{J}_i) = \sigma_2(\mathrm{J}_i) = 1 \tag{4.28}$$

où $\sigma_1(\mathrm{J}_i)$ et $\sigma_2(\mathrm{J}_i)$ sont les valeurs singulières de J_i. Le critère $\mathcal{C}_M$ est donc dans ce cas défini par :

$$\mathcal{C}_M(\mathrm{J}_i) \quad = \quad (\sigma_1(\mathrm{J}_i) - 1)^2 + (\sigma_2(\mathrm{J}_i) - 1)^2 . \tag{4.29}$$

Cependant, les auteurs n'explicitent pas algébriquement ce critère. Ceci les conduit alors à ne minimiser ce dernier qu'avec la méthode sans dérivées de Nelder-Mead, initialisée à partir de $a = (1, 0)^\top$.

Nouveaux critères équivalents. Le premier critère que nous proposons est basé sur l'écriture algébrique de $\mathcal{C}_M$. En effet, le polynôme caractéristique de $\mathrm{J}_i^\top \mathrm{J}_i$ est donné par :

$$\begin{aligned} \chi_{\mathrm{J}_i^\top \mathrm{J}_i}(s) &= s^2 - \mathrm{tr}(\mathrm{J}_i^\top \mathrm{J}_i)s + \det(\mathrm{J}_i^\top \mathrm{J}_i) && (4.30)\\ &= s^2 - \|\mathrm{J}_i\|_F^2 s + \det(\mathrm{J}_i)^2 && (4.31)\\ & && (4.32) \end{aligned}$$

Puisque $\sigma_1(\mathrm{J}_i)$ et $\sigma_2(\mathrm{J}_i)$ sont les racines de $\chi_{\mathrm{J}_i^\top \mathrm{J}_i}$, on a :

$$\sigma_1(\mathrm{J}_i) = \frac{\|\mathrm{J}_i\|_F^2 + \sqrt{\Delta_i}}{2} \quad \text{et} \quad \sigma_2(\mathrm{J}_i) = \frac{\|\mathrm{J}_i\|_F^2 - \sqrt{\Delta_i}}{2}, \tag{4.33}$$

avec :

$$\begin{aligned} \Delta_i &\triangleq \|\mathrm{J}_i\|_F^4 - 4\det(\mathrm{J}_i)^2 \\ &= (\mathrm{j}_{11}^2 - \mathrm{j}_{22}^2)^2 + (\mathrm{j}_{12}^2 - \mathrm{j}_{21}^2)^2 + 2(\mathrm{j}_{11}\mathrm{j}_{12} + \mathrm{j}_{21}\mathrm{j}_{22})^2 + 2(\mathrm{j}_{11}\mathrm{j}_{22} + \mathrm{j}_{12}\mathrm{j}_{21})^2. \end{aligned} \tag{4.34}$$

Par conséquent :

$$(\sigma_1(\mathrm{J}_i) - 1)^2 + (\sigma_2(\mathrm{J}_i) - 1)^2 \quad = \quad \frac{1}{2}\left(\Delta_i + (\|\mathrm{J}_i\|_F^2 - 2)^2\right). \tag{4.35}$$

Ainsi, nous définissons notre premier critère $\mathcal{C}_1$, de degré 4 en a, par :

$$\boxed{\mathcal{C}_1(J_i) \quad \triangleq \quad \Delta_i + (\|\mathrm{J}_i\|_F^2 - 2)^2.} \tag{4.36}$$

Observons qu'une autre façon de satisfaire (4.28) consiste à vérifier le système :

$$\begin{cases} \sigma_1(\mathrm{J}_i)^2 \sigma_2(\mathrm{J}_i)^2 = 1 \\ \sigma_1(\mathrm{J}_i)^2 + \sigma_2(\mathrm{J}_i)^2 = 2 \end{cases}. \tag{4.37}$$

Or ce système se ré-écrit :

$$\begin{cases} \det(\mathrm{J}_i)^2 = 1 \\ \|\mathrm{J}_i\|_F^2 = 2 \end{cases}. \tag{4.38}$$

Par conséquent, en reformulant ce système comme un problème de minimisation, nous définissons notre second critère $\mathcal{C}_2$, de degré 4 en a, par :

$$\boxed{\mathcal{C}_2(J_i) \quad \triangleq \quad \left(\det(\mathrm{J}_i)^2 - 1\right)^2 + \left(\|\mathrm{J}_i\|_F^2 - 2\right)^2.} \tag{4.39}$$

Nouveaux critères relaxés. Les critères précédents ont été proposés en raisonnant uniquement sur la forme générale de J_i et J'_i. Mais, en utilisant leurs formes particulières, nous verrons que ces critères sont sur-déterminés et qu'il est possible de les relaxer. La première étape consiste donc à écrire explicitement J_i et J'_i. Selon (4.18) et (4.24), $\mathrm{f}_{\mathbf{H_a}}$ et $\mathrm{f}_{\mathbf{H'_{a'}}}$ sont données par :

$$\mathrm{f}_{\mathbf{H_a}}(u,v) \;=\; \begin{bmatrix} \dfrac{(a_{11}+a_{12}h_{21})u + a_{12}v}{h_{31}u+1} \\ \\ \dfrac{h_{21}u+v}{h_{31}u+1} \end{bmatrix} \tag{4.40}$$

et

$$\mathrm{f}_{\mathbf{H'_{a'}}}(u',v') \;=\; \begin{bmatrix} \dfrac{(a'_{11}+a'_{12}h'_{21})u' + a'_{12}h'_{23}v'}{h'_{31}u'+h'_{32}v'+h'_{33}} \\ \\ \dfrac{h'_{21}u'+h'_{22}v'+h'_{23}}{h'_{31}u'+h'_{32}v'+h'_{33}} \end{bmatrix}. \tag{4.41}$$

Ainsi, leurs Jacobiennes évaluées aux points de contrôle s'écrivent :

$$\mathrm{J}_i = \begin{bmatrix} \dfrac{a_{11}+\alpha_i a_{12}}{\beta_i^2} & \dfrac{a_{12}}{\beta_i} \\ \\ \dfrac{\alpha_i}{\beta_i^2} & \dfrac{1}{\beta_i} \end{bmatrix} \text{ et } \mathrm{J}'_i = \begin{bmatrix} \dfrac{\lambda_i a'_{11}+\mu_i a'_{12}}{\rho_i^2} & \dfrac{\nu_i a'_{12}-\xi_i a'_{11}}{\rho_i^2} \\ \\ \dfrac{\lambda_i}{\rho_i^2} & \dfrac{\nu_i}{\rho_i^2} \end{bmatrix} \tag{4.42}$$

avec :

$$\begin{array}{rcl} \alpha_i &=& h_{21} - v_i h_{31} \\ \beta_i &=& h_{31}u_i + 1 \\ \lambda_i &=& (h'_{21}h'_{32} - h'_{31}h'_{22})v'_i + (h'_{21}h'_{33} - h'_{31}h'_{23}) \\ \mu_i &=& h'_{32}v'_i + h'_{33} \\ \nu_i &=& (h'_{22}h'_{33} - h'_{23}h'_{32}) - (h'_{21}h'_{32} - h'_{31}h'_{22})u'_i \\ \xi_i &=& h'_{32}u'_i \\ \rho_i &=& h'_{31}u'_i + h'_{32}v_i + h'_{33}. \end{array} \tag{4.43}$$

Par conséquent, on remarque que les dernières lignes de J_i et J'_i sont constantes. A la lumière de ce résultat, analysons plus finement le critère $\mathcal{C}_1$. Il est défini par la somme de deux termes positifs Δ_i et $(\|\mathrm{J}_i\|^2 - 2)^2$. Leur rôle respectif est d'assurer, à l'optimum, d'une part que $\sigma_1(\mathrm{J}_i) = \sigma_2(\mathrm{J}_i)$ et d'autre part que $\sigma_1(\mathrm{J}_i)$ ou $\sigma_2(\mathrm{J}_i)$ soit égal à 1. Développons la condition $\sigma_1(\mathrm{J}_i) = \sigma_2(\mathrm{J}_i)$ à l'aide de l'écriture analytique de J_i et J'_i :

$$\Delta_i = 0$$

$$\Leftrightarrow \quad \begin{cases} \mathrm{j}_{11}^2 = \mathrm{j}_{22}^2 \\ \\ \mathrm{j}_{12}^2 = \mathrm{j}_{21}^2 \\ \mathrm{j}_{11}\mathrm{j}_{12} + \mathrm{j}_{21}\mathrm{j}_{21} = 0 \\ \mathrm{j}_{11}\mathrm{j}_{21} + \mathrm{j}_{12}\mathrm{j}_{22} = 0 \end{cases} \tag{4.44}$$

$$\Rightarrow \quad \mathrm{j}_{11}^2 = \frac{1}{\beta_i^2} \;\text{ et }\; \mathrm{j}_{12}^2 = \frac{\alpha_i^2}{\beta_i^4}. \tag{4.45}$$

Ainsi, dû au fait que les dernières lignes de J_i et J'_i sont constantes, si $\sigma_1(\mathrm{J}_i) = \sigma_2(\mathrm{J}_i)$ alors leurs valeurs sont nécessairement fixées par :

$$\begin{aligned} \|\mathrm{J}_i\|_F^2 &= \sigma_1(\mathrm{J}_i)^2 + \sigma_2(\mathrm{J}_i)^2 = \mathrm{j}_{11}^2 + \mathrm{j}_{12}^2 + \mathrm{j}_{21}^2 + \mathrm{j}_{22}^2 = \frac{2}{\beta_i^2} + \frac{2\alpha_i^2}{\beta_i^4} \\ \|\mathrm{J}'_i\|_F^2 &= 2\frac{\lambda_i^2 + \nu_i^2}{\rho_i^4}. \end{aligned} \tag{4.46}$$

Il y a donc une incohérence à vouloir minimiser $(\|\mathrm{J}_i\|^2 - 2)^2$ dans $\mathcal{C}_1$ puisque rien n'assure, a priori, qu'il puisse être ramené à 0 à l'optimum. Il est par conséquent légitime de vouloir ignorer ce terme, ce qui nous conduit ainsi à définir un troisième critère $\mathcal{C}_\Delta$, de degré 4 en a, par :

$$\boxed{\mathcal{C}_\Delta(J_i) \triangleq \|\mathrm{J}_i\|_F^4 - 4\det(\mathrm{J}_i)^2.} \tag{4.47}$$

Afin de diminuer le degré du critère précédent, il est possible de définir un autre critère qui, lorsqu'il est minimisé, fournit à l'optimum une solution spéciale de (4.44). Pour définir un tel critère, remarquons qu'en appliquant le théorème de Wielandt-Hoffman [Golub 1996] à $\mathrm{J}_i, \mathrm{J}_i(a) - I_2$, on obtient l'inégalité suivante :

$$(\sigma_1(J_i) - 1)^2 + (\sigma_2(J_i) - 1)^2 \leqslant \|\mathrm{J}_i - I_2\|_F^2. \tag{4.48}$$

Ainsi, en minimisant le terme $\|\mathrm{J}_i - I_2\|_F^2$, il est possible de contraindre les valeurs singulières de J_i. En outre, puisque le sous-espace des matrices symétriques $\mathcal{S}_2(\mathbb{R})$ et celui des matrices anti-symétriques $\mathcal{A}_2(\mathbb{R})$ sont en somme directe et orthogonaux pour le produit scalaire Frobenius, on a :

$$\|\mathrm{J}_i - I_2\|_F^2 = \| \underbrace{\frac{(\mathrm{J}_i + \mathrm{J}_i^\top)}{2}}_{\triangleq \mathcal{S}_i} - I_2\|_F^2 + \| \underbrace{\frac{(\mathrm{J}_i - \mathrm{J}_i^\top)}{2}}_{\triangleq \mathcal{A}_i} \|_F^2. \tag{4.49}$$

Explicitons alors les matrices $\mathcal{S}_i - I_2$ et $\mathcal{A}_i$:

$$\mathcal{S}_i - I_2 = \begin{bmatrix} \mathrm{j}_{11} - 1 & \dfrac{\mathrm{j}_{12} + \mathrm{j}_{21}}{2} \\ \dfrac{\mathrm{j}_{12} + \mathrm{j}_{21}}{2} & \mathrm{j}_{22} - 1 \end{bmatrix} \qquad \mathcal{A}_i = \begin{bmatrix} 0 & \dfrac{\mathrm{j}_{12} - \mathrm{j}_{21}}{2} \\ -\dfrac{\mathrm{j}_{12} - \mathrm{j}_{21}}{2} & 0 \end{bmatrix}. \tag{4.50}$$

Par conséquent, fixer tous les paramètres de $\mathcal{S}_i$ permet de déterminer complètement J_i (ce qui n'est pas le cas de $\mathcal{A}_i$) . Plus précisément :

$$\|\mathcal{S}_i - I_2\|_F^2 = 0 \quad \Rightarrow \quad \begin{cases} \mathrm{j}_{11} = 1 \\ \mathrm{j}_{12} = -\mathrm{j}_{21} \\ \mathrm{j}_{22} = 1 \end{cases}. \tag{4.51}$$

Les solutions de ce système sont aussi solutions de (4.44). Cependant, j_{22} est indépendant des paramètres d'optimisation et peut, a priori, ne pas être égal à 1 à l'optimum. Pour J_i et J'_i cette condition, $\mathrm{j}_{22} = 1$, se développe en :

$$\frac{1}{u_i h_{31} + 1} = 1 \quad \text{et} \quad \frac{(h'_{22}h'_{33} - h'_{23}h'_{32}) - (h'_{21}h'_{32} - h'_{31}h'_{22})u'_i}{h'_{31}u'_i + h'_{32}v_i + h'_{33}} = 1 \qquad \forall i = 1 \dots n. \tag{4.52}$$

La diminution du degré de $\mathcal{C}_\Delta$ se fait donc en relachant encore les contraintes imposées sur les jacobiens J_i et J'_i, au risque que $\mathbf{H}_{a^\star}$ et $\mathbf{H}'_{a'^\star}$ soient aberrantes. On définit cependant notre quatrième critère $\mathcal{C}_\mathcal{S}$, de degré deux en a, par :

$$\boxed{\mathcal{C}_\mathcal{S}(J_i) \triangleq \|\frac{(J_i + J_i^\top)}{2} - I_2\|_F^2.} \tag{4.53}$$

Nouveau critère basé sur le conditionnement Une autre façon de contrôler numériquement J_i consiste à stabiliser son conditionnement numérique κ. Il existe plusieurs manières de définir κ qui sont fonctions de la norme utilisée. Mais, afin de disposer d'un critère algébrique, nous choisissons d'utiliser la norme de Frobenius $\| \|_F$. Ainsi, $\kappa_F(\mathrm{J}_i)^2$ s'écrit :

$$\kappa_F(\mathrm{J}_i)^2 \quad = \quad \frac{\mathrm{tr}(\mathrm{J}_i\mathrm{J}_i^\top)^2}{\det \mathrm{J}_i{}^2}. \tag{4.54}$$

Nous définissons ainsi le cinquième critère $\mathcal{C}_\kappa$ par :

$$\boxed{\mathcal{C}_\kappa(J_i) \quad \triangleq \quad \frac{\mathrm{tr}(\mathrm{J}_i\mathrm{J}_i^\top)^2}{\det \mathrm{J}_i{}^2}.} \tag{4.55}$$

Cependant, étant donnée la forme particulière de J_i et J'_i, on a :

$$\det(\mathrm{J}_i) = \frac{a_{11}}{\beta_i^3} \qquad \text{et} \qquad \det(\mathrm{J}'_i) = \frac{\lambda_i(\nu_i - \xi_i)a'_{11}}{\rho_i^4}. \tag{4.56}$$

Ainsi les fonctions coûts $\sum_{i=1}^n \kappa(\mathrm{J}_i)$ et $\sum_{i=1}^n \kappa(\mathrm{J}'_i)$ peuvent être mises sous la forme d'un quotient de polynômes :

$$\sum_{i=1}^n \mathcal{C}_\kappa(\mathrm{J}_i) = \frac{1}{a_{11}^2}\sum_{i=1}^n \left[(a_{11}+\alpha_i a_{12})^2 + (\beta_i a_{12})^2 + (\alpha_i^2+\beta_i^2)\right]$$
$$\sum_{i=1}^n \mathcal{C}_\kappa(\mathrm{J}'_i) = \frac{1}{a'^2_{11}}\sum_{i=1}^n \frac{\rho_i^4}{\lambda_i^2(\nu_i-\xi_i)^2}\left[(\lambda_i a'_{11}+\nu_i a'_{12})^2 + (\nu_i a'_{12}-\xi_i a'_{11})^2 + (\lambda_i^2+\nu_i^2)\right].$$

Bilan. Nous avons défini en tout cinq critères : trois polynômes de degré 4, un de degré 2, et une fraction rationnelle. Ces critères sont rappelés sur le tableau 4.1.

Critère proposé	Formulation mathématique
$\mathcal{C}_1(J_i)$	$\Delta_i + (\|\mathrm{J}_i\|_F^2 - 2)^2$
$\mathcal{C}_2(J_i)$	$\big(\det(\mathrm{J}_i)^2 - 1\big)^2 + \big(\|\mathrm{J}_i\|_F^2 - 2\big)^2$
$\mathcal{C}_\Delta(J_i)$	$\|\mathrm{J}_i\|_F^4 - 4\det(\mathrm{J}_i)^2$
$\mathcal{C}_\mathcal{S}(J_i)$	$\|\frac{(J_i + J_i^\top)}{2} - I_2\|_F^2$
$\mathcal{C}_\kappa(J_i)$	$\frac{\mathrm{tr}(\mathrm{J}_i\mathrm{J}_i^\top)^2}{\det \mathrm{J}_i{}^2}$

TABLE 4.1 – Bilan des critères proposés.

Les critères $\mathcal{C}_1$ (à l'aide de l'expression directe des valeurs singulières) et $\mathcal{C}_2$ (à l'aide du produit et de la somme des valeurs singulières) cherchent à satisfaire de manière directe l'équation :

$\sigma_1(\mathrm{J}_i) = \sigma_2(\mathrm{J}_i) = 1$. Nous avons ensuite démontré que, à cause de la forme particulière de J_i et J'_i, il était légitime d'ignorer le terme $(\|\mathrm{J}_i\|_F^2 - 2)^2$ dans $\mathcal{C}_1$ puisqu'il est implicitement constant. Nous avons donc défini $\mathcal{C}_\Delta$ en supprimant ce dernier, ce qui revient à simplement imposer que $\sigma_1(\mathrm{J}_i) = \sigma_2(\mathrm{J}_i)$, i.e. $\Delta_i = 0$. Afin de disposer d'un critère de moindre degré, et donc plus facilement minimisable, nous avons défini $\mathcal{C}_\mathcal{S}$ qui fournit à l'optimum une solution particulière de $\Delta_i = 0$. Nous abaissons ainsi le degré à 2, ce qui permet de résoudre directement (c.-à-d. sans méthode itérative) $\min_a \sum_{i=1}^n \mathcal{C}_\mathcal{S}(\mathrm{J}_i)$ en cherchant les zéros du gradient. Cependant, avec ce critère, nous prenons le risque d'exhiber des solutions qui ne vérifient pas $\Delta_i = 0$. Enfin, on définit un tout autre type de critère basé sur le conditionnement, κ_F, de Frobenius. Nous montrons que, grâce à la forme spéciale de J_i et J'_i, minimiser la somme des conditionnements κ_F revient à minimiser une seule fraction rationnelle.

4.1.2.3 Résultats numériques.

Le but de cette section est d'évaluer la qualité des cinq critères proposés. Mais, puisque $\mathcal{C}_M$ est égal à $\mathcal{C}_1$, la minimisation globale de $\mathcal{C}_1$ correspond à une amélioration de la méthode proposée par **Mallon**. Ainsi l'évaluation de nos critères est implicitement faite vis-à-vis de la méthode proposée dans [Mallon 2005]. Par ailleurs, le lecteur trouvera dans cette dernière référence une étude comparative de la méthode de **Mallon** avec certaines méthodes détaillées dans l'état de l'art. Cependant, afin d'obtenir de meilleurs résultats finaux, nous évaluerons la matrice fondamentale $\mathbf{F}$ à l'aide d'une estimation itérative (la minimisation du critère de Sampson présenté dans la section suivante) plus précise que celle utilisée dans [Mallon 2005]. Ceci expliquera la différence avec les résultats présentés dans cette référence. Dans un premier paragraphe, nous introduirons les différents critères d'évaluations utilisés. Les suivants seront entièrement consacrés à l'étude comparative. Enfin, nous conclurons dans le dernier paragraphe sur l'efficacité de chacun des critères.

Critères d'évaluations. Plusieurs critères (cf. [Mallon 2005, Loop 1999, Wu 2007]) ont été étudiés afin de caractériser la déformation induite par $\mathbf{H}$ et $\mathbf{H}'$ sur les images rectifiées. Le premier, appelé critère **d'orthogonalité** [Loop 1999, Wu 2007] et noté E_o, évalue l'invariance des angles entre les images originales et les images rectifiées. Il est défini, à partir des milieux des côtés de l'image initiale $(\mathsf{p}_i)_{i=1..4}$, par la relation :

$$E_o = \cos^{-1}\left(\frac{\langle \mathrm{f}_\mathbf{H}(\mathsf{p}_2) - \mathrm{f}_\mathbf{H}(\mathsf{p}_4) \mid \mathrm{f}_\mathbf{H}(\mathsf{p}_1) - \mathrm{f}_\mathbf{H}(\mathsf{p}_3)\rangle}{\|\mathrm{f}_\mathbf{H}(\mathsf{p}_2) - \mathrm{f}_\mathbf{H}(\mathsf{p}_4)\|.\|\mathrm{f}_\mathbf{H}(\mathsf{p}_1) - \mathrm{f}_\mathbf{H}(\mathsf{p}_3)\|}\right).$$

Idéalement, l'orthogonalité E_o doit être égale à $90°$. Le second, introduit dans [Mallon 2005], est appelé **le rapport d'aspect**. Il est noté E_a et évalue l'invariance de la largeur et de la hauteur entre les images originales et les images rectifiées. Il est défini à partir des quatre coins $(\mathsf{q}_i)_{i=1..4}$ de l'image initiale :

$$E_a = \frac{\|\mathrm{f}_\mathbf{H}(\mathsf{q}_2) - \mathrm{f}_\mathbf{H}(\mathsf{q}_4)\|}{\|\mathrm{f}_\mathbf{H}(\mathsf{q}_1) - \mathrm{f}_\mathbf{H}(\mathsf{q}_3)\|}.$$

Idéalement, l'aspect ratio E_a doit être égal à 1. La figure 4.4 fournit une interprétation géométrique de ces deux critères. Enfin, le critère (qui sera détaillé dans la section suivante) de distance moyenne des points de contrôle aux droites épipolaires, noté E_r, permet d'évaluer la qualité de la géométrie épipolaire estimée à partir de $\mathbf{H}'\mathbf{F}_0\mathbf{H}$.

Afin d'évaluer la précision des cinq critères proposés, nous les expérimentons sur plusieurs types d'images. Tout d'abord, nous utilisons les images tests issues d'un banc stéréoscopique. Ensuite, dans le but d'évaluer la robustesse des critères vis-à-vis de l'ampleur du mouvement, nous comparons leurs performances en fonction de l'angle entre les vues d'un même objet.

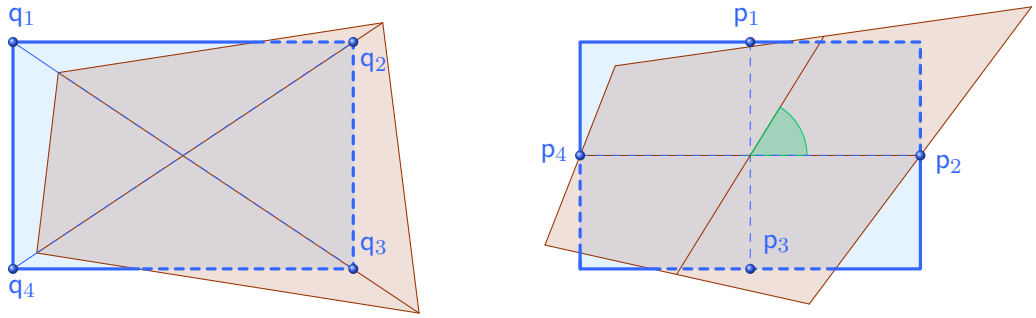


FIGURE 4.4 – Distorsions avant et après rectification : à gauche, les déformations dues à une modification d'orthogonalité, et à droite, les déformations dues à une modification du rapport d'aspect.

Enfin, afin de se placer dans des cas plus défavorables et d'approcher ainsi les limites de leurs domaines de validité, nous minimisons les cinq critères pour des images tests quelconques de la communauté vision artificielle.

Evaluation sur des images stéréoscopiques. Cette première analyse comparative est effectuée sur les images de la publication [Mallon 2005]. Elles ont été acquises par un banc de stéréovision (convergent) classique dans des contextes différents : intérieur, extérieur...L'ensemble de ces images sont fournies dans l'Annexe A. Pour chacun des couples d'images, un ensemble de points appariés, identique à ceux utilisés dans [Mallon 2005] est fourni. Le tableau 4.2 présente les résultats obtenus sur cette série d'images l'aide les cinq critères $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_\Delta, \mathcal{C}_\mathcal{S}$ et $\mathcal{C}_\kappa$. La première conclusion concerne la comparaison entre les critères $\mathcal{C}_1, \mathcal{C}_\Delta$ et $\mathcal{C}_\mathcal{S}$. En effet, ces trois critères sont issus de la même approche : le premier est l'écriture algébrique du critère de Mallon, alors que les deux autres en sont des relaxations plus ou moins fortes. Ainsi, nous vérifions que notre hypothèse sur la sur-détermination de $\mathcal{C}_1$ se vérifie, puisque ce dernier est, quel que soit l'échantillon, moins précis que $\mathcal{C}_\Delta$ et $\mathcal{C}_\mathcal{S}$ (aussi bien sur l'orthogonalité que sur le rapport d'aspect). Il se rélève, en outre, être d'une précision très approximative sur l'échantillon **Roof**, soulignant ainsi que toutes les valeurs singulières des jacobiens ne peuvent pas être simultanément égales à 1. Sur tous les échantillons, $\mathcal{C}_\Delta$ apparaît comme le plus précis des trois, tandis que $\mathcal{C}_\mathcal{S}$ lui est au mieux équivalent. Ce dernier point met en avant que dans quatre cas (**Arch**, **Boxes**, **Yard** et **Drive**), la solution particulière calculée à partir de $\mathcal{C}_\mathcal{S}$ est très proche de la solution optimale. Cependant, sur les cas **Roof** et **Slate**, cette solution particulière n'est pas, malgré sa simplicité de calcul, suffisamment proche de la solution optimale (i.e. il doit exister des jacobiens tels que $j_{22} \neq 1$). La deuxième constatation est que le critère $\mathcal{C}_2$, bien qu'équivalent à $\mathcal{C}_1$, est toujours mal conditionné (autrement dit, le rapport entre les coefficients de $\mathcal{C}_2$ est très important). Ceci rend sa minimisation globale difficile et on constate ainsi qu'il ne fournit jamais de solution convenable. Enfin, $\mathcal{C}_\kappa$ se revèle être, avec $\mathcal{C}_\Delta$, le critère le plus précis. Il semble, de plus, être le plus robuste car il est plus performant que $\mathcal{C}_\Delta$ sur un cas défavorable comme **Roof**. Pour conclure, remarquons qu'aucun critère ne modifie la précision de la géométrie épipolaire finale.

Précision en fonction de l'amplitude du mouvement Afin de confirmer nos premières conclusions, les critères ont été utilisés sur des images non stéréoscopiques issues du mouvement d'une caméra autour d'un objet. Ce mouvement de rotation est effectué dans un plan, le centre optique de la caméra se déplace donc dans un plan. Les images font partie de la série **House** (9 vues) mise à disposition par le Visual Geometry Group d'Oxford et sont four-

nies dans l'Annexe B. Pour cette série, un fichier de points appariés ainsi que les matrices de projection perspectives sont également disponibles. Pour ce test, nous avons voulu étudier l'influence de l'angle entre deux déplacements de la caméra sur la précision de chacun des critères. Pour cela, nous avons utilisé le mode opératoire suivant : nous avons rectifié toutes les paires d'images possibles $0-1, 1-2, 2-3, \ldots$ puis, nous avons calculé chacun des critères d'évaluation pour toutes ces rectifications. Ceci nous fournit donc la moyenne des critères d'évaluation en fonction de l'angle moyen entre les centres optiques des caméras (l'angle entre deux vues pouvant être extrait des matrices de projections perspectives). Ensuite, nous renouvelons le processus pour les vues $0-2, 1-3, 2-4, \ldots$ de sorte que l'angle moyen entre les centres optiques augmente et ainsi de suite jusqu'aux paires où nous disposons d'au moins 10 appariements. La Figure 4.5 résume ce processus. Les résultats obtenus sont exposés dans

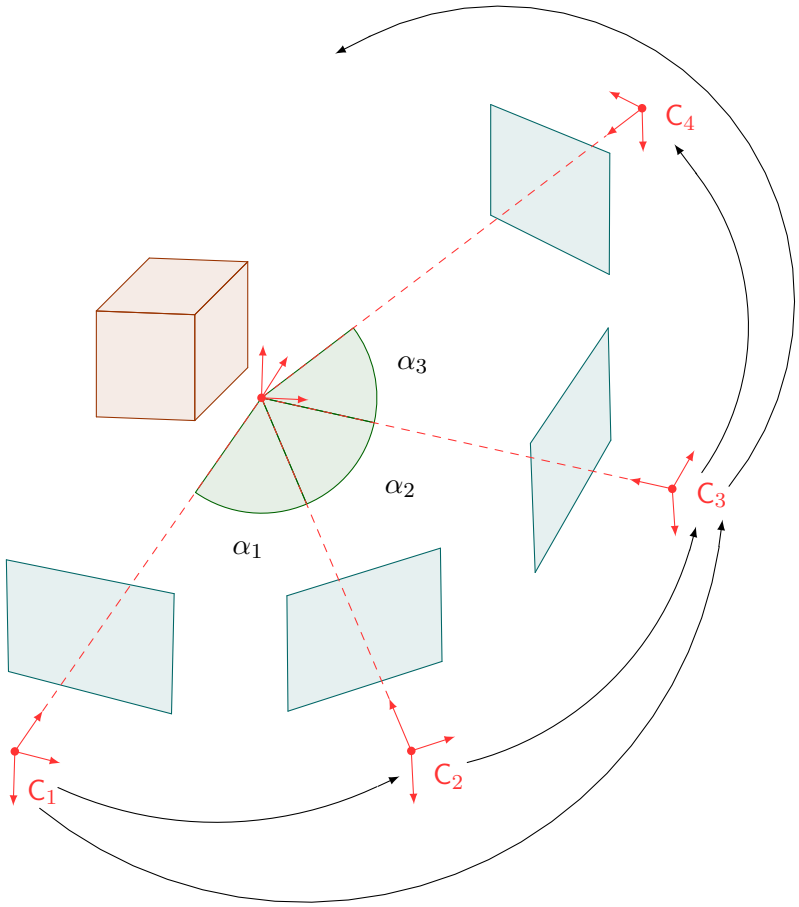


FIGURE 4.5 – Schéma explicatif du processus d'évaluation de la sensibilité des critères de rectification en fonction de l'angle moyen. Les centres optiques $(\mathsf{C}_i)_{i=1,2,3,4}$ étant coplanaires, on calcule l'angle $\alpha_i = \widehat{\mathsf{C}_i\mathsf{C}_{i+1}}$ dans le plan des abscisses. Puis, on effectue une rectification entre toutes les vues i et $i+1$ (flèches noires), puis toutes les vues i et $i+2$.

le tableau 4.3. Ils confirment les constatations faites précédemment. En effet, $\mathcal{C}_2$ n'est jamais compétitif, quel que soit l'angle. Ensuite, au delà de 22°, la performance de $\mathcal{C}_1$ se dégrade et finit par échouer sur un écart supérieur à 37.2°. Ceci semble indiquer que les valeurs singulières des jacobiens ne sont en général pas égales à 1 lorsque la précision de la géométrie épipolaire se détériore. $\mathcal{C}_\Delta$ et $\mathcal{C}_\mathcal{S}$ sont moins sensibles au décalage. Cependant, comme nous l'avons déjà observé, la solution issue de $\mathcal{C}_\mathcal{S}$ est moins robuste que celle calculée par $\mathcal{C}_\Delta$. En effet, $\mathcal{C}_\Delta$ fournit des résultats convenables jusqu'à un angle moyen de 32.8° alors que $\mathcal{C}_\mathcal{S}$ se dégrade à partir de 22°. Enfin le critère basé sur le conditionnement se révèle être le plus robuste puisque la performance des résultats est stable jusqu'à 41.4°.

Evaluation sur des images quelconques Afin d'évaluer le comportement des deux approches sur des images quelconques, les critères ont été appliqués sur des images non stéréoscopiques avec un changement d'échelle, une translation et une rotation dues à de forts déplacements de la caméra. Comme pour la série précedente, ces deux séries d'images, **Library** et **Merton**, sont fournies, avec leurs appariements, par le Visual Geometry Group d'Oxford. Ces images sont représentées dans l'Annexe C. Les résultats obtenus sont présentés dans le Tableau 4.4. Les erreurs de rectification E_o et E_a montrent que $\mathcal{C}_\Delta$ et $\mathcal{C}_\mathcal{S}$ se comportent correctement lorsque la configuration se rapproche d'une configuration stéréoscopique. Dès qu'on s'éloigne légèrement de cette configuration, seul $\mathcal{C}_\Delta$ continue à fournir des résultats acceptables. Enfin, on remarque que $\mathcal{C}_\kappa$ reste performant quel que soit la paire d'images.

4.1.3 Bilan

Cette section décrit l'application de l'optimisation polynomiale au problème de la minimisation de distorsion projective dans le processus de rectification. Pour le problème que nous considérons, il n'existait pas de critère analytique. Nous avons alors proposé cinq nouveaux critères. Pour tous ces critères, nous avons effectué, sur plusieurs séries d'images, des comparaisons numériques à l'aide d'indicateurs de performance de la littérature. Ainsi, nous avons montré que le plus performant et robuste d'entre eux se formule comme une fraction rationnelle. Cependant, pour des images issues d'un banc stéréoscopique, nous avons montré qu'il existait deux autres critères polynomiaux apportant une précision suffisante. En outre, un de ces deux critères peut être minimisé globalement sans faire appel à une méthode itérative. Bien que tous ces critères puissent être minimisés globalement par d'autres outils, l'optimisation polynomiale nous a obligé ici à formuler plusieurs critères algébriques qui constituent, en eux-mêmes, une contribution dans le domaine de la vision artificielle.

Echantillon	Critère	Orthogonalité E_o		Rapport d'aspect E_a		Erreur Rectification E_r	
		$\mathbf{H}_{a\star}$	$\mathbf{H}'_{a'\star}$	$\mathbf{H}_{a\star}$	$\mathbf{H}'_{a'\star}$	Moyenne	Ecart type
Arch	$\mathcal{C}_1$	90.055	91.424	1.001	1.021	0.197	0.227
	$\mathcal{C}_2$	101.52	105.98	1.209	1.313	0.197	0.227
	$\mathcal{C}_\Delta$	89.963	90.580	0.999	1.007	0.197	0.227
	$\mathcal{C}_\mathcal{S}$	89.996	90.982	1.000	1.013	0.197	0.227
	$\mathcal{C}_\kappa$	89.901	90.392	0.998	1.004	0.197	0.227
Slate	$\mathcal{C}_1$	87.933	88.520	0.965	0.975	0.121	0.109
	$\mathcal{C}_2$	93.600	93.194	1.065	1.057	0.121	0.109
	$\mathcal{C}_\Delta$	89.551	89.732	0.992	0.995	0.121	0.109
	$\mathcal{C}_\mathcal{S}$	88.865	89.198	0.981	0.986	0.121	0.109
	$\mathcal{C}_\kappa$	89.612	89.755	0.994	0.996	0.121	0.109
Roof	$\mathcal{C}_1$	87.371	83.644	0.955	0.970	0.798	0.887
	$\mathcal{C}_2$	95.453	62.933	1.091	0.708	0.798	0.887
	$\mathcal{C}_\Delta$	89.263	86.802	0.987	0.990	0.798	0.887
	$\mathcal{C}_\mathcal{S}$	88.411	84.714	0.973	1.026	0.798	0.887
	$\mathcal{C}_\kappa$	89.695	90.326	0.995	1.090	0.798	0.887
Boxes	$\mathcal{C}_1$	88.472	89.043	0.975	0.983	0.101	0.089
	$\mathcal{C}_2$	73.124	77.148	0.747	0.799	0.101	0.089
	$\mathcal{C}_\Delta$	90.474	90.983	1.008	1.015	0.101	0.089
	$\mathcal{C}_\mathcal{S}$	89.234	89.229	0.988	0.986	0.101	0.089
	$\mathcal{C}_\kappa$	90.547	90.919	1.009	1.014	0.101	0.089
Yard	$\mathcal{C}_1$	90.493	89.677	1.008	0.995	0.352	0.244
	$\mathcal{C}_2$	86.177	89.323	0.940	0.989	0.352	0.244
	$\mathcal{C}_\Delta$	90.281	89.977	1.005	1.000	0.352	0.244
	$\mathcal{C}_\mathcal{S}$	90.382	89.827	1.006	0.997	0.352	0.244
	$\mathcal{C}_\kappa$	90.296	89.977	1.005	1.000	0.352	0.244
Drive	$\mathcal{C}_1$	90.486	90.892	1.008	1.014	0.499	0.757
	$\mathcal{C}_2$	95.398	99.180	1.094	1.166	0.499	0.757
	$\mathcal{C}_\Delta$	90.373	90.848	1.006	1.013	0.499	0.757
	$\mathcal{C}_\mathcal{S}$	90.426	90.838	1.007	1.013	0.499	0.757
	$\mathcal{C}_\kappa$	90.364	90.794	1.006	1.012	0.499	0.757

TABLE 4.2 – Comparaison des critères proposés sur les images tests issues de [Mallon 2005].

Angle moyen	Critère	Orthogonalité moyenne		Rapport d'aspect moyen		Erreur Rectification moyenne	
		$\mathbf{H}_{a^\star}$	$\mathbf{H}'_{a'^\star}$	$\mathbf{H}_{a^\star}$	$\mathbf{H}'_{a'^\star}$	Moyenne	Ecart type
11.6°	$\mathcal{C}_1$	89.640	90.462	0.994	1.007	0.252	0.238
	$\mathcal{C}_2$	83.395	99.762	0.897	1.177	0.252	0.238
	$\mathcal{C}_\Delta$	89.799	90.296	0.997	1.005	0.252	0.238
	$\mathcal{C}_\mathcal{S}$	89.729	90.376	0.996	1.006	0.252	0.238
	$\mathcal{C}_\kappa$	89.814	90.322	0.997	1.005	0.252	0.238
16.9°	$\mathcal{C}_1$	89.468	90.638	0.991	1.017	0.374	0.340
	$\mathcal{C}_2$	82.684	67.519	0.886	0.679	0.374	0.340
	$\mathcal{C}_\Delta$	89.799	90.502	0.997	1.015	0.374	0.340
	$\mathcal{C}_\mathcal{S}$	89.634	90.669	0.994	1.017	0.374	0.340
	$\mathcal{C}_\kappa$	89.813	90.776	0.997	1.019	0.374	0.340
22°	$\mathcal{C}_1$	88.034	91.846	0.969	1.034	0.372	0.336
	$\mathcal{C}_2$	72.500	113.930	0.753	1.528	0.372	0.336
	$\mathcal{C}_\Delta$	89.535	90.432	0.992	1.010	0.372	0.336
	$\mathcal{C}_\mathcal{S}$	88.894	91.131	0.982	1.022	0.372	0.336
	$\mathcal{C}_\kappa$	89.646	90.574	0.994	1.013	0.372	0.336
27.7°	$\mathcal{C}_1$	85.998	93.825	0.939	1.069	0.409	0.397
	$\mathcal{C}_2$	63.706	120.532	0.643	1.753	0.409	0.397
	$\mathcal{C}_\Delta$	89.241	90.677	0.987	1.016	0.409	0.397
	$\mathcal{C}_\mathcal{S}$	87.937	92.091	0.968	1.039	0.409	0.397
	$\mathcal{C}_\kappa$	89.472	90.790	0.991	1.018	0.409	0.397
32.8°	$\mathcal{C}_1$	82.143	96.936	0.887	1.112	0.419	0.407
	$\mathcal{C}_2$	53.023	122.853	0.518	1.800	0.419	0.407
	$\mathcal{C}_\Delta$	89.241	90.898	0.987	1.011	0.419	0.407
	$\mathcal{C}_\mathcal{S}$	86.684	93.417	0.953	1.049	0.419	0.407
	$\mathcal{C}_\kappa$	89.627	90.816	0.994	1.009	0.419	0.407
37.2	$\mathcal{C}_1$	76.064	102.272	0.809	1.208	0.511	0.524
	$\mathcal{C}_2$	45.813	129.560	0.424	2.110	0.511	0.524
	$\mathcal{C}_\Delta$	88.927	91.319	0.982	1.018	0.511	0.524
	$\mathcal{C}_\mathcal{S}$	84.684	95.674	0.929	1.082	0.511	0.524
	$\mathcal{C}_\kappa$	89.499	91.169	0.992	1.015	0.511	0.524
40.2	$\mathcal{C}_1$	49.311	118.951	0.465	1.400	0.554	0.560
	$\mathcal{C}_2$	37.644	139.317	0.539	2.136	0.554	0.560
	$\mathcal{C}_\Delta$	89.017	91.838	0.983	0.945	0.554	0.560
	$\mathcal{C}_\mathcal{S}$	78.922	101.425	0.880	1.049	0.554	0.560
	$\mathcal{C}_\kappa$	89.536	91.031	0.992	0.932	0.554	0.560
41.4	$\mathcal{C}_1$	41.641	0.000	0.516	0.223	0.563	0.580
	$\mathcal{C}_2$	76.280	0.000	0.791	0.223	0.563	0.580
	$\mathcal{C}_\Delta$	89.554	91.574	0.992	0.867	0.563	0.580
	$\mathcal{C}_\mathcal{S}$	73.685	105.730	0.850	0.971	0.563	0.580
	$\mathcal{C}_\kappa$	89.274	90.851	0.988	0.856	0.563	0.580

TABLE 4.3 – Moyenne des critères de comparaison en fonction de l'angle moyen entre les centres optiques des caméras.

