Robust Feature Correspondence and Pattern Detection for Façade Analysis

David Ok

École des Ponts
ParisTech

UNIVERSITÉ
PARIS-EST

École Doctorale Paris-Est
Mathématiques & Sciences et Technologies
de l'Information et de la Communication

THÈSE DE DOCTORAT
DE L'ÉCOLE DES PONTS PARISTECH
Domaine : Traitement du Signal et des Images

présentée par **David OK**
pour obtenir le grade de
DOCTEUR DE L'ÉCOLE DES PONTS PARISTECH

Mise en Correspondance Robuste et Détection d'Éléments Visuels Appliquées à l'Analyse de Façades

Soutenue publiquement le 25 mars 2013 devant le jury composé de :

Jean-Yves AUDIBERT	Capital Fund Management	Examinateur
Thomas CHAPERON	Trimble Navigation	Examinateur
Vincent LEPETIT	École Polytechnique Fédérale de Lausanne	Rapporteur
Renaud MARLET	École des Ponts ParisTech	Directeur de Thèse
Dimitris SAMARAS	Stony Brook University	Rapporteur
Peter STURM	INRIA Grenoble, Rhône-Alpes	Président
Olivier TOURNAIRE	Centre Scientifique & Technique du Bâtiment	Examinateur

École des Ponts ParisTech
CERTIS/IMAGINE
6, Av Blaise Pascal - Cité Descartes
Champs-sur-Marne
77455 Marne-la-Vallée cedex 2
France

Université Paris-Est Marne-la-Vallée
École Doctorale Paris-Est MSTIC
Département Études Doctorales
6, Av Blaise Pascal - Cité Descartes
Champs-sur-Marne
77454 Marne-la-Vallée cedex 2
France

École des Ponts
ParisTech

UNIVERSITÉ
PARIS-EST

Robust Feature Correspondence and Pattern Detection for Façade Analysis

By
David Ok

Submitted to the
École Doctorale Paris-Est
Mathématiques & Sciences et Technologies
de l'Information et de la Communication

In partial fulfillment of the requirement for the degree of
Doctor of Philosophy in Computer Science

At the
École des Ponts ParisTech

Defended on March 25th, 2013 in front of the committee composed of:

Jean-Yves Audibert	Capital Fund Management	Examiner
Thomas Chaperon	Trimble Navigation	Examiner
Vincent Lepetit	École Polytechnique Fédérale de Lausanne	Reviewer
Renaud Marlet	École des Ponts ParisTech	Thesis Supervisor
Dimitris Samaras	Stony Brook University	Reviewer
Peter Sturm	INRIA Grenoble, Rhône-Alpes	Chair
Olivier Tournaire	Centre Scientifique & Technique du Bâtiment	Examiner

Mes encadrants. Tout d'abord, je tiens à remercier vivement mes encadrants Jean-Yves Audibert, Renaud Marlet et Olivier Tournaire pour tous leurs précieux conseils et leur super soutien pendant la thèse, surtout dans les moments difficiles. Un grand grand merci pour toute l'énergie et tout le temps que vous investis pour moi, je ne l'oublierai pas. Encore désolé pour toutes ces malheureuses nuits blanches que je vous ai fait subir à l'approche des deadlines de conférences! Je remercie aussi Renaud Keriven de m'avoir accueilli au sein du laboratoire IMAGINE et de m'avoir accordé sa confiance pour ce travail de recherche.

Thesis committee. Second, I would like to thank reviewers Dimitris Samaras and Vincent Lepetit, chairman Peter Sturm, and examinator Thomas Chaperon for having taken the time to read and evaluate my work, for their relevant comments and questions and for the fruitful remarks they provided in their reports and during the defense. I was very honored of your presence and I thank you again for coming from far away for the defense.

PhD candidates and friends in the lab. Special thanks to my collaborator Mateusz Koziński: thank you for helping me to make it for Zürich. Merci à mes compagnons de fortune, Ferran Espuny et Pierre Moulon, pendant le séjour en Corée. C'était génial avec vous ! Le golf et les restos, on remet ça quand ? Je remercie également les membres du laboratoire qui ont animé le labo dans la bonne humeur. Je pense à d'anciens compagnons de galère dont, en vrac, Anne-Laure Chauve, Jérôme Courchay, Anne-Marie Tousch, Hiep Vu, Cédric Allène, Antoine Salomon, Nicolas Thorstensen, Jamil Drareni, Laetitia Comminges. Je pense également aux autres compagnons actuellement en galère, dont je souhaite bon courage, comme Olivier Collier, Pierre Moulon, Victoria Rudakova, Gadde Raghudeep, Francisco Suzano Massa, Alexandre Boulc'h, Mateusz Kozinski, Zhe Liu, Amine Bourki, Loïc Landrieu. You'll gonna make it guys! It was a real pleasure being with you and talking with you about random things. J'en oublie certainement d'autres, j'espère être pardonné.

Permanent members of the lab. Je remercie également les membres permanents pour tous leurs conseils et suggestions qu'ils m'ont prodigués. Je remercie Nikos Paragios de m'avoir aiguillé sur le parsing amélioré de façades et toutes ses suggestions ou conseils. Merci à Arnak Dalalyan pour les suggestions proposées pour essayer de booster la fonction de mérite pour le parsing de façades. Merci à Pascal Monasse pour toutes les discussions et les suggestions que ce soit pour des questions d'ordre technique ou pour l'enseignement des élèves 1A aux Ponts. Merci à Jean-Philippe Pons pour son expertise sur CGAL et en géométrie algorithmique. Merci à Guillaume Obozinski pour son expertise en optimisation convexe et sur la sparsité et pour ses discussions à propos des choix de carrière. Merci à Bertrand Neveu, Nikos Komodakis, Martin de la Gorce pour leurs remarques prodiguées pour la soutenance. Enfin je remercie aussi Brigitte Mondou pour ton super travail qui nous facilite le quotidien. Discuter de tout et de rien m'as aussi permis de de déconnecter un peu de la recherche.

Les amis hors du labo. Hors du domaine professionnel, je remercie à tous mes amis dont notamment Mickaël, Luis, Cécilien, Romain, Marie-Lise et Tony et d'autres que j'oublie pour m'avoir permis de m'échapper de la recherche. Les multiples activités que j'ai partagées avec vous ont été une sacrée bouffée d'oxygène. En particulier, merci à Mickaël, Luis, Cécilien de m'avoir supporté pendant la première semaine de séjour en Chine où je terminais la rédaction de la thèse.

Famille. Enfin pour clore cette page de remerciements, je remercie également tous les membres de ma famille. Maman, Papa, Denis, Virginie, sans votre soutien indéfectible, rien de tout cela n'aurait été possible. En particulier, mille mercis à Maman pour m'avoir supporté dans les moments difficiles ! Merci à tous Tata Édouard, Pou Pol, Tata Victor, Tata Stéphane, Mak Yeay pour m'avoir facilité la vie pendant la thèse et le jour de la soutenance. Merci à mes super cousins dont Victor, Éd, Alex, Lora, Kenory... pour leur bonne humeur, insouciance, et de m'avoir réconforté à plusieurs moments pendant ma thèse.

Depuis quelques années, avec l'émergence de larges bases d'images comme *Google Street View*, la capacité à traiter massivement et automatiquement des données, souvent très contaminées par les faux positifs et massivement ambiguës, devient un enjeu stratégique notamment pour la gestion de patrimoine et le diagnostic de l'état de façades de bâtiment.

Sur le plan scientifique, ce souci est propre à faire avancer l'état de l'art dans des problèmes fondamentaux de vision par ordinateur. Notamment, nous traitons dans cette thèse les problèmes suivants: la mise en correspondance robuste, algorithmiquement efficace de caractéristiques visuelles et l'analyse d'images de façades par grammaire. L'enjeu est de développer des méthodes qui doivent également être adaptées à des problèmes de grande échelle.

Tout d'abord, nous proposons une formalisation mathématique de la cohérence géométrique qui joue un rôle essentiel pour une mise en correspondance robuste de caractéristiques visuelles. À partir de cette formalisation, nous en dérivons un algorithme de mise en correspondance qui est algorithmiquement efficace, précise et robuste aux données fortement contaminées et massivement ambiguës. Expérimentalement, l'algorithme proposé se révèle bien adapté à des problèmes de mise en correspondance d'objets déformés, et à des problèmes de mise en correspondance précise à grande échelle pour la calibration de caméras.

En s'appuyant sur notre algorithme de mise en correspondance, nous en dérivons ensuite une méthode de recherche d'éléments répétés, comme les fenêtres. Celle-ci s'avère expérimentalement très efficace et robuste face à des conditions difficiles comme la grande variabilité photométrique des éléments répétés et les occlusions. De plus, elle fait également peu d'hallucinations.