Echantillon	Critère	Orthogonalité E_o		Rapport d'aspect E_a		Erreur Rectification E_r	
		$\mathbf{H}_{a^\star}$	$\mathbf{H}'_{a'^\star}$	$\mathbf{H}_{a^\star}$	$\mathbf{H}'_{a'^\star}$	Moyenne	Ecart type
Library	$\mathcal{C}_1$	96.917	91.398	1.127	1.032	0.321	0.285
	$\mathcal{C}_2$	105.422	84.895	1.241	0.921	0.321	0.285
	$\mathcal{C}_\Delta$	90.806	90.168	1.014	1.010	0.321	0.285
	$\mathcal{C}_\mathcal{S}$	93.467	90.924	1.062	1.023	0.321	0.285
	$\mathcal{C}_\kappa$	90.317	90.331	1.005	1.013	0.321	0.285
Merton1	$\mathcal{C}_1$	79.294	0.000	0.831	0.566	0.205	0.160
	$\mathcal{C}_2$	68.269	0.000	0.685	0.566	0.205	0.160
	$\mathcal{C}_\Delta$	85.147	82.620	0.920	0.878	0.205	0.160
	$\mathcal{C}_\mathcal{S}$	82.101	73.462	0.871	0.764	0.205	0.160
	$\mathcal{C}_\kappa$	86.564	86.508	0.944	0.942	0.205	0.160
Merton2	$\mathcal{C}_1$	90.636	91.162	1.011	1.028	0.309	0.239
	$\mathcal{C}_2$	98.575	112.69	1.153	1.506	0.309	0.239
	$\mathcal{C}_\Delta$	90.499	91.468	1.008	1.033	0.309	0.239
	$\mathcal{C}_\mathcal{S}$	90.557	91.472	1.009	1.034	0.309	0.239
	$\mathcal{C}_\kappa$	90.500	91.610	1.008	1.035	0.309	0.239
Merton3	$\mathcal{C}_1$	179.66	180.00	1.245	1.425	0.303	0.223
	$\mathcal{C}_2$	69.517	180.00	0.851	1.425	0.303	0.223
	$\mathcal{C}_\Delta$	83.000	87.002	0.939	0.986	0.303	0.223
	$\mathcal{C}_\mathcal{S}$	114.54	92.368	1.137	1.080	0.303	0.223
	$\mathcal{C}_\kappa$	86.868	87.867	0.949	1.003	0.303	0.223

TABLE 4.4 – Comparaison des critères proposés sur des images quelconques.

4.1.4 Estimation de la matrice fondamentale

4.1.4.1 Etat de l'art

Comme nous l'avons précédemment décrit dans le paragraphe §4.1.1.1, déterminer le mouvement entre deux points de vue via la géométrie épipolaire consiste à estimer la matrice fondamentale $\mathbf{F}$ à partir d'un ensemble de n points appariés (ou appariements) $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots n}$. Ce problème a été abondamment étudié dans la littérature et plusieurs approches existent :

- méthodes itératives ou non-itératives
- seulement à partir de sept appariements
- en imposant la contrainte de rang dans l'optimisation ou directement dans le paramétrage de $\mathbf{F}$
- en normalisant ou non les appariements $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots n}$

...

Ces approches, souvent décrites par des problèmes d'optimisation, sont classiquement réparties en deux catégories (cf. [Armangué 2003]) : les méthodes **linéaires directes** et les méthodes **non-linéaires itératives**. Cependant, il convient d'insister sur le fait que ces approches ont toutes en commun trois points clés : **le filtrage des données**, **la normalisation des données** et **la contrainte de rang 2**. Nous exposerons dans un premier temps ces trois étapes essentielles puis nous détaillerons les deux grandes familles de méthodes.

Filtrage des appariements. L'étape préalable à tout processus d'estimation de paramètres est le filtrage des appariements. Ce filtrage est réalisé au moyen **d'estimateurs robustes** dont la finalité est de classer les données en bon (**inliers**) et mauvais (**outliers**) appariements. L'idée directrice est de trouver, par échantillonnage dans le jeu de appariements initiaux $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots n}$, le sous-ensemble $(\mathsf{q}_j, \mathsf{q}'_j)_{j\in I\subset[\![1,n]\!]}$ qui s'ajustera le mieux possible (c.-à-d. vis-à-vis d'une certaine norme) au sens d'un critère donné. La méthode utilisée pour estimer $\mathbf{F}$ est en général une méthode linéaire directe, telle que décrite ci-après. Plusieurs approches existent dans la littérature et différent selon la manière d'établir un consensus dans le choix de $\mathbf{F}$. La méthode **RanSaC** [Fischler 1981] calcule, pour chaque estimation de $\mathbf{F}$, un nombre d'**inliers** puis établit un consensus sur celle qui maximise le nombre d'inliers. Notons que la méthode **MLESaC** [Torr 2000] généralise cette approche. La méthode **LMedS** [Zhengyou 1998] calcule pour chaque estimation de $\mathbf{F}$ la distance médiane entre les appariements et les lignes épipolaires correspondantes, puis établit un consensus autour de celle qui minimise cette quantité.

Remarquons que ces méthodes peuvent aussi fournir une estimation de $\mathbf{F}$ à partir des **inliers**, mais elles peuvent souffrir d'un manque de répétitivité dû au caractère aléatoire de la sélection des sous-ensembles de points. Ainsi, le but principal de ces approches reste le filtrage et la détection de mauvais appariements ce qui constitue un problème distinct du calcul de la matrice fondamentale. Par conséquent, nous excluons les approches robustes de notre étude.

Normalisation des appariements. Dans toutes les approches que nous décrirons par la suite, les inconnues sont les paramètres de la matrice $\mathbf{F}$ tandis que les appariements fournissent, après **manipulations**, les coefficients des fonctionnelles que l'on cherche à minimiser ou à annuler. Ces fonctionnelles sont, en général, des polynômes ou des sommes de fonctions rationnelles et leur paramètres sont, d'ordinaire, le résultat d'une succession d'opérations simples (multiplications, divisions, additions...) appliquées aux coordonnées des appariements entre eux. On comprend dès lors que ces coefficients peuvent non seulement devenir très grands, ce qui pénalise les solveurs numériques, mais surtout avoir des ordres de grandeur très hétérogènes dans le cas où le rapport des variances des appariements est élevé (cf. Figure 4.6).

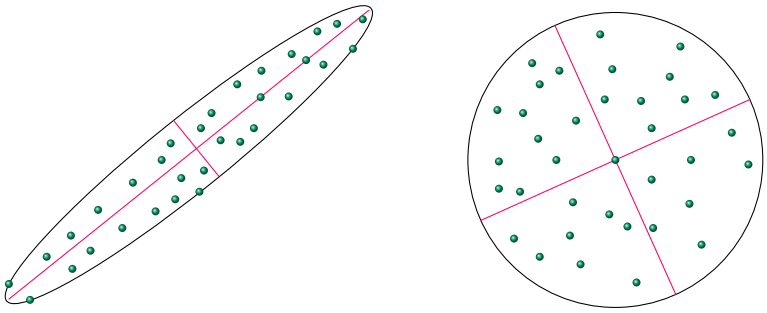


FIGURE 4.6 – Illustration d'appariements ayant des variances en équilibre (à droite) et d'appariements ayant un rapport de variance trop élevé (à gauche)

Ce dernier point aura une influence directe sur le conditionnement et la stabilité (en particulier inverse) des fonctionnelles que l'on étudie. Il est par conséquent légitime de vouloir imposer aux appariements, à l'aide de changements de base, de bonnes propriétés numériques. Ces changements de bases se font en appliquant des transformations inversibles, notées $\mathbf{T}$ et $\mathbf{T}'$, aux appariements avant l'utilisation de chaque méthode. Cependant, la matrice $\hat{\mathbf{F}}^\star$ obtenue après exécution correspond aux appariements modifiés :

$$\hat{\mathsf{q}}_i = \mathbf{T}\mathsf{q}_i \quad \text{et} \quad \hat{\mathsf{q}}'_i = \mathbf{T}'\mathsf{q}'_i. \tag{4.57}$$

Par conséquent, $\hat{\mathbf{F}}^\star$ doit être transformée afin de retrouver la matrice fondamentale originale $\mathbf{F}^\star$. Ce retour dans la base initiale s'effectue via l'équation suivante :

$$\mathbf{F} \quad = \quad \mathbf{T}'^\top \hat{\mathbf{F}}^\star \mathbf{T} \tag{4.58}$$

Il existe plusieurs sortes de normalisation. Les deux plus classiques sont celles proposées dans [Zhengyou 1998] et dans [Hartley 1995]. La première permet de ramener l'ensemble des coordonnées des appariements dans $[-1, 1]$. Dans la seconde, les correspondances sont transformées de telle sorte que la moyenne de leur distance à l'origine soit égale à $\sqrt{2}$.

Contrainte de rang 2. Due à la définition des épipoles (4.6), la matrice fondamentale $\mathbf{F}$ a nécessairement un rang égal à 2. Si cette condition n'est pas réalisée, les droites épipolaires ne s'intersectent pas en un seul point et les épipoles ne peuvent être correctement déterminés. La manière la plus naturelle d'imposer cette condition consisterait à l'intégrer directement en tant que contrainte du problème d'optimisation étudié. Cependant, comme nous l'avons déjà souligné, le choix d'un point inital dans l'espace des contraintes est une étape délicate qui complexifie fortement la minimisation locale. De même, l'ajout d'une contrainte non-linéaire, rendra toute tentative de minimisation globale beaucoup plus difficile. Deux approches sont classiquement utilisées pour contourner l'adjonction de cette contrainte.

La première consiste à forcer le rang de $\mathbf{F}$ **a posteriori**. $\mathbf{F}$ est alors remplacée par $\check{\mathbf{F}}$, sa projection sur la variété $\{M \in \mathcal{M}_{3,3}(\mathbb{R}) \mid \det(M) = 0\}$. Cette projection est déterminée en résolvant le problème de minimisation suivant :

$$\min_{\text{s.l.c. } \det(\check{\mathbf{F}})=0} \|\mathbf{F} - \check{\mathbf{F}}\|. \tag{4.59}$$

Dans le cas particulier où la norme matricielle $\|\ \|$ est la norme de Frobenius, ce problème peut être résolu directement à l'aide d'une **Décomposition en Valeurs Singulières** [Tsai 1984, Hartley 1995]. En effet, si

$$\mathbf{F} \quad = \quad U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^\top \tag{4.60}$$

est une décomposition en valeurs singulières de $\mathbf{F}$ avec $\sigma_1 \geqslant \sigma_2 \geqslant \sigma_3$, alors :

$$\check{\mathbf{F}} = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top \tag{4.61}$$

est une solution de (4.59), et donc la matrice de rang 2 la plus proche de $\mathbf{F}$ au sens de la norme de Frobenius. Cependant, $\check{\mathbf{F}}$ peut être loin d'une solution, même locale, d'un problème de minimisation où la condition de rang serait directement imposée en tant que contrainte.

La seconde approche repose sur l'idée d'imposer, **a priori**, le rang de $\mathbf{F}$ directement dans son paramétrage [Hartley 2003]. La manière la plus simple de réaliser un tel paramétrage consiste à exprimer, par exemple, la troisième colonne de $\mathbf{F}$ comme une combinaison linéaire des deux autres :

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & \alpha f_{11} + \beta f_{12} \\ f_{21} & f_{22} & \alpha f_{21} + \beta f_{22} \\ f_{31} & f_{32} & \alpha f_{31} + \beta f_{32} \end{bmatrix}. \tag{4.62}$$

On remarquera que, dans ce cas, $(\alpha, \beta, -1)^\top$ est l'épipole droit de $\mathbf{F}$ attendu qu'il vérifie (4.6). Bien que ce paramétrage assure la singularité de $\mathbf{F}$ avec un nombre minimal de coefficients, cette approche n'est valide que si les deux vecteurs colonnes $(f_{11}, f_{21}, f_{31})^\top$ et $(f_{12}, f_{22}, f_{32})^\top$ sont linéairement indépendants. Cette condition peut ne pas être vérifiée comme par exemple dans le cas où l'épipole droit se trouve à l'infini. En outre, le choix d'employer les deux premières colonnes est totalement arbitraire. Nonobstant que l'épipole ne se trouve pas sur l'un des axes de coordonnée, n'importe quelle autre paire de colonnes peut être utilisée. En pratique, ces configurations défavorables peuvent être détectées durant la minimisation ce qui permet de changer de paramétrage lorsque les épipoles sont mal positionnés. Notons que le nombre total de configurations possibles [Hartley 2003] à l'aide d'un épipole s'élève à 18. Il est de surcroît possible d'utiliser les deux épipoles dans le paramétrage. Ceci conduit à l'expression de $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & \mathsf{e}_1 f_{11} + \mathsf{e}_2 f_{12} \\ f_{21} & f_{22} & \mathsf{e}_1 f_{21} + \mathsf{e}_2 f_{22} \\ \mathsf{e}'_1 f_{11} + \mathsf{e}'_2 f_{21} & \mathsf{e}'_1 f_{12} + \mathsf{e}'_2 f_{22} & (\mathsf{e}_1 f_{11} + \mathsf{e}_2 f_{12})\mathsf{e}'_1 + (\mathsf{e}_1 f_{21} + \mathsf{e}_2 f_{22})\mathsf{e}'_2 \end{bmatrix}. \tag{4.63}$$

Il conviendra, dans ce cas aussi, de détecter les configurations défavorables durant l'optimisation afin de passer vers des paramétrages non dégénérés. Le nombre de total de configurations à deux épipoles s'élève à 36 [Hartley 2003].

Les trois points que nous venons de détailler sont des étapes communes à toutes les méthodes que nous décrivons dans les paragraphes suivants. Elles sont en général appliquées de manière successive avant et après l'exécution de chaque méthode : on effectue premièrement un filtrage des appariements, puis une normalisation de ces derniers et enfin on corrige le rang soit **a priori**, soit **a posteriori**. Il est important de souligner que, sans ces étapes, toutes les approches décrites ci-dessous peuvent donner de très mauvais résultats.

Les méthodes linéaires directes. Les méthodes linéaires sont basées sur la résolution directe du système d'équations linéaires (4.5). Appliqué à un couple $\mathsf{q}_i = (u_i, v_i, 1)$ et $\mathsf{q}'_i = (u'_i, v'_i, 1)$ de point appariés, ce système fournit l'équation linéaire :

$$u'_i u_i f_{11} + u'_i v_i f_{12} + u'_i f_{13} + v'_i u_i f_{21} + v'_i v_i f_{22} + v'_i f_{23} + u_i f_{31} + v_i f_{32} + f_{33} = 0, \tag{4.64}$$

avec $\mathbf{F} = (f_{i,j})$. Ainsi, à partir des n appariements, on obtient le système linéaire :

$$\mathsf{A}\mathsf{f} = 0, \tag{4.65}$$

où chaque ligne de $\mathsf{A} \in \mathcal{M}_{n\times 9}(\mathbb{R})$ est définie par $(u'_i u_i, u'_i v_i, u'_i, v'_i u_i, v'_i v_i, v'_i, u_i, v_i, 1)^\top$, et $\mathsf{f} \triangleq (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^\top$. Notons que les solutions de ce système sont aussi les candidats à l'optimalité du problème :

$$\min_{\mathbf{F}\in\mathcal{M}_{3,3}(\mathbb{R})} \sum_{i=1}^{n} \left(\mathsf{q}_i'^\top \mathbf{F} \mathsf{q}_i\right)^2 , \tag{4.66}$$

attendu que $\nabla\left(\sum_{i=1}^{n}\left(\mathsf{q}_i'^\top \mathbf{F}\mathsf{q}_i\right)^2\right) = \left(2\mathsf{q}_i'^\top \mathbf{F}\mathsf{q}_i\right)_{i=1\ldots n}$. Cette formulation, initialement résolue pour huit appariements et connue sous le nom **d'algorithme des 8-points**, a été introduite dans [Longuet-Higgins 1981]. Cependant, le nombre d'appariements, en pratique largement supérieur à huit, ainsi que l'erreur liée à leur calcul, font que le système (4.65) est surdéterminé et généralement mal conditionné. Ainsi, plusieurs adaptations [Tsai 1984, Hartley 1995, Torr 2002, Kanatani 2000, Chojnacki 2003] permettent de contraindre le conditionnement de A en imposant un changement d'échelle sur les appariements $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots n}$. Ce changement d'échelle est habituellement réalisé **a priori** au moyen des matrices de normalisation $\mathbf{T}$ et $\mathbf{T}'$ (cf. (4.57)). La correction de rang est, quant à elle, appliquée **a posteriori**.

Les méthodes non-linéaires itératives. Les méthodes itératives peuvent être classifiées en deux groupes : celles qui minimisent une distance géométrique et celles dites **de gradients**. Dans les deux cas, on est ramené à résoudre un problème d'optimisation de la forme :

$$\min_{\mathbf{F}\in\mathcal{M}_{3,3}(\mathbb{R})} \sum_{i=1}^{n} f_i(\mathbf{F}) , \tag{4.67}$$

où les $f_i : \mathcal{M}_{3,3}(\mathbb{R}) \to \mathbb{R}^+$ sont spécifiques à la technique considérée. Pour la première catégorie de méthodes, ces fonctions sont des distances géométriques. La distance la plus pertinente est évidemment la distance entre un appariement $(\mathsf{q}_i, \mathsf{q}'_i)$ et ses lignes épipolaires correspondantes $\mathsf{l}_{\mathsf{q}'_i}$ et $\mathsf{l}'_{\mathsf{q}_i}$ (cf. Figure 4.7).

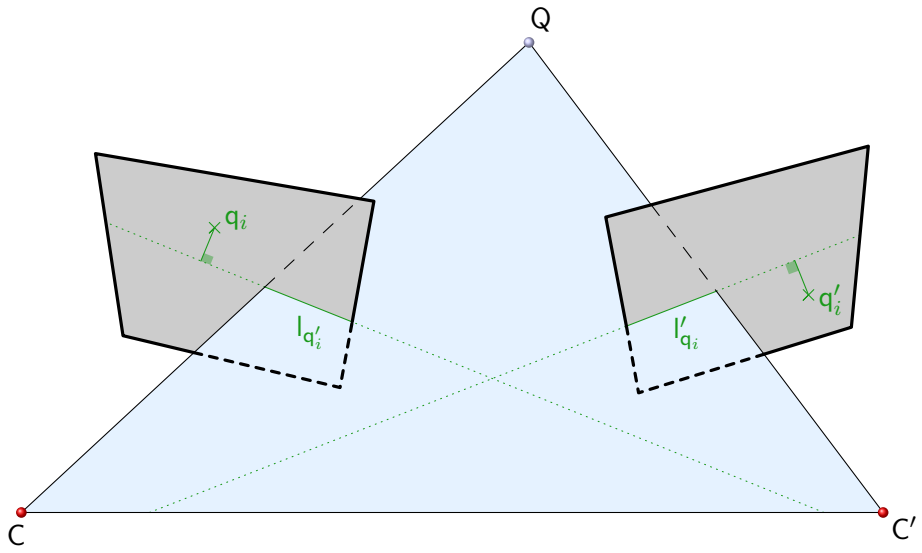


FIGURE 4.7 – Illustration des distances épipolaires (en vert) entre un couple de points appariés (croix) et leurs lignes épipolaires correspondantes

Cette distance, initialement introduite dans [Luong 1996a], est appelée **distance épipolaire croisée**, et conduit à la définition des f_i suivantes :

$$f_i(\mathbf{F}) = \left(\frac{1}{\pi_1\left(\mathbf{F}\mathsf{q}_i\right)^2 + \pi_2\left(\mathbf{F}\mathsf{q}_i\right)^2} + \frac{1}{\pi_1\left(\mathbf{F}^\top\mathsf{q}_i\right)^2 + \pi_2\left(\mathbf{F}^\top\mathsf{q}_i\right)^2}\right)\left(\mathsf{q}_i'^\top \mathbf{F}\mathsf{q}_i\right)^2 \tag{4.68}$$

$\pi_j : (x_1, x_2, x_3) \rightarrow x_j$ est la $j^{\text{ème}}$ projection canonique de $\mathbb{R}^3$. La résolution classique du problème (4.67), avec les f_i ainsi définies, consiste à appliquer directement une méthode d'optimisation locale du type **quasi-Newton**, **région de confiance**, ou **Levenberg-Marquardt**. Pour plus de détails sur les applications des méthodes d'optimisation locales à ce problème le lecteur pourra se référer à [Salvi 1999, Chen 2010].

Le second groupe de méthodes est basé sur la résolution du problème (4.66). Comme nous l'avons déjà souligné, lorsque le rapport des variances des appariements est élevé, le résultat de la minimisation de ce type de problème peut se révèler très imprécis. Une manière d'éviter cet inconvénient, sans nécessairement passer par la normalisation des données, consiste à quotienter chaque résidu $\mathsf{q}_i'^{\top}\mathbf{F}\mathsf{q}_i$ de (4.66) par une fonction $g_i : \mathcal{M}_{3,3}(\mathbb{R}) \rightarrow \mathbb{R}^+$ dépendant des appariements :

$$\min_{\mathbf{F}\in\mathcal{M}_{3,3}(\mathbb{R})} \sum_{i=1}^{n} \frac{\left(\mathsf{q}_i'^{\top}\mathbf{F}\mathsf{q}_i\right)^2}{g_i(\mathbf{F})} \tag{4.69}$$

Chaque méthode est alors caractérisée par le choix des g_i. Dans [Luong 1996a], les g_i sont par exemple définies par :

$$g_i(\mathbf{F}) = \pi_1(\mathbf{F}\mathsf{q}_i)^2 + \pi_2(\mathbf{F}\mathsf{q}_i)^2 + \pi_1(\mathbf{F}^{\top}\mathsf{q}_i')^2 + \pi_2(\mathbf{F}^{\top}\mathsf{q}_i')^2. \tag{4.70}$$

Notons que, dans ce cas particulier, le $i^{\text{ème}}$ rapport

$$\frac{\left(\mathsf{q}_i'^{\top}\mathbf{F}\mathsf{q}_i\right)^2}{\pi_1(\mathbf{F}\mathsf{q}_i)^2 + \pi_2(\mathbf{F}\mathsf{q}_i)^2 + \pi_1(\mathbf{F}^{\top}\mathsf{q}_i')^2 + \pi_2(\mathbf{F}^{\top}\mathsf{q}_i')^2} \tag{4.71}$$

correspond à une approximation de la distance du point $(\mathsf{q}_i, \mathsf{q}_i')$ à la surface $\mathcal{S}$ paramétrée implicitement par le vecteur f. Cette approximation est appelée **approximation** ou **distance de Sampson**. Pour plus de détails sur cette approche, le lecteur pourra se référer à [Hartley 2003]. Une approximation du maximum de vraisemblance pouvant être avantageusement appliquée aux problèmes d'estimation en vision artificielle a été introduite dans [Chojnacki 2002]. Son utilisation pour le calcul de la matrice fondamentale a été exposée dans [Van Den Hengel 2002]. Dans cette approche, les g_i sont définis par :

$$g_i(\mathbf{F}) = \mathsf{q}_i^{\top}\mathbf{F}\mathbf{F}^{\top}\mathsf{q}_i + \mathsf{q}_i'^{\top}\mathbf{F}\mathbf{F}^{\top}\mathsf{q}_i'. \tag{4.72}$$

La minimisation de la fonction coût ainsi obtenue est réalisée à l'aide de la méthode de **Newton** . Cependant, l'étape de résolution du système (2.59) est **ré-arrangée** de manière à intégrer la covariance Λ_q des appariements. Deux ré-arrangements sont proposés, chacun définissant une méthode différente : **FNS** pour **Fundamental Numerical Scheme** et **CNFS** pour **Constrained Fundamental Numerical Scheme**.

Bilan. Nous venons de présenter deux catégories de méthodes. Les méthodes **linéaires directes** sont très rapides puisqu'elles ne nécessitent aucun processus itératif (résolution d'un système linéaire) mais souffrent d'imprécision lorsque leur conditionnement n'est pas contraint. Cependant, ces techniques peuvent être très largement améliorées en appliquant successivement les trois étapes que nous avons décrites. Les **méthodes non-linéaires**, présentent l'avantage de minimiser des critères plus réalistes et peuvent, pour certaines d'entre elles, compenser une mauvaise répartition des appariements. Cependant, les techniques d'optimisation locales utilisées pour leur résolution ne fournissent aucun certificat d'optimalité global. De plus, leur sensibilité au choix de l'initialisation complexifie l'introduction de la contrainte de rang directement dans le problème d'optimisation. Enfin, intégrer la contrainte de rang 2 dans le paramètrage renforce la non-linéarité de la fonction coût et rend donc l'optimisation plus difficile.

Algorithme 18 Algorithme des **8-points**

Précondition : n appariements $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots n}$

1: Normalisation des appariements de sorte que leurs coordonnées soient dans $[-\sqrt{2}; \sqrt{2}]^2$

2: Calcul de $\tilde{\mathbf{F}}$ en résolvant le problème d'optimisation convexe :

$$\tilde{\mathbf{F}} \triangleq \underset{\mathbf{F}\in\mathcal{M}_{3,3}(\mathbb{R})}{\arg\min} \sum_{i=1}^{n} \left(\mathsf{m}_i'^{\top}\mathbf{F}\mathsf{m}_i\right)^2. \tag{4.73}$$

3: Projection de $\mathbf{F}$ sur la variété $\{M \in \mathcal{M}_{3,3}(\mathbb{R}) \mid \det(M) = 0\}$ en résolvant, à l'aide d'une Décomposition en valeurs singulières, le problème :

$$\mathbf{F}_{8\text{pt}} \triangleq \underset{\mathbf{F}\in\mathcal{M}_{3,3}(\mathbb{R})}{\arg\min} \|\mathbf{F} - \tilde{\mathbf{F}}\|^2. \tag{4.74}$$

4: Normalisation de $\mathbf{F}_{8\text{pt}}$ de sorte que $\|\mathbf{F}_{8\text{pt}}\|^2 = 1$.

5: **retour** $\mathbf{F}_{8\text{pt}}$

Ainsi, pour l'ensemble des comparaisons que nous allons mener, nous choisissons d'utiliser comme référence l'adaptation de l'algorithme des **8-points** suivant :
Il a été montré [Hartley 1995] que cet algorithme simple fonctionne très bien en pratique et ce, sans subir les défauts d'un processus itératif (p.e. choix d'un point initial).

4.1.4.2 Utilisation de l'optimisation polynomiale.

Nous nous concentrons dans cette section à appliquer la méthode d'optimisation polynomiale au problème de l'estimation de la matrice $\mathbf{F}$. Dans toutes les approches d'estimation que nous avons mentionnées, le principal désavantage est la difficulté d'intégrer la contrainte de rang 2 car elle empêche d'une part toute résolution linéaire directe et complexifie, d'autre part, le calcul d'une estimée initiale pour les méthodes itératives. Cependant, bien que les méthodes d'optimisation globale ne nécessitent pas d'estimée initiale, leur principal défaut est la difficulté de prendre en compte des contraintes fortement non-linéaires. Comme nous l'avons déjà mentionné, l'optimisation polynomiale permet d'éviter ce problème. Mais nous ne pouvons minimiser globalement avec cette technique que des polynômes ou des sommes de fonctions rationnelles. Or, selon [Hartley 2003], si les appariements sont correctement filtrés et normalisés la minimisation des **distances épipolaires croisées** et la résolution du problème (4.66) donnent des résultats similaires. De surcroit, comme nous l'avons montré, la programmation rationnelle reste plus coûteuse en temps de calcul. Il est donc légitime de ne se concentrer que sur la résolution globale du problème (4.66).

Afin de simplifier l'espace de recherche, il est possible de supprimer la contraintes de rang 2 et de l'inclure directement dans la fonction coût. Cependant, il est nécessaire, dans ce cas, d'intégrer une contrainte additionnelle permettant d'éviter la solution triviale $\mathbf{F} = 0$. Ceci est habituellement réalisé en fixant, **a priori**, un des coefficients f_{ij} de $\mathbf{F}$ à 1. Bien que ce procédé permette de supprimer une variable, il n'assure pas que les autres coefficients de $\mathbf{F}$ soient bornés, ce qui est une condition nécessaire pour assurer la convergence de notre méthode d'optimisation polynomiale. Par conséquent, afin d'assurer cette majoration et d'éviter la solution nulle, nous rajoutons la contrainte complémentaire $\|\mathbf{F}\|_F^2 = 1$.

Comme nous l'avons déjà exposé, la condition de rang peut être introduite dans le problème d'optimisation de deux manières. Concentrons nous tout d'abord sur l'approche consistant à paramétrer $\mathbf{F}$ à l'aide des épipoles. Dans le cas où $\mathbf{F}$ est paramétrée avec un seul épipole, on

obtient le problème suivant :

$$\begin{cases} \min\limits_{\substack{(f_{k,l}) \in \mathbb{R}^6 \\ (\alpha,\beta) \in \mathbb{R}^2}} & \sum\limits_{i=1}^{n} \left(\mathsf{q}_i'^{\top} \begin{bmatrix} f_{11} & f_{12} & \alpha f_{11} + \beta f_{12} \\ f_{21} & f_{22} & \alpha f_{21} + \beta f_{22} \\ f_{31} & f_{32} & \alpha f_{31} + \beta f_{32} \end{bmatrix} \mathsf{q}_i \right)^2 \\ \text{s.l.c.} & \begin{cases} \alpha^2 + \beta^2 = 1 \\ \sum\limits_{k,l} f_{k,l}^2 = 1 \end{cases} \end{cases}, \tag{4.75}$$

Notons que ces contraintes sont une simplification de la contrainte $\|\mathbf{F}\|_F^2 = 1$, mais elles permettent d'imposer que la norme de l'épipole soit égale à 1. Une tentative de résolution d'un problème similaire :

$$\min_{\substack{(f_{k,l}) \in \mathbb{R}^6 \\ (\alpha,\beta) \in [0,1]^2}} \quad \sum_{i=1}^{n} \left(\mathsf{q}_i'^{\top} \begin{bmatrix} \alpha f_{11} + \beta f_{21} & \alpha f_{12} + \beta f_{22} & \alpha f_{13} + \beta f_{23} \\ f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & 1 \end{bmatrix} \mathsf{q}_i \right)^2, \tag{4.76}$$

à l'aide de l'optimisation polynomiale a été proposée dans [Xuelian 2010]. Cependant, cette approche souffre de plusieurs inconvénients. Premièrement, l'absence de bornes sur les coefficients de $\mathbf{F}$ ne permet pas de garantir la convergence de la méthode d'optimisation. Deuxièmement, l'introduction des inconnues α et β augmente le degré du polynôme objectif ainsi que la taille des relaxations. Ceci a pour conséquence d'accroître le temps de calcul de manière importante. Enfin, comme nous l'avons déjà mentionné, le choix de ce paramétrage est arbitraire et peut ne pas être valide dans toutes les configurations. En toute rigueur, il faudrait que l'ensemble des configurations puissent être minimisées. Mais, dans ce cas la contrainte $(\alpha, \beta) \in [0,1]^2$ devrait être remplacée par $\alpha^2 + \beta^2 = 1$ de manière à éviter que, pour le paramétrage (4.62) par exemple, l'épipole ne soit nul. Par ailleurs, nous avons essayé de résoudre (4.75) sur plusieurs configurations synthétiques et réelles mais, en plus d'un temps de calcul élevé, la convergence de la méthode s'est révélée très instable.

Ensuite, dans le cas où la matrice fondamentale est paramétrée à l'aide de deux épipoles, on est amené à résoudre le problème suivant :

$$\begin{cases} \min\limits_{\substack{(f_{k,l}) \in \mathbb{R}^4 \\ (\mathsf{e},\mathsf{e}') \in \mathbb{R}^4}} & \sum\limits_{i=1}^{n} \left(\mathsf{q}_i'^{\top} \begin{bmatrix} f_{11} & f_{12} & \mathsf{e}_1 f_{11} + \mathsf{e}_2 f_{12} \\ f_{21} & f_{22} & \mathsf{e}_1 f_{21} + \mathsf{e}_2 f_{22} \\ \mathsf{e}'_1 f_{11} + \mathsf{e}'_2 f_{21} & \mathsf{e}'_1 f_{12} + \mathsf{e}'_2 f_{22} & (\mathsf{e}_1 f_{11} + \mathsf{e}_2 f_{12})\mathsf{e}'_1 \\ & & +(\mathsf{e}_1 f_{21} + \mathsf{e}_2 f_{22})\mathsf{e}'_2 \end{bmatrix} \mathsf{q}_i \right)^2 \\ \text{s.l.c.} & \begin{cases} \|\mathsf{e}\|^2 = 1 \\ \|\mathsf{e}'\|^2 = 1 \\ \sum\limits_{k,l} f_{k,l}^2 = 1 \end{cases} \end{cases}. \tag{4.77}$$

Bien que ce paramétrage soit théoriquement optimal, il n'est pas raisonnable d'envisager l'utilisation de cette approche sur des cas concrets. En effet, il serait nécessaire de couvrir l'ensemble de paramétrages possibles (36 en tout) ce qui semble contradictoire avec une utilisation pratique à cause de temps de calcul prohibitifs (plus de 3 minutes pour une relaxation

sur un MacBook Pro Core i7 avec 8G de mémoire vive). Ces temps de calcul considérables sont liés à l'augmentation du degré du polynôme objectif induite par l'introduction des deux épipoles dans le paramétrage de $\mathbf{F}$.

Pour toutes les raisons que nous venons de mentionner, il n'est pas pertinent de considérer l'introduction de la contrainte de rang via le paramétrage de $\mathbf{F}$. L'autre solution consiste donc à introduire cette condition directement dans les contraintes. Ceci nous conduit au problème de minimisation de **la distance algébrique** défini par :

$$\begin{cases} \min\limits_{(f_{k,l}) \in \mathbb{R}^9} \sum\limits_{i=1}^{n} \left(\mathsf{q}_i'^{\top} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \mathsf{q}_i \right)^2 \\ \text{s.l.c.} \begin{cases} \det(\mathbf{F}) = 0 \\ \|\mathbf{F}\|_F^2 = 1 \end{cases} \end{cases} \tag{4.78}$$

Cette approche a été étudiée dans [Chesi 2002], où un problème similaire a été résolu à l'aide d'une hiérarchie de problèmes convexes. Cependant, plusieurs observations peuvent être faites sur cette méthode. Premièrement, les variables d'optimisation ne sont pas bornées car, comme dans [Xuelian 2010], la contrainte $\|\mathbf{F}\|_F^2 = 1$ est supprimée au profit de l'hypothèse **a priori** que l'un des coefficients f_{ij} est égal à 1. Deuxièmement, la contrainte $\det(\mathbf{F}) = 0$ n'est prise en compte qu'au prix de l'introduction de nouvelles variables d'optimisation. Enfin, il n'existe pas de preuve de la convergence de la suite des solutions de la hiérarchie des relaxations convexes vers le minimum global. On notera cependant que les résultats obtenus dans cette étude semblent très prometteurs et ce même s'il ne sont pas testés sur un panel exhaustif d'images.

Notons enfin que dans [Kahl 2007], le problème de la minimisation de la distance de Sampson est résolu à l'aide d'une hiérarchie de relaxation convexes construites à partir de la formulation par épigraphes de (4.71). Cependant, bien que la correction de rang soit satisfaite via l'ajout de la contrainte $\det(\mathbf{F}) = 0$, l'exclusion de la solution triviale nulle est réalisée en fixant $f_{33} = 1$ ce qui ne permet pas de borner les autres coefficients de $\mathbf{F}$. De surcroit, comme nous l'avons déjà mentionné, la formulation par épigraphe nécessite l'ajout d'autant de variables d'optimisation que d'appariements, ce qui ne permet généralement pas de considérer un grand nombre d'appariements. La prise en compte des variables d'écart se fait, dans cette approche, au détriment de la convergence vers un minimum global qui n'est pas garanti. Nous avons, pour notre part, essayé de résoudre (4.71) à l'aide de la Programmation Rationnelle. Cependant, la solution globale n'était pas, dans tous nos tests, accessible dès la première relaxation et la seconde hors de portée des solveurs actuels. Afin d'atteindre cette deuxième relaxation, nous avons fixé deux coefficients de $\mathbf{F}$ à l'aide d'une estimation complémentaire via la résolution globale de (4.78). Nous avons alors remarqué que les coefficients de $\mathbf{F}$ ainsi obtenus varient très peu par rapport à la solution initialement fournie par (4.78). Ceci semble confirmer, comme souligné dans [Hartley 2003], que la solution fournie par les problèmes de minimisation de distance de Sampson et la distance algébrique fournissent des résultats similaires. Cependant, pour certifier rigoureusement cette observation, il faudrait pouvoir considérer la totalité des variables, ce qui n'est pas à notre portée au vu des solveurs actuels.

Au bilan, pour toutes les raisons que nous venons de détailler, nous appliquons l'optimisation polynomiale par théorie des moments au problème de la minimisation de la distance algébrique défini par (4.78). D'un point de vue mathématique, ce problème ne peut être résolu

globalement à l'aide de techniques simples comme c'est le cas pour les critères de minimisation des distorsions projectives exposés dans la section précédente. En effet, bien qu'ayant un polynôme objectif de degré 2, ce problème comporte 9 variables et deux contraintes non linéaires de degré 2 et 3, ce qui le classe dans une catégorie de problème difficile pour l'optimisation globale. Un alternative à notre méthode pourrait être l'utilisation des bases de Groebner pour énumérer tous les zéros du Gradient du Lagrangien, mais cette technique n'a pas, à notre connaissance, été étudiée.

4.1.4.3 Résultats numériques

Cette section est consacrée à l'étude des résultats fournis par la minimisation globale de la **distance algébrique** (4.78) comparativement à l'algorithme des **8-points** tel que décrit par l'Algorithme 18. Nous présentons dans un premier paragraphe la procédure d'évaluation des deux méthodes. Les paragraphes suivants seront consacrés à l'exposé des résultats sur des données de synthèse ainsi que sur des cas réels.

Procédure d'évaluation. L'évaluation de la qualité d'une matrice fondamentale n'est pas triviale. Différents critères ont été proposés dans [Zhengyou 1998]. Nous proposons d'évaluer une estimation de la matrice fondamentale $\mathbf{F}$ comparativement au comportement de son raffinement par l'ajustement de faisceaux projectifs. Il est important de noter qu'une matrice fondamentale est équivalente, à une base projective près, à une paire de matrices représentant des caméras perspectives non calibrées. En effet, comme proposé dans [Luong 1996b], on peut extraire de $\mathbf{F}$ deux matrices de projections perspectives $\mathbf{P}$ et $\mathbf{P}'$ définies par :

$$\mathbf{P} \sim \left[\mathbf{I}_{(3\times3)}\ 0_{(3\times3)}\right] \quad \text{et} \quad \mathbf{P}' \sim \left[\mathbf{H}^*\ \gamma \mathsf{e}'\right] \tag{4.79}$$

où $\mathbf{H}$ est l'homographie canonique définie par $\mathbf{H}^* \sim [\mathsf{e}]_\times \mathbf{F}$ et γ un réel qui peut être fixé, sans perte de généralité, à 1. En d'autres termes, une matrice fondamentale permet implicitement une reconstruction projective 3D. Ainsi, notre premier critère d'évaluation de la qualité d'une estimation de la matrice fondamentale est l'erreur de reprojection linéaire, notée $e_{\text{Init}}(\mathbf{F})$.

Un **ajustement de faisceaux** (« bundle adjustment » en anglais) projectifs à deux vues est mathématiquement décrit par un problème de minimisation dont la fonction objectif est la moyenne des erreurs de reprojection des appariements $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots n}$ et les inconnues les points 3D à reconstruire $(\mathsf{Q}_i)_{i=1\ldots n}$ ainsi que les matrices de projection $\mathbf{P}, \mathbf{P}'$:

$$\min_{\mathbf{P},\mathbf{P}',\mathsf{Q}_i} \sum_{i=1}^{n} \|\mathsf{q}_i - \mathbf{P}\mathsf{Q}_i\|^2 + \|\mathsf{q}'_i - \mathbf{P}'\mathsf{Q}_i\|^2. \tag{4.80}$$

Ce problème est résolu à l'aide d'un algorithme de Levenberg-Marquardt initialisé à partir des matrices extraites de la matrice $\mathbf{F}$. Deux indicateurs importants de l'ajustement de faisceaux sont ainsi la valeur de la fonction coût à l'optimum (c.-à-d. l'erreur de reprojection finale) et le nombre d'itérations nécessaires à la convergence. Ces deux critères permettent d'évaluer si les estimations des matrices de projections $\mathbf{P}$ et $\mathbf{P}'$ calculées à partir de $\mathbf{F}$ sont dans un bon bassin d'attraction. En effet, le nombre d'itérations donne une indication de la distance entre l'estimation et l'optimum alors que l'erreur finale fournit une indication sur la qualité de l'optimum. Il est donc naturel d'utiliser ces deux indicateurs en tant que deuxième et troisième critères d'évaluation de la qualité d'une estimation de la matrice fondamentale. Pour le reste de cette étude, nous les noterons respectivement $e_{\text{BA}}(\mathbf{F})$ et $\text{Iter}(\mathbf{F})$. L'algorithme 19 résume l'ensemble de notre procédure d'évaluation.

Algorithme 19 Algorithme d'évaluation d'une estimation de la matrice fondamentale

Précondition : $\mathbf{F}$ et n appariements $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots n}$

1: Calcul du second épipole en résolvant $\mathbf{F}^\top \mathsf{e}' \sim 0_{(3\times 1)}$
2: Calcul de l'homographie canonique $\mathbf{H}^* \sim [\mathsf{e}]_\times \mathbf{F}$
3: $\mathbf{P} \leftarrow \left[\mathbf{I}_{(3\times 3)}\ 0_{(3\times 3)}\right]$
4: $\mathbf{P}' \leftarrow [\mathbf{H}^*\ \gamma \mathsf{e}']$
5: Trianguler chaque point 3D indépendamment en minimisant son erreur de reprojection
6: Calculer $e_{\text{Init}}(\mathbf{F})$
7: Résoudre l'ajustement de faisceaux projectifs à deux vues en utilisant $\mathbf{F}$ et les points triangulés comme initialisation
8: Calculer $e_{\text{BA}}(\mathbf{F})$ et $\text{Iter}(\mathbf{F})$
9: **retour** $e_{\text{Init}}(\mathbf{F})$, $e_{\text{BA}}(\mathbf{F})$ et $\text{Iter}(\mathbf{F})$

Résultats sur données synthétiques. Les données synthétiques nous permettent d'évaluer la sensibilité des deux méthodes en fonction du bruit présent dans les extractions des appariements et du nombre d'appariements considérés dans la fonction coût. Ainsi, nous évaluons tout d'abord le comportement de notre résolution globale par rapport au bruit. Nous nous donnons donc une matrice de paramètres internes et plusieurs mouvements afin de déterminer complètement deux matrices de projection $\mathbf{P}$ et $\mathbf{P}'$. Nous générons ensuite un ensemble de points 3D, noté $(\mathsf{Q}_i)_i$, que nous projetons dans les images. Puis nous perturbons ces projections à l'aide d'un bruit Gaussien de variance σ^2. Les couples $(\mathsf{q}_i, \mathsf{q}'_i)_i$ ainsi obtenus servent d'appariements pour l'estimation de $\mathbf{F}$ avec les deux méthodes. Enfin, nous mesurons, via notre algorithme d'évaluation, l'erreur de cette estimation en fonction du niveau de la perturbation (c.-à-d. la variance σ).

Concrètement, nous simulons une matrice de projection perspective donnée par :

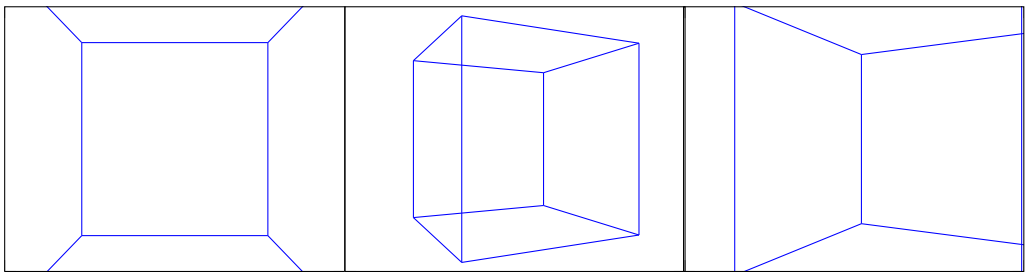
$$\mathbf{P} = \begin{bmatrix} 700 & 0 & 320 \\ 0 & 700 & 240 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.81}$$

ainsi que des points situés dans un cube de côté égal à 10. La première caméra regarde au centre du cube et se situe à 15 de celui-ci. Afin d'être le plus exhaustif possible, nous appliquons deux mouvements à la caméra. Un avec un angle important entre les centres optiques des caméras et un autre plus faible. Leurs matrices de rotation, notées respectivement $\mathbf{R}_1$ et $\mathbf{R}_2$, sont definies par :

$$\mathbf{R}_k \triangleq \begin{bmatrix} \cos(\theta_k) & 0 & \sin(\theta_k) \\ 0 & 1 & 0 \\ -\sin(\theta_k) & 0 & \cos(\theta_k) \end{bmatrix} \text{ avec } \begin{cases} \theta_1 = \dfrac{\pi}{3} \\ \text{and} \\ \theta_2 = \dfrac{\pi}{6} \end{cases}, \tag{4.82}$$

et leur vecteurs de translation par $t_1 = (20, 0, 5)^\top$ et $t_2 = (6, 0, 0)^\top$. La figure 4.8 montre la projection du cube dans chacune des positions de la caméra.
Nous déterminons aléatoirement (c.-à-d. avec un probabilité uniforme) 50 points 3D Q_i dans ce cube. Nous perturbons ensuite les projections $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots 50}$ de ces points dans les images en ajoutant à chacune de leurs coordonnées pixels $(u_i, v_i)_{i=1\ldots 50}$ et $(u'_i, v'_i)_{i=1\ldots 50}$ un bruit Gaussien centré. Nous faisons enfin évoluer la variance σ^2 de ce bruit de 0 à 2 pixels, de sorte que les appariements subissent un déplacement maximal relatif de 6 pixels pour 98% d'entre eux. Pour obtenir des données statistiques significatives, nous effectuons 100 tirages par niveaux de bruits.

Nous étudions ensuite l'influence du nombre d'appariements sur la qualité de l'estimation de $\mathbf{F}$. Ainsi nous considérons comme appariements 100 projections $(\mathsf{q}_i, \mathsf{q}'_i)_{i=1\ldots 100}$ de points



(I) (II) (III)

FIGURE 4.8 – Projection du cube dans la caméra en position initiale (I) et dans la caméra après application des transformations rigides $[\mathbf{R}_1\ t_1]$ (II) et $[\mathbf{R}_2\ t_2]$ (III).

3D $(\mathsf{Q}_i)_{i=1\ldots100}$ déplacées avec un bruit Gaussien de variance 0.5. Nous évaluons ensuite $\mathbf{F}$ en augmentant progressivement le nombre d'appariements.

Soulignons enfin que pour l'ensemble des résultats présentés, la méthode globale converge et fournit un certificat d'optimalité global.

La Figure 4.9 présente l'ensemble des résultats obtenus. La première ligne montre tout d'abord l'influence du bruit sur les erreurs de reprojection avant et après raffinement par ajustement de faisceaux. La deuxième ligne montre le nombre d'itérations effectuées par ce dernier en fonction du bruit. La première colonne concerne le premier mouvement $[\mathbf{R}_1\ t_1]$, et la deuxième le mouvement $[\mathbf{R}_2\ t_2]$.

Premièrement, nous constatons que les deux méthodes ont la même précision car, à de très légères différences près, elles fournissent les mêmes erreurs de reprojection finales. De plus, nous remarquons que les deux méthodes ont un comportement similaire : les erreurs de reprojection initiale ($8pt - Init$ et $Gp - Init$) avant et après ajustement de faisceaux ($8pt - BA$ et $Gp - BA$) sont presque identiques et varient linéairement avec le bruit. Ceci indique donc que, pour les deux mouvements, la solution initiale est très proche du minimum local du problème de l'ajustement de faisceaux. Ceci indique que le bruit introduit perturbe peu la précision finale. Cependant, les courbes de la deuxième ligne mettent en évidence que le nombre d'itérations nécessaires à l'ajustement de faisceaux pour converger sont différentes pour les deux méthodes. L'estimée initiale de la triangulation calculée à partir de $\mathbf{F}_{Gp}$ est plus proche du minimum que celle fournie par $\mathbf{F}_{8pt}$. Pour le mouvement $[\mathbf{R}_1\ t_1]$ le nombre d'itérations de la méthode globale (en vert) reste inférieur à celui de la méthode des **8-points** (en bleu), bien qu'il ait tendance à se réduire pour un bruit élevé ($\sigma^2 > 1$). Ceci indique donc que, pour un mouvement important, la qualité de l'estimation de $\mathbf{F}$ par la méthode globale reste meilleure même si elle se dégrade avec le bruit. A l'inverse, pour le mouvement $[\mathbf{R}_2\ t_2]$, les méthodes sont équivalentes car la différence ne devient significative que pour un bruit élevé ($\sigma^2 > 1$). Au bilan, la méthode des **8-points** fournit donc une solution de qualité équivalente à celle fournie par une minimisation globale lorsque le mouvement n'est pas trop grand. Cependant, elle a tendance à s'en éloigner, tout en restant dans le même bassin d'attraction, lorsque le mouvement augmente.

De manière similaire les deux dernières lignes montrent, pour une perturbation Gaussienne de variance égale à 0.5 pixels, l'influence du nombre points utilisés pour estimer $\mathbf{F}$ sur les erreurs de reprojection et le nombre d'itérations. Nous pouvons remarquer que, pour les deux mouvements, si on considère un nombre suffisant d'appariements (>50), les deux méthodes ont des erreurs de reprojection, avant et après raffinage par ajustement de faisceaux, relativement proches et ce pour un nombre d'itérations équivalents. Cependant, pour un nombre de points inférieurs, la qualité de l'estimation par minimisation globale semble meilleure puisque le nombre d'itérations nécessaires à la convergence est moindre et ce pour une erreur de re-

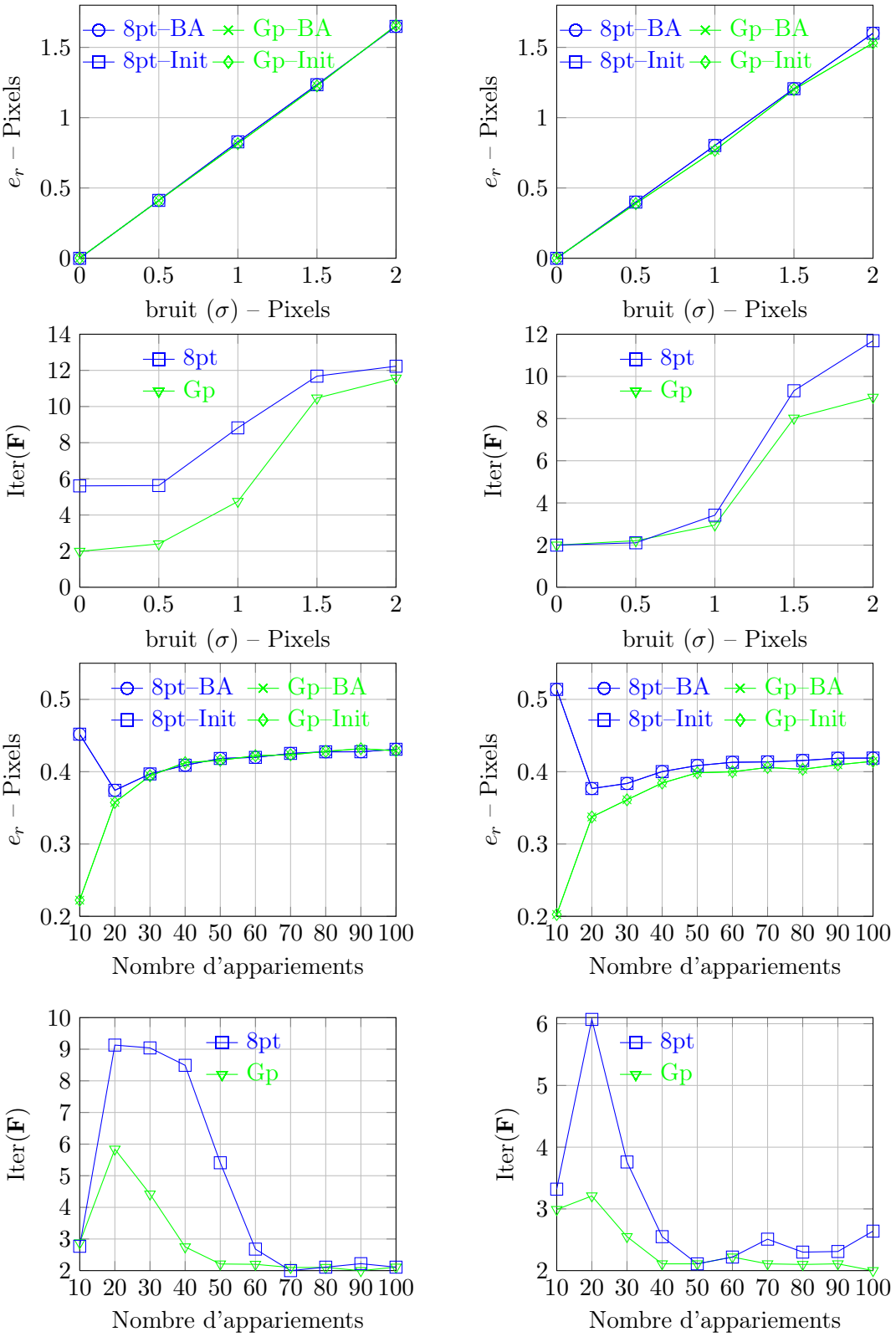


FIGURE 4.9 – Pour les deux mouvements, $[\mathbf{R}_1\ t_1]$ (colonne de gauche) et $[\mathbf{R}_2\ t_2]$ (colonne de droite), erreurs de reprojection (e_r) et nombre d'itérations en fonction du bruit (1ère et 2ème lignes) et du nombre d'appariements (3ème et 4ème lignes).

projection équivalente. Ceci indique que, quel que soit le nombre d'appariements utilisées, l'estimée initiale issue de la triangulation calculée avec $\mathbf{F}_{8pt}$ et celle calculée à partir de $\mathbf{F}_{Gp}$ sont dans le même bassin d'attraction. Cependant, lorsque le nombre d'appariements utilisés diminue, alors l'estimée initiale calculée avec $\mathbf{F}_{8pt}$ s'éloigne de ce bassin, alors que celle calculée à partir de $\mathbf{F}_{Gp}$ semble y demeurer. Afin de confimer ce comportement, nous avons testé, pour le mouvement le plus grand $[\mathbf{R}_1\ t_1]$, l'influence du nombre de points sur les erreurs de reprojection et le nombre d'itérations mais avec un bruit gaussien de variance égale à 1 pixel. Les résultats sont présentés sur la Figure 4.10.

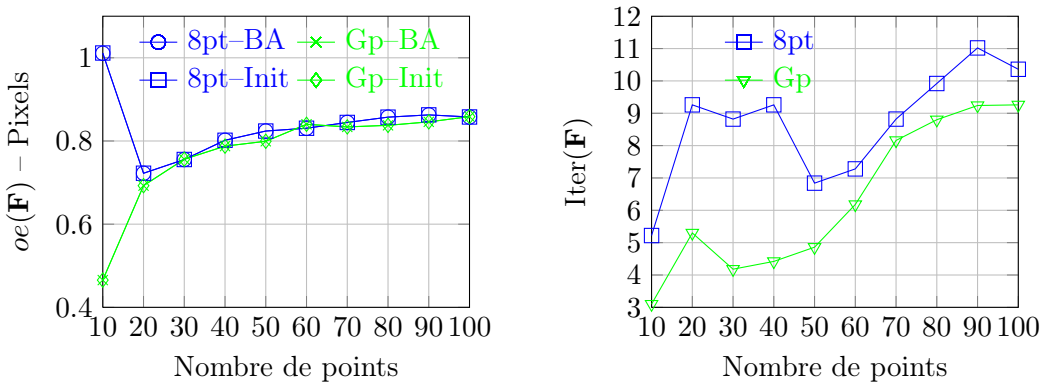


FIGURE 4.10 – Pour le mouvement $[\mathbf{R}_1\ t_1]$, erreurs de reprojections (e_r) et nombre d'itérations en fonction du nombre de points pour un bruit gaussien de variance égale à 1.

Ces dernières courbes confirment notre analyse puisque l'estimée initiale calculée à partir de $\mathbf{F}_{Gp}$ est toujours plus proche du minimum local que celle calculée à partir de $\mathbf{F}_{Gp}$. En effet, quelque soit le nombre de points, le nombre d'itérations est plus faible. Cependant, ce phénomène s'atténue lorsque le nombre de points augmente.

Cette première étude souligne que la qualité des deux méthodes est presque identique lorsque :

- l'angle entre les prises de vues n'est pas trop important
- le nombre d'appariements est suffisant
- le bruit n'est pas trop élevé

Lorsqu'un de ces paramètres varie trop, l'algorithme des **8-points** semble perdre en précision tandis que l'approche globale reste compétitive. Ceci est logique d'un point vue de mathématique car la méthode des **8-points** fournit la projection d'un minimum global sur l'espace des contraintes. Celui-ci a tendance à s'éloigner du minimum global fourni par notre méthode lorsque le conditionnement numérique de (4.78) se dégrade.

Résultats sur cas réels. Afin de confirmer nos premières constatations, nous mesurons $e_{\text{Init}}(\mathbf{F})$, $e_{\text{BA}}(\mathbf{F})$ et $\text{Iter}(\mathbf{F})$ sur plusieurs cas réels. Tout d'abord nous utilisons les images présentées dans la Figure 4.11, car elles permettent de couvrir toutes les configurations épipolaires possibles (épipoles droit ou gauche à l'infini...). Pour toutes les paires d'images, nous disposons de 60 appariements pour réaliser les estimations de $\mathbf{F}$. Les résultats sont présentés dans le tableau 4.11. En outre, nous avons choisi, pour cette série d'images, de faire apparaitre le temps de calcul nécessaire à chaque algorithme pour fournir une estimation.

Ces résultats soulignent que, quel que soit la paire d'images, les erreurs de reprojections et le nombre d'itérations sont toujours meilleurs lorsque $\mathbf{F}_{Gp}$ est utilisée comme initialisation. De plus, notre hypothèse semble se confirmer car, sur 3 configurations différentes ($\mathsf{A}-\mathsf{C}$, $\mathsf{A}-\mathsf{D}$ et $\mathsf{C}-\mathsf{D}$), l'initialisation issue de la méthode des **8-points** n'est pas dans le bon bassin

d'attraction, contrairement à celle calculée via $\mathbf{F}_{Gp}$. Pour quatre configurations ($\mathsf{A}-\mathsf{B}$, $\mathsf{B}-\mathsf{C}$, $\mathsf{B}-\mathsf{D}$ et $\mathsf{B}-\mathsf{D}$), les deux initialisations sont dans le bon bassin d'attraction, mais le nombre d'itérations souligne que la solution globale est toujours plus proche de l'optimum. Remarquons de surcroit que le temps de calcul de notre méthode (2 secondes en moyenne), bien que beaucoup plus élevé que la méthode des **8-points**, reste compatible avec une utilisation pratique.

Dans le but d'être le plus exhaustifs possible et de nous confronter à des cas difficiles, nous avons réalisé les mêmes tests sur plusieurs séries d'images, **Library** et **Merton**, mises à disposition par le Visual Geormetry Group d'Oxford (disponibles sur le site `www.robots.ox.ac.uk/~vgg/data/data-mview.html`). Ces résultats sont présentés sur les figures 4.12, 4.13,4.14 et 4.15. Nous avons également effectué les mêmes tests sur deux autres paires, moins conventionnelles, d'images. Pour la première paire, nous avons utilisé les images d'un cylindre étalon, utilisé pour évaluer la précision des reconstitutions en 3D, gracieusement fournies par la société NOOMEO (`http://www.noomeo.eu`). Les points appariés sont calculés à l'aide d'une méthode de corrélation d'images numériques. Ils sont situés dans une fenêtre au centre du cylindre. Ainsi, nous obtenons 6609 paires $(\mathsf{q}_i, \mathsf{q}_i)_i$ appariées avec une précision sous-pixelique. Les résultats sont présentés dans le tableau 4.11. Pour la seconde paire, nous avons utilisé des images prises par un endoscope et dont les points appariés se situent sur une zone décentrée de l'image. Le tableau 4.39 présente les résultats obtenus sur ce cas difficile.

L'ensemble des résultats présentés confirme, sur des cas difficiles, que la qualité de la solution globale est meilleure, même si dans certains cas elle est très proche de la solution proposée par l'algorithme des **8-points**. En effet, sur la série **Library**, les résultats sont très proches, ce qui souligne le fait que $\mathbf{F}_{8pt}$, la projection de la solution du problème sans contraintes, fournie par l'algorithme des **8-points** est très proche de $\mathbf{F}_{Gp}$, la solution globale du problème avec contraintes fournie par notre méthode. Par contre, sur certaines paires ($1-3$ de la série **Merton1**, $1-2, 1-3$ de la série **Merton2** et $1-2, 1-3$ de la série **Merton3**) la solution $\mathbf{F}_{8pt}$ est éloignée de la solution globale $\mathbf{F}_{Gp}$. Par conséquent, les matrices de projections extraites de $\mathbf{F}_{8pt}$ ne permettent par à l'ajustement de faisceaux projectif de converger vers une solution aussi précise que lorsque ce dernier est initialisé avec les matrices de projections extraites de $\mathbf{F}_{Gp}$. Ceci souligne donc que l'écart à l'optimum global dans le problème de l'estimation de la matrice fondamentale peut, dans certains cas, se traduire par un écart important des solutions pour l'ajustement de faisceaux projectif. Pour le test réalisé sur le cylindre, nous observons que le temps de calcul de la méthode des **8-points** est supérieur à la seconde. Ceci est dû au grand nombre de points appariés qui conduisent à la résolution d'un système linéaire de grande taille. Cependant, comme les points sont précisément appariés, ce système est bien conditionné. Mais la qualité de la matrice fondamentale estimée avec la méthode des **8-points** n'est pas suffisante pour initialiser correctement l'ajustement de faisceaux. Dans le même temps, même si le nombre d'itérations est plus grand, notre méthode globale fournit une bonne estimation, car l'erreur de reprojection finale est de 0.25 pixels. En outre, le temps de calcul reste constant (environ 2 secondes). Pour la dernière paire d'images prises par un endoscope, nous observons que la matrice fondamentale estimée par notre méthode globale est de bonne qualité, car l'erreur de reprojection finale est de 0.93 pixels. Dans le même temps, l'ajustement de faisceaux met plus d'itérations à converger vers une solution moins précise lorsque nous l'initialisons avec les matrices de projections issues de $\mathbf{F}_{8pt}$.

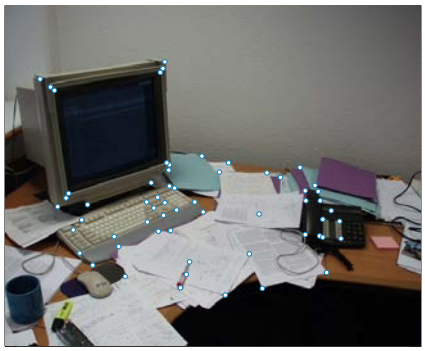
Afin d'approfondir notre étude, nous avons testé l'influence de l'amplitude du mouvement sur la qualité des estimations proposées par les deux approches. Pour cela nous avons utilisé deux séries pour lesquelles les images ont été prises par une caméra dont les centres optiques sont coplanaires. Ensuite, nous avons utilisé le mode opératoire suivant : nous avons estimé

la matrice fondamentale pour toutes les paires d'images possibles $0-1, 1-2, 2-3, \ldots$ puis nous avons calculé les trois critères d'évaluation pour chacune des matrices obtenues. Ceci nous fournit donc la moyenne des critères d'évaluation en fonction d'un angle moyen entre les centres optiques des caméras (l'angle entre deux vues pouvant être extrait des matrices de projections perspectives). Ensuite, nous renouvelons le processus pour les paires d'images $0-2, 1-3, 2-4, \ldots$ et ainsi de suite jusqu'aux paires où nous disposons d'au moins 10 appariements. Comme dans la sous-section 4.1.2, nous pouvons, avec ce procédé, mesurer l'influence de l'angle moyen sur les moyennes des critères d'évaluation pour chacune des deux méthodes. La figure 4.5 résume ce processus. Nous avons réalisé ce test sur deux séries d'images, **Dinosaur** (36 vues) et **House** (9 vues). Ces séries sont mises à disposition par le Visual Geometry Group d'Oxford et fournies dans les Annexes B et D. Les figures de 4.18 à 4.23 montrent la moyenne des erreurs reprojection ainsi que le nombre d'itérations moyen par rapport à l'angle moyen pour les deux séries d'images.

Nous pouvons remarquer que l'erreur de reprojection finale est toujours plus faible lorsque l'ajustement de faisceaux est initialisé avec $\mathbf{F}_{Gp}$. Ensuite, on observe logiquement que plus l'angle moyen augmente, plus l'erreur de reprojection finale se dégrade. Cependant, ce comportement est atténué lorsque l'ajustement de faisceaux est initialisé avec $\mathbf{F}_{Gp}$. Nous observons également que, dans certains cas, l'erreur de reprojection initiale est plus faible avec $\mathbf{F}_{8pt}$ qu'avec $\mathbf{F}_{Gp}$. Ceci peut être dû au fait que l'initialisation issue de $\mathbf{F}_{8pt}$ est dans un bassin d'attraction d'un minimum local de moins bonne qualité que celui du bassin d'attraction associée à $\mathbf{F}_{Gp}$. Mais, dans ce cas, la distance séparant l'estimée initiale et le minimum local est plus importante pour $\mathbf{F}_{Gp}$ que pour $\mathbf{F}_{8pt}$, car le nombre d'itération est plus importante pour $\mathbf{F}_{Gp}$ que pour $\mathbf{F}_{8pt}$.

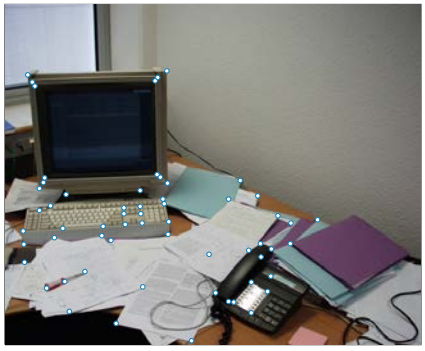
4.1.4.4 Conclusion

Nous avons, dans cette section, exposé le couplage de notre méthode d'optimisation globale à un problème d'estimation de la matrice fondamentale. Après avoir exposé l'état de l'art des méthodes d'estimation, nous avons motivé le choix du problème sur lequel nous avons réalisé ce couplage. Nous avons ensuite détaillé la méthode concurrente que nous avons choisie pour comparer notre algorithme, ainsi que les critères utilisés pour évaluer leurs performances respectives. Les résultats, issus de cas réels et synthétiques, ont montré que l'algorithme global se révélait plus précis et plus robuste. De surcroit, les temps de calculs, qui sont souvent le défaut des méthodes d'optimisation globale, restent raisonnables et autorisent une utilisation pratique de notre approche.



— A —

— B —

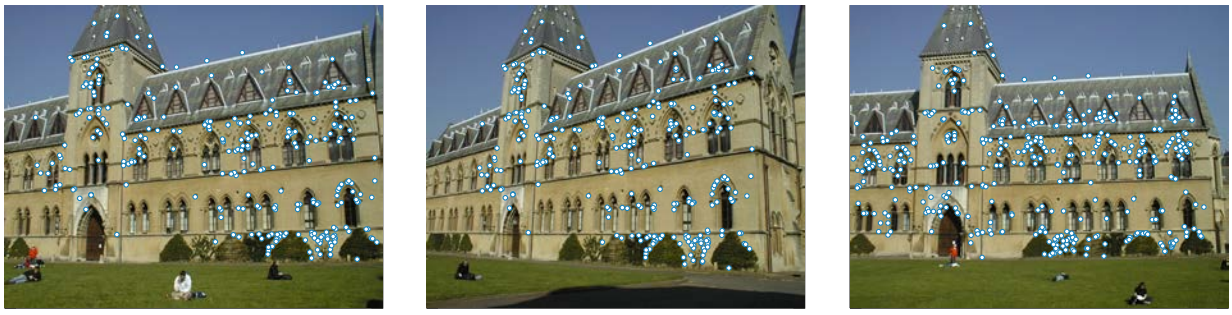


— C —

— D —

Vues		Epipoles		e_{Init}		e_{BA}		Iter($\mathbf{F}$)		Temps	
		e	e′	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$
A	B	$\not\infty$	∞	0.597617	0.65270	0.00252	0.00252	6	6	0.017	2.39
A	C	$\not\infty$	∞	5.61506	5.61996	2.48258	0.00342	122	175	0.017	1.98
A	D	$\not\infty$	$\not\infty$	21.0855	21.5848	4.74837	0.00344	105	30	0.017	2.12
B	C	∞	∞	2.49098	1.91136	0.00260	0.00260	17	12	0.018	1.97
B	D	∞	$\not\infty$	22.0071	23.6253	0.00268	0.00268	122	81	0.018	1.92
C	D	∞	$\not\infty$	28.6586	28.6174	16.6507	0.25921	39	1001	0.018	2.1

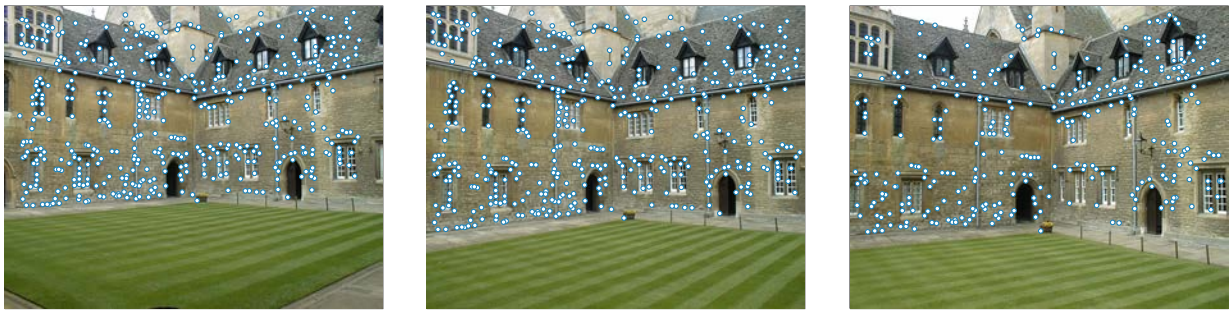
FIGURE 4.11 – Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images A, B, C et D



1 2 3

Vues		nb Points	e_{Init}		e_{BA}		Iter($\mathbf{F}$)		Temps	
			$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$
1	2	310	18.27449	18.32095	11.39809	11.287591	65	57	0.035	2.05
2	3	439	61.84817	63.79435	41.81854	40.2224	29	28	0.020	2.22
1	3	439	42.67438	42.23867	28.65971	27.90128	40	32	0.043	2.10

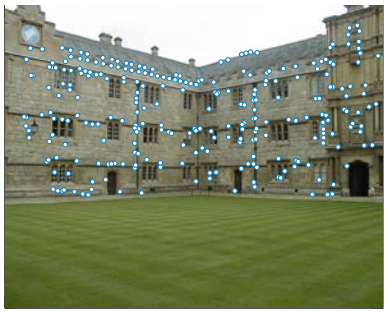
FIGURE 4.12 – Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images de la série **Library**



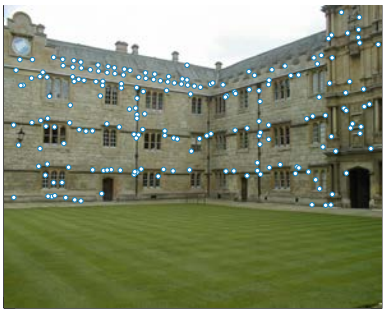
1 2 3

Vues		nb Points	e_{Init}		e_{BA}		Iter($\mathbf{F}$)		Temps	
			$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$
1	2	485	15.1771	15.3126	9.8859	9.7360	77	55	0.048	2.02
2	3	439	61.84817	63.79435	41.81854	40.2224	29	28	0.020	2.22
1	3	384	51.3128	2.3690	7.5245	0.56180	5	52	0.040	2.71

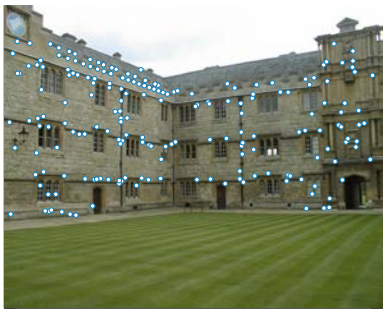
FIGURE 4.13 – Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images de la série **Merton1**



1



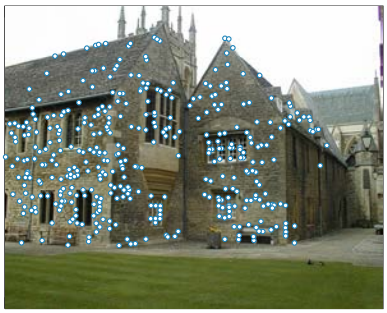
2



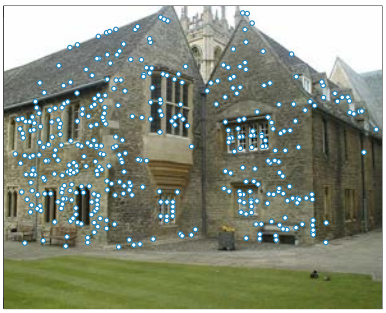
3

Vues		nb Points	e_{Init}		e_{BA}		Iter($\mathbf{F}$)		Temps	
			$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$
1	2	345	24.7158	29.6530	14.0481	2.7599	69	42	0.037	2.91
2	3	197	143.431	154.880	73.8776	72.8977	14	19	0.026	2.52
1	3	270	59.8109	77.9734	30.3824	15.6367	38	32	0.031	2.37

FIGURE 4.14 – Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images de la série **Merton2**



1

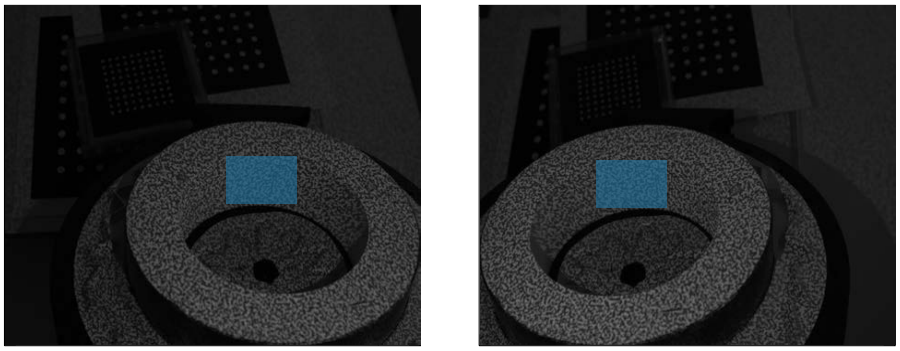


2

3

Vues		nb Points	e_{Init}		e_{BA}		Iter($\mathbf{F}$)		Temps	
			$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$
1	2	400	62.7290	68.6030	52.6557	23.1740	5	32	0.041	3.03
2	3	197	135.950	140.801	83.3365	76.8614	13	14	0.025	2.52
1	3	264	116.7659	118.1754	24.7184	13.5496	9	11	0.032	2.39

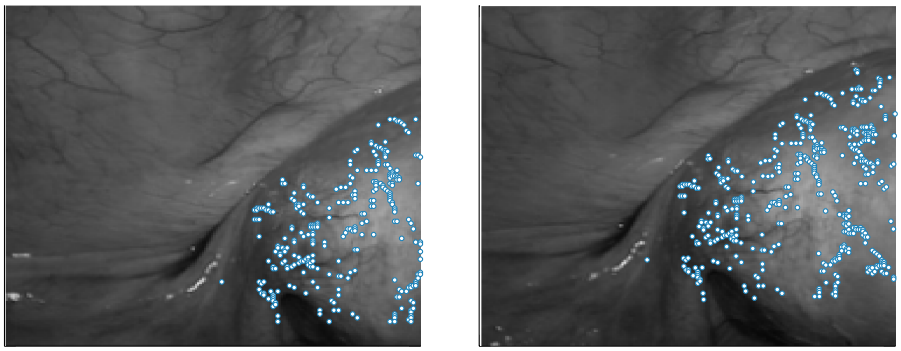
FIGURE 4.15 – Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus en combinant les paires d'images de la série **Merton3**



1 2

Vues		nb Points	e_{Init}		e_{BA}		Iter($\mathbf{F}$)		Temps	
			$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$
1	2	6609	6.3142	6.4021	1.4701	0.2531	401	237	1.23	2.33

FIGURE 4.16 – Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus pour la paire d'images **Cylindre**. Les points appariés sont situés dans la boite englobante bleue.



1 2

Vues		nb Points	e_{Init}		e_{BA}		Iter($\mathbf{F}$)		Temps	
			$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$	$\mathbf{F}_{8Pt}$	$\mathbf{F}_{Gp}$
1	2	730	5.6624	5.6366	3.8566	0.9386	163	401	0.07	1.69

FIGURE 4.17 – Erreur de reprojection avant (e_{Init}) et après (e_{BA}) ajustement de faisceaux, nombre d'itérations (Iter($\mathbf{F}$)), et temps de calcul pour estimer $\mathbf{F}$ (Temps), obtenus pour la paires d'images **Endoscope**

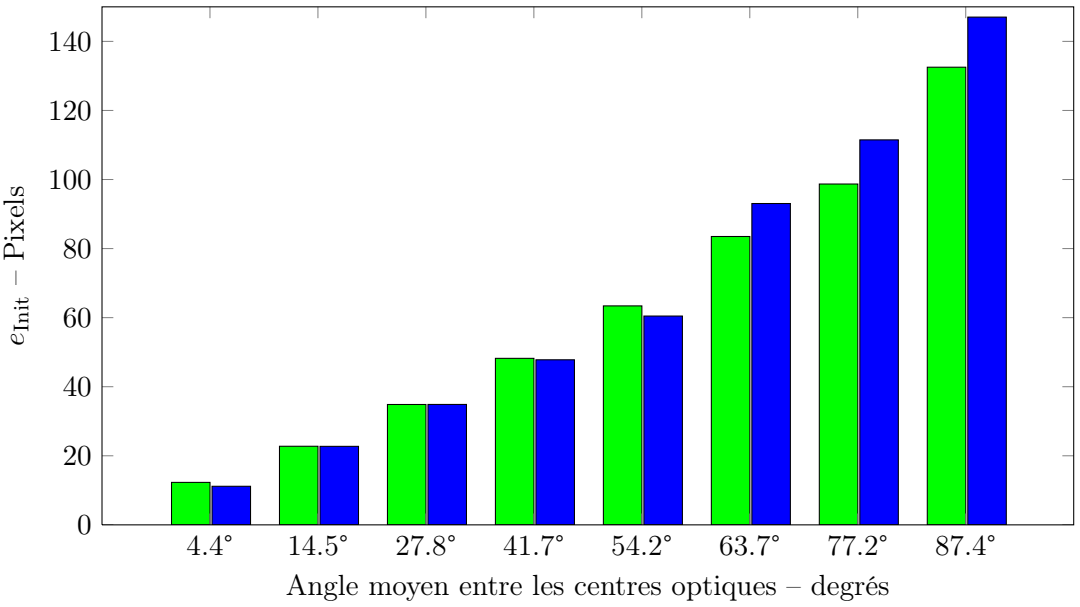


FIGURE 4.18 – Moyenne des erreurs de reprojection initiales en fonction de l'angle moyen entre les centres optiques des caméras pour la série **Dinosaur**

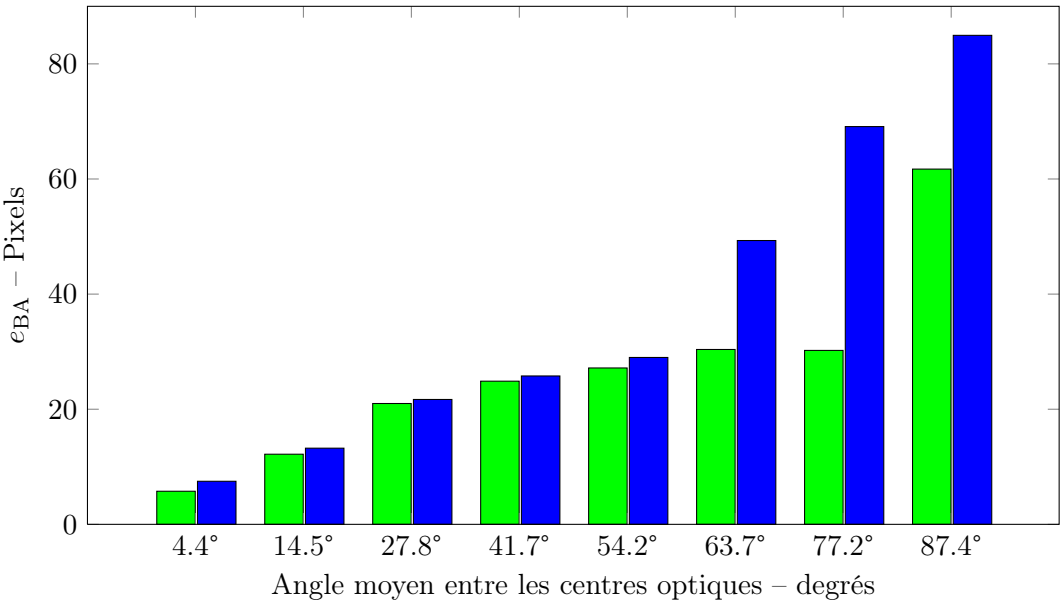


FIGURE 4.19 – Moyenne des erreurs de reprojection finales en fonction de l'angle moyen entre les centres optiques des caméras pour la série **Dinosaur**

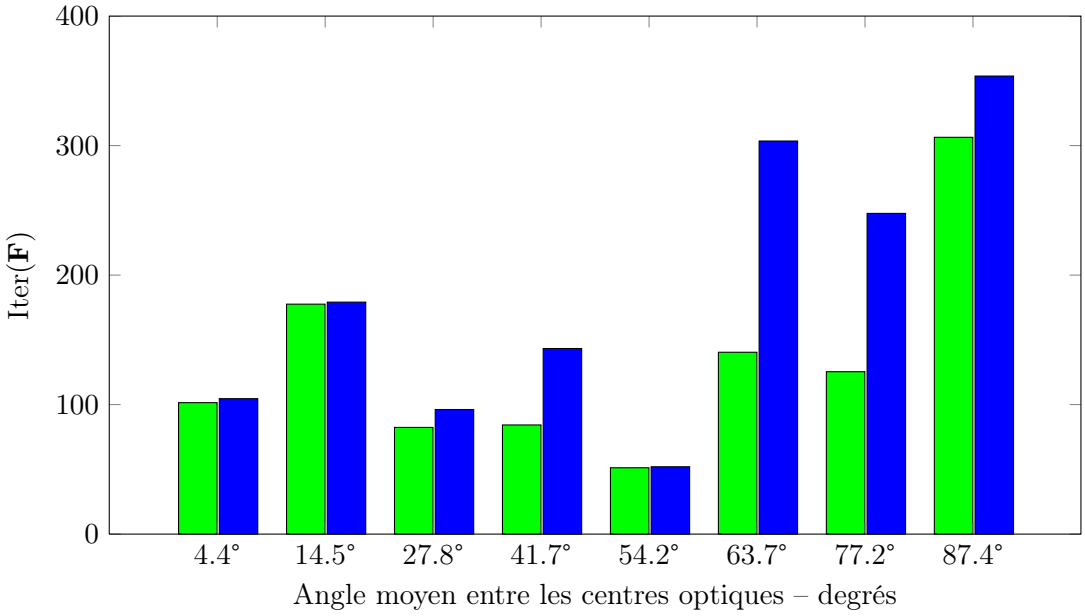


FIGURE 4.20 – Nombre d'itérations moyen effectués par l'ajustement de faisceaux en fonction de l'angle moyen entre les centres optiques des caméras pour la série **Dinosaur**

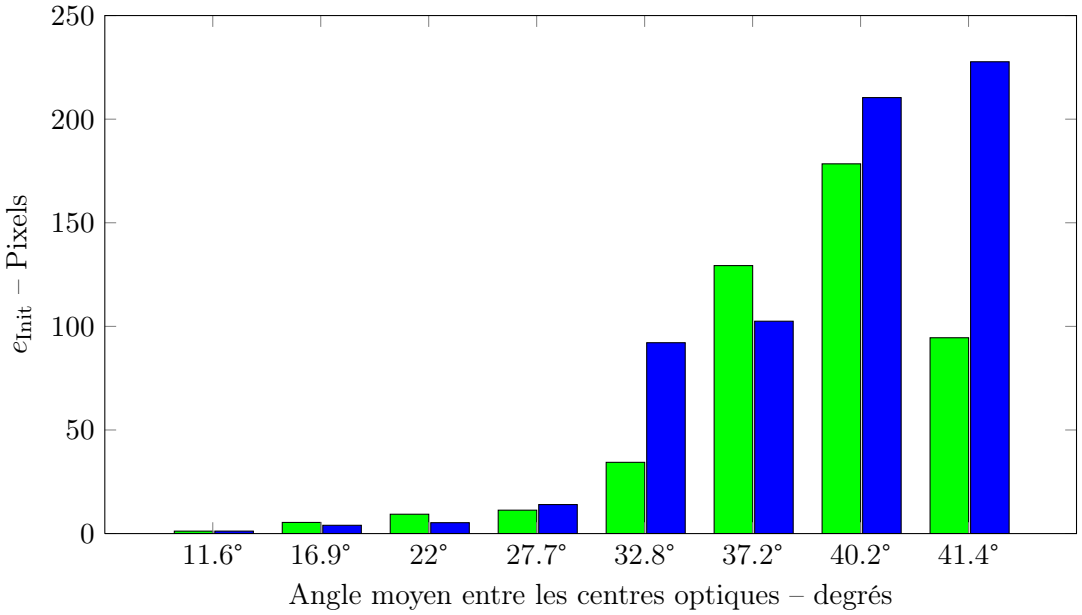


FIGURE 4.21 – Moyenne des erreurs de reprojection initiales en fonction de l'angle moyen entre les centres optiques des caméras pour la série **House**

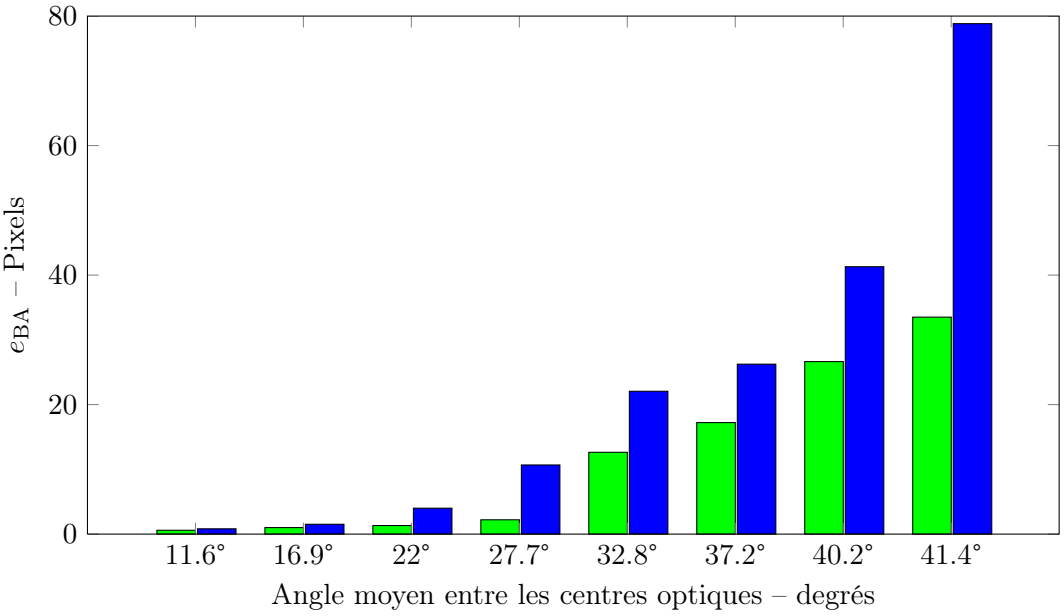


FIGURE 4.22 – Moyenne des erreurs de reprojection finales en fonction de l'angle moyen entre les centres optiques des caméras pour la série **House**

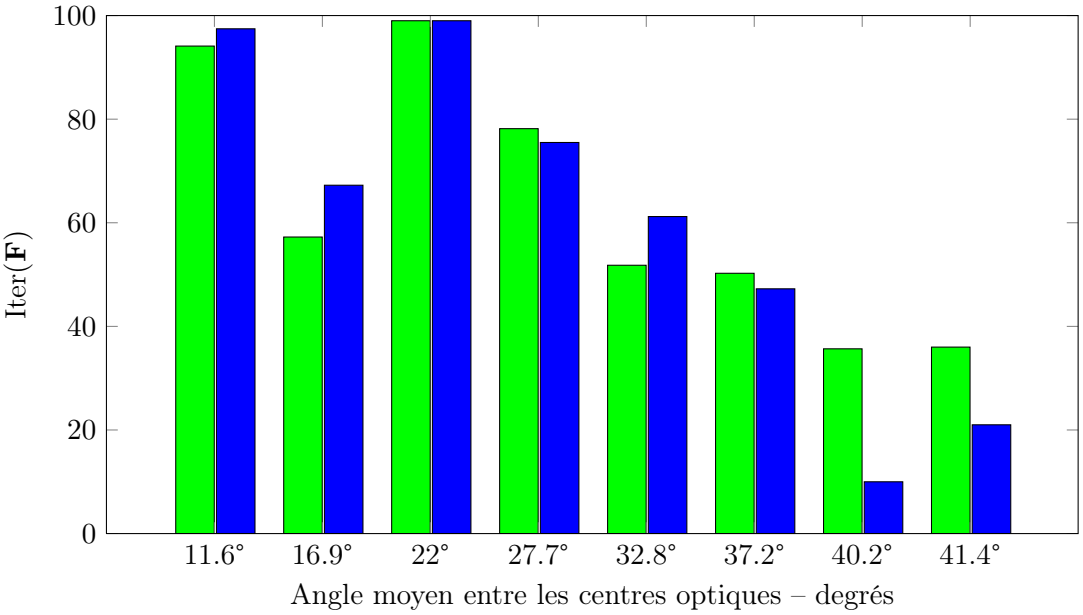


FIGURE 4.23 – Nombre d'itérations moyen effectués par l'ajustement de faisceaux en fonction de l'angle moyen entre les centres optiques des caméras pour la série **House**

4.2 Application de la programmation rationnelle à la triangulation multivues

4.2.1 Rappels de reconstruction 3D multivues

L'objectif de ce paragraphe est de rappeler les principes mathématiques de la reconstruction 3D multivues. Plus largement, nous présenterons dans ce paragraphe la modélisation d'une caméra puis son extension au cas générique de n caméras. Le but de l'étape de reconstruction 3D est de déterminer les coordonnées, dans un repère donné, de chaque point d'un nuage 3D. Nous distinguerons ici deux types de problèmes qui, une fois résolus, fournissent les coordonnées 3D recherchées : **l'ajustement de faisceaux** et la **triangulation**. Mathématiquement, ces deux problèmes sont issus du même problème d'estimation au sens des moindres carrés dans lequel plus ou moins d'inconnues sont fixées. Pour l'ajustement de faisceaux certains, voire tous les paramètres autres que les coordonnées 3D sont des inconnues, tandis que pour la triangulation, seules les coordonnées sont des variables d'optimisation. Ces autres coefficients sont dits de **calibrage** dont on distingue deux catégories : les paramètres **intrinsèques** et les paramètres **extrinsèques**. Les paramètres **intrinsèques** représentent les données internes d'une caméra tels que la taille et la forme des pixels ou les coefficients permettant de corriger les distorsions géométriques induites par l'objectif. Les paramètres **extrinsèques**, représentés par une transformation **rigide** (i.e. composée d'une rotation et d'une translation) décrivent la position et l'orientation d'une caméra par rapport au repère de la scène. Ces paramètres sont nécessaires à l'obtention d'un point 3D par triangulation à partir de points appariés dans différentes images. Cependant, comme nous le verrons par la suite, il peuvent aussi être estimés en même temps que les points 3D.