Enfin, nous proposons des contributions méthodologiques qui exploitent efficacement les résultats de détections d'éléments répétés pour l'analyse de façades par grammaire, qui devient substantiellement plus précise et robuste.

Mots-clefs

vision par ordinateur; analyse d'images; façades; grammaires; segmentation sémantique; détection d'objets; mise en correspondance; caractéristiques visuelles; éléments répétés; ambiguïté massive; contamination; faux positifs; cohérence géométrique; contrainte d'ordre 4; cohérence affine locale; représentation hiérarchique.

For a few years, with the emergence of large image database such as *Google Street View*, designing efficient, scalable, robust and accurate strategies have now become a critical issue to process very large data, which are also massively contaminated by false positives and massively ambiguous. Indeed, this is of particular interest for property management and diagnosing the health of building façades.

Scientifically speaking, this issue puts into question the current state-of-the-art methods in fundamental computer vision problems. More particularly, we address the following problems: (1) robust and scalable feature correspondence and (2) façade image parsing.

First, we propose a mathematical formalization of the geometry consistency which plays a key role for a robust feature correspondence. From such a formalization, we derive a novel match propagation method. Our method is experimentally shown to be robust, efficient, scalable and accurate for highly contaminated and massively ambiguous sets of correspondences. Our experiments show that our method performs well in deformable object matching and large-scale and accurate matching problem instances arising in camera calibration.

We build a novel repetitive pattern search upon our feature correspondence method. Our pattern search method is shown to be effective for accurate window localization and robust to the potentially great appearance variability of repeated patterns and occlusions. Furthermore, our pattern search method makes very few hallucinations.

Finally, we propose methodological contributions that exploit our repeated pattern detection results, which results in a substantially more robust and more accurate façade image parsing.

Keywords

computer vision; image analysis; façades; grammars; semantic segmentation; object detection; feature correspondence; repeated visual patterns; massive ambiguity; outlier; geometry consistency; 4th order constraint; local affine consistency; hierarchical representation.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction (French)

Financé par le Centre Technique et Scientifique du Bâtiment (CSTB), cette thèse s'inscrit dans une volonté de développer des méthodologies permettant de lier automatiquement l'aspect extérieur d'un bâtiment et ses caractéristiques énergétiques. À terme, de simples prises de vues à partir de la rue permettraient de classer les bâtiments suivant divers typologies, par exemple, par âge, par type de matériaux utilisés, repérant ainsi les besoins les plus criants en réhabilitation énergétique.

Face à ces objectifs ambitieux, nous limitons toutefois cette thèse à la détection automatique d'éléments simples et répétés d'une façade de bâtiment, comme, par exemple, les fenêtres. En effet, il faut insister sur le fait que détecter des éléments de façades répétés, comme par exemple, les fenêtres, constitue déjà en lui-même un défi. Une fois mises au point, de telles méthodes de détection, à la fois efficaces et robustes, permettra ensuite d'obtenir une analyse de façades par grammaire avec une fiabilité substantiellement accrue.

D'autre part, si les méthodes robustes permettant de localiser précisément les fenêtres sont des outils précieux pour l'analyse de scènes urbaines, l'analyse d'images de façades de bâtiments par grammaire reste indispensable pour obtenir le pourcentage de surface vitrée, qui est un indicateur précieux pour les spécialistes du bâtiment. En effet, le pourcentage de surface vitrée constitue un paramètre clé pour évaluer les performances thermales d'un bâtiment.

Sur le plan socio-économique, les techniques developpées auront également des retombées concrètes puisque l'analyse automatisée de façades permettra, entre autres, de rationaliser la gestion de patrimoine en identifiant automatiquement les zones urbaines nécessitant une réhabilitation énergétique. Une telle connaissances est particulièrement stratégique pour les collectivités territoriales soucieux de l'état de leur parc.

Sur le plan scientifique, cette problématique soulève des défis propres à faire avancer l'état de l'art en vision par ordinateur. Depuis quelques années, avec l'émergence de très larges bases d'images urbaines comme *Google Street View*, la conception de méthodes à la fois précises, efficaces, robustes et algorithmiquement efficaces et adaptées pour des problèmes de grande dimension revêt une importance cruciale pour traiter massivement

les données.

En particulier, une grande partie de ces données peuvent être traitée par mise en correspondence d'éléments répétées au moyen de mise en correspondance de caractéristiques visuelles et la segmentation de façades. Cependant, la mise en correspondance d'éléments répétées devient difficile à cause de la contamination massive par des correspondances aberrantes et de l'ambiguïté massive, en grande partie due à la répétition d'éléments similaires. Concernant la tâche de segmentation de façades, le défi principal reste de minimiser les étiquetages de pixels à cause des indices visuelles souvent peu discriminants. Le but de cette thèse est de mettre au point une méthode pour la mise en correspondance robuste avec des ensembles de correspondances particulièrement contaminée et ambigus. On pourra ensuite en dériver une méthode robustes pour la détection d'objets répétés, comme les fenêtres. En conséquence, les détections de fenêtres, fiables, fournies par le détecteur permettra d'améliorer substantiellement la segmentation et l'analyse de façades par grammaire, et ainsi de calculer le pourcentage de surface vitrée sur une façade de bâtiment.

Contents

1.1 Le Problème de Mise en Correspondance de Caractéristiques Visuelles

Le problème de mise en correspondance entre des ensembles de caractéristiques visuelles est omniprésent en vision par ordinateur, comme en témoigne son historique particulièrement riche dans la littérature. Il apparaît dans de nombreuses applications comme

- le suivi de caractéristiques visuelles dans les vidéos, étudié par exemple dans les travaux de SHI and TOMASI (1994) et de BIRCHFIELD (2007),
- l'assemblage cohérent et "sans heurts" de photographies pour la construction de panorama, comme présenté par exemple dans les travaux de BROWN and LOWE (2007),

- la stéréovision multi-vues, en particulier la calibration de caméras à partir de photographies touristiques pour la reconstruction 3D numérique de lieux touristiques très fréquentés (voir notamment les travaux de SNAVELY et al. (2008)) et la fusion de données 3D (voir par exemple les travaux de PRITCHETT et al. (1998)),

- la détection et reconnaissance d'object (voir par exemple les travaux de BERG et al. (2005)),

- la classification d'images (voir par exemple les travaux de LAZEBNIK et al. (2006)) ou la recherche dans des bases d'images (voir par exemple les travaux de SCHMID and MOHR (1997)).

Dans les situations où les ensembles de correspondances sont très ambigus, par exemple, lorsque des objets similaires apparaissent de nombreuses fois, comme les fenêtres sur une façade, ou quand les scènes sont peu texturées, ce qui est courant au niveau des façades, une mise en correspondance robuste de caractéristiques visuelles passe nécessairement, par exemple, par l'utilisation de la cohérence géométrique des positions des caractéristiques.

Notons que le problème de mise en correspondances de caractéristiques visuelles est assez large et la pertinence d'une méthode de mise en correspondance dépend non seulement de la tâche considérée parmi celles énumérées plus haut. De plus, la pertinence d'une méthode particulière peut également dépendre du type de caractéristique visuelle considérée, à savoir, entre autres:

- les points d'intérêts comme les coins, qui peuvent être détectés par exemple par la méthode de HARRIS and STEPHENS (1988);

- les morceaux de lignes, dont GROMPONE VON GIOI et al. (2010) en présentent un détecteur performant;

- les bords, détectés par exemple les méthodes de MARR and HILDRETH (1980) ou de CANNY (1986), ou encore les contours d'objets, utilisés par exemple dans les travaux de GRAUMAN and DARRELL (2004);

- les points échantillonnés sur les contours d'objets, utilisés par exemple dans les travaux de BELONGIE et al. (2002) et de BERG et al. (2005);

- les régions localement invariantes (*blob*), en particulier les extréma locaux de différence de Gaussiennes (*DoG*) (WEICKERT et al. 1999; LINDEBERG 1991; LOWE 2004), les régions covariantes affines (MIKOLAJCZYK et al. 2005) et les régions extrémales maximisant un critère de stabilité (*MSER*) (MATAS et al. 2002).

Dans cette thèse, nous portons plus particulièrement notre attention sur la mise en correspondance de régions localement invariantes.

1.1.1 Les Approches par Comparaison Photométrique

L'explosion combinatoire est le défi qui apparaît naturellement dans les problèmes de mise en correspondance. De fait, concevoir des méthodes applicables à des problèmes de mise en correspondance à grande échelle est d'une importance cruciale. D'une part, le nombre de correspondances possibles est quadratique en le nombre de caractéristiques. D'autre part, exiger, en outre, la cohérence géométrique dans la mise en correspondance multiplie le nombre de combinaisons possibles qui dépend du nombre de régions cohérentes et est principalement responsable de l'explosion combinatoire.