Nous introduirons dans un premier temps la modélisation mathématique permettant de transformer un point 3D de la scène à observer en un pixel de l'image. Puis, nous étendrons cette modélisation au cas d'un banc stéréoscopique (2 caméras) et enfin au cas de n caméras. L'ensemble de ces modélisations nous permettra d'introduire le principe du calibrage. Nous détaillerons pour chaque cas le problème d'optimisation numérique associé.

La formation d'une image. Une caméra est un système composé de deux parties distinctes : un détecteur et un système optique (objectif) focalisant la source (scène à observer) sur le détecteur. Si dans un premier temps on néglige les effets géométriques du système optique, on obtient une modélisation parfaite appelée **sténopé**. Dans cette modélisation, la transformation $\mathbf{P}$ qui associe un point Q de l'espace 3D à observer $\mathcal{R}^w$ à un pixel q de l'image, est linéaire dans l'espace projectif $\mathbb{P}^3$. Elle est composée de trois transformations distinctes. Tout d'abord, une transformation $\mathbf{T}$ qui déplace Q du répère de la scène $\mathcal{R}^w$ vers le repère de la caméra $\mathcal{R}^c$ dont l'origine est son centre optique C. On notera $\tilde{\mathsf{Q}}$, le point 3D ainsi obtenu. Cette transformation est définie par la composée d'une rotation $\mathbf{R} \triangleq (r_{ij})_{i,j}$ et d'une translation $\mathsf{t} \triangleq (t_x, t_y, t_z)$ et representée, dans $\mathbb{P}^3$, par la matrice $\mathbf{T}$:

$$\mathbf{T} \triangleq \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.83}$$

Les coefficients de $\mathbf{T}$ sont les paramètres **extrinsèques** décrivant la position et l'orientation de la caméra par rapport au repère de la scène. Ensuite, une transformation, fonction de l'optique, projettant $\tilde{\mathsf{Q}}$ sur le repère du capteur, appelé plan **rétinien**. Cette transformation,

notée $\mathbf{P}_f$, est linéaire de $\mathbb{P}^3$ dans $\mathbb{P}^2$. Elle est donc représentée par matrice 3×4 définie par :

$$\mathbf{P}_f \triangleq \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.84}$$

où f, la distance du plan image au centre optique, donne le facteur d'échelle de la projection. Enfin, une dernière transformation linéaire $\mathbf{A}$, de $\mathbb{P}^2$ dans $\mathbb{P}^2$, représentant la conversion des coordonnées rétiniennes en pixels. Elle est définie par :

$$\mathbf{A} \triangleq \begin{pmatrix} \alpha_x & s & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{4.85}$$

où :

- α_x et α_y représentent le nombre de pixels par unité de mesure (exprimé en pixels par millimètres) suivant les vecteurs directeurs du plan image $\overrightarrow{u}$ et $\overrightarrow{v}$.
- Le paramètre $s \triangleq \alpha_x \cot(\widehat{u\,v})$, appelé **facteur d'inclinaison** (**skew factor** en anglais), exprime le défaut d'orthogonalité des vecteurs directeurs du plan image $\overrightarrow{u}$ et $\overrightarrow{v}$. Cet angle est généralement considéré comme étant égal à 90°, conduisant à fixer s à zéro.
- Le point $c = (c_x, c_y)$ est appelé **le point principal** et correspond à la projection du centre optique C de la caméra sur le plan image. Sa position est, en pratique, proche du centre de l'image.

La composition des deux transformations précédentes génère $\mathbf{K}$ la matrice des paramètres **intrinsèques** d'une caméra :

$$\mathbf{K} \triangleq \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cong \mathbf{A}\mathbf{P}_f, \tag{4.86}$$

avec $f_x \triangleq \alpha_x f$ et $f_y \triangleq \alpha_y f$. L'ensemble des variables (f_x, f_y, c_x, c_y, s) ainsi sont appelés les paramètres **intrinsèques** de la caméra. Ainsi, la composition de ces trois transformations permet d'associer Q un point 3D de la scène observée, à q, un pixel de l'image. Elle est linéaire de $\mathbb{P}^3$ dans $\mathbb{P}^2$ et s'écrit :

$$\begin{aligned} &\mathbb{P}^3 \mapsto \mathbb{P}^2 \\ &\mathsf{Q} \mapsto \mathsf{q} \cong \underbrace{\mathbf{A}\mathbf{P}_\mathbf{f}\mathbf{T}}_{\triangleq \mathbf{P}} \mathsf{Q}. \end{aligned} \tag{4.87}$$

La matrice $\mathbf{P} \in \mathcal{M}_{3,4}(\mathbb{R})$ est appelée **matrice de projection perspective** associée à la caméra. Cette application peut aussi être vue comme une fonctionelle rationnelle de $\mathbb{R}^3$ dans $\mathbb{R}^2$, notée $f_\mathbf{P}$, définie par :

$$\begin{aligned} \mathrm{f}_\mathbf{P} : \mathbb{R}^3 &\mapsto \mathbb{R}^2 \\ \mathsf{Q} &\mapsto \mathsf{q} = \begin{pmatrix} f_x \dfrac{r_{11}\mathsf{Q}_1 + r_{12}\mathsf{Q}_2 + r_{13}\mathsf{Q}_3 + t_x}{r_{31}\mathsf{Q}_1 + r_{32}\mathsf{Q}_2 + r_{33}\mathsf{Q}_3 + t_z} + c_x \\ \\ f_y \dfrac{r_{21}\mathsf{Q}_1 + r_{22}\mathsf{Q}_2 + r_{23}\mathsf{Q}_3 + t_y}{r_{31}\mathsf{Q}_1 + r_{32}\mathsf{Q}_2 + r_{33}\mathsf{Q}_3 + t_z} + c_y \end{pmatrix}. \end{aligned} \tag{4.88}$$

ou de manière équivalente à l'aide la matrice de projection perspective $\mathbf{P} = (p_{i,j})_{i,j}$:

$$\begin{aligned} \mathrm{f}_\mathbf{P} : \mathbb{R}^3 &\mapsto \mathbb{R}^2 \\ \mathsf{Q} &\mapsto \mathsf{q} = \begin{pmatrix} \dfrac{p_{11}\mathsf{Q}_1 + p_{12}\mathsf{Q}_2 + p_{13}\mathsf{Q}_3 + p_{14}}{p_{31}\mathsf{Q}_1 + p_{32}\mathsf{Q}_2 + p_{33}\mathsf{Q}_3 + p_{34}} \\ \\ \dfrac{p_{21}\mathsf{Q}_1 + p_{22}\mathsf{Q}_2 + p_{23}\mathsf{Q}_3 + p_{24}}{p_{31}\mathsf{Q}_1 + p_{32}\mathsf{Q}_2 + p_{33}\mathsf{Q}_3 + p_{34}} \end{pmatrix}. \end{aligned} \tag{4.89}$$

Notons que pour la suite du document, on identifiera indifféremment $\mathsf{f}_{\mathbf{P}}$ avec la fonctionnelle $(\mathbf{K}, \mathbf{T}, \mathsf{Q}) \rightarrow \mathsf{f}_{\mathbf{P}}(\mathbf{K}, \mathbf{T}, \mathsf{Q})$. L'ensemble de ces trois transformations sont représentées sur la figure 4.24.

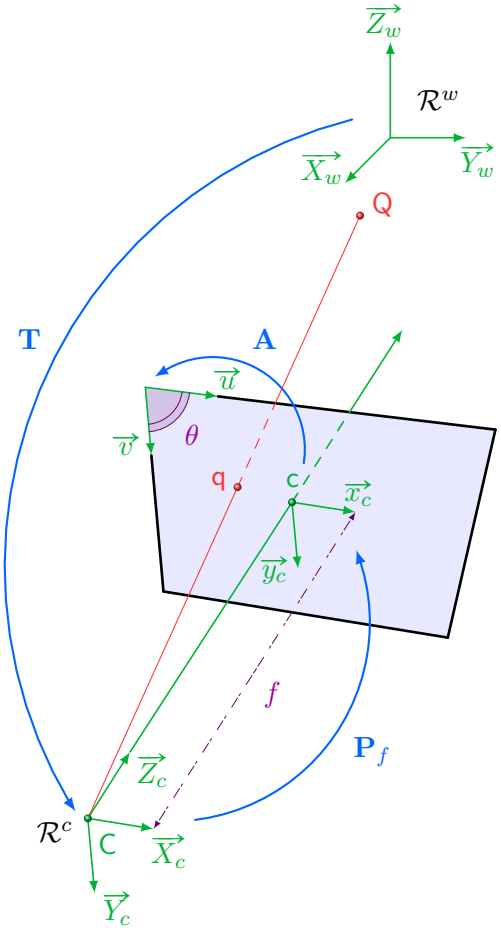


FIGURE 4.24 – Les trois transformations linéaires du modèle **sténopé**. La première envoie le repère $\mathcal{R}^w$ de la scène dans celui centré sur la caméra $\mathcal{R}^c$. La seconde projette $\mathcal{R}^c$ dans le plan **rétinien**. Enfin la dernière, exprime le repère ainsi projeté dans le repère de l'image.

Cependant, ce modèle linéaire ne permet pas de prendre en compte les distorsions générées par les lentilles de l'objectif. Bien qu'il existe plusieurs modèles pour corriger ces défauts [Cornille 2005], nous nous limiterons, pour les besoins applicatifs de notre approche globale, au modèle dit **paramétrique**. Ce modèle permet de corriger plusieurs types de distorsions : **radiales**, **prismatiques** et de **décentrage**. Dans ce modèle, un point $\mathsf{m} = (x, y)^\top$ dans le repère du capteur $(\overrightarrow{x_c}, \overrightarrow{y_c})$ est distordu en un point $\overline{\mathsf{m}} = (\overline{x}, \overline{y})^\top$ au moyen de la transformation polynomiale suivante :

$$\begin{array}{rccl} \mathsf{D}: & \mathbb{R}^2 & \mapsto & \mathbb{R}^2 \\ & \mathsf{m} & \mapsto & \overline{\mathsf{m}} \triangleq \mathsf{m} + \Delta_r(\mathsf{m}) + \Delta_d(\mathsf{m}) + \Delta_p(\mathsf{m}) \end{array} \tag{4.90}$$

où :

– Δ_r défini par :

$$\Delta_r(\mathsf{m}) \triangleq \begin{pmatrix} x \sum_{i=1}^{3} \kappa_i (x^2 + y^2)^i \\ y \sum_{i=1}^{3} \kappa_i (x^2 + y^2)^i \end{pmatrix} \tag{4.91}$$

représente le vecteur de correction radiale associé au vecteur de distorsions $(\kappa_1, \kappa_2, \kappa_2)^\top$

– Δ_d défini par :

$$\Delta_d(\mathsf{m}) \triangleq \begin{pmatrix} 2d_1xy + d_2(3x^2 + y^2) \\ 2d_1xy + d_2(x^2 + 3y^2) \end{pmatrix} \tag{4.92}$$

représente le vecteur de correction de **décentrage** associé au vecteur de distorsions $(d_1, d_2)^\top$

– Δ_d défini par :

$$\Delta_p(\mathsf{m}) \triangleq \begin{pmatrix} p_1(x^2 + y^2) \\ p_2(x^2 + y^2) \end{pmatrix} \tag{4.93}$$

représente le vecteur de correction **prismatique** associé au vecteur de distorsions $(p_1, p_2)^\top$.

Ce modèle est généralement désigné sous la forme d'un acronyme (cf. [Cornille 2005]). Par exemple, pour le modèle complet que nous venons de présenter, l'acronyme est **R3D1P1**. En pratique, on se contente souvent d'utiliser un modèle radial, noté **R3**, variant de 1 à 3 paramètres. Lorsque les distorsions sont prises en compte, le vecteur de coefficients les représentant vient s'ajouter aux variables (f_x, f_y, c_x, c_y, s) dans le vecteur de paramètres **intrinsèques**. L'application de ces corrections intervient après l'application de l'opérateur de projection canonique Ψ_2 de $\mathbb{P}^2$ dans $\mathbb{R}^2$ (cf. (4.19)) et avant l'application de la matrice de paramètres intrinsèques $\mathbf{K}$, donnant lieu à la transformation suivante :

$$\begin{array}{rcl} \mathbb{P}^3 & \mapsto & \mathbb{P}^2 \\ \mathsf{Q} & \mapsto & \mathsf{q} \cong \mathbf{K} \begin{pmatrix} \mathsf{D}(\Psi_2(\mathbf{TQ})) \\ 1 \end{pmatrix}. \end{array} \tag{4.94}$$

Cette application n'est donc plus linéaire de $\mathbb{P}^3$ dans $\mathbb{P}^2$, mais se développe en une fonctionnelle rationnelle de $\mathbb{R}^3$ dans $\mathbb{R}^2$, notée f_D, donnée par :

$$\begin{array}{rcl} \mathrm{f}_\mathsf{D} : \mathbb{R}^3 & \mapsto & \mathbb{R}^2 \\ \mathsf{Q} & \mapsto & \mathsf{q} = \begin{pmatrix} f_x\mathsf{D}_1(\Psi_2(\mathbf{TQ})) + c_x \\ f_y\mathsf{D}_2(\Psi_2(\mathbf{TQ})) + c_y \end{pmatrix} \end{array} \tag{4.95}$$

avec $\mathsf{D}_i \triangleq \pi_i \circ \mathsf{D}$ où π_i désigne la i$^{\text{ème}}$ projection canonique. A l'instar de $\mathrm{f}_\mathbf{P}$, on identifiera indifféremment f_D et $(\mathbf{K}, \kappa, \mathbf{T}, \mathsf{Q}_i) \rightarrow \mathrm{f}_\mathsf{D}(\mathbf{K}, \kappa, \mathbf{T}, \mathsf{Q})$. Précisons ensuite que, dans la définition de f_D, $\mathbf{TQ}$ est exprimé dans $\mathbb{R}^3$ puisque sa dernière coordonnée n'est jamais nulle. En effet, si tel était le cas, le point $\mathbf{TQ}$ serait dans le plan du capteur $(\mathsf{C}, \overrightarrow{X}_c, \overrightarrow{Y}_c)$, ce qui est physiquement irréaliste. Il est, en outre, important de constater que les degrés des numérateurs et dénominateurs de la fonctionnelle f_D entrainent des comportements fortement non-linéaires que nous développerons par la suite.

Suivant la modélisation introduite précédemment, **calibrer** une caméra, représentée à l'aide d'un modèle radial **R3**, consiste à déterminer les paramètres (f_x, f_y, c_x, c_y, f), les paramètres de distorsions radiales $(\kappa_1, \kappa_2, \kappa_3)$ ainsi que la matrice de rotation $\mathbf{R}$ et le vecteur de translation t. En pratique, certains paramètres peuvent être fixés comme (c_x, x_y) que l'on peut contraindre à se situer au centre de l'image. Notons que dans le cas où l'on néglige les distorsions de l'objectif, on peut déterminer directement les 12 coefficients de la matrice $\mathbf{P}$. Pratiquement, afin d'estimer ces paramètres, on dispose d'un ensemble de p points 3D $(\mathsf{Q}_i)_{i=1\cdots p}$ fournis par une mire étalon. Cette mire est déplacée q fois devant la caméra. A chaque position j de la mire est associé $\mathcal{R}_j^w$ un repère 3D et, par conséquent, une transformation rigide $\mathbf{T}_j$ permettant de localiser $\mathcal{R}_j^w$ par rapport à $\mathcal{R}^c$. Notons que, pour chaque déplacement, $\mathcal{R}_j^w$ est un repère associé à la mire dans lequel elle se situe sur le plan $(\overrightarrow{X}_{w_j}, \overrightarrow{Y}_{w_j}, 0)$. On suppose

connues, pour chaque position j de la mire, les p projections $(\mathsf{q}_i^j)_{i=1\dots p}$ des points 3D $(\mathsf{Q}_i)_{i=1\cdots p}$. Il y a donc en tout $p \times q$ données mesurées. Dénombrons alors le nombre d'inconnues : $(9+3)q$ pour les paramètres de chaque transformation rigide $\mathbf{T}_j$, 5 paramètres intrinsèques et les $2p$ coordonnées spatiales des points de la mire (puisque chaque coordonnée selon $\overrightarrow{Z}_{w_j}$ est nulle). Il y a donc en tout $2pq$ équations pour $12q + 2p + 4$ inconnues. Remarquons que ce nombre diminue lorsque la rotation $\mathbf{R}_j$ de $\mathbf{T}_j$ est paramétrée à l'aide d'un vecteur de rotation instantané ou d'un quaternion. Ainsi, pour une expérience classique avec une mire comportant 81 points et déplacée 10 fois, il y a 1620 équations pour 286 inconnues. Par conséquent, si on néglige les distorsions géométriques, on dispose alors d'un système linéaire d'inconnues $(f_x, f_y, c_x, c_y, (\mathsf{Q}_i)_i)$ largement surdéterminé :

$$\mathsf{q}_i^j \quad \cong \quad \mathbf{K}\mathbf{T}_j\mathsf{Q}_i \qquad \forall (i,j) \in \{1,\dots,p\} \times \{1,\dots,q\}\,. \tag{4.96}$$

Ce système, résolu à l'aide d'une méthode classique d'algèbre linéaire comme la Décomposition en Valeurs Singulières, fournit une première estimation des paramètres intrinsèques. Cependant, attendu que les données mesurées peuvent être entachées d'erreurs, ce système peut être mal conditionné. Il convient donc de raffiner les solutions de (4.96) en résolvant le problème d'estimation au sens des moindres carrés suivant :

$$\min_{\mathbf{K},\mathbf{T}_j,\mathsf{Q}_i} \sum_{j=1}^{q}\sum_{i=1}^{p} \|\mathsf{q}_i^j - \mathrm{f}_{\mathbf{P}}(\mathbf{K},\mathbf{T}_j,\mathsf{Q}_i)\|_2^2. \tag{4.97}$$

Il est aussi possible de rajouter le vecteur de distorsions radiales κ parmi les inconnues en utilisant la fonctionnelle f_{D}. On obtient alors le problème :

$$\min_{\mathbf{K},\kappa,\mathbf{T}_j,\mathsf{Q}_i} \sum_{j=1}^{q}\sum_{i=1}^{p} \|\mathsf{q}_i^j - \mathrm{f}_{\mathsf{D}}(\mathbf{K},\kappa,\mathbf{T}_j,\mathsf{Q}_i)\|_2^2. \tag{4.98}$$

Ces deux problèmes sont classiquement résolus à l'aide de l'algorithme de Levenberg-Marquardt initialisé avec la solution de (4.96) et $\kappa = (0,0,0)^\top$.

Une fois la caméra calibrée, **trianguler** consiste à retrouver les coordonnées 3D d'un point connaissant ses projections dans les images. Remarquons que dans le cas mono-caméra, connaître la projection q dans l'image d'un point 3D Q ne suffit pas à retrouver ses coordonnées spatiales. En effet, on ne disposerait alors que de deux équations pour estimer trois inconnues. Afin de pallier ce défaut et d'introduire le problème de la triangulation, étudions la modélisation d'un banc stéréoscopique.

Modélisation d'un banc stéréoscopique. Considérons un système de numérisation composé de deux caméras, notées caméra 1 et caméra 2. La modélisation de ce système se fait en utilisant la modélisation mono-caméra précédente pour chacune des deux caméras et en ajoutant une transformation rigide, dite **stéréoscopique** et notée $\mathbf{T}_{1\to 2}$, permettant de déplacer le repère $\mathcal{R}_1^c$ de la caméra 1 vers $\mathcal{R}_2^c$, celui de la caméra 2. On note :

- $(f_x^k, f_x^k, c_x^k, c_y^k, \kappa^k)$ les paramètres intrinsèques de la $k^{\text{ème}}$ caméra (θ_1 et θ_2 étant considérés comme égaux à 90°). Dans le cas sans distorsions, la matrice de paramètres intrinsèques de la caméra k est notée $\mathbf{K}_k$.
- (r'_{ij}) et (t'_x, t'_y, t'_z) les coefficients de $\mathbf{T}_{1\to 2}$ correspondant respectivement à la rotation $\mathbf{R}'$ et à la translation t'.

L'ensemble de ces transformations sont rappelées sur la figure 4.25. Si on néglige dans un premier temps les distorsions géométriques des deux caméras, la transformation permettant

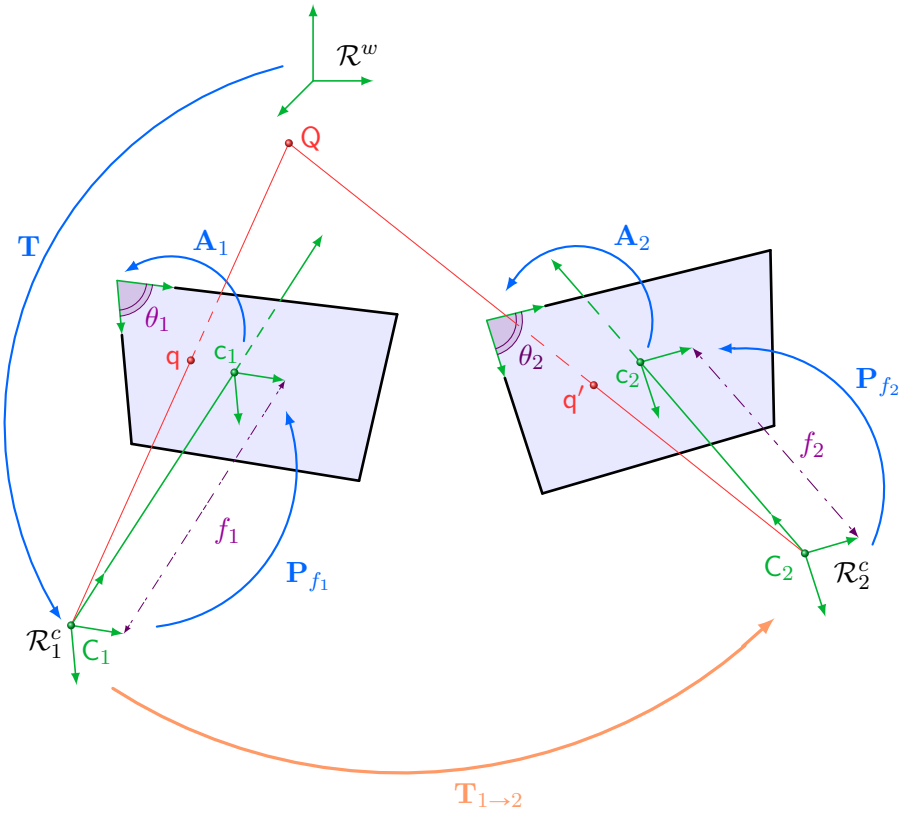


FIGURE 4.25 – Transformations linéaires d'un banc stéréoscopique.

d'associer Q à ses deux projections q et q' dans les images 1 et 2 est linéaire de $\mathbb{P}^3$ dans $\mathbb{P}^2 \times \mathbb{P}^2$. Elle est donnée par l'expression suivante :

$$\begin{aligned} &\mathbb{P}^3 \mapsto \mathbb{P}^2 \times \mathbb{P}^2 \\ &\mathsf{Q} \mapsto \begin{pmatrix} \mathsf{q} \\ \mathsf{q}' \end{pmatrix} \cong \begin{pmatrix} \mathbf{K}_1 \mathbf{T} \mathsf{Q} \\ \mathbf{K}_2 \mathbf{T}_{1\to 2} \mathbf{T} \mathsf{Q} \end{pmatrix}. \end{aligned} \tag{4.99}$$

En outre, il est possible de construire une matrice fondamentale $\mathbf{F}$ à partir de ces transformations :

$$\mathbf{F} = \mathbf{K}_1^{-1} \mathbf{E} \mathbf{K}_2, \tag{4.100}$$

$$\tag{4.101}$$

où la matrice $\mathbf{E}$ est définie par :

$$\mathbf{E} \triangleq \underbrace{\begin{pmatrix} r'_{11} & r'_{12} & r'_{13} \\ r'_{21} & r'_{22} & r'_{23} \\ r'_{31} & r'_{32} & r'_{33} \end{pmatrix}}_{\mathbf{R}'} \underbrace{\begin{pmatrix} 0 & -t'_z & t'_y \\ t'_z & 0 & -t'_x \\ -t'_y & t'_x & 0 \end{pmatrix}}_{\triangleq [\mathsf{t}']_\times}. \tag{4.102}$$

La matrice $\mathbf{E}$ est appelée **matrice essentielle**. A l'instar de la matrice fondamentale, si l'on connait les paramères intrinsèques de chacune des caméras, il existe plusieurs techniques numériques pour estimer $\mathbf{E}$ et retrouver ainsi $\mathbf{R}'$ et t'. Nous ne développerons pas ici ces méthodes mais, pour plus de détails sur ces approches et les propriétés théoriques de $\mathbf{E}$, le lecteur pourra se référer à [Hartley 2003]. Par analogie au cas mono-caméra, la fonctionnelle

rationnelle induite s'écrit :

$$\begin{aligned} &\mathbb{R}^3 \mapsto \mathbb{R}^2 \times \mathbb{R}^2 \\ &\mathsf{Q} \mapsto \begin{pmatrix} \mathsf{q} \\ \mathsf{q}' \end{pmatrix} = \begin{pmatrix} \mathrm{f}_{\mathbf{P}}(\mathbf{K}_1, \mathbf{T}, \mathsf{Q}) \\ \mathrm{f}_{\mathbf{P}}(\mathbf{K}_2, \mathbf{T}_{1\to 2}\mathbf{T}, \mathsf{Q}) \end{pmatrix}. \end{aligned} \tag{4.103}$$

En considérant ensuite les distorsions géométriques, on obtient la fonctionnelle rationnelle :

$$\begin{aligned} &\mathbb{R}^3 \mapsto \mathbb{R}^2 \times \mathbb{R}^2 \\ &\mathsf{Q} \mapsto \begin{pmatrix} \mathsf{q} \\ \mathsf{q}' \end{pmatrix} = \begin{pmatrix} \mathrm{f}_{\mathrm{D}}(\mathbf{K}_1, \kappa^1, \mathbf{T}, \mathsf{Q}) \\ \mathrm{f}_{\mathrm{D}}(\mathbf{K}_2, \kappa^2, \mathbf{T}_{1\to 2}\mathbf{T}, \mathsf{Q}) \end{pmatrix}. \end{aligned} \tag{4.104}$$

où D_i^k désigne la i$^{\text{ème}}$ projection de la fonction de distorsions (4.90) associée à la $k^{\text{ème}}$ caméra. Par analogie au cas mono-caméra, pour une expérience de calibrage où une mire de p points est déplacée q fois devant un banc stéréoscopique, on doit tout d'abord résoudre le système linéaire à $4pq$ équations et $12q + 2p + 2 \times 4 + 12$ inconnues (les douze nouvelles inconnues étant les coefficients de $\mathbf{T}_{1\to 2}$) défini par :

$$\begin{cases} \mathsf{q}_i^j \cong \mathbf{K}_1 \mathbf{T}_j \mathsf{Q}_i \\ \mathsf{q}'^j_i \cong \mathbf{K}_2 \mathbf{T}_{1\to 2} \mathbf{T}_j \mathsf{Q}_i \end{cases} \qquad \forall (i,j) \in \{1,\dots,p\} \times \{1,\dots,q\}. \tag{4.105}$$

La solution obtenue est peut ensuite être raffinée en résolvant :

$$\min_{\mathbf{K}_k, \mathbf{T}_{1\to 2}, \mathbf{T}_j, \mathsf{Q}_i} \sum_{j=1}^{q} \sum_{i=1}^{p} \|\mathsf{q}_i^j - \mathrm{f}_{\mathbf{P}}(\mathbf{K}_1, \mathbf{T}_j, \mathsf{Q}_i)\|_2^2 + \|\mathsf{q}'^j_i - \mathrm{f}_{\mathbf{P}}(\mathbf{K}_2, \mathbf{T}_{1\to 2}\mathbf{T}_j, \mathsf{Q}_i)\|_2^2 \tag{4.106}$$

ou utilisée comme initialisation du problème d'estimation au sens des moindres carrés suivant :

$$\min_{\mathbf{K}_k, \kappa^k, \mathbf{T}_{1\to 2}, \mathbf{T}_j, \mathsf{Q}_i} \sum_{j=1}^{q} \sum_{i=1}^{p} \|\mathsf{q}_i^j - \mathrm{f}_{\mathsf{D}}(\mathbf{K}_1, \kappa^1, \mathbf{T}_j, \mathsf{Q}_i)\|_2^2 + \|\mathsf{q}'^j_i - \mathrm{f}_{\mathsf{D}}(\mathbf{K}_2, \kappa^2, \mathbf{T}_{1\to 2}\mathbf{T}_j, \mathsf{Q}_i)\|_2^2. \tag{4.107}$$

Ce problème, très classique en vision par ordinateur, est appelé **ajustement de faisceaux** (**Bundle Adjustement** en anglais), car les termes $\mathrm{f}_{\mathsf{D}}(\mathbf{K}_1, \kappa^1, \mathbf{T}_j, \mathsf{Q}_i)$ et $\mathrm{f}_{\mathsf{D}}(\mathbf{K}_2, \kappa^2, \mathbf{T}_{1\to 2}\mathbf{T}_j, \mathsf{Q}_i)$ représentent deux faisceaux issus des centres optiques de chacune des caméras et s'intersectant en Q_i. Le but de la résolution de (4.107) est donc d'ajuster itérativement les faisceaux pour que leur intersection avec les plans images soient les plus proches possibles des données mesurées et ce, en jouant sur les paramètres de calibrage et les points 3D. Lorsque les caméras sont calibrées, trianguler un point 3D Q consiste alors à retrouver l'intersection des faisceaux à partir de ses projections q et q'. Classiquement la reconstruction 3D s'effectue dans le repère de la première caméra, la transformation $\mathbf{T}$ est donc égale à l'identité. Ainsi, la formulation linéaire est donnée par :

$$\begin{cases} \mathsf{q} \cong \mathbf{K}_1 \mathsf{Q} \\ \mathsf{q}' \cong \mathbf{K}_2 \mathbf{T}_{1\to 2} \mathsf{Q} \end{cases}. \tag{4.108}$$

où $\mathbf{P}_1$, $\mathbf{P}_2$ et $\mathbf{T}_{1\to 2}$ sont fixées. Par suite, l'étape de raffinage non-linéaire se formule :

$$\min_{\mathsf{Q}} \left(\mathsf{q}_1 - \frac{(\mathbf{K}_1\mathsf{Q})_1}{(\mathbf{K}_1\mathsf{Q})_3}\right)^2 + \left(\mathsf{q}_2 - \frac{(\mathbf{K}_1\mathsf{Q})_2}{(\mathbf{K}_1\mathsf{Q})_3}\right)^2 + \left(\mathsf{q}'_1 - \frac{(\mathbf{K}_2\mathbf{T}_{1\to 2}\mathsf{Q})_1}{(\mathbf{K}_2\mathbf{T}_{1\to 2}\mathsf{Q})_3}\right)^2 + \left(\mathsf{q}'_2 - \frac{(\mathbf{K}_2\mathbf{T}_{1\to 2}\mathsf{Q})_2}{(\mathbf{K}_2\mathbf{T}_{1\to 2}\mathsf{Q})_3}\right)^2. \tag{4.109}$$

A l'instar du calibrage, selon que l'on prenne en compte ou non les distorsions, en résolvant :

$$\min_{\mathbf{K}_k, \mathbf{T}_{1\to 2}, \mathsf{Q}_i} \sum_{i=1}^{p} \|\mathsf{q}_i - \mathrm{f}_{\mathbf{P}}(\mathbf{K}_1, I_4, \mathsf{Q}_i)\|_2^2 + \|\mathsf{q}'_i - \mathrm{f}_{\mathbf{P}}(\mathbf{K}_2, \mathbf{T}_{1\to 2}, \mathsf{Q}_i)\|_2^2, \tag{4.110}$$

ou

$$\min_{\mathbf{K}_k,\kappa^k,\mathbf{T}_{1\to2},\mathsf{Q}_i} \sum_{i=1}^{p} \|\mathsf{q}_i - \mathrm{f}_{\mathsf{D}}(\mathbf{K}_1,\kappa^1,\mathsf{Q}_i)\|_2^2 + \|\mathsf{q'}_i - \mathrm{f}_{\mathsf{D}}(\mathbf{K}_2,\kappa^2,\mathbf{T}_{1\to2},\mathsf{Q}_i)\|_2^2, \tag{4.111}$$

il est possible de reconstruire tous les points par ajustement de faisceaux tout en réestimant l'ensemble des paramètres de calibrage. Remarquons que cette réestimation peut améliorer de manière significative la précision des paramètres de calibrage. Il est important de souligner que la précision finale de (4.110) ou de (4.111) dépend directement de la précision avec laquelle les appariements $(\mathsf{q}_i,\mathsf{q'}_i)_i$ sont calculés. Or, les techniques de corrélation d'images numériques [Sutton 2009, Harvent 2010] permettent, à partir d'un système calibré, des mises en correspondances beaucoup plus fines que les méthodes classiques d'extraction de points de mires. De plus, on dispose à l'aide de ces techniques d'un nombre beaucoup plus important d'appariements. On comprend dès lors tout l'intérêt d'effectuer un recalibrage en même temps que la reconstruction 3D d'un nuage de points. Dans le cas sans distorsions, il est aussi possible de reformuler (4.106) comme un problème d'estimation de pose à l'aide des matrices de projection perspective $\mathbf{P}_1$ et $\mathbf{P}_2$:

$$\min_{\mathbf{P}_1,\mathbf{P}_2,\mathsf{Q}_i} \sum_{i=1}^{p} \left(\mathsf{q}_1 - \frac{(\mathbf{P}_1\mathsf{Q}_i)_1}{(\mathbf{P}_1\mathsf{Q}_i)_3}\right)^2 + \left(\mathsf{q}_2 - \frac{(\mathbf{P}_1\mathsf{Q}_i)_2}{(\mathbf{P}_1\mathsf{Q}_i)_3}\right)^2 + \left(\mathsf{q}'_1 - \frac{(\mathbf{P}_2\mathsf{Q}_i)_1}{(\mathbf{P}_2\mathsf{Q}_i)_3}\right)^2 + \left(\mathsf{q}'_2 - \frac{(\mathbf{P}_2\mathsf{Q}_i)_2}{(\mathbf{P}_2\mathsf{Q}_i)_3}\right)^2. \tag{4.112}$$

Ce problème possède légèrement plus d'inconnues que (4.106) : $24 + 3p$ au lieu de $19 + 3p$. Notons que (4.112) correspond au problème (4.80) succinctement énoncé dans la sous-section 4.1.4.

Par analogie au cas mono-caméra, nous présentons dans le paragraphe suivant la reconstruction 3D multivues comme une extension de la modélisation du système stéréoscopique.

Reconstruction 3D multivues. La modélisation mathématique de l'étape de reconstruction 3D multivues se fait de manière analogue à la modélisation d'un banc stéréoscopique à partir du cas mono-caméra, c'est-à-dire en utilisant le paramétrage mono-caméra auquel on ajoute des transformations rigides afin de localiser le repère des caméras par rapport à un repère de référence. Supposons, sans perte de généralité, que la reconstruction 3D s'effectue dans le repère de la première caméra $\mathcal{R}_1^c$. Raisonnons de manière incrémentale et considérons la modélisation d'un banc stéréoscopique tel que nous l'avons défini plus haut. Si on ajoute une nouvelle caméra à ce système, il est nécessaire de définir une transformation rigide $\mathbf{T}_{1\to3}$ permettant de localiser $\mathcal{R}_3^c$ par rapport à $\mathcal{R}_1^c$. Il existe deux manières de définir une telle transformation : soit directement, soit en définissant une transformation intermédiaire $\mathbf{T}_{2\to3}$ qui repère $\mathcal{R}_3^c$ par rapport à $\mathcal{R}_2^c$ de sorte que $\mathbf{T}_{1\to3} = \mathbf{T}_{2\to3} \circ \mathbf{T}_{1\to2}$. L'ensemble de ces transformations est représenté sur la figure 4.26. Théoriquement ces deux approches sont équivalentes. Mais en pratique, le choix est étroitement lié à la configuration spatiale du système et au choix du repère. En effet, plus le mouvement entre les caméras 1 et 3 sera important ou assujetti à une importante rotation, plus le nombre de correspondants $(\mathsf{q}_i^j,\mathsf{q}''^j_i)$ dont on disposera pour estimer $\mathbf{T}_{1\to3}$ sera faible et entaché d'erreurs (dues par exemple à des variations d'illumination ou une rotation importante de l'image), auquel cas on aura tout intérêt à calculer $\mathbf{T}_{2\to3}$. Cependant, introduire dans la modélisation des compositions de transformations (et donc des multiplications entre leurs coefficients) se fait au détriment de l'augmentation du degré (et donc de la non-linéarité) de la fonctionnelle à minimiser. De plus, lors de la phase de triangulation issue d'une modélisation utilisant de telles compositions, les erreurs d'estimation de $\mathbf{T}_{1\to2}$ et de $\mathbf{T}_{2\to3}$ vont se cumuler sur le calcul des faisceaux de la troisième caméra. On comprend dès lors que si le mouvement entre la caméra 1 et la caméra 3 reste équivalent à celui entre la caméra 2 et la caméra 3, il sera plus avantageux ne pas utiliser

$\mathbf{T}_{2\to3}$ dans la modélisation. Pour un exemple de modélisation et de calibrage d'un système à quatre caméras dans cette dernière configuration, le lecteur pourra se référer à [Orteu 2011].