Une première approche pour maîtriser cette combinatoire est de concevoir un descripteur pour un type de caractéristique donné et une mesure de similarité pour comparer les descripteurs associés à chaque caractéristique visuelle. Une telle stratégie permet alors d'éliminer un certain nombre de faux positifs, autrement dit des correspondances aberrantes (*outliers*, en anglais). Il en résulte alors un ensemble de correspondances, non seulement significativement réduit, mais aussi présentant une contamination moindre par les correspondances aberrantes. Ceci permet ensuite d'obtenir un problème de correspondance à la fois soluble informatiquement et plus facile à résoudre.

En particulier, les régions localement invariantes peuvent être décrites de facon robutes, par exemple, avec SIFT (LOWE 2004). D'autres descripteurs concurrents, plus efficaces à calculer, sont également proposés comme SURF (BAY et al. 2008), DAISY (TOLA et al. 2010). Tous les descripteurs mentionnnés possèdent une remarquable robustesse par rapport aux changements de point de vue, contraste, rotation, changements d'échelle et à diverses conditions de flou et compression d'images. En quelques années, les régions localement invariantes sont devenues omniprésentes dans les approches modernes de vision par ordinateur. En particulier, ils constituent un ingrédient bas-niveau fondamental dans toutes les applications de vision mentionnées précédemment.

Notons que les autres types de caractéristiques comme les morceaux de lignes, bords ou contours, mentionnés précédemment, sont plus difficiles à décrire de façon robuste et de fait, semble moins utilisés. D'une part, ils ne possèdent *a priori* pas de propriétés d'invariance locale évidentes comme l'invariance au changement de point de vue. D'autre part, la conception de descripteurs contextuels ou photométriques spécifiques semble moins explorée ou peu évidente bien qu'il en existe déjà (voir, par exemple, les travaux de BELONGIE et al. (2002) et de WANG et al. (2009)). Toutefois, l'intérêt de ces caractéristiques ne doit pas être minimisé car elles sont en général plus parcimonieuses et plus pertinentes. En effet, la calibration de caméra devient particulièrement difficile pour des scènes urbaines, qui contiennent, entre autres, des fenêtres : ces dernières ont une très grande variabilité photométrique à cause leur surface spéculaire.

1.1.2 Nécessité de la Cohérence Géométrique

En général pour des situations simples, comme des situations où les scènes présentent une richesse texturelle et peu de répétitions de motifs visuels, les comparaisons photométriques permettent alors de produire un ensemble de correspondances très peu contaminées par des correspondances aberrantes.

De plus, les approches de type "sac-de-caractéristiques" (*bag-of-features* en anglais), peuvent être employées pour améliorer la robustesse des descripteurs par rapport au variation intra-classe, robustesse nécessaire notamment pour la catégorisation d'images et pour la recherche dans les bases d'images (voir par exemple les travaux de LAZEBNIK et al. (2006)).

Or, les situations simples sont souvent éloignées de la réalité et les comparaisons photométriques ne suffisent plus à produire des ensembles de correspondances suffisamment propres.

Dans les situations où changements de point de vue ou de contraste entre les images sont forts, les comparaisons entre descripteur n'est plus en mesure de garantir une certaine robustesse.

D'autre part, l'ambiguïté dans la mise en correspondance survient lorsque les images présente beaucoup de répétitions de texture ou de motifs visuels, auquel cas la comparaison photométrique seule ne peut pas résoudre l'ambiguïté. Ainsi, il peut en résulter de nombreuses correspondances qui paraissent correcte du point de vue *local et purement photométrique*, mais en réalité fausse en terme de cohérence globale.

Garantir la cohérence géométrique des correspondances semble être le seul moyen d'éliminer efficacement les trop nombreuses correspondances aberrantes. Ainsi, pour améliorer la robustesse de la mise en correspondance, la cohérence géométrique peut être garantie à trois niveaux d'échelle croissant, à savoir

- à l'échelle locale d'une caractéristique visuelle donnée,
- à l'échelle du voisinage contenant des caractéristiques visuelles proches de la caractéristique visuelle donnée,
- à l'échelle d'une plus large région de caractéristiques visuelles.

Afin de préserver la fluidité de ce chapitre, nous reverrons dans un chapitre ultérieur (Chapitre 3) l'état de l'art sur la mise en correpondance robuste de caractéristiques visuelles préservant la cohérence géométrique.

1.2 La Détection d'Objets Répétés pour l'Analyse des Images par Grammaire

La segmentation d'images et l'analyse d'images par grammaire sont des problèmes clés pour la compréhension des image.

La segmentation d'images cherche à retrouver des segments d'images par groupement de pixels d'images. L'intérêt de la segmentation est qu'elle simplifie la représentation d'une image, facilitant ainsi l'interprétation sémantique de l'image.

L'analyse d'images par grammaire fait plus que de la segmentation. Elle cherche à fournir une représentation hiérarchique d'une image étant donné une grammaire. La littérature sur l'analyse d'images par grammaire est particulièrement vaste et nous nous contentons de pointer le lecteur vers l'étude de ZHU and MUMFORD (2006). Ici, nous nous

intéressons en particulier à l'analyse de façades. En particulier, nous cherchons à retrouver et à paramétriser la structure hiérarchique d'un bâtiment, par exemple, dénombrer le nombre d'étages, déterminer la taille et position de chaque étage, localiser les fenêtres dans chaque étage et ainsi de suite. Notons que les travaux de TEBOUL et al. (2010); TEBOUL et al. (2011) démontrent la robustesse de l'analyse de façades par grammaire pour la tâche de segmentation de façades.

1.2.1 Étiquetage Erronné Dues aux Informations Bas-Niveau en Segmentation

En segmentation, l'étiquetage erronné des pixels provient surtout du fait que l'information *a priori* élémentaires, de bas-niveau, finissent par avoir des limites, comme les contours ou la couleur. Dans les cas simples, de telles informations suffisent à segmenter les images de manière satisfaisante. Or, en général, plusieurs problèmes apparaissent au sein d'une image. En particulier, les variations d'illuminations et la présence d'occlusions partielles ou complètes apparaissent fréquemment au sein d'une image de façade (TEBOUL 2010; TEBOUL et al. 2010; TEBOUL et al. 2011). De plus, les lignes peuvent devenir difficilement détectables car leur saillance peut être amoindrie au niveau des zones très illuminées de la façades (TEBOUL 2010; TEBOUL et al. 2010; TEBOUL et al. 2011).

Quand les variations d'illumination apparaissent au sein d'une image, les approches par comparaison photométrique simples comme la corrélation croisée normalisée, ne permettent plus d'étiqueter correctement certaines parties de l'image. Par exemple, dans les images de façade, les façades sont souvent plus illuminées au niveau du dernier étage qu'au niveau du rez-de-chaussée (TEBOUL 2010; TEBOUL et al. 2010; TEBOUL et al. 2011). En particulier, les pixels correspondant au mur, dans la partie supérieure de l'image, peuvent devenir aussi brillant que le ciel. Ainsi, les méthodes naïves de segmentation peuvent étiqueter à tort les pixels 'mur' comme étant des pixels de 'ciel'.

Des informations *a priori* plus robustes peuvent être mises au point pour mieux guider la segmentation comme par exemple:

- la mise au point de descripteurs photométriques plus puissants (MALIK et al. 1999; BREIMAN and SCHAPIRE 2001; LOWE 2004; TOLA et al. 2010; SHOTTON et al. 2008)
- l'apprentissage d'un nouveau descripteur hybride à partir d'une combinaison de plusieurs descripteurs différents (CHENG et al. 2011)
- l'utilisation jointe de plusieurs informations géométriques simples comme les contours (CANNY 1986), les lignes (GROMPONE VON GIOI et al. 2010), les cartes de gradient (KASS et al. 1988; COOTES et al. 1995).

1.2.2 Informations Haut-Niveau Fournies par un Détecteur d'Élements Répétitifs Robuste

L'analyse de façade par grammaire peut être plus précise et robuste si l'on utilise des informations *a priori* de haut-niveau. En particulier, on peut penser à la détection de

fenêtres, ce qui aide grandement à retrouver la composition d'un bâtiment. Les détections de fenêtres peuvent être fournies par des méthodes de reconnaissance d'objet comme VIOLA and JONES (2004); BLASCHKO and LAMPERT (2008).

Cependant, les méthodes de reconnaissance d'objets (VIOLA and JONES 2004; BLASCHKO and LAMPERT 2008) sont d'une certaine manière générique. Ils n'exploitent pas le fait que les éléments peuvent se répéter au sein d'une image, comme les fenêtres sur la façade d'un bâtiment. Dans ce cas, il arrive qu'ils font souvent des détections erronnées, c'est-à-dire, les éléments recherchées ne sont pas détectés ou sont hallucinées ou ne sont pas localisées correctement. Comme ces méthodes ont souvent un paramètre réglable qui favorise soit la précision ou le rappel, il est ici préférable de régler les détecteurs de telle façon que les fenêtres détectées soient précisément localisées sans se soucier de savoir si des fenêtres ont été manquées. Elles pourront être retrouvées en détectant des caractéristiques visuelles et en mettant en correspondance les caractéristiques visuelles sur les fenêtres avec les autres caractéristiques visuelles sur le reste de l'image. Les fenêtres alors retrouvées constituent alors des informations très précieuses et robustes pour l'analyse de façades par grammaire.