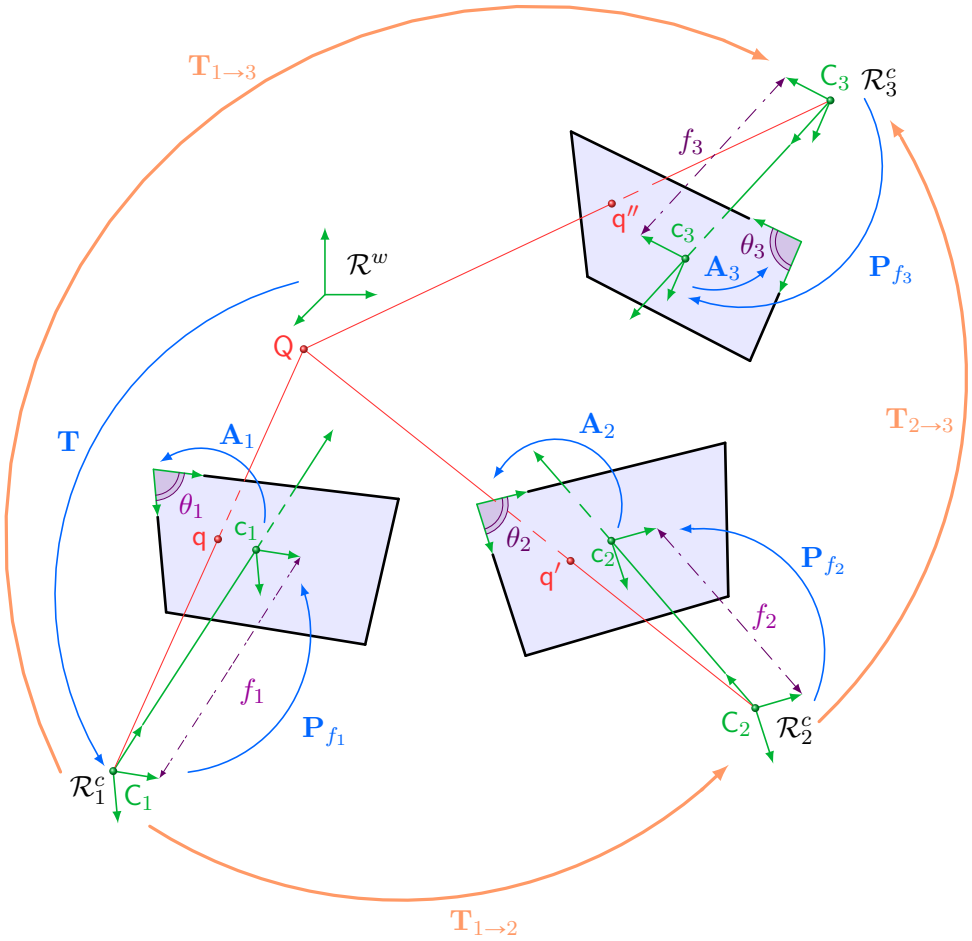


FIGURE 4.26 – Transformations linéaires d'un système multivues. Le repère $\mathcal{R}_c^1$ se déplace vers le repère $\mathcal{R}_c^3$ soit directement par la transformation $\mathbf{T}_{1\to3}$ soit par la composée des applications $\mathbf{T}_{2\to3} \circ \mathbf{T}_{1\to2}$

Comme nous venons de souligner, pour la $k^{\text{ème}}$ caméra, le choix d'introduire ou non, des transformations rigides intermédiaires doit être fait selon la configuration du système de vision. Par conséquent, nous supposons, sans perte de généralité, que ce choix a été fait a priori et notons $\mathbf{T}_k^l$ la matrice de localisation de $\mathcal{R}_k^c$ par rapport au repère de reconstruction. Suivant cette nouvelle nomenclature, pour n vues, la transformation linéaire qui à un point 3D Q associe ses n projections $(\mathsf{q}^k)_{k=1\cdots n}$ dans toutes les vues, est donnée par :

$$\begin{aligned} &\mathbb{P}^3 \mapsto \mathbb{P}^{2n} \\ &\mathsf{Q} \mapsto (\mathsf{q}^k)_{k=1\cdots n} \cong \left(\mathbf{K}_k \mathbf{T}_k^l \mathsf{Q}\right)_{k=1\cdots n}, \end{aligned} \tag{4.113}$$

et la fonctionnelle induite par :

$$\begin{aligned} &\mathbb{R}^3 \mapsto \mathbb{R}^{2n} \\ &\mathsf{Q} \mapsto (\mathsf{q}^k)_{k=1\cdots n} = \left(\mathrm{f_P}(\mathbf{K}_k, \mathbf{T}_k^l, \mathsf{Q})\right)_{k=1\cdots n}. \end{aligned} \tag{4.114}$$

En tenant compte des distorsions, la fonctionnelle associant un point 3D Q à ses projections

s'écrit :

$$\begin{aligned} &\mathbb{R}^3 \mapsto \mathbb{R}^{2n} \\ &\mathsf{Q} \mapsto (\mathsf{q}^k)_{k=1\cdots n} = \Big(\mathrm{f}_{\mathrm{D}}(\mathbf{K}_k, \kappa^k, \mathbf{T}_k^l, \mathsf{Q})\Big)_{k=1\cdots n}. \end{aligned} \tag{4.115}$$

Par conséquent les problèmes d'ajustement de faisceaux stéréoscopiques à p points (4.106) et (4.107) s'étendent au cas de n vues sous les formes :

$$\min_{\mathbf{K}_k, \mathbf{T}_k^l, \mathsf{Q}_i} \sum_{i=1}^{p} \sum_{k=1}^{n} \|\mathsf{q}_i^k - \mathrm{f}_{\mathbf{P}}(\mathbf{K}_k, \mathbf{T}_k^l, \mathsf{Q}_i)\|_2^2 \tag{4.116}$$

et

$$\min_{\mathbf{K}_k, \kappa^k, \mathbf{T}_k^l, \mathsf{Q}_i} \sum_{i=1}^{p} \sum_{k=1}^{n} \|\mathsf{q}_i^k - \mathrm{f}_{\mathsf{D}}(\mathbf{K}_k, \kappa^k, \mathbf{T}_k^l, \mathsf{Q}_i)\|_2^2, \tag{4.117}$$

Notons que le problème de triangulation d'un point Q (4.109) se généralise quant à lui en :

$$\min_{\mathsf{Q}} \sum_{k=1}^{n} \left(\mathsf{q}_1^k - \frac{(\mathbf{K}_k \mathbf{T}_k^l \mathsf{Q})_1}{(\mathbf{K}_k \mathbf{T}_k^l \mathsf{Q})_3}\right)^2 + \left(\mathsf{q}_2^k - \frac{(\mathbf{K}_k \mathbf{T}_k^l \mathsf{Q})_2}{(\mathbf{K}_k \mathbf{T}_k^l \mathsf{Q})_3}\right)^2 \tag{4.118}$$

et le problème d'estimation de pose en :

$$\min_{\mathbf{P}_k, \mathsf{Q}_i} \sum_{i=1}^{p} \sum_{k=1}^{n} \left(\mathsf{q}_1^k - \frac{(\mathbf{P}_k \mathsf{Q})_1}{(\mathbf{P}_k \mathsf{Q})_3}\right)^2 + \left(\mathsf{q}_2^k - \frac{(\mathbf{P}_k \mathsf{Q})_2}{(\mathbf{P}_k \mathsf{Q})_3}\right)^2. \tag{4.119}$$

où $\mathbf{P}_k$ désigne la $k^{\text{ème}}$ matrice de projection perspective. De manière analogue au cas stéréoscopique ces problèmes sont initialisés avec les paramètres de calibrage et les solutions du système linéaire :

$$\mathsf{q}_i^k \cong \mathbf{P}_k \mathbf{T}_k^l \mathsf{Q}_i \qquad \forall (i,k) \in \{1,\ldots,p\} \times \{1,\ldots,n\}. \tag{4.120}$$

Nous avons posé dans ces paragraphes les formulations des problèmes de calibrage et de reconstruction 3D. Ces problèmes sont formulés comme la minimisation d'une somme de fractions rationnelles dont les degrés dépendent des modèles de caméra choisis. La section suivante est consacrée à l'état de l'art des méthodes numériques utilisées pour résoudre ces problèmes d'optimisation.

4.2.2 Etat de l'art

Le but de cette section est de présenter l'état de l'art des méthodes d'optimisation numérique utilisées pour résoudre les problèmes de triangulation et l'ajustement de faisceaux. Les méthodes utilisées sont scindées en deux catégories : les méthodes globales et les méthodes locales. Les méthodes locales sont classiquement utilisées pour résoudre les problèmes d'ajustement de faisceaux car leurs formulations utilisent un grand nombre de variables, ce qui est rédhibitoire pour les méthodes globales. Ainsi, les méthodes globales sont généralement plus utilisées pour résoudre les problèmes de triangulation. Nous commencerons par présenter dans cette section les adaptations des méthodes locales utilisées pour résoudre l'ajustement de faisceaux puis nous présenterons les méthodes d'optimisation globales utilisées pour résoudre le problème de triangulation multivues.

Les méthodes locales pour l'ajustement de faisceaux. Les méthodes locales classiquement utilisées pour résoudre les problèmes (4.116) et (4.117) sont les méthodes de régions de confiance (Algorithme 5) ou l'algorithme de Levenberg-Marquardt (Algorithme 7). Cependant, comme nous l'avons souligné, le nombre de variables d'optimisation de ces deux problèmes peut devenir important. Il est donc nécessaire de tenir compte de l'interdépendance des variables entre elles. Rappelons que, dans notre cas d'un modèle radial **R3**, il y a, pour chaque caméra k :

- 7 paramètres intrinsèques : 4 internes et 3 de distorsions radiales contenus respectivement dans la matrice $\mathbf{K}_k$ (identifiée avec un vecteur de $\mathbb{R}^4$) et le vecteur κ_k.
- 9 coefficients pour chaque matrice de localisation $\mathbf{T}_k^l$
- 3 inconnues par points 3D Q_i.

En toute généralité, il peut y avoir un nombre de déplacements et un nombre de caméras différents. Cependant, pour plus de clarté et sans perte de généralités, nous supposerons qu'il n'y a qu'une seule caméra se déplaçant n fois en observant p points. Ainsi, pour (4.116) et (4.117), définissons le vecteur de résidus $r \in \mathbb{R}^{np}$ par :

$$r \triangleq \left(\begin{array}{c} \boxed{\begin{array}{c} \mathsf{q}_1^1 - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_1^l, \mathsf{Q}_1) \\ \vdots \\ \mathsf{q}_p^1 - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_1^l, \mathsf{Q}_p) \end{array}} \\ \boxed{\begin{array}{c} \mathsf{q}_1^2 - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_2^l, \mathsf{Q}_1) \\ \vdots \\ \mathsf{q}_p^2 - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_2^l, \mathsf{Q}_p) \end{array}} \\ \vdots \\ \boxed{\begin{array}{c} \mathsf{q}_1^k - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_k^l, \mathsf{Q}_1) \\ \vdots \\ \mathsf{q}_p^k - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_k^l, \mathsf{Q}_p) \end{array}} \\ \vdots \\ \boxed{\begin{array}{c} \mathsf{q}_1^n - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_n^l, \mathsf{Q}_1) \\ \vdots \\ \mathsf{q}_p^n - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_n^l, \mathsf{Q}_p) \end{array}} \end{array} \right) \tag{4.121}$$

où f désigne la fonctionnelle de projection propre à chaque problème. Avec cette définition, chaque sous-bloc $r_k \in \mathbb{R}^p$ de r défini par :

$$r_k \triangleq \boxed{\begin{array}{c} \mathsf{q}_1^k - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_k^l, \mathsf{Q}_1) \\ \vdots \\ \mathsf{q}_p^k - \mathrm{f}(\mathbf{K}, \kappa, \mathbf{T}_k^l, \mathsf{Q}_p) \end{array}} \tag{4.122}$$

représente les projections des p points dans la $k^{\text{éme}}$ vue. Ainsi, les p lignes correspondant à r_k dans la jacobienne J_r de r sont données par :

$$\begin{array}{|c|c|c|c|}\hline \begin{matrix}\frac{\partial \mathrm{f}}{\partial \mathbf{K}} \\ \vdots \\ \frac{\partial \mathrm{f}}{\partial \mathbf{K}}\end{matrix} & \begin{matrix}\frac{\partial \mathrm{f}}{\partial \kappa} \\ \vdots \\ \frac{\partial \mathrm{f}}{\partial \kappa}\end{matrix} & \begin{matrix}\frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_1)}{\partial \mathbf{T}_1^l} & \cdots & \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_1)}{\partial \mathbf{T}_n^l} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_p)}{\partial \mathbf{T}_1^l} & \cdots & \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_p)}{\partial \mathbf{T}_n^l}\end{matrix} & \begin{matrix}\frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_1)}{\partial \mathrm{Q}_1} & \cdots & \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_1)}{\partial \mathrm{Q}_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_p)}{\partial \mathrm{Q}_1} & \cdots & \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_p)}{\partial \mathrm{Q}_p}\end{matrix} \\ \hline \end{array} \tag{4.123}$$

Les deux premiers blocs de dérivées partielles (en bleu et rouge) sont en général creux (s'il y a plus d'une caméra), mais traités comme des blocs pleins dans notre discussion mono-caméra. Etudions ensuite le bloc de dérivées partielles selon les matrices de localisation $\mathbf{T}_k^l$ (en vert). Pour cela remarquons que, quel que soit $k \in \{1,\ldots,n\}$, le bloc r_k est indépendant de $\mathbf{T}_j^l$ pour $j \neq k$. Par conséquent :

$$\frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_i)}{\partial \mathbf{T}_j^l} = 0 \qquad \forall j \in \{1,\ldots,n\}\setminus\{k\}\,, \quad \forall i \in \{1,\ldots,p\}\,. \tag{4.124}$$

Ainsi, le bloc de dérivées partielles selon les matrices de localisation se réduit en :

$$\begin{array}{|ccccccc|}\hline 0 & \cdots & 0 & \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_1)}{\partial \mathbf{T}_k^l} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_p)}{\partial \mathbf{T}_k^l} & 0 & \cdots & 0 \\ \hline \end{array} \tag{4.125}$$

Enfin, puisque :

$$\frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_j)}{\partial \mathrm{Q}_i} = 0 \qquad \forall (i,j) \in \{1,\ldots,p\}\,, \text{ tels que } i \neq j, \tag{4.126}$$

le bloc des dérivées partielles selon les points Q_i (en jaune) devient :

$$\begin{array}{|ccc|}\hline \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_1)}{\partial \mathrm{Q}_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial \mathrm{f}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathrm{Q}_p)}{\partial \mathrm{Q}_p} \\ \hline \end{array} \tag{4.127}$$

La forme finale de la matrice J_r est ainsi donnée par la figure 4.27. Sous l'hypothèse de Gauss-Newton (2.86), la forme de la matrice hessienne $H_r = J_r^\top J_r$ est donnée par la figure 4.28 où

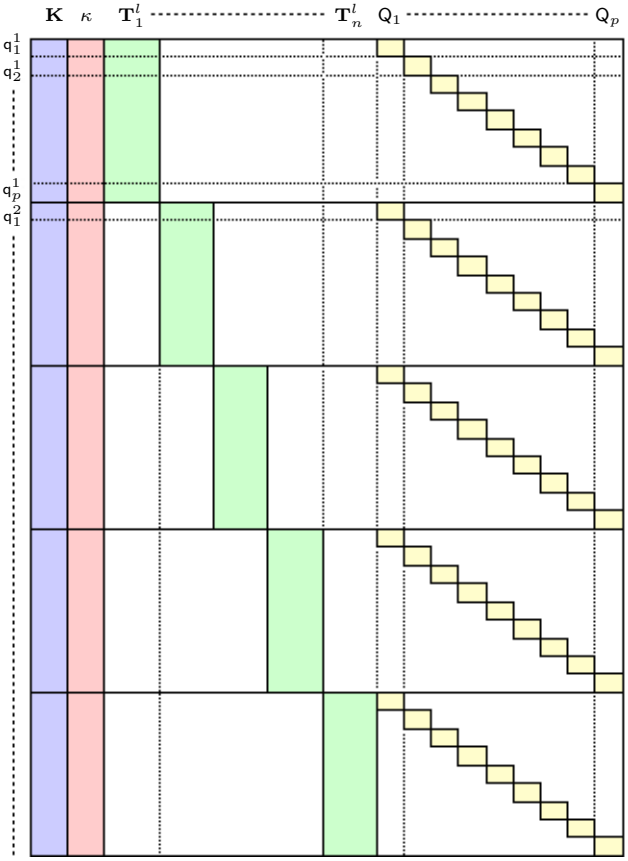


FIGURE 4.27 – Forme générale de la matrice J_r

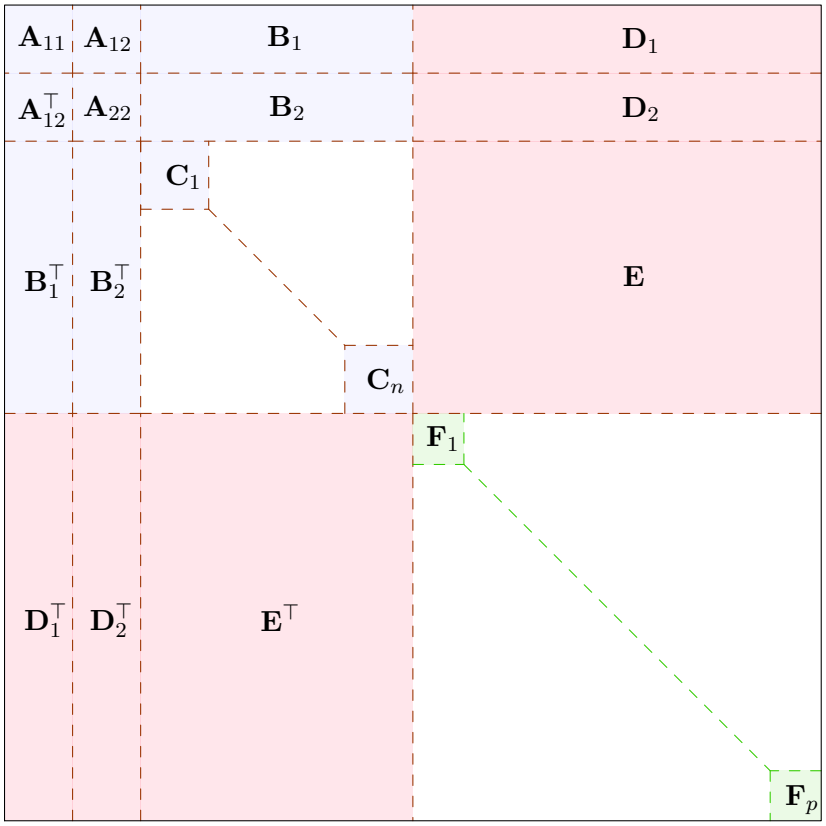


FIGURE 4.28 – Forme générale de la matrice H_r

les matrices $\mathbf{A}_{ik}$, $\mathbf{B}_{1k}$, $\mathbf{B}_{2k}$, $\mathbf{C}_k$, $\mathbf{D}_{1k}$, $\mathbf{D}_{2k}$, $\mathbf{E}$ et $\mathbf{F}_i$ sont définies par :

$$\mathbf{A}_{11} \triangleq \sum_{k=1}^{n}\sum_{i=1}^{p}\left(\frac{\partial \mathrm{f}}{\partial \mathbf{K}}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \mathbf{K}}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{4,4}(\mathbb{R}) \tag{4.128}$$

$$\mathbf{A}_{12} \triangleq \sum_{k=1}^{n}\sum_{i=1}^{p}\left(\frac{\partial \mathrm{f}}{\partial \mathbf{K}}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \kappa}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{4,3}(\mathbb{R}) \tag{4.129}$$

$$\mathbf{A}_{22} \triangleq \sum_{k=1}^{n}\sum_{i=1}^{p}\left(\frac{\partial \mathrm{f}}{\partial \kappa}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \kappa}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{3,3}(\mathbb{R}) \tag{4.130}$$

$$\mathbf{B}_{1,k} \triangleq \sum_{i=1}^{p}\left(\frac{\partial \mathrm{f}}{\partial \mathbf{K}}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \mathbf{T}_k^l}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{5,9}(\mathbb{R}) \tag{4.131}$$

$$\mathbf{B}_{2,k} \triangleq \sum_{i=1}^{p}\left(\frac{\partial \mathrm{f}}{\partial \kappa}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \mathbf{T}_k^l}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{3,9}(\mathbb{R}) \tag{4.132}$$

$$\mathbf{C}_k \triangleq \sum_{i=1}^{p}\left(\frac{\partial \mathrm{f}}{\partial \mathbf{T}_k^l}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \mathbf{T}_k^l}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{9,9}(\mathbb{R}) \tag{4.133}$$

$$\mathbf{D}_{1,i} \triangleq \sum_{k=1}^{n}\left(\frac{\partial \mathrm{f}}{\partial \mathbf{K}}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \mathbf{Q}_i}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{5,3}(\mathbb{R}) \tag{4.134}$$

$$\mathbf{D}_{2,i} \triangleq \sum_{k=1}^{n}\left(\frac{\partial \mathrm{f}}{\partial \kappa}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \mathbf{Q}_i}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{3,3}(\mathbb{R}) \tag{4.135}$$

$$\mathbf{E}_{ki} \triangleq \left(\frac{\partial \mathrm{f}}{\partial \mathbf{T}_k^l}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \mathbf{Q}_i}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{9,3}(\mathbb{R}) \tag{4.136}$$

$$\mathbf{F}_i \triangleq \sum_{k=1}^{n}\left(\frac{\partial \mathrm{f}}{\partial \mathbf{Q}_i}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)^\top\left(\frac{\partial \mathrm{f}}{\partial \mathbf{Q}_i}(\mathbf{K},\kappa,\mathbf{T}_k^l,\mathbf{Q}_i)\right)\in\mathcal{M}_{3,3}(\mathbb{R}). \tag{4.137}$$

Remarquons que, s'il y avait plus d'une caméra, les tailles de ces matrices (mise à part $\mathbf{C}_{kk}$, $\mathbf{E}$ et $\mathbf{F}$) seraient liées à n_c, le nombre de caméras utilisées. La matrice H_r appartient à l'espace des matrices carrées de taille $7+9n+3p$ ($7n_c+9n+3p$ dans le cas général). Cette taille peut donc devenir très importante et on comprend alors qu'il serait défavorable d'utiliser directement $J_r^\top J_r$ dans la résolution modèle quadratique pertubé :

$$\left(J_r(x_k)^\top J_r(x_k)+\mu_k I_n\right)d_k = -J_r(x_k)r(x_k) \tag{4.138}$$

utilisé dans l'Algorithme 7 ou dans la résolution du sous-problème de région de confiance (13) (où ∇f et $\nabla^2 f$ sont respectivement égaux à $J_r r$ et $J_r^\top J_r$). Ainsi, afin de tenir compte de la structure creuse de $J_r^\top J_r$ et donc d'optimiser la résolution des problèmes précédents, on découpe la hessienne de r suivant la figure 4.29. On obtient ainsi les trois matrices suivantes : $\mathbf{H}_{11}\in\mathcal{M}_{7+9n,7+9n}(\mathbb{R})$, $\mathbf{H}_{12}\in\mathcal{M}_{7+9n,3p}(\mathbb{R})$ et $\mathbf{H}_{22}\in\mathcal{M}_{3p,3p}(\mathbb{R})$. Dans le cas de l'algorithme de Levenberg-Marquardt ou dans celui des régions de confiance, on sera alors amené à résoudre un système de la forme :

$$\underbrace{\begin{pmatrix}\mathbf{H}_{11} & \mathbf{H}_{12}\\ \mathbf{H}_{12}^\top & \mathbf{H}_{22}\end{pmatrix}}_{=\nabla^2(\frac{1}{2}\|r\|_2^2)=J_r^\top J_r}\underbrace{\begin{pmatrix}d_1\\ d_2\end{pmatrix}}_{\text{direction de descente } d_k} = \underbrace{\begin{pmatrix}\varepsilon_1\\ \varepsilon_2\end{pmatrix}}_{=\nabla(\frac{1}{2}\|r\|_2^2)=J_r r} \tag{4.139}$$

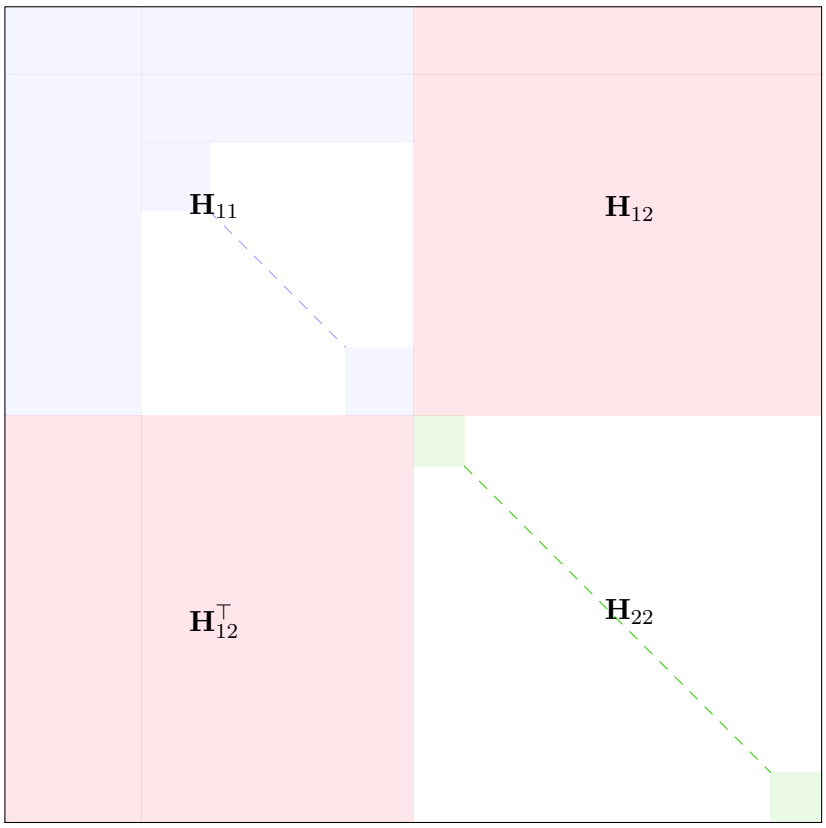


FIGURE 4.29 – Découpage de la matrice H_r

En utilisant **le complément de Schur**, on obtient alors le système suivant :

$$\begin{cases} \left(\mathbf{H}_{11} - \mathbf{H}_{12}\mathbf{H}_{22}^{-1}\mathbf{H}_{12}^{\top}\right) d_1 &= \varepsilon_1 - \mathbf{H}_{11}\mathbf{H}_{22}^{-1}\varepsilon_2 \\ \mathbf{H}_{12}^{\top} d_1 + \mathbf{H}_{22} d_2 &= \varepsilon_2 \end{cases} \tag{4.140}$$

Cette dernière formulation est très favorable car on ne doit inverser que la matrice la plus creuse $\mathbf{H}_{22}$. Il est ensuite possible de simplifier encore ces équations en détaillant chacun des blocs. Cependant, le but étant d'introduire la mise en oeuvre pratique des algorithmes de Levenberg-Marquardt et des régions de confiance au problème de l'ajustement de faisceaux, nous ne détaillerons pas plus avant cette résolution. Pour plus de détail sur l'application de l'algorithme de Levenberg-Marquardt, le lecteur pourra se référer à [Triggs 2000, Lourakis 2010, Lourakis 2004], tandis que dans [Lourakis 2005], il trouvera une adaptation de l'algorithme des régions de confiance avec une recherche **dogleg**. Notons que la recherche sur le thème de l'ajustement de faisceaux reste très intensive, notamment sur sa résolution théorique [Agarwal 2010], l'étude de sa structure [Kushal 2012] ou son implémentation informatique [Wu 2011].

Les méthodes globales pour la triangulation. Dans les décennies précédentes, beaucoup de recherches ont été menées afin de résoudre les problèmes sous forme algébrique. A l'instar de l'algorithme des 8-points, ces approches sont souvent utilisées pour rechercher un point de départ à l'ajustement de faisceaux. Ces méthodes sont basées sur la minimisation de fonctions polynomiales sous des contraintes dont le rôle est d'imposer des propriétés théoriques « idéales » sur la solution recherchée (p. ex. la condition de rang pour la matrice fondamentale). Comme nous l'avons vu précédemment, ces fonctions de coût algébrique sont généralement minimisées à l'aide de méthodes d'algèbre linéaire. Cependant, même s'il est possible d'exhiber une solution globale pour ces problèmes, sa qualité peut, en pratique, s'avérer décevante.

En effet, les fonctions coûts algébriques ne sont souvent pas liées de manière significative à la structure métrique de la géométrie de la scène. Cependant, ces solutions algébriques fournissent des initialisations acceptables pour les problèmes métriques dont les fonctions coût se formulent comme des sommes de fractions rationnelles. Ainsi, un certain nombre de nouvelles méthodes garantissant des solutions globales ont été étudiées ces dernières années. Ces méthodes utilisent des résultats récents dans les domaines de l'optimisation convexe et polynomiale et apportent de nouvelles façons de comprendre les problèmes en vision par ordinateur et robotique. Un certain nombre de ces problèmes ont déjà été résolus avec succès, d'autres sont encore ouverts.

D'un point de vue théorique, tous les problèmes d'optimisation se formulant comme la minimisation d'une somme de fraction rationnelle :

$$\min_{x \subset \mathbb{R}^n} \sum_{i=1}^{l} \frac{p_i(x)}{q_i(x)} \tag{4.141}$$

peuvent être résolus de manière globale. En effet, il suffit de réduire au même dénominateur cette somme :

$$\frac{P(x)}{Q(x)} \triangleq \frac{\sum_{i=1}^{l} \left(p_i(x) \prod_{j \neq i}^{l} q_j(x) \right)}{\prod_{i=1}^{l} q_i(x)} \tag{4.142}$$

et d'énumérer toutes les solutions de $\nabla P(x) = 0$, ce qui se formule comme un système d'équations polynomiales :

$$\sum_{i=1}^{l} \left[\frac{\partial p_i(x)}{\partial x_k} \prod_{j \neq i}^{l} q_j(x) + p_i(x) \sum_{j \neq i}^{l} \frac{\partial q_j(x)}{\partial x_k} \prod_{s \neq j}^{l} q_s(x) \right] = 0 \qquad \forall k \in \{1, \ldots, n\} . \tag{4.143}$$

La complexité de ce système est étroitement liée à l le nombre de fractions et aux degrés des p_i et q_i. Pratiquement, en vision par ordinateur, ces degrés sont tous égaux et généralement faibles (p. ex. égaux à 2 pour le problème de la triangulation sans distorsions). Mais, même pour un nombre de variables supérieur à 3, si le nombre l de fractions rationnelles est supérieur à 5, alors (4.143) devient vite difficile à résoudre dans un temps de calcul raisonnable. Ainsi, bien que cette approche puisse résoudre un large éventail de problèmes, elle n'est appliquée en pratique, à cause de temps de calcul prohibitifs, que sur des problèmes de petite taille. Notons cependant que cette approche a été appliquée avec succès à de nombreux problèmes comme l'estimation de pose [Bujnak 2010, Bujnak 2008, Nister 2007] ou la triangulation à deux [Hartley 1997] et trois vues [Stewénius 2005, Byröd 2007a]. Notons en outre que la résolubilité de (4.143) pour des problèmes de vision par ordinateur a été étudiée d'un point de vue numérique [Byröd 2007b, Byröd 2008] et théorique [Nister 2007].

Comme nous venons de le voir, la recherche des points stationnaires de (4.141) devient rapidement complexe si le nombre de fractions augmente. De plus, ce problème ne prend pas en compte d'éventuelles contraintes sur l'espace de recherche. Ainsi, afin de pallier ce défaut, une deuxième famille de méthodes est basée sur l'utilisation des propriétés théoriques des problèmes quasi-convexes. Rappelons que tout minimum local d'une fonction convexe est aussi un minimum global. Ceci est également vrai pour les fonctions quasi-convexes. De plus, pour une famille finie de fonctions convexes (resp. quasi-convexes) $(f_i)_i$, la fonction $\max(f_i)$

définie par :

$$\begin{aligned} \max(f_i) : \mathbb{R}^n &\mapsto \mathbb{R} \\ x &\mapsto \max_i f_i(x) \end{aligned} \tag{4.144}$$

est convexe (resp. quasi-convexe). La figure 4.30 souligne graphiquement ce résultat pour une famille de trois fonctions quasi-convexes. Notons cependant que la fonction $\max(f_i)$ perd toute propriété de différentiabilité. Ainsi, dans [Kahl 2008b], les auteurs utilisent le fait que, pour la

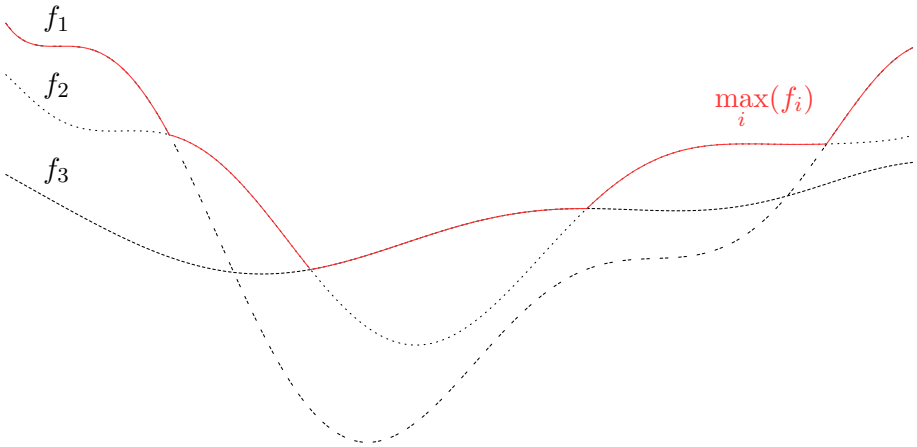


FIGURE 4.30 – La fonction maximum d'une famille de fonctions quasi-convexe est quasi-convexe

fonctionnelle $\mathrm{f}_{\mathbf{P}}$, les coordonnées du vecteur de résidus r définies par (4.121) sont des fonctions quasi-convexes. Ils minimisent alors r au sens de la norme L_∞. Le problème résultant peut se ré-écrire sous la forme :

$$\begin{aligned} \min_x \max_i \ & \frac{\|\mathbf{A}_i x + c_i\|_2^2}{a_i^\top x + b_i} \\ \text{s.l.c.} \quad & a_i^\top x + b_i \geqslant 0. \end{aligned} \tag{4.145}$$

Or, la fonction objectif de ce problème n'est ni différentiable, ni convexe. Cependant, la fonction coût de la formulation sous forme d'épigraphes :

$$\begin{aligned} \min_{x,\gamma} \quad & \gamma \\ \text{s.l.c.} \quad & \|\mathbf{A}_i x + c_i\|_2^2 - \gamma \left(a_i^\top x + b_i\right) \leqslant 0 \\ & \gamma \geqslant 0, \end{aligned} \tag{4.146}$$

possède ces deux propriétés. Remarquons que dans l'utilisation de la norme, L_∞ permet de n'utiliser qu'une seule variable d'écart γ (puisque seul le maximum est recherché). Le problème (4.146) est alors résolu au moyen de technique de Programmation Conique de Second Ordre (SOCP) [Vandenberghe 1996]. Notons que, indépendamment, une approche similaire a été développée dans [Ke 2007]. Cependant, cette approche basée sur la norme L_∞ n'est plus applicable dès lors que la quasi-convexité est perdue (ce qui est le cas pour des problèmes impliquant des rotations). Les algorithmes **Branch and Bound** ont alors été utilisés pour remédier à cet inconvénient. Comme nous l'avons déjà souligné, les algorithmes **Branch and Bound** sont des méthodes d'optimisation globale pour des problèmes non-convexes. Lorsqu'ils convergent, ils fournissent un encadrement à $\varepsilon > 0$ près de la solution globale (pour ε arbitrairement petit). Plus précisément, considérons $\Phi : \mathbb{R}^n \to \mathbb{R}$ et détaillons l'Algorithme 13. La structure de données initiales L est représentée par un rectangle. On note alors :

$$\Phi_{\min}(\mathsf{E}) \quad \triangleq \quad \min_{x \in \mathsf{E}} \Phi \text{ pour tout rectangle } \mathsf{E} \subseteq \mathsf{L}. \tag{4.147}$$

Ensuite, on suppose que l'on est capable de construire, pour tout rectangle $\mathsf{E} \subseteq \mathsf{L}$, la fonction $\Phi_{\mathsf{lb}}(\mathsf{E})$ vérifiant les propriétés suivantes :

1. $\Phi_{\mathsf{lb}}(\mathsf{E})$ permet de calculer une minoration de $\Phi_{\min}(\mathsf{E})$, i.e.

$$\Phi_{\mathsf{lb}}(\mathsf{E}) \leqslant \Phi_{\min}(\mathsf{E}) \tag{4.148}$$

2. l'erreur d'approximation $|\Phi_{\mathsf{lb}}(\mathsf{E}) - \Phi_{\min}(\mathsf{E})|$ tend uniformément vers zéro lorsque la taille de E, plus précisément le maximum des demi-longueurs noté $|\mathsf{E}|$, tend vers 0 :

$$\forall \varepsilon,\ \exists \delta_\varepsilon,\ \text{tel que} : \forall \mathsf{E} \subseteq \mathsf{L},\ |\mathsf{E}| \leqslant \delta_\varepsilon \Rightarrow |\Phi_{\mathsf{lb}}(\mathsf{E}) - \Phi_{\min}(\mathsf{E})| \leqslant \varepsilon. \tag{4.149}$$

Soit $\varepsilon > 0$, l'algorithme commence en choisissant un rectangle $\mathsf{E} \subseteq \mathsf{L}$ et en calculant $\Phi_{\mathsf{lb}}(\mathsf{E})$ et son antécédent $\mathsf{e}^\star$ tel que $\Phi_{\mathsf{lb}}(\mathsf{e}^\star) = \Phi_{\mathsf{lb}}(\mathsf{E})$. Si $|\Phi(\mathsf{e}^\star) - \Phi_{\min}(\mathsf{E})| \leqslant \varepsilon$ alors l'algorithme se termine. Sinon, E est partitionné en une collection finie de rectangles $(\mathsf{E}_i)_i$ de sorte que $\mathsf{E} = \bigsqcup_i \mathsf{E}_i$. Le calcul du minorant de chaque sous-domaine se fait alors en calculant $\Phi_{\mathsf{lb}}(\mathsf{E}_i)$ et son antécédent $\mathsf{e}_i^\star$. On fait alors l'hypothèse que $\Phi(\mathsf{e}^\star)$, avec $\mathsf{e}^\star = \arg\min_i \Phi(\mathsf{e}_i^\star)$, est la meilleure approximation courante de $\Phi_{\min}(\mathsf{E})$. Si $|\Phi(\mathsf{e}^\star) - \Phi_{\min}(\mathsf{E})| \leqslant \varepsilon$ alors l'algorithme se termine, sinon, la partition de E est raffinée en divisant certains sous-domaines et en répétant l'opération précédente. Notons que les rectangles E_i tels que $\Phi_{\mathsf{lb}}(\mathsf{E}_i) > \Phi(\mathsf{e}^\star)$ ne peuvent contenir le miminum global, ils ne sont donc pas pris en compte pour le raffinage et supprimés de la structure de données. L'ensemble de cette adaptation est rappelé par l'Algortihme 20.

Algorithme 20 Algorithme de séparation et évaluation adapté

Précondition : Rectangle initial E, $\varepsilon > 0$

Calcul de $\Phi_{\mathsf{lb}}(\mathsf{E})$ et de son antécédent $\mathsf{e}^\star$
initialisation de la structure de données $\mathsf{L} = \{\mathsf{E}\}$
tant que $|\Phi(\mathsf{e}^\star) - \Phi_{\min}(\mathsf{E})| \leqslant \varepsilon$ **faire**
  Extraire un élément E de L en choisissant E tel que : $\mathsf{E} = \arg\min_{\tilde{\mathsf{E}} \subseteq \mathsf{L}} \Phi_{\mathsf{lb}}(\tilde{\mathsf{E}})$
  Division de E en k sous-domaines $\mathsf{E} = \bigsqcup_i^k \mathsf{E}_i$
  $\mathsf{L} \leftarrow \{\mathsf{L} \backslash \mathsf{E}\} \bigcup \{\mathsf{E}_1, \ldots, \mathsf{E}_k\}$
  pour chaque sous-domaine E_i **faire**
    Calcul d'un minorant de f sur E_i en déterminant $\Phi_{\mathsf{lb}}(\mathsf{E}_i)$ et $\mathsf{e}_i^\star$
    si $\Phi(e_i^\star) < \Phi(e^\star)$ **alors**
      $\mathsf{e} \leftarrow \mathsf{e}_i$ //Le minorant de Φ sur E_i est inférieur au minimum courant
    sinon
      $\mathsf{L} \leftarrow \{\mathsf{L} \backslash \mathsf{E}_i\}$. //Supprimer E_i

L'applicabilité pratique de cet algorithme dépend directement de sa capacité à construire, dans un temps raisonnable, des fonctions Φ_{lb} facilement minimisables. Notons que cette étape correspond, dans le schéma général de l'Algorithme 13, à l'étape d'évaluation (ou bounding) alors que l'étape de séparation (ou branching), correspond à la phase de division des rectangles. Afin d'assurer la convergence de l'algorithme, les fonctions Φ_{lb} à construire doivent nécessairement vérifier les conditions (4.148) et (4.149). Or, pour tout rectangle $\mathsf{E} \subseteq \mathsf{L}$, ces propriétés sont, par définition, inhérentes à $(\mathsf{conv}\,\Phi)_{|\mathsf{E}}$, l'enveloppe convexe (cf. définition 6) de Φ restreinte à E. Cependant, construire l'enveloppe convexe d'une fonction quelconque peut s'avérer aussi difficile que de trouver le minimum global. Mais, dans le cas d'une seule fraction rationnelle, ce problème se ramène à un problème **SOCP**, donc aisément résoluble. De plus, si $\Phi = \sum_{i=1}^{l} \frac{p_i}{q_i}$ alors il est possible de démontrer que :

$$\mathsf{conv}\,\Phi \quad \geqslant \quad \sum_{i=1}^{l} \mathsf{conv}\,\frac{p_i}{q_i}. \tag{4.150}$$

où $\sum_{i=1}^{l} \mathsf{conv}\, \frac{p_i}{q_i}$ est une fonction convexe. Cette dernière fonction est alors choisie pour calculer Φ_{lb}. Cette approche a été utilisée sur les problèmes d'estimation de pose et de triangulation multivues au sens des normes L_1, L_2 dans [Kahl 2008a, Fangfang 2007] et au sens L_∞ dans [Kahl 2008b]. Cependant, il existe un lien entre la quasi-convexité des fractions $\frac{p_i}{q_i}$ et l'erreur d'approximation :

$$\left| \mathsf{conv} \sum_{i=1}^{l} \frac{p_i}{q_i} - \sum_{i=1}^{l} \mathsf{conv}\, \frac{p_i}{q_i} \right|. \tag{4.151}$$

En effet, comme le souligne la figure 4.31, si une fonction de la somme n'est pas quasi-convexe, cette erreur peut être significative.

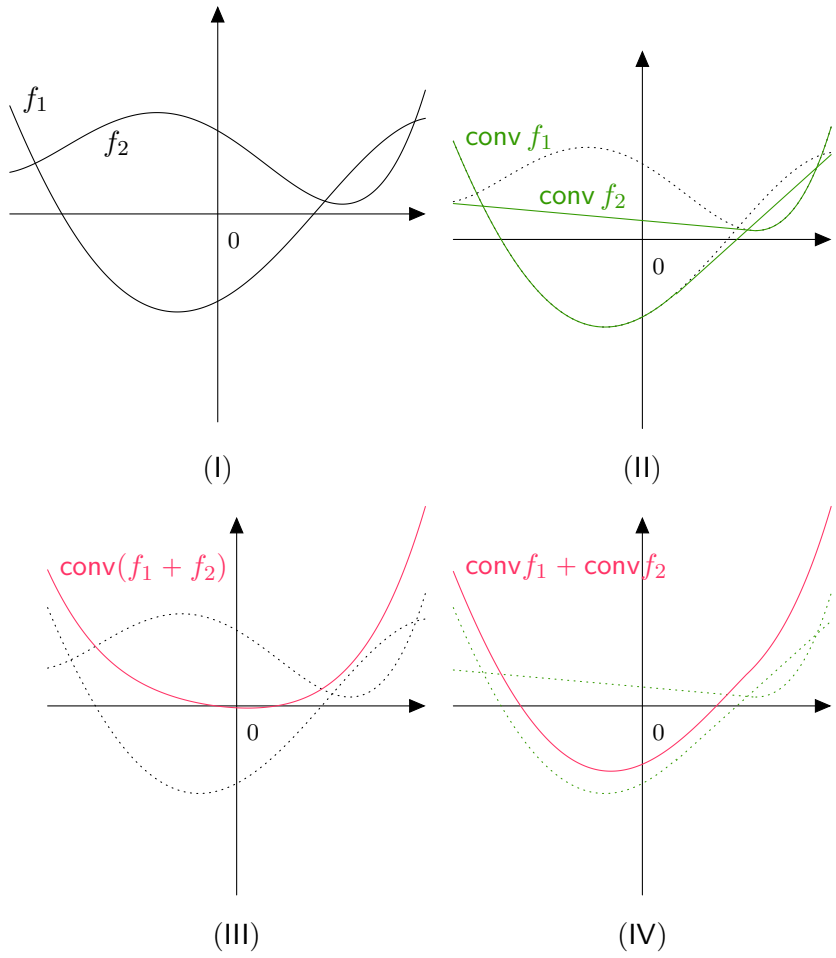


FIGURE 4.31 – Illustation de l'écart entre $\mathsf{conv} f_1 + \mathsf{conv} f_2$ et $\mathsf{conv}(f_1 + f_2)$. Le premier graphique (I) représente les deux courbes f_1 et f_2. Le deuxième (I) représente (en vert) l'enveloppe convexe de f_1, $\mathsf{conv} f_1$, et celle de f_2, $\mathsf{conv} f_2$. (III) représente, $\mathsf{conv}(f_1 + f_2)$, l'enveloppe convexe de $f_1 + f_2$ et (IV) la somme de $\mathsf{conv} f_1$ et $\mathsf{conv} f_2$.