D'autres caractéristiques visuelles peuvent être utilisées pour améliorer la précision de l'analyse de façades. En particulier, les lignes sont particulièrement utiles pour l'analyse d'environnement urbain. L'utilisation jointe de ces informations permet alors d'obtenir une analyse précise de façades.

Notons qu'il est possible également de construire des informations haut-niveau par aggrégation successive comme dans les travaux de SHI and MALIK (2000) et DELONG et al. (2012).

1.3 Contributions de la Thèse

Dans un premier temps, nous revoyons les méthodes existantes pour la mise en correspondeance de caractéristiques visuelles, tout en mentionnant les forces et faiblesses respectives de chaque méthode dans le Chapitre 3. Puis, cette thèse développe principalement quatre contributions apportées dans les domaines suivants: la mise en correspondance de caractéristiques visuelles et l'analyse grammaticale descendante des façades de bâtiments. Nous exposons ces contributions, selon l'organisation de ce manuscrit.

- Le Chapitre 4 revisite minutieusement la répétabilité et la précision moyenne des détecteurs de caractéristiques visuelles. Nous proposons une formalisation mathématique de ces notions. Nous formalisons complètement la notion de cohérence géométrique à partir d'une contrainte locale d'ordre 4 entre des correspondances voisines. La cohérence géométrique est garantie à trois niveaux d'échelle comme mentionné dans le dernier paragraphe de la Sous-Section 2.1.2. En outre, notre formalisation de cohérence géométrique permet de traiter la mise en correspondance d'objets modérément déformés.

- S'appuyant sur la cohérence géométrique exposée dans le Chapitre 4, le Chapitre 5 propose un algorithme efficace de propagation de correspondances locales. Notre

méthode de propagation diffère des autres méthodes de propagation dans la mesure où elle se base sur des contraintes locales d'ordre 4 impliquant une cohérence mutuelle entre correspondances locales voisines. Dans le Chapitre 6, nous démontrons expérimentalement que l'algorithme proposé est plus efficace que les méthodes existantes, et qu'il est adapté aux problèmes de mise en correspondance de grande échelle, tout en résolvant les correspondances ambigües efficacement. Notamment, nous le verrons dans la mise en correspondance d'objet déformés et la mise en correspondance précise pour la calibration de caméras.

- Le Chapitre 7 présente une méthode de recherche d'éléments répétés qui s'appuie sur notre méthode de mise en correspondance. Cette méthode de recherche d'éléments répétés nécessite en entrée un "archétype" visuel, puis elle retrouve tous les modèles visuels similaires à cet archétype, de façon récursive, en faisant une exploration en largeur. Nous démontrons expérimentalement que notre méthode de recherche d'éléments répétés obtient des résultats supérieurs à l'état de l'art notamment pour la localisation précise de fenêtres dans les bases d'images *eTrims* (KORČ and FÖRSTNER 2009) et de l'*École Centrale Paris* (TEBOUL 2010).

- Le Chapitre 8 présente deux techniques pour l'analyse de façades. Elles exploitent efficacement nos résultats de détection de fenêtres obtenues par notre méthode de recherche d'éléments répétés. La première contribution consiste à combiner nos résultats de détection de fenêtres et les résultats d'une segmentation sémantique pixellique médiocre. Cette combinaison produit alors des informations substantiellement améliorée. Ces informations serviront alors à mieux évaluer la qualité de la structure hiérarichique proposée pour la façade. Deuxièmement, nous proposons de combiner nos résultats de détections de fenêtres et les résultats de détections de lignes pour mieux guider l'analyse grammaticale de façade. Sur le plan de l'optimisation, cette contribution améliore substantiellement la vitesse de convergence, précision de l'analyse, et elle limite les dérives entre la solution obtenue et l'optimum global.

Les deux premières contributions sont apportées dans le domaine de la mise en correspondance de caractéristiques visuelles. Elles sont à la fois théoriques et expérimentales et s'avèrent être déterminantes pour une bonne cohérence géométrique. Ainsi, une propagation de correspondances locales, basée sur des contraintes d'ordre 4 entre correspondances locales voisines, se révèlent particulièrement efficace.

La troisième contribution est apportée dans le domaine de la reconnaissance d'objet, qui constitue une étape possible pour améliorer l'analyse de façades. S'appuyant directement sur notre méthode de propagation de correspondances locales, ces contributions permettent alors de fournir des informations de plus haut-niveau, par exemple, des détections de fenêtres, très précieuses pour l'analyse de façades.

En combinant nos détections de fenêtres avec d'informations *a priori* existantes, nous améliorons l'état de l'art dans l'analyse de façades par grammaire.

1.4 Publications

Une partie de ce travail a été publié dans les conférences avec comité de lecture suivantes.

- *Efficient and Scalable 4th-order Match Propagation.*
  DAVID OK, RENAUD MARLET, JEAN-YVES AUDIBERT.
  In 11th Asian Conference on Computer Vision (ACCV 2012), Daejeon, Korea, November 2012.

- *High-Level Bottom-Up Cues for Top-Down Parsing of Facade Images.*
  DAVID OK, MATEUSZ KOZINSKI, RENAUD MARLET, NIKOS PARAGIOS.
  In 2nd Joint 3DIM/3DPVT Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT 2012), Zürich, Switzerland, October 2012.

Chapter 2

Introduction

This thesis is financially supported by the Centre Technique et Scientifique du Bâtiment (CSTB) and is part of CSTB's will to develop methodologies that establish links between a building exterior appearance and its specific energetic features. One of the CSTB's objective is to classify buildings from an analysis of simple street photographs, also taking into account specific criteria such as age or construction materials. Hence, this would enable to automatically diagnose urban zones needing energetic rehabilitation.

Because this objective is rather ambitious, the scope of the thesis is limited to the design of methods which automatically detect simple yet relevant façade elements such as windows. Indeed, robustly detecting repeated façade elements, such as windows, already constitutes a challenging problem. As illustrated in this thesis, the development of such robust methods then leads to a much more reliable parsing of façade images.

Besides, while methods for accurately localizing windows are valuable tools for urban scene analysis, façade image parsing are still needed to obtain the percentage of window area in the building façade, which is a precious data for building specialists. Indeed, the percentage of glass area is a key parameter for assessing the thermal performance of buildings.

Socio-economically speaking, these techniques would then help to rationalize the property management by automatically identifying urban zones that needs energetic rehabilitation. Such a knowledge is particularly helpful for territorial collectivities.

Scientifically speaking, the issue of energetic rehabilitation puts into question the performance of existing computer vision methods. Indeed, for a few years, with the emergence of large image database such as *Google Street View*, designing efficient, scalable, robust and accurate methods have now become of crucial importance to process very large urban-related data. In particular, a large part of urban-related data can be processed through (1) repeated pattern detection using feature correspondence and (2) façade image segmentation. However, repeated pattern detection via feature correspondence is made difficult because of massive contamination by false correspondences and massive ambiguity, which is due to the repetition of patterns in a large part. Regarding façade image segmentation, the main challenge is to minimize the pixel mislabeling because of

weak low-level visual cues.

The goal of the thesis is to devise a robust method for feature correspondence that specifically deals with massively contaminated and ambiguous sets of correspondences. Then, an efficient detector of repeated patterns, such as windows can then be derived from such method. In turn, the window detection provided by the detector greatly helps façade segmentation and parsing, thus allowing to estimate of percentage of glass area in the building building.

Contents

2.1 Feature Correspondence Problem

Establishing correspondences between sets of visual features is a fundamental problem in computer vision. It has been well studied as it arises in many vision tasks such

- feature tracking as addressed in SHI and TOMASI (1994); BIRCHFIELD (2007),
- seamless consistent image stitching for panorama construction as studied in BROWN and LOWE (2007),
- multiple view geometry, more particularly camera calibration from touristic photographs arising in digital 3D reconstruction of highly frequented touristic places (see for example SNAVELY et al. (2008)) and wide-baseline stereo fusion (see for example PRITCHETT et al. (1998)),
- object detection as addressed, for example, in BERG et al. (2005);
- shape matching as addressed, for example, in LEORDEANU and HEBERT (2005); ZHENG and DOERMANN (2006);
- image classification (see for example LAZEBNIK et al. (2006)) or retrieval (see for example SCHMID and MOHR (1997)).

In ambiguous settings, e.g., when similar objects occur several times, like windows on a facade, or when distinctive textures are lacking, which is common in façades, establishing correspondences between sets of visual features needs to be made more robust, e.g., by using the geometric consistency of feature location.

Note that the feature correspondence is a rather large vision problem and the relevance of a matching method usually depends on the task which are among those mentioned above. Besides, its relevance may also depend of the type of visual features, which include, but are not limited to:

- simple interest points such as corners which can be detected with HARRIS and STEPHENS (1988)' famous corner detector;
- line segments, of which GROMPONE VON GIOI et al. (2010) present a reliable detector;
- edges, which can be detected for example with MARR and HILDRETH (1980)'s detector or CANNY (1986)'s, or object contours, which are used in GRAUMAN and DARRELL (2004);
- sampled contour points, which are used in BELONGIE et al. (2002); BERG et al. (2005);
- locally invariant regions or *blobs*, such as local extrema of difference of gaussians (LINDEBERG 1991; LOWE 2004), affine covariant regions (MIKOLAJCZYK et al. 2005) and maximally stable extremal regions (MATAS et al. 2002).

In the thesis, we focus our attention on the correspondence problem between sets of locally invariant regions.

2.1.1 Photometric Comparison-Based Approaches

The feature correspondence problem is by nature a challenging combinatorial problem. Devising scalable robust methods is therefore of crucial importance. Indeed, on the one hand, the number of possible correspondences quadratically in the number of visual features. On the other hand, requiring in addition geometric consistency in the feature correspondence multiplies the combinations depending on the number of correspondences in the consistent regions and is mainly responsible for considering a daunting combinatorial space.

A natural way to avoid such combinatorial explosion is to exploit the appearance cues when possible. Usually, a feature descriptor is devised for a given kind of feature and a similarity measure to compare these descriptors. Such a method can then be used in order to eliminate a large number of false positives, i.e., *outliers*. The resulting set of correspondences is much smaller and much less contaminated by outliers, making the correspondence problem not only computationally more tractable but also easier to solve.

In particular, locally invariant regions can be described very robustly, e.g., with SIFT (LOWE 2004). Alternatively, more computationally efficient descriptors such as SURF (BAY et al. 2008), DAISY (TOLA et al. 2010) have been proposed. These above-mentioned descriptors remain remarkably invariant under a variety of settings such as viewpoint changes, illumination changes, rotation and scale changes, blur, image compression.

Note that previously mentioned kinds of features other than locally invariant regions (in particular line segments and contours) are not easy to describe robustly and thus seems less employed. This is partly because they do not enjoy obvious invariant properties such as relative invariance to viewpoint changes. Nevertheless, their relevance are not to be downplayed as they are generally sparser and are potentially much more relevant visual cues than locally invariant regions. For example, camera calibration becomes notoriously challenging in urban scenes containing, among others, windows which typically have a challenging appearance variability because of its specular surface. For such specific problem, line segments are indeed more relevant than locally invariant regions. Let us however cite BELONGIE et al. (2002) and WANG et al. (2009), that propose contextual signatures or photometric descriptors for such features.

2.1.2 Geometry Consistency

Photometric comparison-based approaches are sufficient in simple settings but, in any case, geometric consistency remains essential. In simple settings, e.g., scenes that are well-textured and present few repetitions, photometric comparison approaches alone produces very good set of correspondences, i.e., the set is very little contaminated by outliers. Additionally, bag-of-features approaches can be used to build invariance of these descriptors to intra-class variation for image categorization and retrieval purposes (see for example LAZEBNIK et al. (2006)). However, the photometric descriptor becomes less reliable in many difficult real-life settings, in which case the number of false positives potentially becomes overwhelming. For instance, such settings corresponds to situations where viewpoint changes or illumination variations between image pairs are too important for the photometric descriptor.

Second, ambiguity also occurs when, for example, correspondences must be established between images picturing a scene with numerous repetitive patterns. Then, feature detectors produce a number of similar features, in which case photometric comparison alone cannot disambiguate correspondences. Therefore, many correspondences may look correct from a *local and purely photometric* standpoint, but are actually false in terms of *global consistency*.

In these cases, in which ambiguous correspondences and outliers are too many, geometric consistency becomes critical to efficiently guide the correspondence task and can be enforced at three levels, i.e., in increasing order

- at the local scale of a given feature correspondence,
- at the "vicinity" of a given feature correspondence, which contains "close" feature correspondences,
- at the level of a whole "region" of correspondences.

We shall specify the meanings of each level of geometric consistency in this thesis.

To avoid breaking the flow of this chapter, we will thoroughly review in Chapter 3 existing methods for feature correspondence.

2.2 Pattern Detection For Façade Image Parsing

Image segmentation and parsing are key problems in image understanding.

Image segmentation aims at perceiving *low-level* grouping and organization as it forms image segments by grouping image pixels. The benefit of image segmentation is that it simplifies the representation of an image, facilitating its semantic interpretation.

Image parsing does more than just image segmentation as it additionally tries to retrieve the hierarchical structure of an image by means of image grammars. The literature on image parsing is vast and we just refer the reader to ZHU and MUMFORD (2006)'s survey. Here, as we specifically focus on façade image parsing, one wants to retrieve and parameterize the hierarchical structure of a building, e.g., number of floors, height, position of each floor, windows in each floor and so on. Note that, as demonstrated in TEBOUL et al. (2010); TEBOUL et al. (2011), image grammars are a powerful top-down information that significantly robustifies the segmentation of façade images.

2.2.1 Mislabeling in Segmentation and Weak Low-Level Cues

In image segmentation, the pixel mislabeling is often due to elementary low-level cues, e.g., contours, colors, which eventually becomes weak in difficult cases. In simple cases, such visual cues suffice to satisfactorily segment images. However, various issues generally occur within a single image. Namely, illumination variations and partial or complete occlusions frequently occur within a façade image (TEBOUL 2010; TEBOUL et al. 2010; TEBOUL et al. 2011). Besides, edges may become harder to detect as their saliency may vanish in highly illuminated parts of the façade in the image (TEBOUL 2010; TEBOUL et al. 2010; TEBOUL et al. 2011).

When illumination variations occur within an image, simple photometric similarity approaches, such as normalized cross-correlation of local patches, become unable to correctly label some parts of the image. For example, in façade images, façades are often more illuminated at the top than at the bottom (TEBOUL 2010; TEBOUL et al. 2010; TEBOUL et al. 2011). In particular, 'wall' pixels on the top of the image may become as bright as the sky. Thus, naive segmentation methods may wrongly label upper 'wall' pixels as 'sky' pixels.

More robust cues can be designed to efficiently guide the segmentation task. They include

- designing more discriminative feature space (MALIK et al. 1999; BREIMAN and SCHAPIRE 2001; LOWE 2004; TOLA et al. 2010; SHOTTON et al. 2008),
- learning a new feature space obtained by combining multiple feature spaces (CHENG et al. 2011)
- or jointly using basic geometric cues such as contours (CANNY 1986), line segments (GROMPONE VON GIOI et al. 2010), gradient map (KASS et al. 1988; COOTES et al. 1995).

2.2.2 High-Level Bottom Cues Provided by a Robust Pattern Detector.

Façade image parsing can be made more accurate and robust if we use high-level bottom-up cues. In particular, one can think of window detection, which will greatly helps to retrieve the compositional structure of the building. Window detection can be provided by object recognition methods such as VIOLA and JONES (2004); BLASCHKO and LAMPERT (2008).

However, the object recognition methods (VIOLA and JONES 2004; BLASCHKO and LAMPERT 2008) are somehow "generic". They do not exploit the fact that patterns can be repeated within an image, like windows in a building façade. In this case, they make erroneous detection, i.e., miss objects, hallucinate objects, or do not localize them properly. Since these methods have parameter to favor either precision or recall, it is here preferable to tune them in such a way that all found windows are accurately localized at the cost of missing many windows. Then retrieving undetected windows can be done by detecting robust visual features and by establishing correspondences between features on windows and other features on the rest of the image. The detected windows constitute very valuable and robust high-level bottom-up cues for façade image parsing.

Other visual features can be used to enhance the accuracy of image representation. In particular, line features are valuable visual cues for analyzing man-made environments. As a result, the joint use of all these cues efficiently yields very precise façade image parsing.

Note that higher level information can be also built in a bottom-up fashion to enforce appearance cues: see for example SHI and MALIK (2000) and DELONG et al. (2012).

2.3 Contributions of the Thesis

After reviewing state-of-the-art feature correspondence methods and pointing out their respective strengths and weaknesses in Chapter 3, this thesis brings four main contributions to the fields of feature correspondence and façade parsing. We now review them, following the organization of the manuscript.

- Chapter 4 thoroughly revisits the expected repeatability and precision of feature detectors, and a formalization on such performance measures is given. Then it completely formalizes the geometry-consistent correspondence problem as a 4^{th}-order local consistency between neighboring matches. The geometric consistency is enforced at three levels as mentioned in Subsection 2.1.2. Furthermore, such a formulation is able to take into account reasonable object deformation as opposed to just rigid scenes.