Il est donc possible que cette approche se révèle inefficace sur certains problèmes dans lesquels toutes les fractions ne sont pas convexes ou quasi-convexes. Dans se sens, une étude complète de la convexité et de la quasi-convexité dans les problèmes de vision a été réalisée dans [Kahl 2008b]. Notons enfin que cette dernière approche ne permet pas de traiter des contraintes non linéaires.

Afin de traiter le problème (4.141) de manière générique, c'est-à-dire sans faire d'hypothèses sur les fractions $\frac{p_i}{q_i}$, et de pouvoir considérer des contraintes non-linéaires, il est présenté dans [Kahl 2007] un cadre général permettant de fournir une estimée initiale au problème (4.141) sous contraintes polynomiales. En effet, comme nous l'avons déjà mentionné pour l'estimation de la matrice fondamentale, cette approche permet de construire une hiérarchie de relaxations convexes construites à partir de la formulation par épigraphes de (4.141). Mais, bien que cette méthode permette de considérer un grand nombre de variables d'écart, la suite des solutions issues de la hiérarchie convexe ne converge pas, a priori, vers le minimum global. Cependant, les comparaisons menées dans [Kahl 2007] sur les problèmes de triangulation multivues et d'estimation de pose au sens de la norme L_2 montrent que l'estimée initiale calculée à l'aide de cette approche est de meilleure qualité que celle calculée par la résolution du système linéaire.

Enfin, il est présenté dans [Hartley 2008], une approche rapide (environ 0.5 ms par points), permettant de tester, a posteriori, si un minimum local du problème de triangulation multivues au sens de la norme L_2 est global ou non.

4.2.3 Applications à la triangulation multivues

Le but de cette section est de présenter les résultats obtenus par notre approche globale, comparativement à une résolution locale, sur des cas synthétiques et réels. Elle sera divisée suivant deux applications distinctes : la triangulation multivues sans distorsions et avec distorsions. Pour toutes les expériences, nous comparerons notre méthode à **une résolution locale du problème à l'aide d'une méthode de régions de confiance** initialisée avec une solution du problème linéaire, appelée L_2 *local*. Notons que nous avons essayé de comparer notre méthode à l'approche présentée dans [Kahl 2008a], mais cette dernière s'est révélée très instable (de trés bonnes convergences sur certaines données et pas de convergence pour d'autres) dans tous nos tests, de sorte que toute conclusion soit impossible. Pour le cas de la triangulation sans distorsions, comme le montrent les études réalisées dans la littérature, les solutions locales peuvent être proches des solutions globales. Cependant, nous nous attacherons à montrer sur des cas réels et synthétiques que ces écarts peuvent être, dans certains cas, significatifs. Enfin, nous mènerons les mêmes types de comparaisons dans le cas de la triangulation avec distorsions. Notons que, dans cette section, nous ne ferons aucune référence au temps de calcul. En effet, de ce point de vue, notre approche globale (avec des temps de calcul de l'ordre de 2 secondes en moyenne) n'est pas compétitive par rapport aux méthodes locales. Nous verrons si l'apport en précision de notre approche globale est suffisant pour pallier ce défaut.

Nous commencerons tout d'abord par présenter les processus de génération de données synthétiques et d'évaluation des résultats qui sont identiques pour les deux applications. Puis, avant de conclure, nous exposerons l'ensemble des résultats obtenus.

4.2.3.1 Triangulation sans prise en compte des distorsions

Génération des données synthétiques. Afin d'être le plus exhaustif possible, nous avons effectué un grand nombre d'expériences avec des images simulées afin de quantifier les performances de l'algorithme. Nos simulations sont basées sur trois configurations : un mouvement latéral de la caméra, un mouvement vers la scène et enfin un mouvement circulaire autour de la scène (i.e. une rotation centrée sur la scène). L'ensemble de ces trois mouvements sont rappelés sur la figure 4.32. Pour les trois configurations, les positions de la caméra sont réparties de manière équidistante le long de chaque trajectoire. Afin de simuler des situations réalistes,

nous utilisons la matrice de paramètres intrinsèques suivante :

$$\mathbf{K} = \begin{bmatrix} 700 & 0 & 320 \\ 0 & 700 & 240 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.152}$$

La scène est composée de points 3D uniformément répartis dans une boule de diamètre 100. Dans la configuration (**1**), la caméra effectue une translation le long d'une droite parallèle à l'axe $\overrightarrow{X}_w$ située à 200 du centre de la boule. Pour la configuration (**2**), la caméra effectue une translation selon l'axe $\overrightarrow{Z}_w$ de sorte que sa dernière position n soit située à 200 du centre de la boule. Notons que cette configuration est une expérience de reconstruction 3D particulièrement difficile. En effet, ce mouvement est défini comme une translation le long de l'axe $\overrightarrow{Z}_w$. Or, dans ce cas, cet axe est confondu avec les axes $\overrightarrow{Z}_c$ des différentes positions de la caméra. Par conséquent, les faisceaux issus des centres optiques et passant par les points 3D situés le long de l'axe de translation ne peuvent s'intersecter. Ainsi, la triangulation des points situés à proximité de cet axe est extrêmement difficile, ce qui se vérifie notamment par des erreurs de reconstruction élevées. Enfin, pour la configuration (**3**), la trajectoire de la caméra est obtenue à partir d'une rotation de rayon égal à 200 et d'angle 45° autour du centre de la scène.

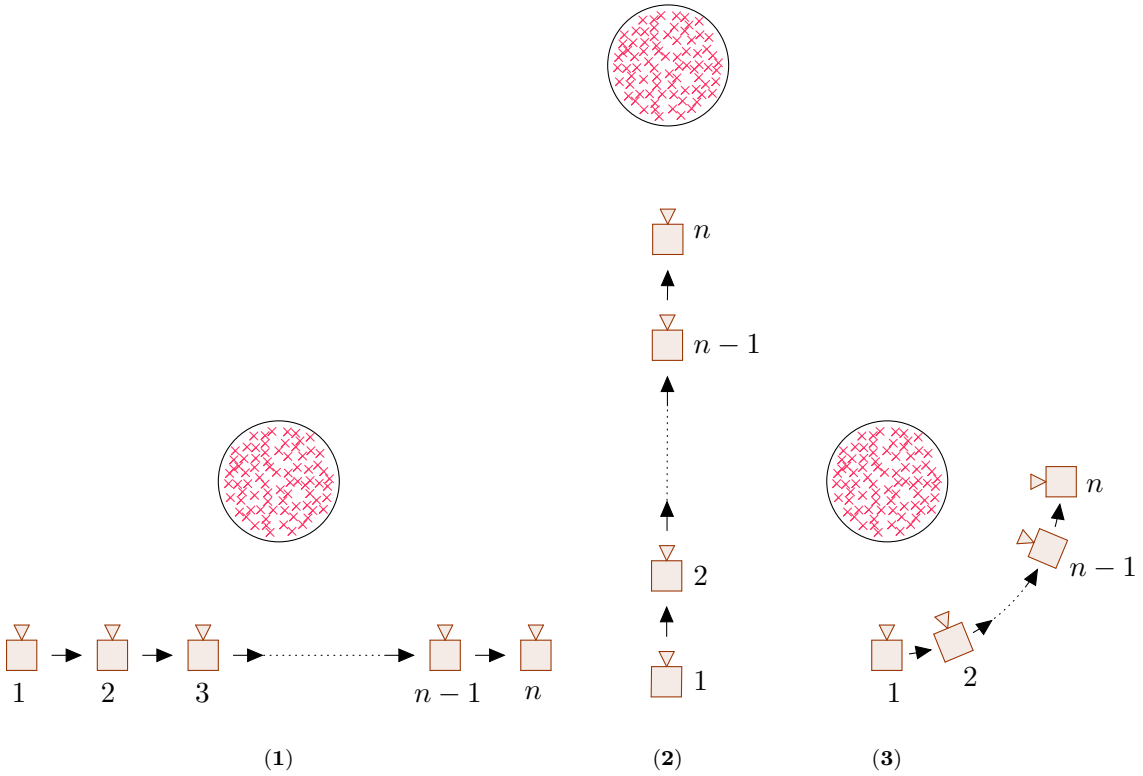


FIGURE 4.32 – Les trois configurations utilisées pour nos simulations : un mouvement latéral (**1**), une translation vers la scène (**1**), et un mouvement circulaire centré sur la scène (**3**).

Pour chaque configuration, nous avons mené trois types d'expériences. Les deux premières ont pour but d'étudier l'influence de deux types de bruit sur la précision de la triangulation. La troisième analyse l'influence du nombre de vues sur la précision de la triangulation. Notons que pour tous les résultats présentés par la suite, les solutions fournies par notre algorithme bénéficient d'un certificat d'optimalité global.

Sensibilité au bruit sur les appariements. La première expérience étudie l'influence du bruit présent dans l'extraction des appariements 2D sur la précision de la triangulation. Elle est réalisée de la manière suivante :

1. Déterminer aléatoirement (c.-à-d. selon une loi uniforme) n points $(\mathsf{Q}_i)_{i=1\cdots n}$ dans la scène.
2. Calculer les projections $(\mathsf{q}_i^j)_{i,j}$ de ces ces points dans les m vues.
3. Ajouter un bruit gaussien centré aux coordonnées pixel de façon à obtenir des projections perturbées $(\widetilde{\mathsf{q}}_i^j)_{i,j}$.
4. Obtenir un nuage de points 3D $(\widetilde{\mathsf{Q}}_i)_{i=1\cdots n}$ à l'aide d'un algorithme de reconstruction 3D utilisant soit notre méthode globale, soit une minimisation locale.
5. Calculer l'erreur de reconstrustion 3D moyenne, notée $\mathcal{E}_{3D}$, à l'aide de la formule :

$$\mathcal{E}_{3D} \triangleq \frac{1}{n}\sum_{i=1}^{n}\|\mathsf{Q}_i - \widetilde{\mathsf{Q}}_i\|_2^2 \tag{4.153}$$

Notons qu'il est aussi possible de calculer l'erreur de reprojection 2D (en pixels), notée $\mathcal{E}_{2D}$, calculée à l'aide de la formule :

$$\mathcal{E}_{2D} \triangleq \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}\|\mathbf{P}^j\widetilde{\mathsf{Q}}_i - \mathsf{q}_i^j\|_2^2. \tag{4.154}$$

Cependant, comme nous disposons des points 3D initiaux, nous avons préféré utiliser le critère 3D qui nous semble, dans ce cas, plus pertinent. Pour cette expérience nous fixons $n = 50$, $m = 6$. Ensuite, nous faisons évoluer la variance σ du bruit de 0 à 2 pixels, de sorte que les projections subissent un déplacement maximal relatif de 6 pixels pour 98% d'entre elles. Enfin, dans le but d'obtenir des résultats statistiquement significatifs, cette expérience est effectuée sur 50 tirages de points 3D différents. Les résultats obtenus pour les 3 configurations sont présentés sur la figure 4.33. Pour cette expérience, nous avons rajouté, à titre d'exemple, les résultats obtenus par l'algorithme de triangulation globale au sens de la norme L_∞ présenté dans [Kahl 2008b]. Les trois expériences soulignent le comportement cohérent des trois méthodes : l'erreur de triangulation augmente linéairement avec le bruit. Analysons alors plus précisément chacun des déplacements. La première courbe présente les erreurs 3D moyennes pour le mouvement latéral. Bien que ces erreurs soient faibles, on constate que les solutions globales au sens de la norme L_∞ et celles fournies par notre algorithme sont toujours plus précises que les solutions locales. On notera de plus que, dans ce cas, les solutions globales au sens de la norme L_∞ et L_2 (i.e. fournies par notre algorithme) sont équivalentes. La deuxième courbe présente les résultats pour le mouvement difficile de translation vers la scène. Sur ce mouvement, on observe que l'écart entre la solution locale et notre algorithme est très important, soulignant ainsi que l'estimée initiale issue de la résolution du système linéaire n'est jamais dans le bassin d'attraction du minimum global. De plus, on remarque que les solutions globales au sens de la norme L_2 fournies par notre méthode sont significativement meilleures que celles calculées au sens de la norme L_∞. Enfin la dernière courbe présente les résultats pour le mouvement circulaire autour de la scène. On observe que, dans ce cas où les faisceaux s'intersectent correctement, les trois approches donnent des résultats identiques. Pour conclure cette première expérience, nous pouvons souligner que notre méthode se révèle, quel que soit le déplacement, la moins sensible au bruit ajouté sur les appariements. Toutefois, l'écart de précision avec les autres approches est faible pour un mouvement classique comme la translation face à la scène ou inexistant avec un mouvement bien défini comme une rotation autour de la scène.

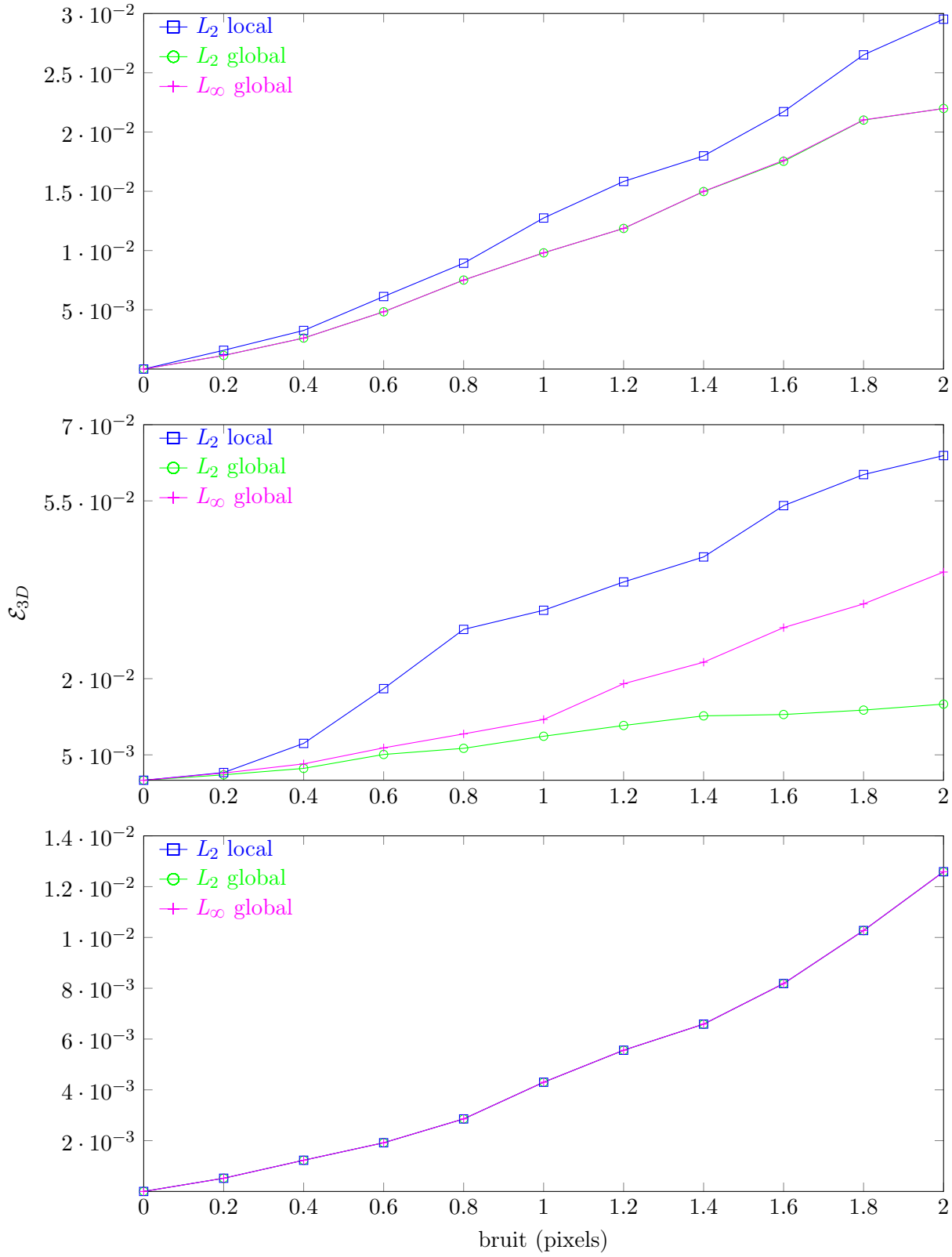


FIGURE 4.33 – Erreurs relatives de reconstruction 3D $\mathcal{E}_{3D}$ en fonction du bruit inséré dans les points projetés pour les trois types de configurations (de haut en bas, la configuration (**1**), puis la (**2**) et enfin la (**3**)).

Sensibilité au bruit sur les matrices de projections La seconde expérience mesure l'influence du bruit présent dans les paramètres de calibrage sur la précision de la triangulation. Elle est réalisée de la manière suivante :

1. Déterminer au hasard n points $(\mathsf{Q}_i)_{i=1\cdots n}$ dans la scène.
2. Ajouter un bruit gaussien aux coefficients des m matrices de projection $\mathbf{P}^j = \mathbf{K}\mathbf{T}_j$ (où les $\mathbf{T}_j$ sont les transformations rigides représentant les déplacements de la caméra) de manière à obtenir des matrices de projection perturbées $\widetilde{\mathbf{P}}^j$.
3. Calculer les projections $(\mathsf{q}_i^j)_{i,j}$ des $(\mathsf{Q}_i)_i$ dans les m vues à l'aide des matrices $\widetilde{\mathbf{P}}^j$.
4. Obtenir un nuage de points 3D $(\widetilde{\mathsf{Q}}_i)_{i=1\cdots n}$ à l'aide d'un algorithme de reconstruction 3D utilisant soit notre méthode globale, soit une minimisation locale.
5. Calculer l'erreur de reconstruction 3D moyenne (en mm) $\mathcal{E}_{3D}$.

Pour cette deuxième expérience nous fixons $n = 50$, $m = 6$. Ensuite, afin de rester dans des cas réels, nous faisons évoluer la variance σ du bruit de 0 à 0,2 de sorte que les coefficients des matrices $\mathbf{P}^j$ subissent un déplacement maximal relatif de 0,6 pour 98% d'entre eux. Dans ce cas, le bruit n'a pas d'unité car nous le rajoutons directement sur les matrices $\mathbf{P}^j$ et donc indistinctement sur les paramètres intrinsèques et extrinsèques. Enfin, dans le but d'obtenir des résultats statistiquement significatifs, cette expérience est effectuée sur 50 tirages de points 3D différents. Les résultats obtenus pour les 3 configurations sont présentés sur la figure 4.34. Signalons que, pour les trois déplacements, la triangulation globale au sens de la norme L_∞ s'est révélée très sensible à ce type de bruit. Ainsi, les erreurs 3D moyennes associées à cette méthode sont trop importantes pour apparaitre conjointement avec celles des autres approches. Par conséquent, nous avons représenté sur les graphiques la méthode locale (en bleu) et notre méthode globale (en vert). Sur le mouvement de translation face à la scène, les deux méthodes sont d'une précision équivalente pour un bruit inférieur à 0,08. Pour un bruit supérieur, l'erreur 3D moyenne liée à l'approche locale s'accroit de façon importante avec le bruit. Comparativement, les erreurs liées à notre approche augmentent linéairement avec le bruit mais selon une pente beaucoup plus faible. Mis à part ce comportement en deux phases, le mouvement difficile de translation vers la scène confirme ce résultat. En effet, les erreurs 3D liées à la méthode locale augmentent linéairement avec le bruit mais avec une pente très largement supérieure à celle que l'on observe pour notre méthode. Enfin, le mouvement circulaire valide notre analyse, car bien que les erreurs soient proches, on observe bien dans les deux cas une augmentation linéaire de l'erreur avec le bruit avec une légére différence de pente en faveur de notre approche. Pour conclure, on observe que notre approche se révèle beaucoup moins sensible au bruit sur les paramètres de calibrage. Ceci corrobore le fait qu'avec ce type de bruit les méthodes locales peuvent rapidement s'éloigner de l'optimum global.

Influence du nombre de vues. Enfin la dernière expérience étudie l'influence du nombre de vues sur la précision de notre algorithme. Elle est réalisée en déterminant au hasard n points $(\mathsf{Q}_i)_{i=1\cdots n}$ dans la scène. Ces points sont ensuite projetés dans les m images et perturbés à l'aide d'un bruit gaussien de variance fixe σ. Enfin, on calcule les reconstrutions 3D à partir de ces projections perturbées en augmentant le nombre de vues de 2 à 20 et en calculant à chaque étape l'erreur $\mathcal{E}_{3D}$. Pour conclure, afin d'obtenir des résultats statistiquement significatifs, ces expériences sont effectuées sur 50 tirages de points 3D différents. Pour l'ensemble de nos tests, les résultats fournis par la méthode de triangulation globale au sens de la norme L_∞ sont équivalent à ceux de notre algorithme. Par conséquent, nous ne les afficherons pas sur les graphiques. Pour la première série de tests nous fixons $n = 50$ et $\sigma = 0.4$. Les résultats sont présentés sur la figure 4.35. Pour le premier mouvement de translation latéral ainsi que pour celui de rotation autour de la scène, les erreurs de reconstruction moyennes de

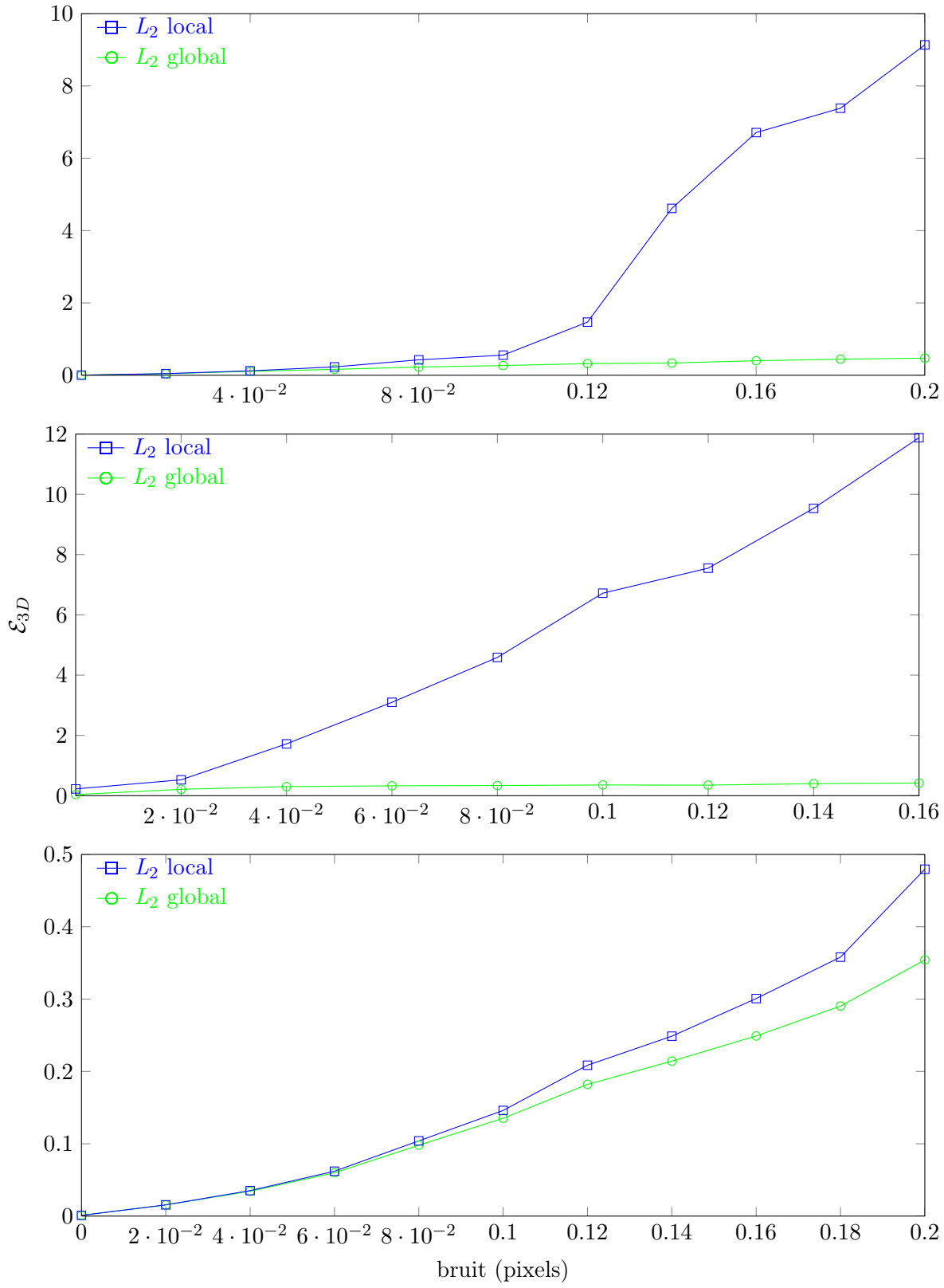


FIGURE 4.34 – Erreurs relatives de reconstruction 3D $\mathcal{E}_{3D}$ en fonction du bruit inséré dans les matrices de projection perspective $\mathbf{P}^j$ pour les trois types de configurations.

la méthode locale et de notre algorithme sont équivalentes et diminuent avec le nombre de caméras. Cependant, bien que les comportements soient identiques (les erreurs diminuent avec l'augmentation du nombre de caméras), les résultats obtenus sur le mouvement de translation vers la scène montrent un écart significatif entre la méthode locale et notre approche globale. Afin de vérifier si notre analyse reste valide pour un bruit plus élevé, nous avons réitéré cette expérience mais en ajoutant un bruit de variance égale à 1 sur les points projetés. Les résultats obtenus sont présentés sur la figure 4.36. Comparativement aux résultats obtenus pour le bruit de variance égale 0.4, on retrouve, sur les mouvements de translation latéral et de rotation, un comportement presque équivalent. En effet, il est probable que les légères différences observées soient dûes à un nombre de tirage insuffisant et qu'elles disparaissent en l'augmentant. Cependant le mouvement de translation vers la scène confirme que, même si le nombre de vues augmente, les solutions fournies par l'approche locale restent loin de l'optimum global. Comme pour les deux autres courbes, un nombre de tirages plus important permettrait de lisser la courbe bleue en supprimant le pic observé. Pour conclure, nous observons que, pour les deux approches, l'augmentation du nombre de vues permet de réduire l'erreur de reconstruction moyenne. Cependant, cette augmentation ne permet pas à la résolution locale d'afficher les mêmes performance que notre algorithme sur des cas difficiles.

Résultats sur des cas réels. Le but de ce paragraphe est de présenter, sur des cas réels, les résultats de notre algorithme comparativement à la résolution locale et à la résolution globale au sens de la norme L_∞. Comme les points 3D originaux ne sont pas connus, il est en général impossible de calculer l'erreur de reprojection 3D $\mathcal{E}_{3D}$. Ainsi, les tests effectués dans la littérature consistent à effectuer plusieurs triangulations à partir d'appariements connus et à comparer l'erreur de reprojection 2D $\mathcal{E}_{2D}$. Nous avons, pour notre part, choisi de comparer des erreurs 3D. Mais pour effectuer cette comparaison, il faut disposer comme nous venons de le mentionner, soit des points 3D initiaux, soit à défaut d'un modèle 3D suffisamment précis. Afin d'obtenir un tel modèle, nous utilisons les images du cylindre étalon décrit dans la sous-section 4.1.4 dédiée à l'estimation de la matrice fondamentale. Ce cylindre (dont le diamètre interne est égal à 70mm) est usiné très précisément et maintenu de telle sorte que, d'une part, il est certifié correspondre, à 3 micromètres près, à son modèle CAO et d'autre part, que sa forme interne ne s'altère que très peu dans le temps. Les images 4.37 et 4.38 montrent les deux images du cylindre ainsi que son modèle CAO. La reconstruction 3D se fera à l'aide d'un banc stéréoscopique préalablement calibré à l'aide de la boite à outils de calibrage de la librairie OPENCV [Bradski 2008]. Les paramètres de calibrage sont les suivants :

– Les paramètres intrinsèques des caméras sont donnés par :

$$\mathbf{K}_1 = \begin{pmatrix} 16358.49 & 0 & 1359.13 \\ 0 & 16369.56 & 983.46 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.155}$$

$$\mathbf{K}_2 = \begin{pmatrix} 16371.74 & 0 & 1183.66 \\ 0 & 16391.63 & 974.07 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.156}$$

– Le paramètre de distorsion radial κ_1 est égal à -0.05 pour les deux caméras.
– La transformation stéréoscopique est donnée par :

$$\mathbf{T} = \begin{pmatrix} 0.9203 & 0.0002 & -0.3910 & 228.0 \\ -0.0008 & -0.9999 & -0.0014 & 0.0435 \\ 0.3910 & 0.0016 & 0.9203 & 51.13 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.157}$$

Méthode	Distance moyenne (mm)	Ecart type (mm)	Distance maximale (mm)
Locale	+0.025/-0.008	0.077	+0.923/-0.633
Globale L_∞	+0.009/-0.005	0.033	+0.528/-0.412
Globale L_2	+0.010/-0.003	0.032	+0.523/-0.117

TABLE 4.5 – Distance moyenne, écart type, et distance maximale obtenues en recalant les points 3D triangulés par les trois méthodes testées sur le cylindre étalon.

On notera que cette expérience correspond bien à une reconstruction sans distorsions puisque $\kappa_1 = -0.05$ correspond à un déplacement en bordure d'image de 0.6 pixels. Afin de disposer de la meilleure précision possible, nous choisissons de calculer des appariements dans une fenêtre située à l'intérieur du cylindre sur laquelle un motif aléatoire a été projeté (cf. Figure 4.39). Ces appariements sont alors calculés avec une méthode de corrélation d'images numériques. Nous disposons ainsi de $m = 9000$ paires de points 2D $(\mathsf{q_i})_i$ appariés avec une précision sous-pixélique. A partir de ces correspondances et des paramètres de calibrage, nous effectuons une triangulation avec les trois méthodes. Nous obtenons ainsi, trois nuages de points 3D. Ces trois nuages sont alors recalés sur le modèle CAO à l'aide du logiciel de recalage 3D Geomagic®. Notons que ce recalage est fait au moyen d'une optimisation numérique et qu'il introduit donc un biais. Cependant, ce biais est identique pour les trois reconstructions et ne nuit donc pas à la comparaison. Ce même logiciel permet ensuite d'afficher l'écart de chaque nuage 3D à la forme nominale (la CAO du cylindre). Les résultats obtenus sont présentés sur les figures (4.40), (4.41), (4.42) et résumés dans la Table 4.5. La première reconstruction, obtenue à l'aide de la méthode locale, montre une dispersion relativement importante. En effet, on observe un écart maximum positif de 0.923mm et négatif de -0.633mm, l'écart moyen étant lui compris entre +0.025mm et -0.008mm avec un écart type de 0.077mm. Les deux reconstructions 3D globales sont plus précises. En effet, pour la reconstruction au sens de la norme L_∞ on observe un écart maximum positif de 0.528mm et négatif de -0.412mm, un écart moyen compris entre +0.009mm et -0.005mm et un écart type de 0.033mm. Pour notre approche globale, on obtient un écart maximum positif de 0.523mm et négatif de -0.117mm, un écart moyen compris entre +0.010mm et -0.003mm et un écart type de 0.032mm. On peut donc conclure que les approches globales apportent un gain de précision non négligeable, notamment au niveau de la régularité de la reconstruction 3D. De plus ce test nous permet de vérifier le résultat suivant : les précisions de la reconstruction 3D résolue globalement avec la norme L_2 et la norme L_∞ semblent équivalentes.

4.2.3.2 Triangulation avec prise en compte d'une distorsion radiale

Ce paragraphe est scindé en deux parties. Tout d'abord, nous effectuons une série de tests sur des données simulées, puis nous testons notre approche globale sur un cas réel. Pour les données simulées, nous utilisons, à de légères différences près, la même procédure que dans le paragraphe précédent (c.-à-d. même nombre de caméras, de points, de tirages...). Nous conservons un processus de génération et de perturbation des points 3D identique. Cependant, nous ne perturbons pas entièrement la matrice $\mathbf{P}$. En effet, nous avons choisi de perturber uniquement les paramètres extrinsèques afin d'étudier leur influence sur la précision de la triangulation. Pour l'ensemble des données simulées, nous fixons le paramètre de distorsion radiale à $\kappa_1 = -0.3$, soit un déplacement en bordure d'image de 27 pixels. Pour tous les tests réalisés, nous comparons notre approche à une méthode de régions de confiance.

Sensibilité au bruit sur les appariements. Les résultats obtenus sur cette expérience sont présentés sur les figures 4.44, 4.45, 4.46 et 4.47. Nous constatons tous d'abord que les com-

portements restent très similaires à la triangulation sans distorsions. En effet, l'ajout de bruit sur les appariements ne permet pas de distinguer les deux méthodes sur les mouvements de rotation et de translation latérale. Cependant, la différence de précision est visible pour le mouvement de translation vers la scène. Ceci semble indiquer que l'estimée initiale, issue de la résolution du système linéaire, n'est pas dans le bassin d'attraction de l'optimum global du problème avec distorsions. Il convient toutefois de nuancer cette observation en remarquant qu'un tel comportement n'est significatif que pour un bruit avec une variance $\sigma \geqslant 1$. Ensuite, nous constatons que les deux méthodes sont équivalentes lorsque le bruit est inséré dans les matrices de rotations. Ceci souligne donc la stabilité de cette configuration. Comme pour le cas sans distorsions, le mouvement de translation vers la scène se révèle très instable. En effet, dès que $\sigma > 0.004$, la différence des erreurs de reprojection 3D devient significative. Ceci indique donc que l'estimée initiale issue de la résolution du système n'est pas dans le bon bassin d'attraction. Pour le mouvement de translation latéral à la scène, il faut des valeurs de bruit importantes $\sigma > 0.1$ pour observer une différence.

Ensuite, afin de disposer de résultats sur un cas réel, nous avons reconstruit, à l'aide d'un banc stéréoscopique, un plan dont les images sont reportées sur la Figure 4.43. Comme pour le cylindre, les appariements sont calculés grâce à une méthode de corrélation d'images numériques et le banc est calibré de manière précise. Les paramètres de calibrage sont les suivants :

– Les paramètres intrinsèques des caméras sont donnés par :

$$\mathbf{K}_1 = \begin{pmatrix} 1718.87 & 0 & 511.551 \\ 0 & 1719.16 & 382.535 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.158}$$

$$\mathbf{K}_2 = \begin{pmatrix} 1720.9 & 0 & 512.95 \\ 0 & 1720.75 & 375.2 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.159}$$

– Le paramètre de distorsion radial κ_1 est égal à -0.23 pour les deux caméras, soit un déplacement en bordure d'image de 10 pixels.
– La transformation stéréoscopique est donnée par :

$$\mathbf{T} = \begin{pmatrix} 0.9697 & -0.0054 & 0.2441 & -130.1 \\ 0.0053 & 0.9999 & 0.0010 & -1.068 \\ -0.2441 & 0.0003 & 0.9697 & 16.10 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.160}$$

Le plan est un objet particulièrement adéquat pour mesurer l'influence des erreurs de triangulation en présence de distorsions radiales. En effet, si nous effectuons une reconstruction 3D idéale de cet objet, tous les points triangulés sont coplanaires. Mais, en pratique ces points ne le sont pas. Ces erreurs de triangulation planaires sont généralement dues aux effets de la distorsion. Ainsi, afin de quantifier l'influence de ce paramètre, nous procédons comme suit : nous triangulons 500 points sur l'ensemble du plan, puis, à l'aide du logiciel Geomagic®, nous déterminons le plan moyen (au sens des moindres carrés) passant par ce nuage. Ceci nous permet de calculer les erreurs suivantes : la distance moyenne et l'écart type des points au plan ainsi que la distance maximum des points au plan. l'ensemble des résultats obtenus est reporté sur la Table 4.6. Même si l'écart est minime, nous observons que les résultats obtenus sont en faveur de la triangulation globale. Ceci souligne donc que, lorsque les paramètres de calibrages sont précisément connus, alors la solution fournie par la résolution du système linéaire est très proche de la solution globale du problème de triangulation avec distorsion radiale.

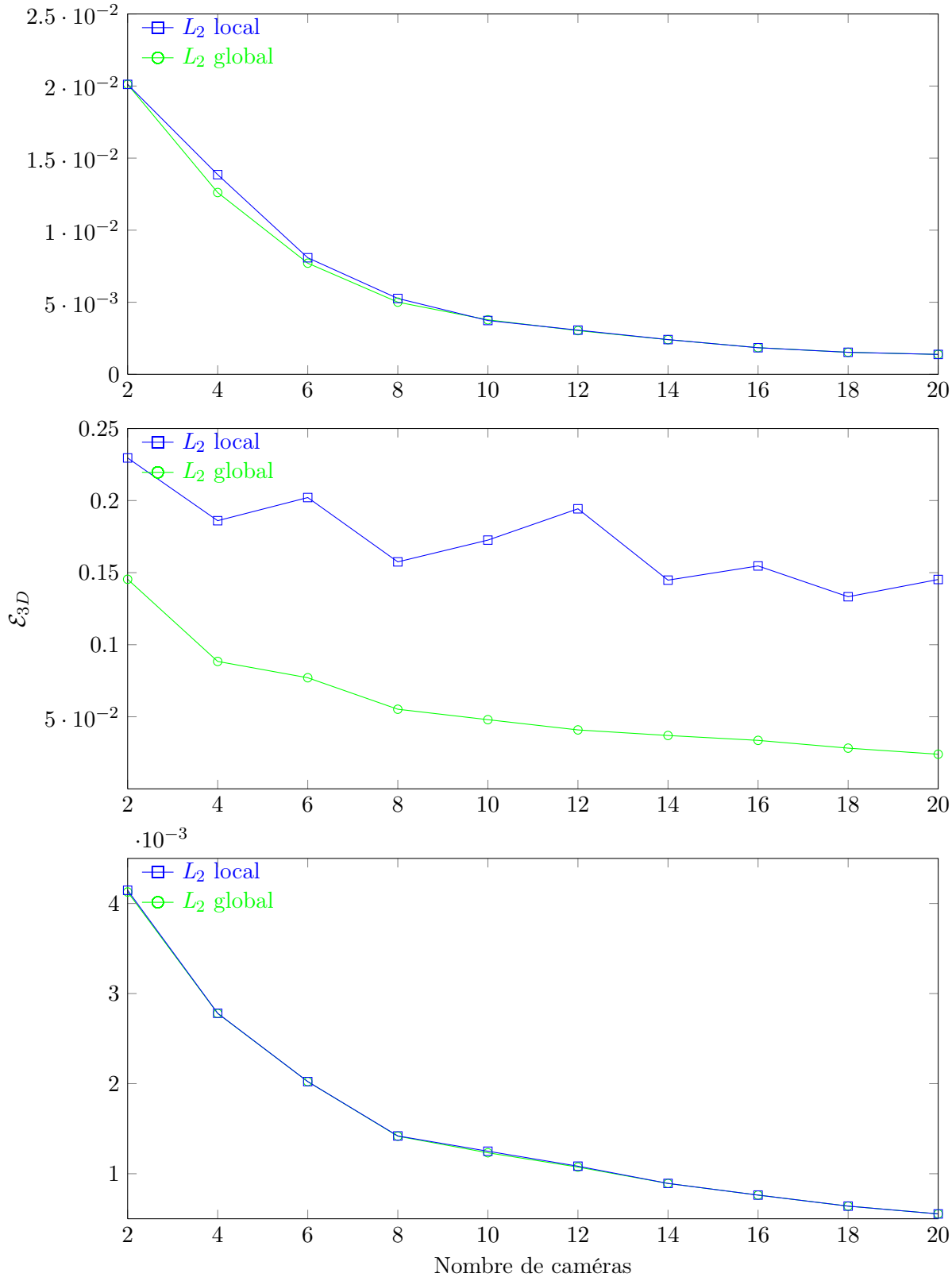


FIGURE 4.35 – Pour les trois types de configurations, erreurs moyennes de reconstruction 3D en fonction du nombre de vues avec un bruit Gaussien de variance $\sigma = 0,4$ inséré dans les points projetés.

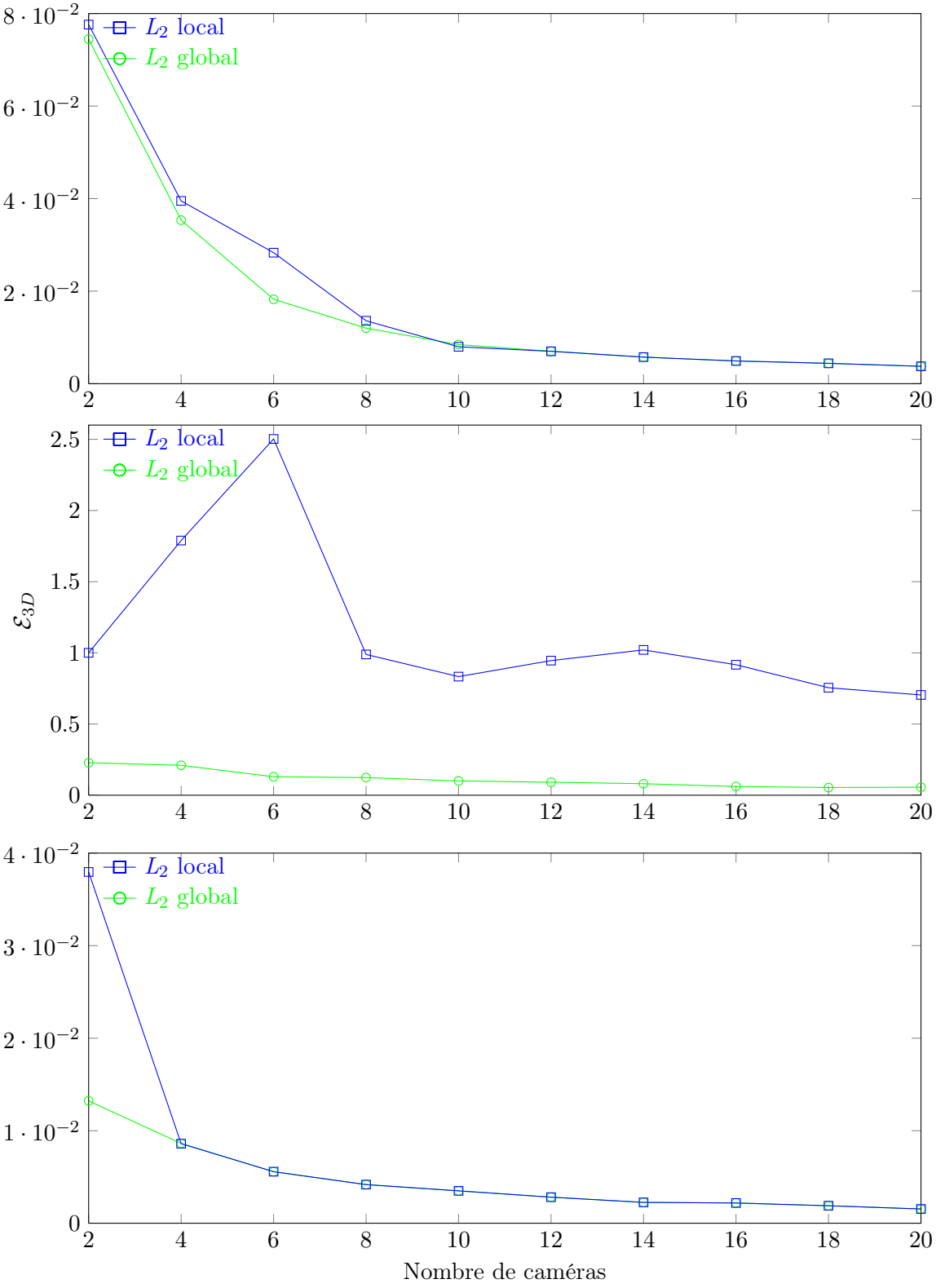


FIGURE 4.36 – Pour les trois types de configurations, erreurs moyennes de reconstruction 3D en fonction du nombre de vues avec un bruit Gaussien de variance $\sigma = 1$ inséré dans les points projetés.