- Chapter 5 derives an efficient match propagation algorithm from such formulation. It departs from other match propagation methods as it completely formalizes the propagation procedure based on high-order local affine consistencies. In Chapter 6, our algorithm is shown experimentally to be more efficient than existing methods in deformable object matching and calibration. It scales to large correspondence problems while efficiently handling ambiguity.

- Chapter 7 presents a repetitive pattern search algorithm based on our match propagation method. Provided that a model object (on which features are detected) is given, our pattern search method recursively searches for similar objects in breadth-first search. It is shown to achieve state-of-the-art window localization results on the *eTrims* datasets (KORČ and FÖRSTNER 2009) and the *École Centrale Paris* datasets (TEBOUL 2010).

- Chapter 8 presents two techniques for efficiently combining our detection results from our pattern search methods in the façade parsing. The first contribution consists in combining our detection results with the bottom-up information obtained from pixel-wise classification. Such combination results in a significantly better bottom-up merit information, which is used to evaluate the quality of a façade parsing. The second contribution leads to (1) a better guiding of the façade parsing process in terms of precision and convergence speed, and (2) a limitation of the convergence deviation in terms of parsing results. This is efficiently achieved by combining our detection results and additional line detections.

The first two contributions are brought to the field of feature correspondence problems, with applications in particular to deformable object matching and camera calibration. They are theoretical and experimental contributions that demonstrates the efficiency of our geometric consistent formulation. These contributions advocate for our 4th-order match propagation algorithm.

The third contributions is brought to the field of object recognition, which is also a first step to improve façade parsing. Directly relying on our match propagation method, they provide reliable high-level bottom-up cues that turn out to be particularly valuable for image parsing. Combining our window detection with other bottom-up cues, we achieve state-of-the-art results on façade image parsing.

2.4 Publications

This work has lead to the following publications.

- *Efficient and Scalable 4th-order Match Propagation.*
  DAVID OK, RENAUD MARLET, JEAN-YVES AUDIBERT.
  In 11th Asian Conference on Computer Vision (ACCV 2012), Daejeon, Korea, November 2012.

- *High-Level Bottom-Up Cues for Top-Down Parsing of Facade Images.*
  DAVID OK, MATEUSZ KOZINSKI, RENAUD MARLET, NIKOS PARAGIOS.
  In 2nd Joint 3DIM/3DPVT Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT 2012), Zürich, Switzerland, October 2012.

PART I

Robust Geometry-Consistent Feature Correspondence

Chapter 3

Existing Approaches for Geometry-Consistent Matching

Establishing correspondences between sets of visual features arises in many vision tasks, e.g., object matching, structure-from-motion and pattern detection. In many cases, distinctive feature descriptors and simple photometric comparison approaches (LOWE 2004) successfully produce a reasonably good set of matches with respect to the task requirements, i.e., they are large enough to carry meaningful information and with a large enough proportion of inliers, i.e., correct correspondences. But in ambiguous settings, e.g., when similar objects occur several times (e.g., windows on a facade, rocks in a landscape) or when distinctive textures are lacking, these matching strategies may fail and jeopardize the whole task. Yet, more robust correspondences can be found using the geometric consistency of feature location. In this chapter, we review existing geometry-consistent approaches. They roughly fall into three main categories:

- model-based consistency methods,
- match propagation methods,
- combinatorial optimization approaches.

To help the reader, we summarize the state-of-the-art in Table 3.1.

In hard settings (many outliers, many ambiguities)	**Model-Based**	**Combinatorial**	**Standard Match Propagation**	**Our Method**
Scalable	++++		+	++++
Accurate	++++	+	+	+++
Outlier-resistant	+	++	+	+++
Deformation-resistant		+++	++	++
Efficient for multiple models	+	(+)	+	+++

Table 3.1: Strengths and weaknesses of state-of-the-art approaches.

Contents

3.1 Model-Based Consistent Methods

What we call model-based consistency methods includes RANSAC-based methods (FISCHLER and BOLLES 1981) and the Hough transform method (DUDA and HART 1972; BROWN and LOWE 2002). These two methods are not strictly exclusive methods as they can be used in conjunction as in BROWN and LOWE (2002)'s work.

They enjoy a great popularity in the vision community as their simplicity of implementation is undeniably unbeatable, at least in their basic form. They are extremely well suited for parametric model fitting. As far as feature correspondence is concerned, parametric models are essentially affinity, homography, fundamental or essential matrix. They are fast and robust if the noise in the data can be estimated and if the percentage of inliers among the set of candidate correspondences is of order 10% or greater.

In what follows, we recall the features of RANSAC-based methods and the Hough transform. Then we explain in which kinds of correspondence problem they become less suited for.

3.1.1 RANSAC and Variants

RANSAC-based methods are iterative methods that consist in estimating a parametric model from sampled elemental subsets.

RANSAC basically proceeds as described in Algorithm 3.1.

Since the work of FISCHLER and BOLLES (1981), continuous improvements have been proposed. In general, RANSAC variants have higher breakdown points than FISCHLER and BOLLES (1981)' standard method, i.e., they are more resistant to higher outlier rates. Without claiming to be exhaustive, let us list some most notable variants of RANSAC.

- Early RANSAC variants such as LMedS (ROUSSEEUW 1984) and MINPRAN (STEWART 1995) maximize some likelihood criterion instead of the number of inliers, which makes them more robust to higher contamination rates. In addition, in contrast to standard RANSAC, they do not require a user threshold that decides

Algorithm 3.1 RANSAC algorithm

```
Initialize the parameter of the current best model as the null model.
Set the current best support count to 0.          // The support is defined in the loop.
repeat
    draw a sample of correspondences with minimal cardinality
    estimate the parameters of the candidate model from the sample
    // Support = number of correspondences fitting with the candidate model
    count the support
    if the model has greater support than the current best one then
        update the current best model and the current best support
    end if
until the maximum number of iterations is reached
return the current best model
```

whether a data point is an inlier. Indeed, they are noise-adaptive, meaning that they try to estimate the noise scale automatically.

- Along the same line of research, TORR and ZISSERMAN (2000) propose a RANSAC variant called MLESAC. This variant proposes two modifications in the standard RANSAC. First, it maximizes a likelihood function instead of maximizing the number of inliers. Second, it uses a more robust cost function than the standard least squares, which leads to better accuracy in the model estimation.

- CHUM et al. (2003) propose LO-RANSAC. Namely, this variant refines the model parameter via a local optimization, which is performed when a drawn sample has the current best support. As a result, the number of sampling is decreased and the model accuracy is improved.

- Better sampling strategies have been proposed to increase the chance of drawing good samples as in NAPSAC (MYATT et al. 2002), ORSA (MOISAN and STIVAL 2004), PROSAC (CHUM and MATAS 2005), BetaSAC (MÉLER et al. 2010) and GroupSAC (NI et al. 2009). Other variants such as DEGENSAC (CHUM and MATAS 2005) or QDEGSAC (FRAHM and POLLEFEYS 2006) specifically deal with degenerate or quasi-degenerate elemental subsets to improve the sampling as well.

- Many more recent variants also estimate the noise scale automatically and adaptively, such as ASSC (WANG and SUTER 2004), RECON (RAGURAM and FRAHM 2011), ORSA (MOISAN and STIVAL 2004).

- Other RANSAC variants, such as the projection based M-estimator (MITTAL et al. 2011), propose to deal with possibly heteroscedastic noise, i.e., a Gaussian noise with varying standard deviation,

3.1.2 The Hough Transform

The Hough transform (DUDA and HART 1972) is a popular voting method, which estimates the sought parameter vector by efficiently pruning the parameter space. It basically proceeds as follows.

First, the parameter space needs to be quantized into well-parameterized *bins*, i.e., a multidimensional interval, such that the set of bins is a partition of the parametric space. The multidimensional array of bins is called an *accumulator*. The voting procedure consists in looping over the set of data points. At each iteration, an accumulator bin gets a new vote each time a data point falls into it. Once the loop has terminated, bins corresponding to local maxima are identified. Let us recall some practical remarks.

Parameterization of the parameter space.

First, a careful one-to-one parameterization of the space is needed to apply the Hough transform. In the line fitting example, the best parameterization is the pair consisting of (1) the distance between the line and the origin and (2) the orientation angle of the line.

Trade-off between accuracy, efficiency and identifiability.

Second, the Hough accumulator becomes more and more memory-intensive as the dimensionality of the parameter space increases. An adaptive coarse-to-fine strategy has been proposed to resolve this issue (ILLINGWORTH and KITTLER 1987).

In presence of noise, the bin size should be chosen carefully. The Hough transform should be able to identify locally maximal bins if they have a significantly more votes than neighboring bins. When bins are too fine, and because of the noise, data points corresponding to the optimal parameter may fall into neighboring bins instead of falling into the correct bin. As a result, the locally optimal parameter may be more difficult to pin down.

Hough clustering as a filtering step.

As far as feature correspondence is concerned, the Hough transform can be used in conjunction with RANSAC to remove spurious correspondences (BROWN and LOWE 2002). By doing so, the set of correspondences is much less contaminated and this facilitates model estimation for RANSAC-based methods.

The idea is to group together feature correspondences that are roughly consistent by coarse affinities. The clustering is done with a Hough transform with a coarse-binned accumulator. Each homography can be estimated in each bin through a RANSAC filtering.

3.1.3 Situations In Which These Methods Are Less Suited For.

Very strong contamination ratio.

While these parametric model-consistency approaches are gold standard methods in multiple-view geometry, they become much less suited for pattern detection and ambigu-

ous feature matching, where true correspondences can be less than 5%. In this case, RANSAC-based methods may take as long as an exhaustive search to eventually find the best elemental subset. As for the Hough transform, the voting results may not identify conclusively local maxima, which are hard to identify in such settings.

Local affine homographies with strong variations.

An alternative is to estimate locally affine homographies to explain local correspondences. In PRITCHETT et al. (1998) and BROWN and LOWE (2002), correspondences are explained by a number of independent homographies, i.e., disjoint planar facets. There is no relation among homographies other than looking for a totally new homography at the periphery of a previous one, which is inappropriate for curved surfaces and deformations. On the contrary, we argue that a continuous chaining of affine consistent matches is needed to cope with curved surfaces and deformations. Second, when deformation is too significant, the quality of a locally affine model is hard to evaluate because the model support depends on the knowledge of the noise. And the noise varies a lot even locally because of the deformation.

Deformable object matching with the thin plate spline model.

It is possible to globally model deformable object matching with thin plate splines as in CHUI and RANGARAJAN (2003). The term "thin plate spline" refers to a physical analogy involving the bending of a thin sheet of metal. By using the thin plate spline model, correct correspondences are assumed to be explained by a smooth function $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ which is supported by a number of correct correspondences. However, if we are to use RANSAC to choose the correspondences supporting ϕ, we are still clueless about setting the right threshold on deformation variations, because it may vary from a point to another. Instead, an energy formulation as proposed in CHUI and RANGARAJAN (2003) is more appropriate.

Another objection against the thin plate model is that the function ϕ is actually not globally smooth, it is piecewise smooth and has discontinuities. For example, urban scenes typically usually contains square or angular building façades, which makes the thin plate spline model appropriate only locally.

3.2 Combinatorial Approaches

In combinatorial approaches, the correspondence problem is usually cast as an NP-hard constrained assignment problem. Global optimality is usually hard to guarantee but a local optimum is very satisfactory in practice and is obtained by using efficient heuristics and approximations.

The assignment problem takes the following general form. Let $\mathcal{X} = \{x_i\}_{i=1}^m$ and $\mathcal{Y} = \{y_a\}_{a=1}^n$ be two sets of visual features, between which correspondences are to be established[1]. The goal is to find an assignment matrix $\mathbf{Z} = (z_{ia}) \in \{0,1\}^{m \times n}$ that

[1]For consistency, i and j are indices reserved for $\mathcal{X}$, and a and b reserved for $\mathcal{Y}$.

minimizes some cost function J subject to some constraints C, i.e.

$$\begin{array}{ll} \underset{\mathbf{Z}\in\{0,1\}^{m\times n}}{\text{minimize}} & J(\mathcal{X},\mathcal{Y},\mathbf{Z}) \\ \text{subject to} & C(\mathbf{Z}) \end{array} \tag{3.1}$$

The matrix component z_{ia} is 1 if and only if feature x_i corresponds to feature y_a and is 0 otherwise.

Generally, J takes the form of

- a linear function in $\mathbf{Z}$ in the linear assignment problem,
- a quadratic polynomial for graph matching problems, registration problems,
- a polynomial of order 3 at least for general hypergraph matching problems.

3.2.1 Appropriateness of combinatorial approaches

Some combinatorial approaches have been applied in very specific multiple view geometry problems and point registration problems. See for example MACIEL and COSTEIRA (2003); CHERTOK and KELLER (2010). However, as far as multiple view geometry problems are concerned and especially since the work of LOWE (2004), they can hardly claim to be as competitive as RANSAC-based methods from a strictly computational point of view. In theory and practice, they do not scale as well as RANSAC in terms of memory space and computational complexity.

Provided the correspondence problem is of reasonable size, their robustness make them well suited (1) when, as explained in Section 3.1, parametric model-consistent methods are not and (2) for object recognition or categorization with strong intra-class variation and with articulated parts. See for example DUCHENNE et al. (2009); LEORDEANU et al. (2009).

Combinatorial approaches include:

- Maciel and Costeira's approach based on a concave programming formulation (MACIEL and COSTEIRA 2003),
- a graph matching formulation as can be found in LEORDEANU and HEBERT (2005),
- hypergraph matching formulations which are more robust and expressive than graph matching formulation (ZASS and SHASHUA 2008; DUCHENNE et al. 2009).

Graph and hypergraph matching seem to be the most popular combinatorial approaches and we choose to only review them.

3.2.2 Outlier-Aware Strategies in Graph Matching

Several outlier-aware strategies have been attempted in graph and hypergraph matching approaches.

Some approaches such as ZHENG and DOERMANN (2004)'s work use a threshold on the computed match confidence, but the confidence value is relative and cannot be easily associated to a geometric, understandable measure, leaving the user clueless for setting a sensible threshold value.

Dummy points can be added to attract outliers as used in GOLD and RANGARAJAN (1996); BERG et al. (2005); ZHENG and DOERMANN (2006); CHO et al. (2010). A correspondence containing a dummy point is considered an outlier. However, introducing dummy points introduces supplementary degrees of freedom. For example, in GOLD and RANGARAJAN (1996), we need to set a score for every correspondence involving dummy points and the user is left clueless for setting a sensible score for correspondences involving dummy points.

Alternatively, ZASS and SHASHUA (2008) proposes a convex formulation in which the outliers are determined automatically and rigorously. However, because it makes strong assumptions, it experimentally underperforms state-of-the-art methods such as DUCHENNE et al. (2009); CHO et al. (2010) in handling the outliers.

3.2.3 Scalability issue.

Finally, scalability is an issue in graph matching approaches and this is especially true for hypergraph matching. Recently, research has been now focusing on this question as far as graph matching is concerned (see CHO and LEE (2012)). However, more particularly, in hypergraph matching, the algorithmic complexity still remains prohibitive: given n points, time $O(n^d \log n)$ has been reported for d-order potentials and after a number of approximations (see for example DUCHENNE et al. (2011)). Hypergraph matching approaches hardly scale to thousands of interest points, which would correspond to huge (gigabytes) affinity tensors, even after sparsification.

3.3 Match Propagation Methods

Unlike combinatorial approaches, propagation methods do not have obvious principled formulation. Yet, they scale much better than combinatorial approaches and enjoy better empirical performance.

They include LHUILLIER and QUAN (2002)'s, KANNALA et al. (2008)'s, FERRARI et al. (2004)'s and CHO et al. (2009)'s work. Basically, they solve many local correspondence problems through simultaneous match propagation. Different seeds are grown and adapt to local affine transformations. However, these approaches basically exploit second-order constraints and heavily depend on the affine shape adaptation (MIKOLAJCZYK et al. 2005). They are thus not or poorly applicable to features that are not affine-covariant, such as DoG-SIFT features (LOWE 2004). Moreover, as shown by our experiments, affine shape determination is not very precise and shape adaptation can thus be significantly noisy. Even if optimized during propagation as in FERRARI et al. (2004), affine shapes lack robustness as we will see in camera calibration experiments in Chapter 6. Some approaches such as KANNALA et al. (2008) and FERRARI et al. (2004) also require the images to be available, as opposed to only working on the set of abstract feature points.

In addition, these methods cope with a reasonable amount of matching ambiguity, but fail to limit false detection when the set of possible correspondences is strongly contaminated by outliers.

3.3.1 Our Method

Our method also belongs to the category of match propagation methods. It tries to overcome the above drawbacks. It is a simple but careful adaptation of the match propagation principle to 4th-order geometric constraints (feature quadruple matching). Our framework explains a set of matches by a continuous network of locally-similar affinities which are determined from neighboring matches rather than by the affine shape of a single match.

We will show in Chapters 4 and 6 that our approach enjoys many good properties. It works on any kind of feature point (not only affine-covariant), and different types of features can even be freely mixed for denser, more uniform or more precise correspondences. Besides, it does not require the image pixels after detection, contrary to most propagation based methods. Although it has no global view of all correspondences (contrary to non-approximating hypergraph matchers), it produces very reliable matches. It can tell inliers from outliers and is robust to high outlier contamination rates. It adapts to scenes that have to be explained by different, separate models as well as by continuous model deformation. Last, it scales to hundreds of thousands of matches, both in time and space.