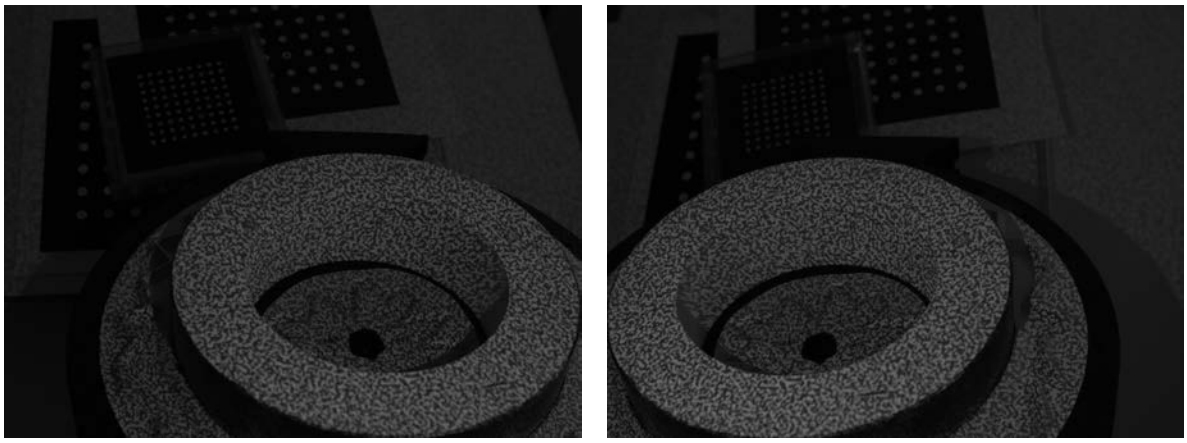


FIGURE 4.37 – Images du cylindre étalon, de 70mm de rayon interne, à partir duquel nous réalisons notre série de tests.

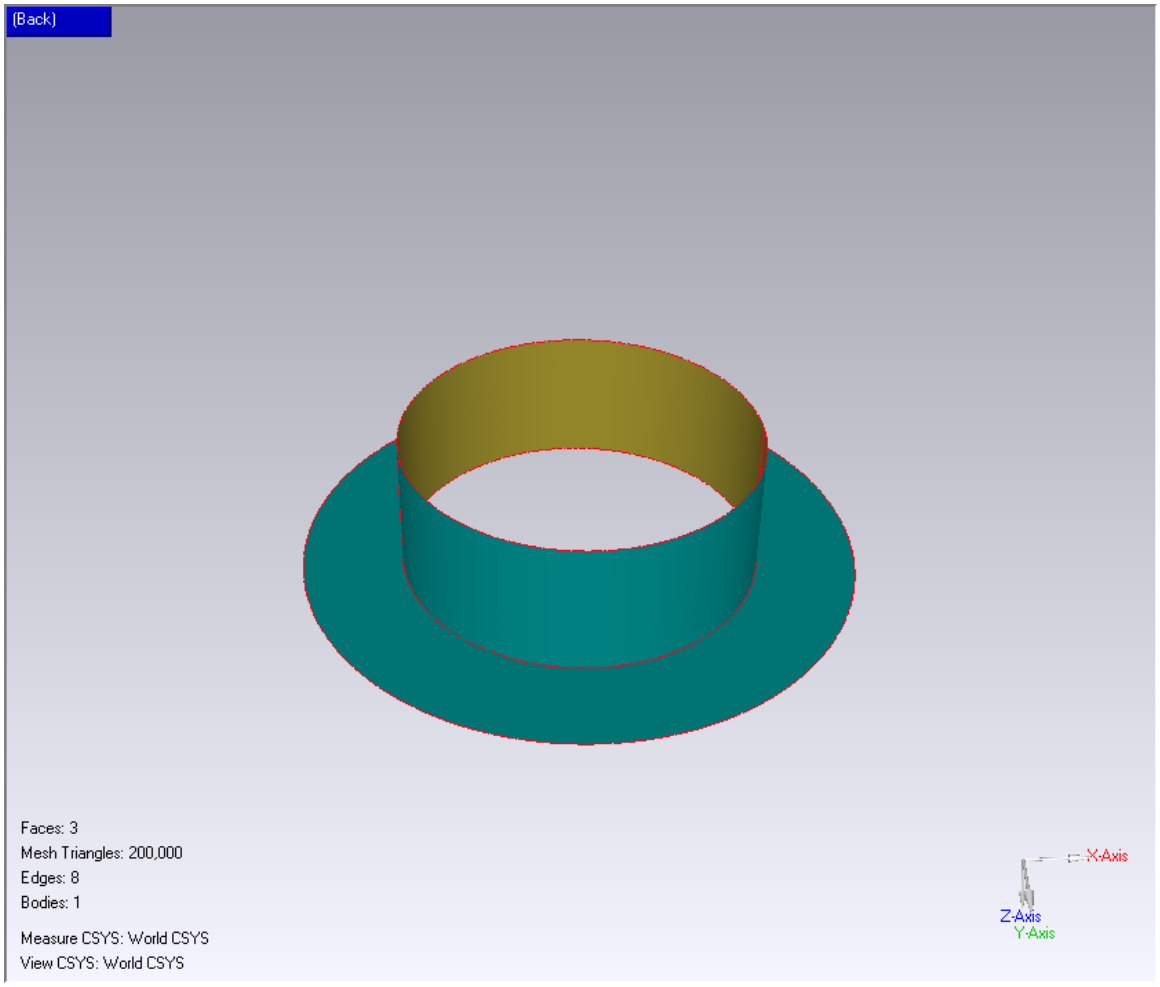
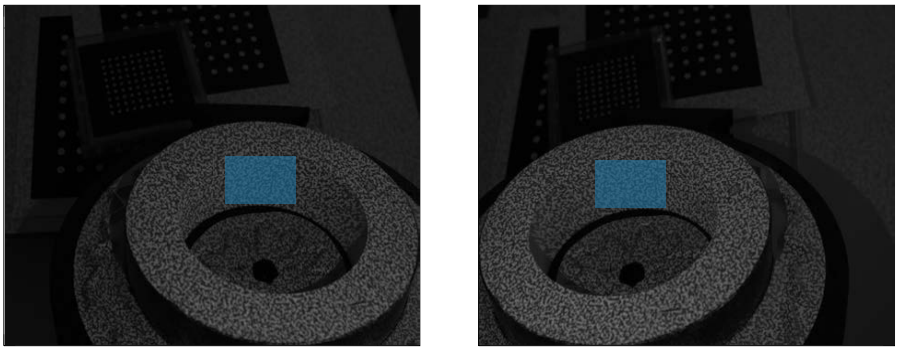


FIGURE 4.38 – CAO du cylindre étalon utilisé pour nos tests.

Méthode	Distance moyenne (mm)	Ecart type (mm)	Distance maximale (mm)
Locale	+0.008/-0.007	0.011	+0.040/-0.078
Globale	+0.005/-0.006	0.007	+0.019/-0.017

TABLE 4.6 – Distance moyenne, écart type, et distance maximale obtenues en calculant le plan moyen passant par les points triangulés de notre objet de test.



1 2

FIGURE 4.39 – Images du cylindre sur lequel un motif aléatoire a été projeté afin de permettre un appariement de points par des méthodes de corrélation d'images numériques. Les appariements se situent dans les boites englobantes bleues.

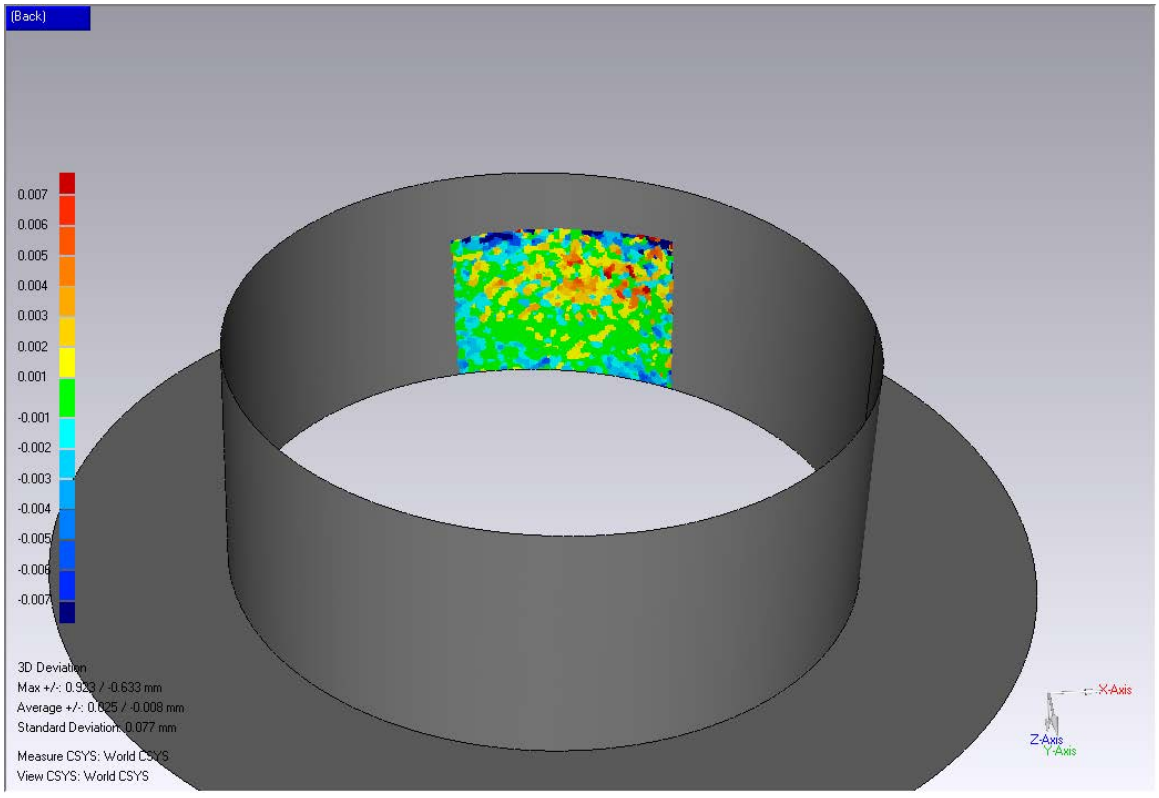


FIGURE 4.40 – Ecart à la forme nominale du nuage 3D triangulé avec l'approche locale

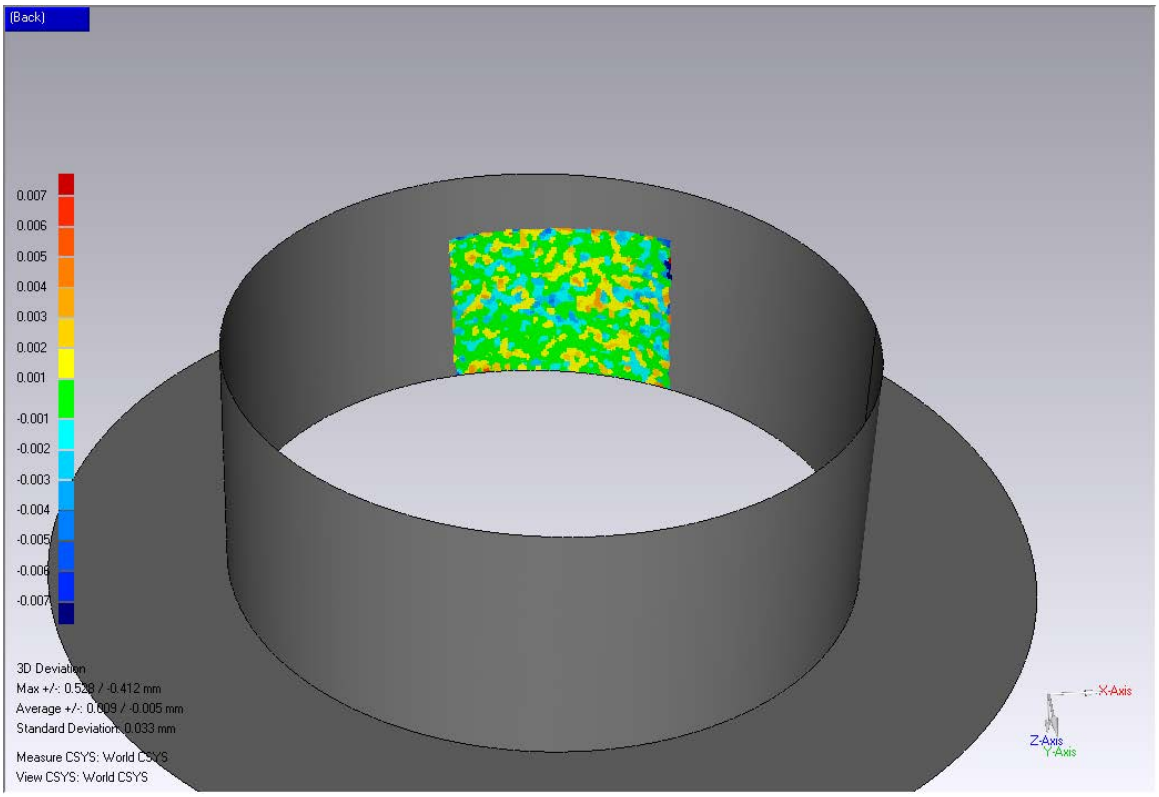


FIGURE 4.41 – Ecart à la forme nominale du nuage 3D triangulé avec l'approche globale au sens de la norme L_∞

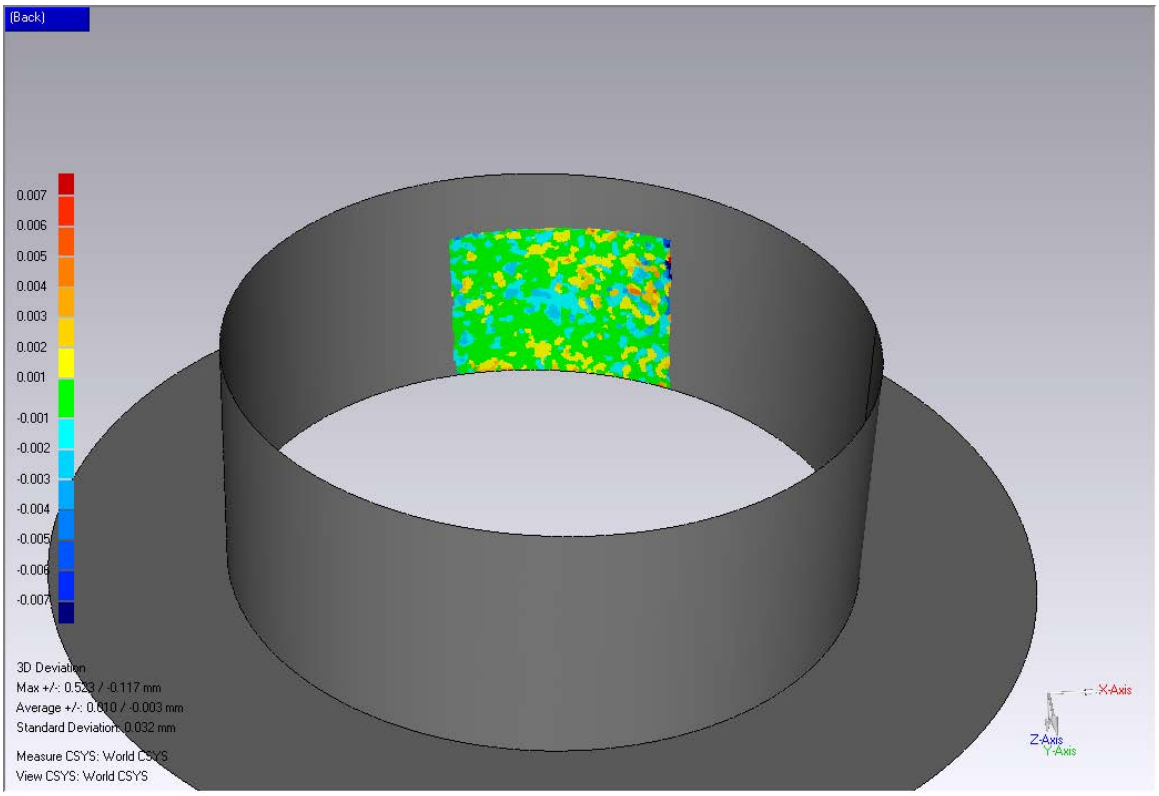


FIGURE 4.42 – Ecart à la forme nominale du nuage 3D triangulé avec notre approche globale

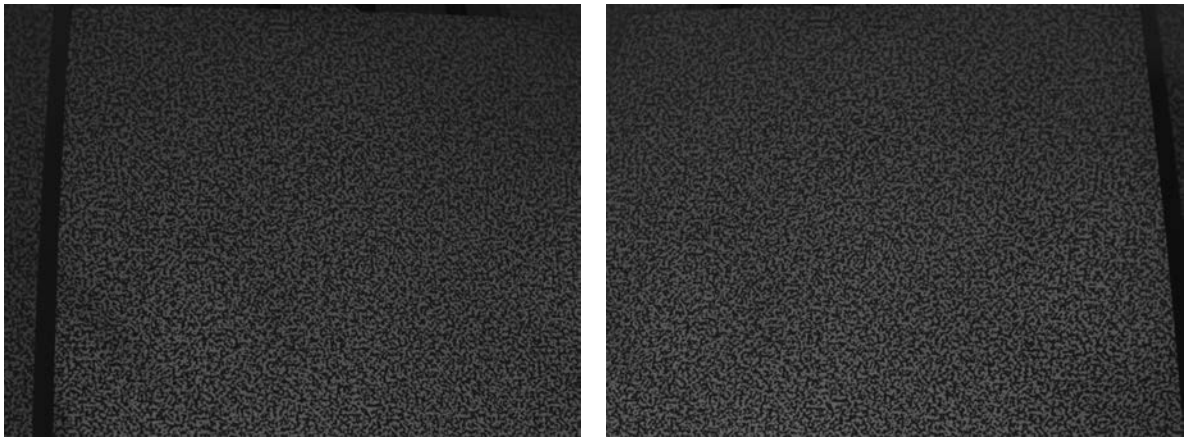


FIGURE 4.43 – Images du plan à partir duquel nous réalisons notre test de reconstruction 3D avec prise en compte des distorsions.

4.2.4 Conclusions

Nous avons dans ce chapitre, appliqué l'optimisation globale polynomiale et rationnelle à la minimisation des distorsions projectives, l'estimation de la matrice fondamentale et la triangulation. Dans la première étude, nous avons mis en avant plusieurs nouveaux critères polynomiaux. La minimisation globale de ces critères a permis de comparer leur efficacité et donc d'exhiber plusieurs nouvelles approches plus précises que l'existant. La deuxième étude était consacrée à l'étude d'un problème classique de vision artificielle : l'estimation de la matrice fondamentale. D'un point de vue mathématique, cette estimation se présente comme un problème de minimisation sous contrainte d'une fonction polynomiale à 9 paramètres. Après avoir proposé une méthode d'évaluation, nous avons montré que notre approche fournissait, sur des cas difficiles, de meilleurs résultats que la méthode classique des 8-points. Enfin, pour le problème de la reconstruction 3D multivues, formulé comme la minimisation d'une somme de fractions rationnelles, nous avons mené deux étude similaires. Dans un premier temps, nous avons testé notre algorithme sur le problème de reconstruction classique (c.-à-d. linéaire) et montré que notre méthode de Programmation Rationnelle obtenait de meilleurs résultats dans des configurations difficiles. De plus, lorsque les mouvements sont stables et que les paramètres de calibrages sont bien connus, nous avons constaté que les méthodes locales étaient équivalentes au méthodes globales. En outre, cette dernière étude a montré, sur un cas réel, que les triangulations globales au sens de la norme L_∞ et L_2 étaient équivalentes. Enfin, nous avons mené la même étude sur le problème de la triangulation avec prise en compte des distorsions. Dans ce cas nous avons constaté que les méthodes locales et globales étaient trés proches, la différence n'étant significative que sur des cas particulièrement difficiles. On retiendra au final que, sur des expériences de triangulation (avec ou sans distorsions) réalisées dans des conditions standards, l'apport de notre approche reste minime en regard de son temps de calcul largement supérieur à celui d'une triangulation résolue avec une méthode locale.

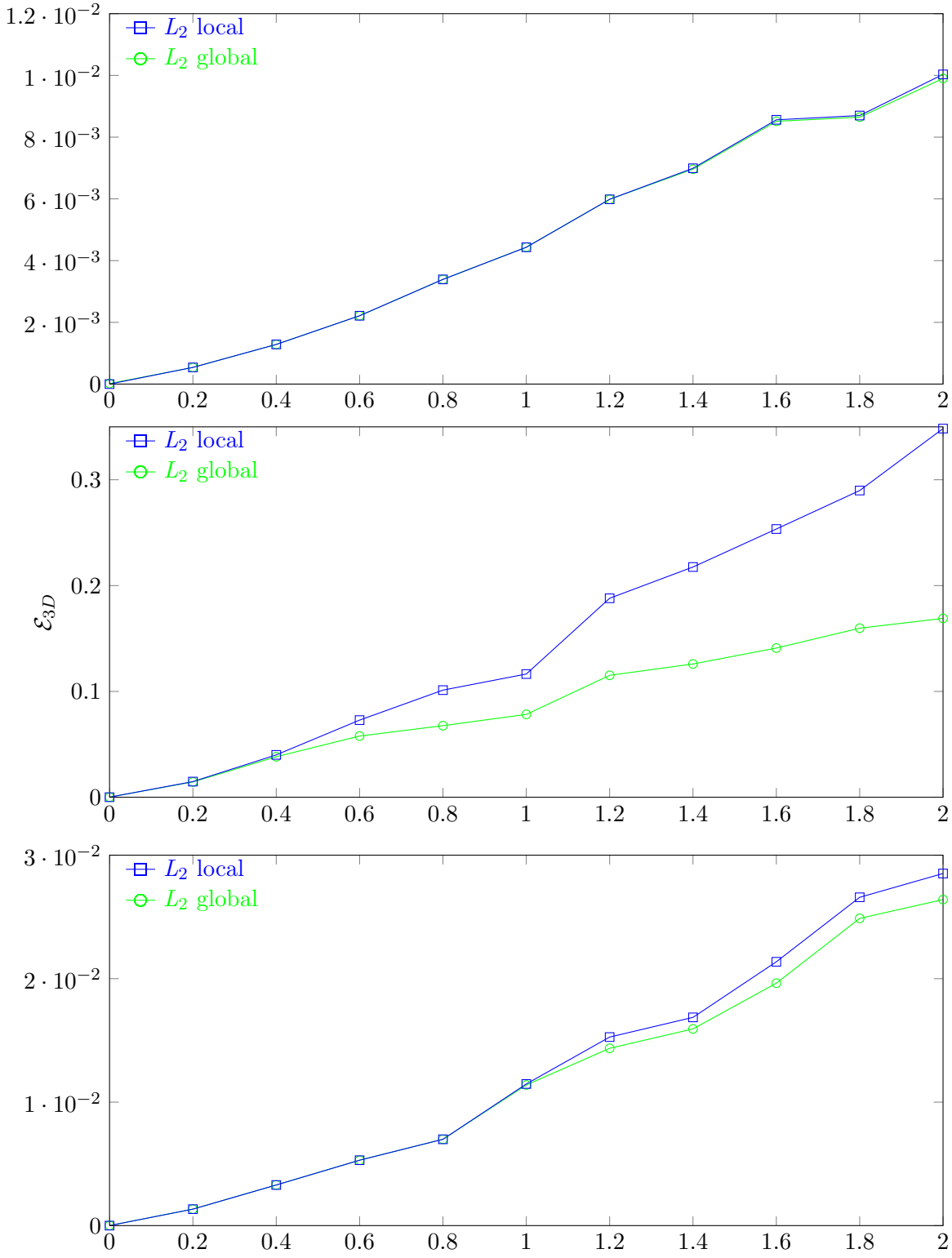


FIGURE 4.44 – Erreurs relatives de reconstruction 3D en fonction du bruit inséré dans les points projetés pour les trois types de configurations.

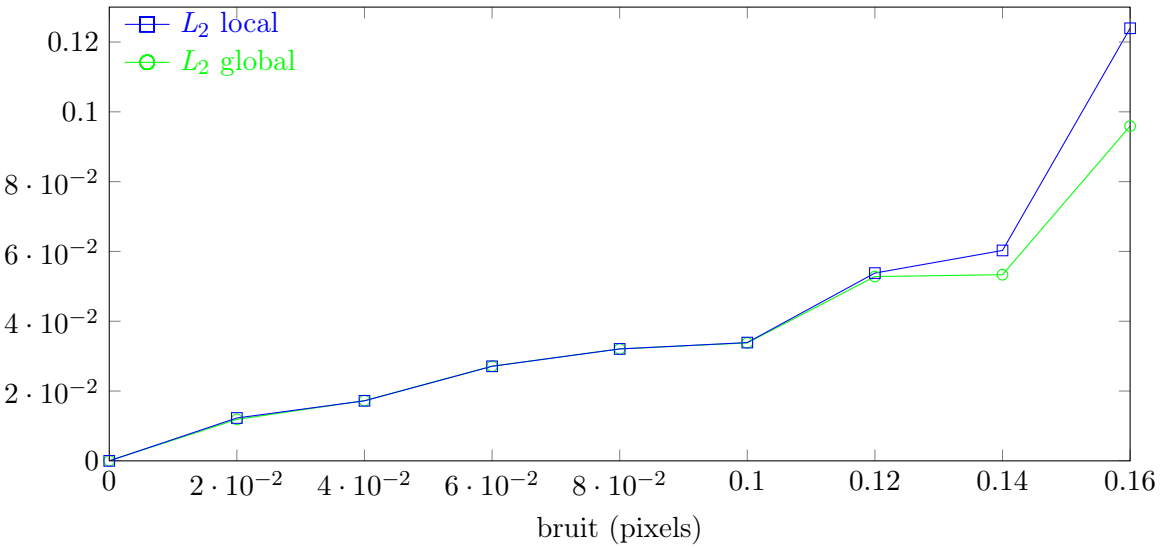


FIGURE 4.45 – Erreurs relatives de reconstruction 3D en fonction du bruit inséré dans les matrices de rotation pour le mouvement de rotation.

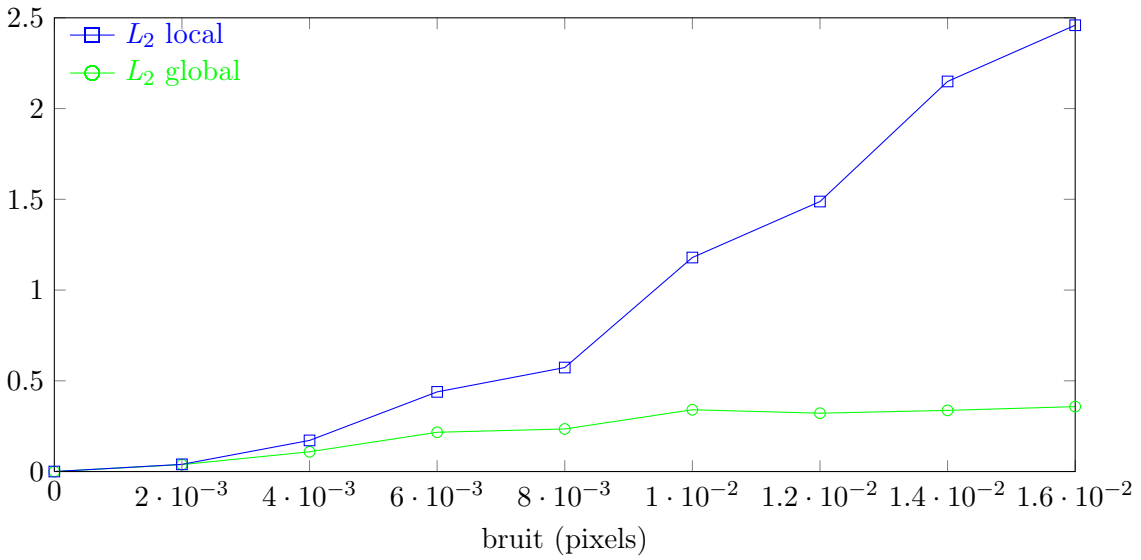


FIGURE 4.46 – Erreurs relatives de reconstruction 3D en fonction du bruit inséré dans le vecteur de translation lors du déplacement vers la scène.

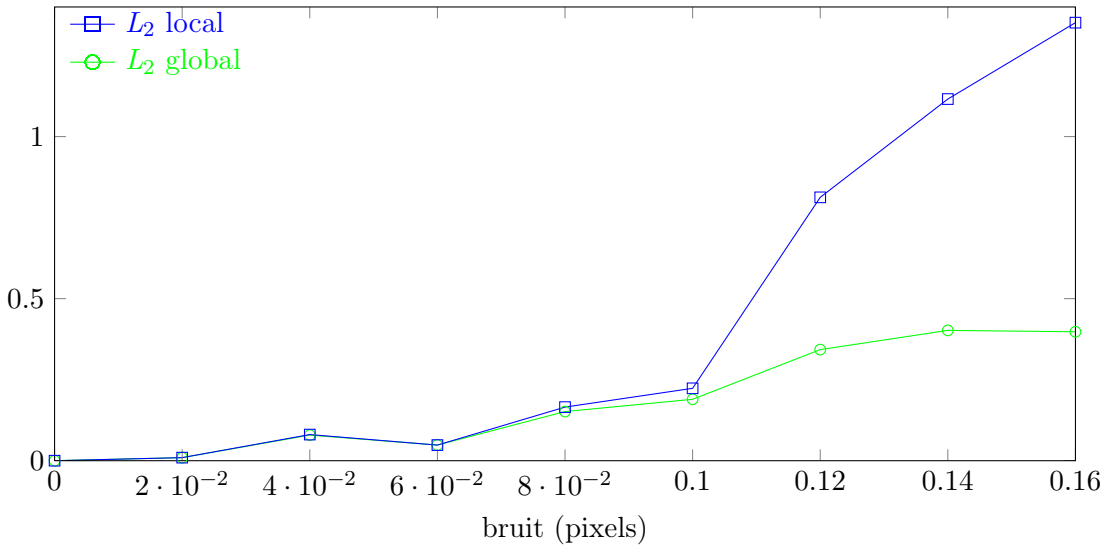


FIGURE 4.47 – Erreurs relatives de reconstruction 3D en fonction du bruit inséré dans le vecteur de translation lors du déplacement latéral à la scène.

5 Conclusions et perspectives

5.1 Bilan des contributions

Une première contribution de cette thèse est constituée d'apports théoriques dans le domaine de l'optimisation globale. Tout d'abord, nous avons étudié l'optimisation polynomiale par théorie des moments. C'est un outil de nature théorique et, par conséquent, peu utilisé en vision par ordinateur. Outre cette complexité théorique, cette méthode ne peut uniquement traiter que des problèmes de minimisation où la fonction coût est un polynôme. Notre principal apport consiste à étendre cette approche à la minimisation globale d'une somme de fractions rationnelles. Grâce à l'application directe de l'optimisation polynomiale, et notamment la prise en compte du faible couplage des variables, nous avons montré qu'il existait une première solution à ce dernier problème. Ensuite, à l'aide de nouveaux éléments théoriques, nous avons démontré qu'il existait une autre solution au problème de la minimisation globale d'une somme de fractions rationnelles. Enfin, nous avons départagé ces deux nouvelles méthodes à l'aide d'une série de tests numériques.

Une deuxième contribution consiste à appliquer l'optimisation polynomiale ainsi que son extension, la programmation rationnelle, à des problèmes classiques en vision par ordinateur. La première application concerne le problème de la minimisation de distorsions projectives issues du processus de rectification d'images. Pour l'approche que nous avons choisie, il n'existait pas de critères polynomiaux. Nous en avons donc proposé quatre différents. Pour ces critères, l'optimisation polynomiale nous a servi à départager le plus performant, sur plusieurs types d'images. La deuxième application concerne l'estimation de la matrice fondamentale. Nous avons montré que ce problème, difficile d'un point de vue mathématique, peut être efficacement résolu par l'optimisation polynomiale par théorie des moments.

5.1.1 Extension de l'optimisation polynomiale à l'optimisation rationnelle

Dans le chapitre 2, nous avons fait un survol des méthodes d'optimisation afin de fournir au lecteur un panorama des méthodes classiquement utilisées en vision par ordinateur. Ces approches se divisent en deux grandes familles : les méthodes locales et les méthodes globales. Parmi les méthodes locales sans contraintes, les plus utilisées sont les méthodes d'ordre 2, car, pour beaucoup des problèmes rencontrés, les formules analytiques des gradients et des hessiennes sont disponibles. Lorsque le problème est sous contraintes, la solution la plus répandue consiste à résoudre le problème sans contraintes puis à projeter la solution obtenue

sur l'espace des contraintes. La faisabilité de cette technique dépend fortement de la capacité à déterminer un opérateur de projection. Dès lors, cette approche reste réservée à des problèmes particuliers. Dans tous les cas, sauf hypothèses de convexités additionnelles, il n'est pas possible avec ces méthodes d'obtenir un certificat d'optimalité. Les méthodes globales pallient cet inconvénient. Cependant, leur mise en œuvre se fait généralement au détriment du temps de calcul et de la généralité. La méthode la plus utilisée pour la vision artificielle est l'algorithme par séparation et évaluation (branch and bound en anglais). Mais cette méthode ne peut être utilisée que lorsque la phase d'évaluation est réalisée de manière précise et rapide. Ainsi, afin de garantir cette propriété, il est nécessaire de tirer parti de la forme particulière du problème. Par exemple, dans le cas de la minimisation d'une somme de fractions rationnelles, il faut utiliser la quasi-convexité des numérateurs et dénominateurs. Nous avons donc vu qu'il n'existait pas de méthode générique pour résoudre globalement la minimisation d'un polynôme ou d'une somme de fractions rationnelles sous contraintes. Ensuite, dans le chapitre 3, nous avons présenté le principe de l'optimisation par théorie des moments. Dans un premier temps, nous avons rappelé que tout problème de minimisation $\min_{x \in K} f$ peut se formuler comme un problème linéaire dans l'espace des mesures. Puis, dans le cas où f est un polynôme et K est un ensemble semi-algébrique de base engendré par des polynômes, nous avons vu que le problème dans l'espace des mesures pouvait être ramené à un problème particulier : le problème des moments sur K. Ce problème peut se formuler ainsi : quand une séquence de nombres représente-t-elle les moments d'une mesure supportée sur K ? Ensuite, nous avons vu que ce dernier problème pouvait être "tronqué" et approché, aussi précisément que l'on veut, par une hiérarchie de problèmes convexes de dimension finie. Ce processus est résumé sur la Figure 5.1.

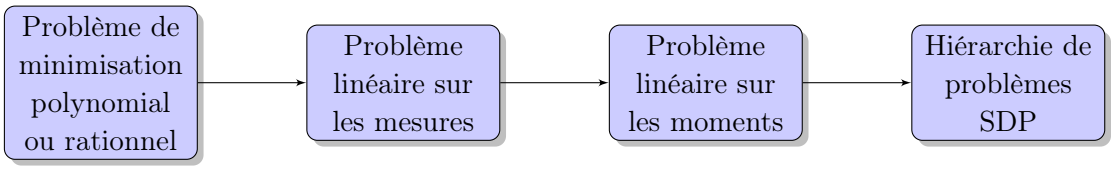


FIGURE 5.1 – Diagramme récapitulatif de l'optimisation par théorie des moments.

Enfin, dans un cas particulier où les variables sont faiblement couplées (c.-à-d. qu'il existe un schéma de parcimonie), nous avons vu qu'un processus identique conduisait à une hiérarchie permettant de considérer un plus grand nombre de variables. Dans le chapitre 3, nous avons ensuite exposé notre contribution qui consiste à étendre la théorie précédente au cas de la minimisation d'une somme de fractions rationnelles. Cette extension a pu être réalisée au moyen de deux méthodes : les Epigraphes Creux et la Programmation Rationnelle. Pour chacune des deux méthodes, nous avons prouvé l'équivalence du problème dans l'espace des mesures avec le problème original ainsi que la convergence de la suite de solutions issue de la hiérarchie SDP associée vers la solution du problème initial. Enfin, nous avons testé les deux approches sur plusieurs problèmes tests. Ces tests ont souligné le net avantage de la Programmation Rationnelle au détriment des Epigraphe Creux. Nous avons donc conclu que la Programmation Rationnelle était la meilleure des deux méthodes pour résoudre globalement la minimisation, sous contraintes, d'une somme de fractions rationnelles.

5.1.2 Applications à la vision multivues

Dans le chapitre 4, nous avons appliqué l'optimisation globale par théorie des moments à trois problèmes classiques en vision par ordinateur. Le premier concerne la minimisation des distorsions liées à la rectification d'images projectives. Ce processus consiste à améliorer la qualité des homographies de rectification en minimisant les distorsions qu'elles induisent

dans les images. Pour cela, nous avons vu qu'il existait plusieurs familles de méthodes. Pour les problèmes sur lesquels nous avons choisi d'appliquer notre méthode, il n'existait pas de critères analytiques. Nous avons alors proposé quatre nouveaux critères : trois polynomiaux et un rationnel. Puis, l'optimisation polynomiale nous a permis de déterminer lequel d'entre eux était le plus efficace. La deuxième étude était consacrée à un problème classique de vision artificielle : l'estimation de la matrice fondamentale. Dans le cas d'une paire d'images prise à l'aide d'un banc stéréoscopique, cette matrice permet d'associer un point d'une image à sa droite épipolaire dans l'autre image. D'un point de vue mathématique, cette estimation se présente comme un problème de minimisation sous contraintes d'une fonction polynomiale à 9 paramètres, pour lequel il est parfois difficile de trouver une bonne estimée initiale. Sur l'ensemble des tests présentés, nous avons montré que notre approche fournissait de meilleurs résultats que la méthode classique des 8-points. Enfin, dans la dernière partie nous avons étudié le problème de reconstruction 3D multivues. Ce problème se présente comme la minimisation d'une somme de fractions rationnelles à 3 inconnues, le nombre de fractions dépendant du nombre de vues considérées. Nous avons considéré deux variantes de ce problème. Dans la première, nous avons traité le problème sans prendre en compte les distorsions. Il résulte alors un problème avec des numérateurs et dénominateurs de degrés égaux à 2, pour lequel il existe des résolutions globales essentiellement basées sur des algorithmes de séparation et d'évaluation. Dans la deuxième, nous avons pris en compte une distorsion radiale d'ordre 1. Il résulte alors un problème avec des numérateurs et dénominateurs de degrés égaux à 6, pour lequel aucune méthode globale n'a été mise en œuvre. Dans les deux cas, nous avons montré, sur des cas réels et simulés que notre algorithme donnait de meilleurs résultats que l'approche locale classique. En outre, dans le premier cas, en nous comparant avec un autre algorithme global, nous avons pu montrer que la triangulation au sens de la norme L_∞ et la triangulation au sens de la norme L_2 donnaient des résultats équivalents.

5.2 Perspectives et potentiel de la méthode d'optimisation

Tout d'abord, afin d'évaluer pleinement le potentiel de notre méthode il conviendrait de tester les résultats qu'elle peut fournir lorsque l'on considère des modèles de distorsion plus poussés. Ainsi, il faudrait, tout d'abord, augmenter l'ordre de la distorsion radiale jusqu'à trois. Ensuite, il conviendrait de considérer un modèle avec des distorsions prismatiques et de décentrage. Cependant, comme ces modèles sont rarement utilisés en pratique, il faudrait se tourner vers des problèmes de triangulation particuliers. Ainsi, à l'aide de changements de variables, il serait intéressant de considérer des modèles de distorsion plus complexes comme ceux d'optiques "fisheyes". De manière plus générale, tout au long de ce mémoire, nous avons insisté sur le fait que notre approche était générique, c'est-à-dire qu'elle pouvait résoudre n'importe quel type de problème formulé comme une somme de fractions rationnelles. Dans ce cadre-là, nous avons commencé des investigations sur un problème pour lequel il n'existe pas de méthode proposant une résolution globale : la triangulation de ligne. Ce problème (cf. [Bartoli 2005]) peut être formulé, à l'aide des coordonnées de Plücker, de la manière suivante :

$$\begin{aligned} &\min_{\mathbf{L}\in\mathbb{R}^6} \sum_{i=1}^{n} \frac{1}{\left(\mathbf{L}^\top \widetilde{\mathbf{P}}_i^1\right)^2 + \left(\mathbf{L}^\top \widetilde{\mathbf{P}}_i^2\right)^2} \sum_{k=1}^{l(i)} \mathsf{q}_{i,k}^\top \widetilde{\mathbf{P}}_i \mathbf{L} \\ &\text{s.l.c} \quad \|\mathbf{L}\|_F^2 = 1 \\ &\qquad \mathbf{L}^\top \mathbf{K} \mathbf{L} = 0 \text{ avec } \mathbf{K} \triangleq \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}, \end{aligned} \tag{5.1}$$

où $\widetilde{\mathbf{P}}_i \in \mathbb{R}^{3\times 6}$ représente la matrice de projection de lignes dans la $i^{\text{ème}}$ vue et $\widetilde{\mathbf{P}}_i^j$ sa $j^{\text{ème}}$ ligne, $\mathsf{q}_{i,k}^\top$ représente le $k^{\text{ème}}$ point de contour de la $i^{\text{ème}}$ vue et $l(i)$ le nombre de points de

contour dans la $i^{\text{ème}}$ vue. Ce problème comporte donc deux contraintes non-linéaires pour 6 variables d'optimisation. Les premiers résultats obtenus sont encourageants, mais manquent encore de recul pour figurer dans ce mémoire.

Cependant, le potentiel le plus intéressant de la méthode de Programmation Rationnelle réside dans sa capacité à prendre en compte des schémas de parcimonie structurée vérifiant la propriété d'intersection courante. Or cette propriété est très présente en vision par ordinateur. En effet, on va chercher souvent à estimer un objet mathématique en fonction de données mesurées. Cependant, une approche courante, appelée génériquement « ajustement de faisceaux » consiste à réestimer les données mesurées durant le processus d'optimisation afin d'augmenter la précision finale. Mais il y a, en général, beaucoup de données mesurées. Cependant, comme nous l'avons vu dans le chapitre 4, cette approche n'est possible que grâce à la structure creuse de la matrice hessienne. Or, cette structure est révélatrice d'un schéma de parcimonie qui pourrait être utilisé. Considérons un cas simple : l'estimation d'une homographie planaire $\mathbf{H} = (h_{ij})_{1\leqslant i,j\leqslant 3}$ à l'aide de N points 2D appariés $(\mathsf{q}_i, \mathsf{q}'_i) = (u_i, v_i, u'_i, v'_i)$. Dans ce cas, on dispose d'un problème :

$$\begin{aligned} \min_{\mathbf{H}} \sum_{i=1}^{N} \left[\left(u'_i - \frac{h_{11}u_i + h_{12}v_i + h_{13}}{h_{31}u_i + h_{32}v_i + h_{33}} \right)^2 + \left(v'_i - \frac{h_{21}u_i + h_{22}v_i + h_{23}}{h_{31}u_i + h_{32}v_i + h_{33}} \right)^2 \right] \\ \text{s.l.c.} \quad \|\mathbf{H}\|_F^2 = 1. \end{aligned} \tag{5.2}$$

Considérons le schéma de parcimonie structurée définit par $I_k = \{\mathbf{H}, \mathsf{q}_k, \mathsf{q}'_k\}$. Comme $I_k \cap I_h = \{\mathbf{H}\}\ \forall k \neq h$, on peut facilement montrer que celui-ci vérifie la propriété d'intersection courante. Ainsi, il est possible de réestimer tous les points q_i, c'est-à-dire de résoudre le problème :

$$\begin{aligned} \min_{\mathbf{H},(\mathsf{q}_i,\mathsf{q}'_i)_{i=1,\dots,N}} \sum_{i=1}^{N} \left[\left(u'_i - \frac{h_{11}u_i + h_{12}v_i + h_{13}}{h_{31}u_i + h_{32}v_i + h_{33}} \right)^2 + \left(v'_i - \frac{h_{21}u_i + h_{22}v_i + h_{23}}{h_{31}u_i + h_{32}v_i + h_{33}} \right)^2 \right] \\ \text{s.l.c.} \quad \|\mathbf{H}\|_F^2 = 1. \end{aligned} \tag{5.3}$$

De manière similaire, dans tout problème où l'on peut former un schéma de parcimonie $I_k = \{\mathbf{M}_k, \mathsf{m}_k\}$ avec $\mathbf{M}_k$ une inconnue commune à toutes les fractions et m_k une inconnue propre à chacune des fractions, alors ce schéma vérifie la propriété d'intersection courante et il est possible de réestimer les données mesurées dans le processus de minimisation. Bien que la taille des matrices utilisées dépende fortement de la taille du bloc de variables $\mathbf{M}_k$, un tel processus ouvre, à terme, la voie vers la résolution globale du problème d'ajustement de faisceaux.

De manière plus générale, ces travaux soulignent l'intérêt de mener des investigations à l'intersection entre la vision artificielle et l'optimisation globale. En effet, la vision par ordinateur constitue un champ applicatif idéal pour les méthodes d'optimisation globales. Contrairement à beaucoup de problèmes rencontrés dans l'industrie, les problèmes issus de la vision par ordinateur sont formulés de manière analytique, avec des propriétés théoriques fortes (p. ex. quasi-convexité) et un nombre limité de variables. En outre, de la recherche de zéros à des problèmes de programmation entière, ces problèmes couvrent un large éventail de formulations. Enfin, leurs structures théoriques particulières permettent de mener des études théoriques très en amont de l'utilisation pratique[Aholt 2011, Aholt 2012]. On aura donc compris que, loin de n'être qu'un simple domaine applicatif, la vision artificielle permet, au travers des problèmes qu'elle soumet, de faire émerger de nouvelles adaptations de techniques d'optimisation globale.

A

Images tests stéréoscopiques

Ces images tests sont disponibles sur la page web http://www.vsg.dcu.ie/code.html.

FIGURE A.1 – Paire d'images **Arch**.

FIGURE A.2 – Paire d'images **Drive**.

FIGURE A.3 – Paire d'images **Lab**.

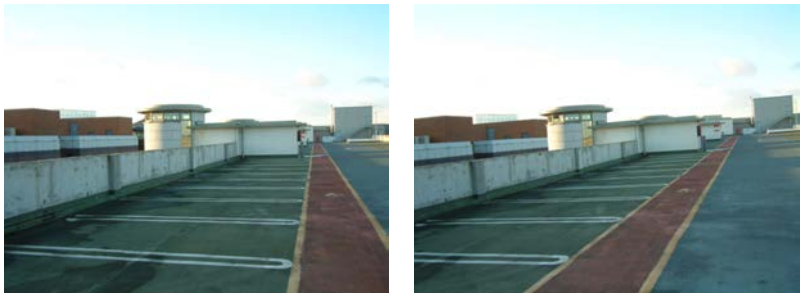


FIGURE A.4 – Paire d'images **Roof**.

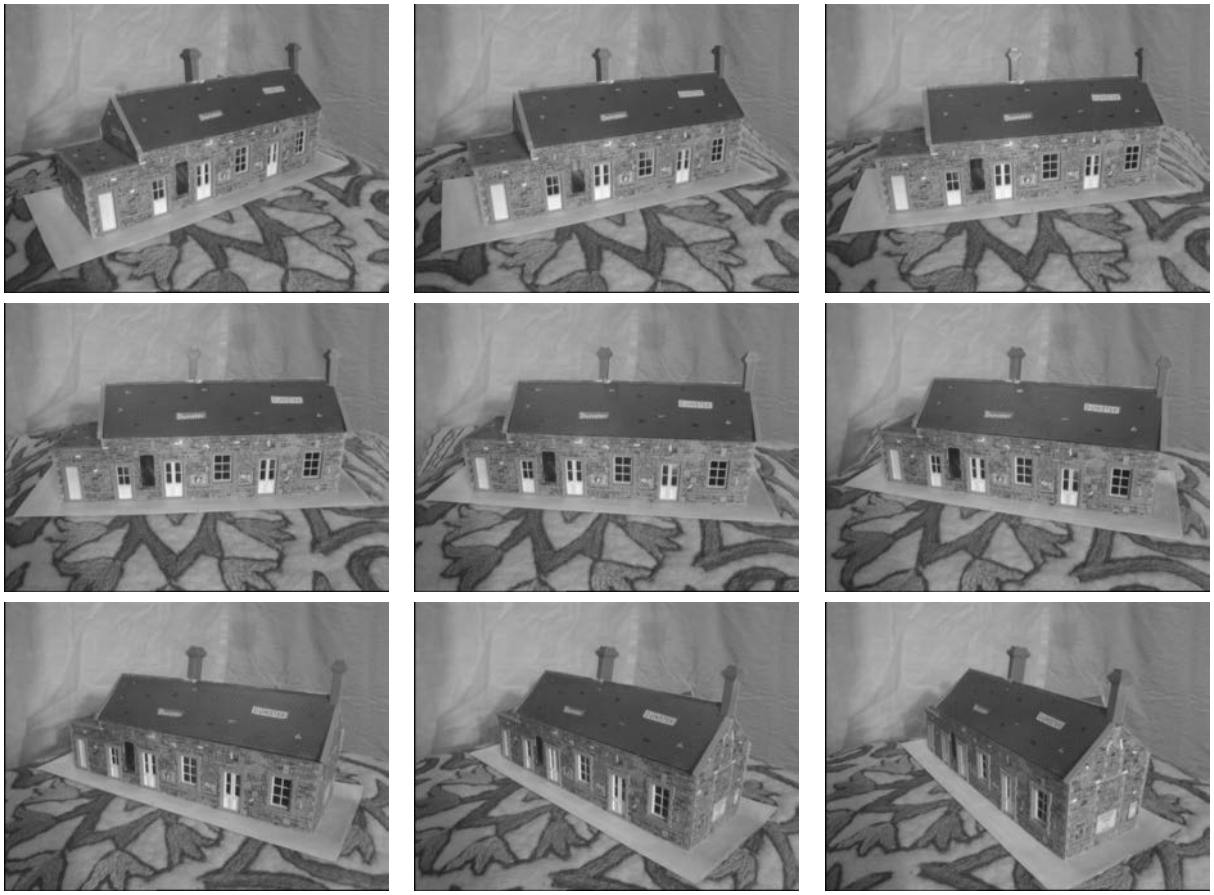
FIGURE A.5 – Paire d'images **Boxes**.

FIGURE A.6 – Paire d'images **Yard**.

B

Images tests de la série House

Ces images sont aussi disponibles sur le site www.robots.ox.ac.uk/~vgg/data/data-mview.html.



C

Paires d'images tests Library et Merton

Ces images sont aussi disponibles sur le site www.robots.ox.ac.uk/~vgg/data/data-mview.html.

FIGURE C.1 – Paire d'images **Library**.

FIGURE C.2 – Paire d'images **Merton1**.

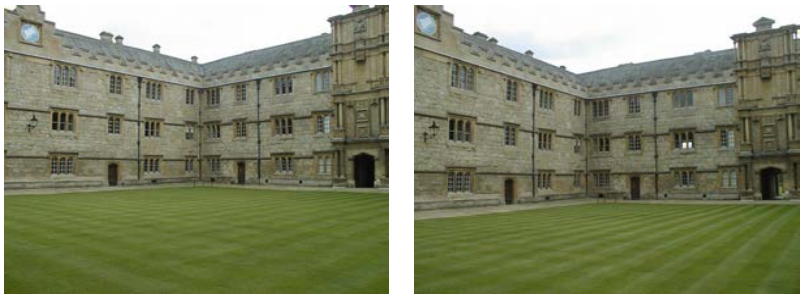


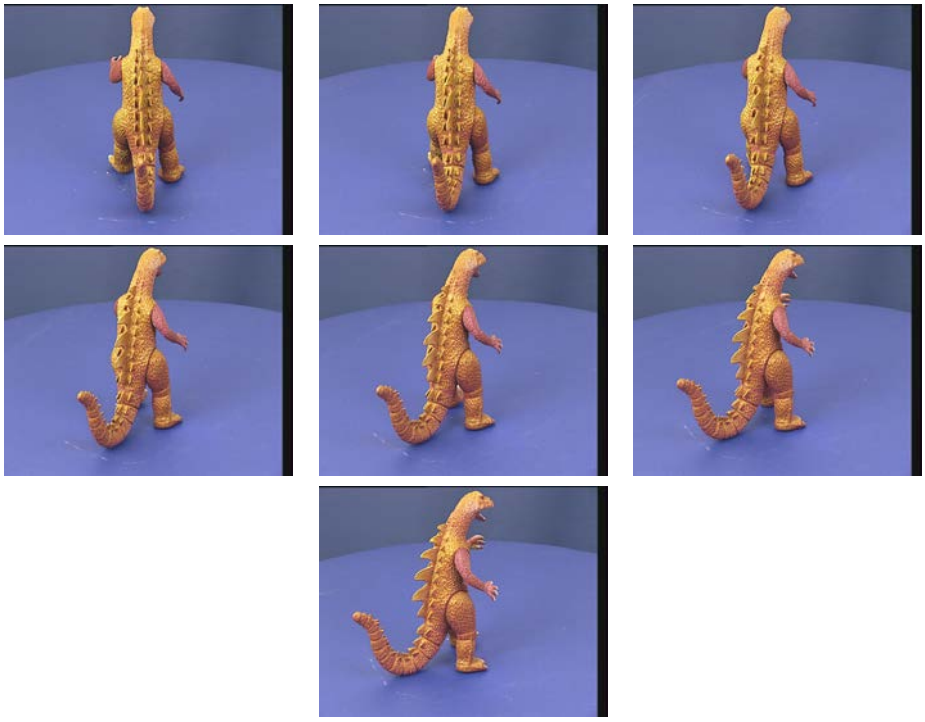
FIGURE C.3 – Paire d'images **Merton2**.

FIGURE C.4 – Paire d'images **Merton3**.

D

Images tests de la série Dinosaure

Ces images sont aussi disponibles sur le site www.robots.ox.ac.uk/~vgg/data/data-mview.html.



E

Matrice et vecteur du deuxième problème de Shekel

$$A = \begin{pmatrix}
9.681 & 0.667 & 4.783 & 9.095 & 3.517 & 9.325 & 6.544 & 0.211 & 5.122 & 2.020 \\
9.400 & 2.041 & 3.788 & 7.931 & 2.882 & 2.672 & 3.568 & 1.284 & 7.033 & 7.374 \\
8.025 & 9.152 & 5.114 & 7.621 & 4.564 & 4.711 & 2.996 & 6.126 & 0.734 & 4.982 \\
2.196 & 0.415 & 5.649 & 6.979 & 9.510 & 9.166 & 6.304 & 6.054 & 9.377 & 1.426 \\
8.074 & 8.777 & 3.467 & 1.863 & 6.708 & 6.349 & 4.534 & 0.276 & 7.633 & 1.567 \\
7.650 & 5.658 & 0.720 & 2.764 & 3.278 & 5.283 & 7.474 & 6.274 & 1.409 & 8.208 \\
1.256 & 3.605 & 8.623 & 6.905 & 4.584 & 8.133 & 6.071 & 6.888 & 4.187 & 5.448 \\
8.314 & 2.261 & 4.224 & 1.781 & 4.124 & 0.932 & 8.129 & 8.658 & 1.208 & 5.762 \\
0.226 & 8.858 & 1.420 & 0.945 & 1.622 & 4.698 & 6.228 & 9.096 & 0.972 & 7.637 \\
7.305 & 2.228 & 1.242 & 5.928 & 9.133 & 1.826 & 4.060 & 5.204 & 8.713 & 8.247 \\
0.652 & 7.027 & 0.508 & 4.876 & 8.807 & 4.632 & 5.808 & 6.937 & 3.291 & 7.016 \\
2.699 & 3.516 & 5.874 & 4.119 & 4.461 & 7.496 & 8.817 & 0.690 & 6.593 & 9.789 \\
8.327 & 3.897 & 2.017 & 9.570 & 9.825 & 1.150 & 1.395 & 3.885 & 6.354 & 0.109 \\
2.132 & 7.006 & 7.136 & 2.641 & 1.882 & 5.943 & 7.273 & 7.691 & 2.880 & 0.564 \\
4.707 & 5.579 & 4.080 & 0.581 & 9.698 & 8.542 & 8.077 & 8.515 & 9.231 & 4.670 \\
8.304 & 7.559 & 8.567 & 0.322 & 7.128 & 8.392 & 1.472 & 8.524 & 2.277 & 7.826 \\
8.632 & 4.409 & 4.832 & 5.768 & 7.050 & 6.715 & 1.711 & 4.323 & 4.405 & 4.591 \\
4.887 & 9.112 & 0.170 & 8.967 & 9.693 & 9.867 & 7.508 & 7.770 & 8.382 & 6.740 \\
2.440 & 6.686 & 4.299 & 1.007 & 7.008 & 1.427 & 9.398 & 8.480 & 9.950 & 1.675 \\
6.306 & 8.583 & 6.084 & 1.138 & 4.350 & 3.134 & 7.853 & 6.061 & 7.457 & 2.258 \\
0.652 & 2.343 & 1.370 & 0.821 & 1.310 & 1.063 & 0.689 & 8.819 & 8.833 & 9.070 \\
5.558 & 1.272 & 5.756 & 9.857 & 2.279 & 2.764 & 1.284 & 1.677 & 1.244 & 1.234 \\
3.352 & 7.549 & 9.817 & 9.437 & 8.687 & 4.167 & 2.570 & 6.540 & 0.228 & 0.027 \\
8.798 & 0.880 & 2.370 & 0.168 & 1.701 & 3.680 & 1.231 & 2.390 & 2.499 & 0.064 \\
1.460 & 8.057 & 1.336 & 7.217 & 7.914 & 3.615 & 9.981 & 9.198 & 5.292 & 1.224 \\
0.432 & 8.645 & 8.774 & 0.249 & 8.081 & 7.461 & 4.416 & 0.652 & 4.002 & 4.644 \\
0.679 & 2.800 & 5.523 & 3.049 & 2.968 & 7.225 & 6.730 & 4.199 & 9.614 & 9.229 \\
4.263 & 1.074 & 7.286 & 5.599 & 8.291 & 5.200 & 9.214 & 8.272 & 4.398 & 4.506 \\
9.496 & 4.830 & 3.150 & 8.270 & 5.079 & 1.231 & 5.731 & 9.494 & 1.883 & 9.732 \\
4.138 & 2.562 & 2.532 & 9.661 & 5.611 & 5.500 & 6.886 & 2.341 & 9.699 & 6.500
\end{pmatrix}$$

$$c = \begin{pmatrix} 0.806 & 0.517 & 0.100 & 0.908 & 0.965 & 0.669 & 0.524 & 0.902 & 0.531 & 0.876 & 0.462 \\ 0.491 & 0.463 & 0.714 & 0.352 & 0.869 & 0.813 & 0.811 & 0.828 & 0.964 & 0.789 & 0.360 \\ 0.369 & 0.992 & 0.332 & 0.817 & 0.632 & 0.883 & 0.608 & 0.326 & & & \end{pmatrix}.$$

Bibliographie

[Agarwal 2010] S. Agarwal, N. Snavely, S. Seitz et Richard S. *Bundle Adjustment in the Large.* In European Conference in Computer Vision (ECCV), 2010. (Cité en page 138.)

[Aholt 2011] C. Aholt, B. Sturmfels et R. Thomas. *A Hilbert Scheme in Computer Vision.* Rapport technique, arxiv number 1107.2875, 2011. (Cité en page 164.)

[Aholt 2012] C. Aholt et L. Oeding. *The ideal of the trifocal variety (arxiv number 1205.3776).* Rapport technique, arxiv number 1107.2875, 2012. (Cité en page 164.)

[Ali 2005] M. M. Ali, C. Khompatraporn et Z. B. Zabinsky. *A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems.* Journal of Global Optim, vol. 31, no. 4, pages 635–672, 2005. (Cité en pages 80, 81 et 82.)

[Armangué 2003] X. Armangué et J. Salvi. *Overall view regarding fundamental matrix estimation.* Image and Vision Computing, vol. 21, pages 205–220, 2003. (Cité en page 102.)

[Bartoli 2005] A. Bartoli et P. Sturm. *Structure-from-motion using lines : Representation, triangulation, and bundle adjustment.* Computer Vision and Image Understanding, vol. 100, no. 3, pages 416–441, 2005. (Cité en page 163.)

[Benson 2005] S. J. Benson et Y. Ye. *DSDP5 User Guide — Software for Semidefinite Programming.* Rapport technique, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 2005. (Cité en page 45.)

[Bersini 1996] H. Bersini, M. Dorigo, S. Langerman, G. Seront et L. Gambardella. *Results of the first international contest on evolutionary optimisation.* In Proceedings of IEEE International Conference on Evolutionary Computation, pages 611 – 615, 1996. (Cité en page 80.)

[Bocquillon 2007] B. Bocquillon, A. Bartoli, P. Gurdjos et A. Crouzil. *On Constant Focal Length Self-Calibration From Multiple Views.* In International Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, États-Unis. IEEE, juin 2007. (Cité en page 34.)

[Bonnans 2006] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal et C. Sagastizábal. Numerical optimization – theoretical and practical aspects. Springer Verlag, Berlin, Germany, 2006. (Cité en pages 14 et 15.)

[Bradski 2008] Gary Bradski et Adrian Kaehler. Learning opencv : Computer vision with the opencv library. O'Reilly, Cambridge Mass, USA, 2008. (Cité en page 149.)

[Brian 1999] B. Brian. *CSDP, a C library for semidefinite programming.* Optimization Methods and Software, vol. 11, pages 613–623, 1999. (Cité en page 45.)

[Bujnak 2008] M. Bujnak, Z. Kukelova et T. Pajdla. *A general solution to the P4P problem for camera with unknown focal length.* In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008. (Cité en page 139.)

[Bujnak 2010] M. Bujnak, Z. Kukelova et T. Pajdla. *New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion.* In Asian Conference in Computer Vision (ACCV), 2010. (Cité en page 139.)

[Byröd 2007a] M. Byröd, K. Josephson et K. Åström. *Fast Optimal Three View Triangulation*. In Asian Conference in Computer Vision (ACCV), Tokyo, Japan, Nov. 2007. (Cité en page 139.)

[Byröd 2007b] M. Byröd, K. Josephson et K. Åström. *Improving numerical accuracy of Gröbner basis polynomial equation solver*. In International Conference in Computer Vision (ICCV), 2007. (Cité en page 139.)

[Byröd 2008] M. Byröd, K. Josephson et K. Åström. *A Column-Pivoting Based Strategy for Monomial Ordering in Numerical Gröbner Basis Calculations*. In European Conference in Computer Vision (ECCV), 2008. (Cité en page 139.)

[Chandraker 2007] M. K. Chandraker, S. Agarwal, F. Kahl, D. Nistér et D. J. Kriegman. *Autocalibration via Rank-Constrained Estimation of the Absolute Quadric*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, 2007. (Cité en page 4.)

[Chen 2010] P. Chen. *Why not use the LM method for fundamental matrix estimation ?* IET computer vision, vol. 4, no. 4, pages 286–294, 2010. (Cité en page 106.)

[Chesi 2002] G. Chesi, A. Garulli, A. Vicino et R. Cipolla. *Estimating the Fundamental Matrix via Constrained Least-Squares : A Convex Approach*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pages 397–401, 2002. (Cité en page 109.)

[Chojnacki 2002] W. Chojnacki, M.J. Brooks, A. Van Den Hengel et D. Gawley. *A New Approach to Constrained Parameter Estimation Applicable to Some Computer Vision Problems*. In Statistical Methods in Video Processing Workshop held in conjunction with ECCV'02, Copenhagen, Denmark, 2002. (Cité en page 106.)

[Chojnacki 2003] W. Chojnacki, M. J. Brooks et A. Van Den Hengel. *Revisiting Hartley's Normalized Eight-Point Algorithm*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, pages 1172–1177, 2003. (Cité en page 105.)

[Conn 2000] A. R. Conn, N. I. M. Gould et P. L. Toint. Trust-region methods. Society for Industrial and Applied Mathematics, Philadelphia, USA, 2000. (Cité en pages 22 et 23.)

[Conn 2009] A.R. Conn, K. Scheinberg et L.N. Vicente. Introduction to derivative free optimization. SIAM, Philadelphia, USA, 2009. (Cité en page 26.)

[Cornille 2005] N. Cornille. *Accurate 3D Shape and Displacement Measurement using a Scanning Electron Microscope*. PhD thesis, Ecole des Mines d'Albi et University of South Carolina, Albi, France, 2005. (Cité en pages 126 et 127.)

[Cox 2007] D. A. Cox, J. Little et D. O'Shea. Ideals, varieties, and algorithms, an introduction to computational algebraic geometry and commutative algebra. Springer Verlag, Berlin, Germany, 2007. (Cité en page 33.)

[de Klerk 2006] E. de Klerk et D. Jibetean. *Global optimization of rational functions : a semidefinite programming approach*. Math. Programming, vol. 106, pages 93–109, 2006. (Cité en page 58.)

[Devernay 1997] F. Devernay. *Vision stéréoscopique et propriétés différentielles des surfaces*. PhD thesis, Ecole polytechnique, Paris, France, 1997. (Cité en pages 87 et 88.)

[Dorigo 2005] M. Dorigo et C. Blum. *Ant Colony Optimization Theory : A Survey*. Theoretical Computer Science, vol. 344, no. 2-3, 2005. (Cité en page 37.)

[Dreo 2003] J. Dreo, A. Petrowski, P. Siarry et E. Taillard. Métaheuristique pour l'optimisation difficile. Hermès, Cachan, France, 2003. (Cité en page 35.)

[Elkadi 2007] M. Elkadi et B. Mourrain. Introduction à la résolution des systèmes polynomiaux. Springer Verlag, Berlin, Germany, 2007. (Cité en page 33.)

[Fangfang 2007] L. Fangfang et R. Hartley. *A Fast Optimal Algorithm for L $_2$ Triangulation*. In Asian Conference in Computer Vision (ACCV), Tokyo, Japan, Nov. 2007. (Cité en pages 4 et 142.)

[Fischler 1981] M. A. Fischler et R. C. Bolles. *Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography*. Commun. ACM, vol. 24, pages 381–395, June 1981. (Cité en page 102.)

[Fletcher 1964] R. Fletcher et C. M. Reeves. *Function minimization by conjugate gradients*. Computer Journal, vol. 7, no. 2, pages 148–154, 1964. (Cité en page 17.)

[Fujisawa 2003] K. Fujisawa, M. Fukuda, M. Kojima, K. Nakata, M. Nakata, M. Yamashita, K. Fujisawa, M. Fukuda et K. Kobayashi. *SDPA (Semidefinite Programming Algorithm) – User's Manual*. Rapport technique, Dept. Math. and Comp. Sciences – Tokyo Institute of Technology, 2003. (Cité en page 45.)

[Glover 1986] F. Glover. *Future paths for integer programming and links to artificial intelligence*. Comput. Oper. Res., vol. 13, no. 5, pages 533–549, 1986. (Cité en page 35.)

[Gluckman 2001] J. Gluckman et S. K. Nayar. *Rectifying transformations that minimize resampling effects*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pages 111–117, 2001. (Cité en pages 87, 89 et 90.)

[Golub 1996] G. H. Golub et C. F. Van Loan. Matrix computations, 3rd edn. The John Hopkins University Press, Baltimore, USA, 1996. (Cité en page 93.)

[Hansen 1990] P. Hansen et B. Jaumard. *Algorithms for the maximum satisfiability problem*. Computing, vol. 44, no. 4, pages 279–303, 1990. (Cité en page 35.)

[Hansen 2003] E. R. Hansen et G. W. Walster. Global optimization using interval analysis, 2nd edn. Marcel Dekker, New York, USA, 2003. (Cité en page 33.)

[Hartley 1995] Richard Hartley. *In Defence of the 8-point Algorithm*. In 5^{th} International Conference on Computer Vision (ICCV'95), pages 1064–1070, Boston (MA, USA), Jun 1995. (Cité en pages 103, 105 et 107.)

[Hartley 1997] R. I. Hartley et P. Sturm. *Triangulation*. Comput. Vis. Image Underst., vol. 68, no. 2, pages 146–157, 1997. (Cité en page 139.)

[Hartley 1999] R. Hartley. *Theory and Practice of Projective Rectification*. International Journal of Computer Vision, vol. 35, pages 115–127, 1999. (Cité en pages 87, 88, 89 et 90.)

[Hartley 2003] R. Hartley et A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, Cambridge, United Kingdom, 2003. (Cité en pages 72, 85, 104, 106, 107, 109 et 129.)

[Hartley 2008] R. Hartley et Y. Seo. *Verifying global minima for L2 minimization problems*. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008. (Cité en page 143.)

[Harvent 2010] J. Harvent. *Mesure de formes par corrélation multi-images : application à l'inspection de pièces aéronautiques à l'aide d'un système multi-caméras*. PhD thesis, Institut Clément Ader et LAAS-CNRS, Albi, France, 2010. (Cité en page 131.)

[Henrion 2002] D. Henrion et J.B. Lasserre. *GloptiPoly : Global Optimization over polynomials with Matlab and SeDuMi*. Proceedings of IEEE Conference on Decision and Control, Dec. 2002. (Cité en pages 51 et 67.)

[Hestenes 1952] M.R. Hestenes et E. Stiefel. *Methods of conjugate gradients for solving linear systems*. Journal Of Research Of The National Bureau Of Standards, vol. 49, no. 6, pages 409–436, 1952. (Cité en page 16.)

[Hiriart-Urruty 1996] J.-B. Hiriart-Urruty. L'optimisation. Que sais-je. PUF, Paris, France, 1996. (Cité en page 14.)

[Hiriart-Urruty 2007] J.-B. Hiriart-Urruty. Les mathématiques du mieux faire. vol. 1 : Premiers pas en optimisation. Collection Opuscules. ELLIPSES, Paris, France, 2007. (Cité en page 14.)

[Holland 1992] J. Holland. Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Cambridge Mass.,USA, 1st MIT press ed. édition, 1992. (Cité en page 36.)

[Kahl 2007] F. Kahl et D. Henrion. *Globally optimal estimates for geometric reconstruction problems.* International Journal of Computer Vision, vol. 74, pages 3–15, 2007. (Cité en pages 4, 109 et 143.)

[Kahl 2008a] F. Kahl, S. Agarwal, M. K. Chandraker, D. Kriegman et S. Belongie. *Practical Global Optimization for Multiview Geometry.* Int. J. Comput. Vision, vol. 79, no. 3, pages 271–284, 2008. (Cité en pages 4, 142 et 143.)

[Kahl 2008b] F. Kahl et R. Hartley. *Multiple-view geometry under the Linfinity-norm.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 9, pages 1603–1617, 2008. (Cité en pages 4, 140, 142 et 145.)

[Kanatani 2000] K. Kanatani. *Optimal Fundamental Matrix Computation : Algorithm and Reliability Analysis.* In Proc. 6th Symp. Sensing via Image Inf, pages 291–298, 2000. (Cité en page 105.)

[Ke 2007] Q. Ke et T. Kanade. *Quasiconvex optimization for robust ge- ometric reconstruction.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 10, page 1834–1847, 2007. (Cité en page 140.)

[Kearfott 2009] R. B. Kearfott. *GlobSol user guide.* Optimization Methods Software, vol. 24, no. 4-5, pages 687–708, 2009. (Cité en page 34.)

[Kim 2009] S. Kim, M. Kojima et H. Waki. *Exploiting Sparsity in SDP Relaxation for Sensor Network Localization.* SIAM Journal on Optimization, vol. 20, no. 1, pages 192–215, 2009. (Cité en page 56.)

[Kirkpatrick 1983] S. Kirkpatrick et M. P. Gelatt Jr. C. D. nd Vecchi. *Optimization by Simulated Annealing.* Science Magazine, vol. 220, no. 4598, pages 671–680, 1983. (Cité en page 35.)

[Kojima 2010] M Kojima. *Exploiting Structured Sparsity in Large Scale Semidefinite Programming Problems.* In ICMS, pages 4–9, 2010. (Cité en page 56.)

[Kushal 2012] A. Kushal et S. Agarwal. *Visibility Based Preconditioning For Bundle Adjustment.* In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012. (Cité en page 138.)

[Land 1960] A. H. Land et A. G. Doig. *An Automatic Method of Solving Discrete Programming Problems.* Econometrica, vol. 28, pages 497–520, 1960. (Cité en page 34.)

[Las 2008] *Multistart algorithms for seeking feasibility.* Computers and Operations Research archive, vol. 35, pages 1379–1393, 2008. (Cité en page 35.)

[Lasserre 2001] J.B. Lasserre. *Global Optimization with polynomials and the problem of moments.* SIAM Journal on Optimization, vol. 11, pages 796–817, 2001. (Cité en page 51.)

[Lasserre 2010] J. B. Lasserre. Moments, positive polynomials and their applications, volume 1 of *Imperial College Press Optimization Series.* Imperial College Press, London, United Kingdom, 2010. (Cité en pages 51, 55, 56, 58, 63, 64, 65 et 66.)

[Laurent 2009] M. Laurent. *Sums of squares, moment matrices and optimization over polynomials. In Emerging Applications of Algebraic Geometry.* IMA Volumes in Mathematics and its Applications, vol. 149, pages 157–270, 2009. (Cité en pages 45, 49, 50 et 54.)

[Longuet-Higgins 1981] H. C. Longuet-Higgins. *A computer algorithm for reconstructing a scene from two projections.* Nature, vol. 293, pages 133–135, Sep 1981. (Cité en page 105.)

[Loop 1999] C. Loop et Z. Zhang. *Computing Rectifying Homographies for Stereo Vision.* Rapport technique, Microsoft Research, 1999. (Cité en pages 87 et 95.)

[Lourakis 2004] M.L.A. Lourakis et A. A. Argyros. *The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm.* In Technical Report, page updated August 2009, 2004. (Cité en page 138.)

[Lourakis 2005] M.L.A. Lourakis et A.A. Argyros. *Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment ?* In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 2, pages 1526 –1531 Vol. 2, oct. 2005. (Cité en page 138.)

[Lourakis 2010] M.I.A. Lourakis. *Sparse Non-linear Least Squares Optimization for Geometric Vision.* In European Conference on Computer Vision (ECCV), volume 2, pages 43–56, 2010. (Cité en page 138.)

[Luenberger 2008] D. Luenberger. Linear and nonlinear programming. Springer Verlag, Berlin, Germany, third édition, 2008. (Cité en page 15.)

[Luong 1996a] Q.-T. Luong et O. Faugeras. *The fundamental matrix : Theory, algorithms, and stability analysis.* International Journal of Computer Vision, vol. 17, no. 1, pages 43–76, 1996. (Cité en pages 105 et 106.)

[Luong 1996b] Q.-T. Luong et T. Viéville. *Canonical representations for the geometries of multiple projective views.* Comput. Vis. Image Underst., vol. 64, pages 193–229, September 1996. (Cité en page 110.)

[Löfberg 2004] J. Löfberg. *YALMIP : A Toolbox for Modeling and Optimization in MATLAB.* In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004. (Cité en page 45.)

[Mallon 2005] J. Mallon et P. Whelan. *Projective Rectification from the fundamental matrix.* Image and Vision Computing, vol. 23, pages 643–650, 2005. (Cité en pages 87, 89, 90, 91, 95, 96 et 99.)

[McKinnon 1999] K.I.M. McKinnon. *Convergence Of The Nelder-Mead Simplex Method To A Nonstationary Point.* SIAM Journal on Optimization, vol. 9, pages 148–158, 1999. (Cité en page 26.)

[Michalewicz 2004] Z. Michalewicz et D ; B. Fogel. How to solve it : Modern heuristics. Springer Verlag, Berlin, Germany, 2nd édition, 2004. (Cité en page 34.)

[Mituhiro] F. Mituhiro, K Mituhiro, K Masakazu, M Kazuo et N Kazuhide. *Exploiting Sparsity in Semidefinite Programming via Matrix Completion I : General Framework.* SIAM Journal on Optimization, vol. 11, pages 647–674. (Cité en page 56.)

[Mongeau 2007] M. Mongeau. *Introduction à l'optimisation globale.* vol. 19, pages 9–12, 2007. (Cité en pages 1, 3 et 33.)

[Moore 2009] R. E. Moore, R. B. Kearfott et M. J. Cloud. Introduction to interval analysis. SIAM, Philadelphia, USA, 2009. (Cité en page 33.)

[Nelder 1965] J. A. Nelder et R. Mead. *A Simplex Method for Function Minimization.* The Computer Journal, vol. 7, no. 4, pages 308–313, jan 1965. (Cité en page 26.)

[Nister 2007] D. Nister, R. Hartley et H. Stewenius. *Using Galois Theory to Prove Structure from Motion Algorithms are Optimal*. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2007. (Cité en page 139.)

[Nocedal 2006] J. Nocedal et S.J Wright. Numerical optimization. Springer Verlag, Berlin, Germany, 2006. (Cité en page 15.)

[Orteu 2011] J.-J. Orteu, F. Bugarin, J. Harvent, L. Robert et V. Velay. *Multiple-Camera Instrumentation of a Single Point Incremental Forming Process Pilot for Shape and 3D Displacement Measurements : Methodology and Results*. Experimental Mechanics, vol. 51, pages 625–639, 2011. 10.1007/s11340-010-9436-1. (Cité en page 132.)

[Pearl 1984] J. Pearl. Heuristics : Intelligent search strategies for computer problem solving. Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1984. (Cité en page 34.)

[Polak 1969] E. Polak et G. Ribiere. *Note sur la convergence de méthodes de directions conjuguées*. Revue Française d'Informatique et de Recherche Opérationnelle, vol. 16, pages 35–43, 1969. (Cité en page 17.)

[Rayward-Smith 1996] I. H. Rayward-Smith V. J.and Osman, C. R. Reeves et G. D. Smith. Modern heuristic search methods. John Wiley & Sons Ltd., New York, USA, 1996. (Cité en page 34.)

[Renders 1995] J-M. Renders. Algorithmes génétiques et réseaux de neurones. Hermès, Cachan, France, 1995. (Cité en pages 36 et 37.)

[Rinnooy Kan 1987] A. H. G Rinnooy Kan et G. T. Timmer. *Stochastic global optimization methods. part 1 : clustering methods*. Math. Program., vol. 39, no. 1, pages 27–56, 1987. (Cité en page 35.)

[Sahinidis 2010] N. V. Sahinidis et M. Tawarmalani. *BARON 9.0.4 : Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual. Rapport technique, 2010. (Cité en pages 34 et 67.)

[Salvi 1999] J. Salvi. *An Approach to Coded Structured Light to Obtain Three Dimensional Information*. PhD thesis, University of Girona, Girona, Spain, 1999. (Cité en page 106.)

[Schaible 2003] S. Schaible et J. Shi. *Fractional programming : the sum-of-ratios case*. Optimization Methods and Software, vol. 18, no. 2, pages 219–229, 2003. (Cité en page 58.)

[Sommese 2005] A. J. Sommese et C. W. Wampler II. The numerical solution of systems of polynomials arising in engineering and science. World Scientific, Singapore, Republic of Singapore, 2005. (Cité en page 33.)

[Stewénius 2005] H. Stewénius, F. Schaffalitzky et D. Nistér. *How hard is three-view triangulation really ?* In International Conference in Computer Vision (ICCV), pages 686–693, 2005. (Cité en page 139.)

[Sturm 1999] J.F. Sturm. *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*. Optimization Methods and Software, vol. 11–12, pages 625–653, 1999. Version 1.05 available from `http://fewcal.kub.nl/sturm`. (Cité en page 45.)

[Sutton 2009] M. Sutton, J.-J. Orteu et H. Schreier. Image correlation for shape, motion and deformation measurements : Basic concepts, theory and applications. Springer Verlag, Berlin, Germany, 2009. (Cité en page 131.)

[Tawarmalani 2005] M. Tawarmalani et N. V. Sahinidis. *A polyhedral branch-and-cut approach to global optimization*. Mathematical Programming, vol. 103, pages 225–249, 2005. (Cité en page 67.)

[Toh 2003] K.C. Toh, M.J. Todd et R.H. Tutuncu. *Solving semidefinite-quadratic-linear programs using SDPT3*. Mathematical Programming, vol. 95, pages 189–217, 2003. (Cité en page 45.)

[Torr 2000] P. H. S. Torr et A. Zisserman. *MLESAC : A New Robust Estimator with Application to Estimating Image Geometry*. Computer Vision and Image Understanding, vol. 78, page 2000, 2000. (Cité en page 102.)

[Torr 2002] P. H. S. Torr et A. W. Fitzgibbon. *Invariant fitting of two view geometry or "In defiance of the 8 point algorithm"*. 2002. (Cité en page 105.)

[Triggs 2000] B. Triggs, P. Mclauchlan, R. Hartley et A. Fitzgibbon. *Bundle Adjustment - A Modern Synthesis*. In Proceedings of the International Workshop on Vision Algorithms : Theory and Practice, ICCV '99, pages 298–372, London, UK, 2000. Springer-Verlag. (Cité en page 138.)

[Tsai 1984] R. Y. Tsai et T. S. Huang. *Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces*. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 6, pages 13–26, 1984. (Cité en pages 103 et 105.)

[Van Den Hengel 2002] A. Van Den Hengel, W. Chojnacki, M. J. Brooks et D. Gawley. *A New Constrained Parameter Estimator : Experiments In Fundamental Matrix Computation*. In In Proceedings of the 13th British Machine Vision Conference, September 2002. (Cité en page 106.)

[Vandenberghe 1996] L. Vandenberghe et S. Boyd. *Semidefinite Programming*. SIAM review, vol. 38, pages 49–95, 1996. (Cité en pages 45, 51 et 140.)

[Vaz 2007] A. I. Vaz et L. N. Vicente. *A particle swarm pattern search method for bound constrained global optimization*. J. of Global Optimization, vol. 39, no. 2, pages 197–219, 2007. (Cité en page 37.)

[Verschelde 2008] J. Verschelde et Yun Guan. *PHClab : A MATLAB/Octave interface to PHCpack*. In IMA : Software for Algebraic Geometry, vol. 148, pages Pages 15–32, 2008. (Cité en page 33.)

[Weise 2007] T. Weise. Global optimization algorithms - theory and application (e-book). Thomas Weise, 2007. (Cité en page 33.)

[Wu 2007] H-H. Wu et C-C. Chen. Scene reconstruction, pose estimation and traking. I-Tech Education and Publishing, Vienna, Austria, 2007. (Cité en page 95.)

[Wu 2011] C. Wu, S. Agarwal, S. M. Seitz et B. Curless. *Multicore Bundle Adjustment*. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011. (Cité en page 138.)

[Xuelian 2010] X. Xuelian. *New fundamental matrix estimation method using global optimization*. In Proceedings of International Conference on Computer Application and System Modeling (ICCASM), Taiyuan, China, 22–24 October 2010. (Cité en pages 108 et 109.)

[Zhengyou 1998] Z. Zhengyou. *Determining the epipolar geometry and its uncertainty : A review*. International Journal of Computer Vision, vol. 27, pages 161–195, 1998. (Cité en pages 102, 103 et 110.)

Vision 3D multi-images : contribution à l'obtention de solutions globales par optimisation polynomiale et théorie des moments

Résumé : L'objectif général de cette thèse est d'appliquer une méthode d'optimisation polynomiale basée sur la théorie des moments à certains problèmes de vision artificielle. Ces problèmes sont en général non convexes et classiquement résolus à l'aide de méthodes d'optimisation locale. Ces techniques ne convergent généralement pas vers le minimum global et nécessitent de fournir une estimée initiale proche de la solution exacte. Les méthodes d'optimisation globale permettent d'éviter ces inconvénients. L'optimisation polynomiale basée sur la théorie des moments présente en outre l'avantage de prendre en compte des contraintes. Dans cette thèse nous étendrons cette méthode aux problèmes de minimisation d'une somme d'un grand nombre de fractions rationnelles. De plus, sous certaines hypothèses de « faible couplage » ou de « parcimonie » des variables du problème, nous montrerons qu'il est possible de considérer un nombre important de variables tout en conservant des temps de calcul raisonnables. Enfin nous appliquerons les méthodes proposées aux problèmes de vision par ordinateur suivants : minimisation des distorsions projectives induites par le processus de rectification d'images, estimation de la matrice fondamentale, reconstruction 3D multi-vues avec et sans distorsions radiales.

Mots clés : Optimisation Globale, Optimisation polynomiale, Théorie des moments, Reconstruction 3D

Contribution to the global resolution of minimization problems in computer vision by polynomial optimization and moments theory

Abstract : The overall objective of this thesis is to apply a polynomial optimization method, based on moments theory, on some vision problems. These problems are often nonconvex and they are classically solved using local optimization methods. Without additional hypothesis, these techniques don't converge to the global minimum and need to provide an initial estimate close to the exact solution. Global optimization methods overcome this drawback. Moreover, the polynomial optimization based on moments theory could take into account particular constraints. In this thesis, we extend this method to the problems of minimizing a sum of many rational functions. In addition, under particular assumptions of « sparsity », we show that it is possible to deal with a large number of variables while maintaining reasonable computation times. Finally, we apply these methods to particular computer vision problems : minimization of projective distortions due to image rectification process, Fundamental matrix estimation, and multi-view 3D reconstruction with and without radial distortions.

Keywords : Global Optimization, Polynomial Optimization, Moments Theory, 3D Reconstruction