3.3.2 Differences Between our Work and Existing Propagation Methods

As it will be seen in Chapters 4 and 5, our approach uses known ideas for matching under affinity constraint (MIKOLAJCZYK et al. 2005).

Yet, despite a possible feeling of déjà-vu, we consider it includes original ingredients and, as a whole, provides a unique blend. First, let us highlight the differences as follows. Our propagation is based on *local affinities* like KANNALA et al. (2008), FERRARI et al. (2004), CHO et al. (2009), but not on

- *pixel adjacency* as in LHUILLIER and QUAN (2002) and CECH et al. (2011),
- *flow* as in LHUILLIER and QUAN (2002),
- *similitude* transformation as in HACOHEN et al. (2011).

Our affinities are computed from *match triples* (any kind of feature points, possibly in combination), but not necessarily from

- *affine correspondences* as in the works of FERRARI et al. (2004); CHO et al. (2009),
- 2^{nd} *moment matrix plus gradient orientation* as in KANNALA et al. (2008),
- *patch transformations* as in the works of LHUILLIER and QUAN (2002); HACOHEN et al. (2011).

Our affinity constraint is 4^{th}-*order and sensitive to feature scale*, but not

- 2^{nd}-*order* as in the works of LEORDEANU and HEBERT (2005); CHOI and KWEON (2009); CHO et al. (2010),
- 3^{rd}-*order* as in CHERTOK and KELLER (2010) and *photometric* as in (DUCHENNE et al. 2011),
- 4^{th}-*order reduced to points* as in the works of ZASS and SHASHUA (2008); CHERTOK and KELLER (2010).

Now, the following points advocates for our match propagation approach.

- For precision and robustness, each point of our growing regions selects nearby scale-consistent candidates; each candidate (best first) then looks for a nearby consistent triple in the region.
- It is simpler that the expansion-contraction phases of (FERRARI et al. 2004).
- Our propagation is isotropic, image-order insensitive, scale-invariant and adapts to detection density like CHO et al. (2009), contrary to fixed-size grid in model image FERRARI et al. (2004), fixed-size pixel neighborhood as in the works of KANNALA et al. (2008); LHUILLIER and QUAN (2002); CECH et al. (2011) or reference image as in HACOHEN et al. (2011).
- We are purely based on features, like CHO et al. (2009)'s method, rather than photometric similarity. We do not require images (pixels) after feature detection, unlike the works of KANNALA et al. (2008); FERRARI et al. (2004); LHUILLIER and QUAN (2002); CECH et al. (2011); HACOHEN et al. (2011), nor a regular flow of images as in CECH et al. (2011) or epipolarly rectified image pairs as in CECH et al. (2011).

These characteristics are crucial for robustness and precision for the difficult scenes we address.

Chapter 4

Geometry Consistency in Feature Correspondence

As previously mentioned in Chapter 2, the geometry consistency of the feature correspondence can be enforced at three levels. The lowest level for geometry consistency is at the level of a single feature match. Namely, if (x, y) is a good feature match, the image around feature x should be similar to the image around feature y. This photometric criterion translates into features having "close enough" descriptors. The next two higher level of geometry consistency are (1) at the level of the "vicinity" of a feature match and (2) at the level of a whole geometry-consistent "region", as previously mentioned in Chapter 2. In this chapter, we propose a formalization of the geometric consistency at these two higher levels.

Contents

Notations To begin with, we first lay down some notations that will be useful in the rest of the chapter. Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets of features which are respectively extracted from two images, and let $\mathcal{M} \subseteq \mathcal{X} \times \mathcal{Y}$ be a given set of possible matches. In the following, we denote a match by $m = (x, y)$. It is typically a pair of features whose descriptors are close, or close enough compared to other close descriptors. Note that $\mathcal{M}$ may include numerous ambiguities, i.e., any number of matches m with the same feature x or y. $\mathcal{M}$ can even be $\mathcal{X} \times \mathcal{Y}$. We denote by ϕ a geometric (feature) mapping that maps any point $\mathbf{x}$ in image 1 to a point $\mathbf{y} = \phi(\mathbf{x})$ in image 2. We also call a set of matches $R \subset \mathcal{M}$ a *region*. Such a term will appear natural for our geometry-consistent formulation.

4.1 Information of Locally Invariant Regions

Before moving to the geometry consistency formulation, we will review the common information carried by locally invariant regions. Besides, the sets of features and matches that we consider can freely mix detectors and descriptors of different kinds, e.g., Harris-affine or Hessian-affine regions (MIKOLAJCZYK and SCHMID 2002), DoG+SIFT blobs and descriptors (LOWE 2004), MSER regions (MATAS et al. 2002). But a meaningful match can only involve a detector-descriptor pair of the same kind.

For each kind f of feature (a detector-descriptor pair), and each feature x of kind f, we assume that the following information is available:

- the position $\mathbf{x} \in \mathbb{R}^2$ of feature x in the associated image, which we note by a *bold font* change for readability;
- the shape $\mathcal{S}_x$, which represents the *possibly anisotropic* scale of feature x and is defined by
$$\mathcal{S}_x \stackrel{\text{def}}{=} \left\{ \mathbf{x}' \in \mathbb{R}^2 \;\middle|\; (\mathbf{x}' - \mathbf{x})^T \boldsymbol{\Sigma}_x^{-1} (\mathbf{x}' - \mathbf{x}) \leq 1 \right\} \tag{4.1}$$
where $\boldsymbol{\Sigma}_x \in \mathbb{R}^{2\times 2}$ is a scale matrix (also a positive definite matrix), typically provided by the detector;
- the orientation vector $\mathbf{o}_x$, typically provided by the descriptor;
- the feature descriptor $\mathbf{v}_x$.

Note that, while affine-covariant keypoint scales are elliptic, e.g., with a Harris-affine detector, others such as DoG scales are isotropic, i.e., circular as shown in Figure 4.1. Examples of elliptic feature scales are shown in Figure 4.2. Let us make a few comments about the scale matrix. In terms of dimensional analysis, the scale matrix $\boldsymbol{\Sigma}_x$ corresponds to the square of a length. From a statistical point of view, the scale matrix $\boldsymbol{\Sigma}_x$ can be viewed as a covariance matrix of a Gaussian distribution. In the image processing literature, the scale matrix corresponds to the inverse of the second moment matrix

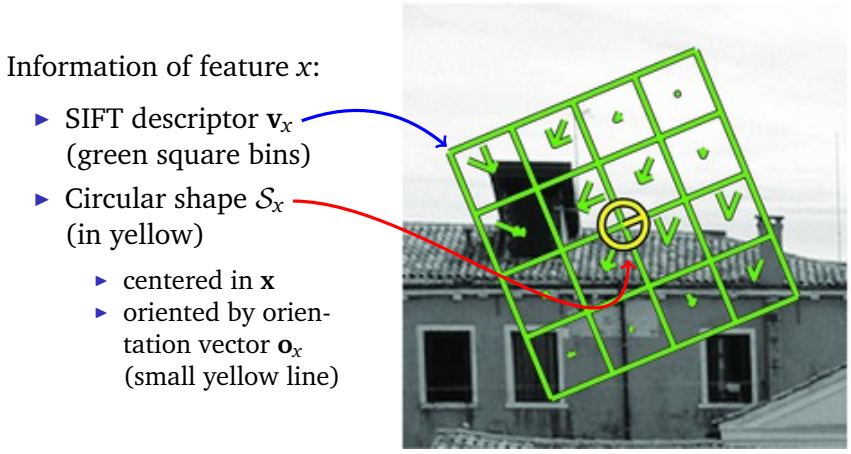


Figure 4.1: DoG+SIFT feature information. Credits: http://www.vlfeat.org/overview/sift.html.

which characterizes the local shape around feature x (see for example LINDEBERG and GÅRDING (1997), MIKOLAJCZYK and SCHMID (2002), MIKOLAJCZYK and SCHMID (2004), MIKOLAJCZYK et al. (2005)).

Let us also comment on the orientation vector $\mathbf{o}_x$, which orients the elliptic shape $\mathcal{S}_x$ in a *photometrically invariant* manner as illustrated in Figure 4.5. It must not be confused with the orientation of the ellipse axes with respect to the x-axis. It is defined from the dominant gradient direction around x. Additionally, it must be stressed that $\mathbf{o}_x$ does not necessarily correspond to the dominant gradient direction in a general affine scale space. However, it is the case only in an isotropic scale space.

The feature descriptor abstracts the image around x, also at some appropriate scale and some appropriate orientation, for comparison with other detected features. In particular, the SIFT descriptor corresponds to a concatenation of histograms of local gradients on a patch at some appropriate size and scale and centered on the feature position as illustrated in Figures 4.1 and 4.3.

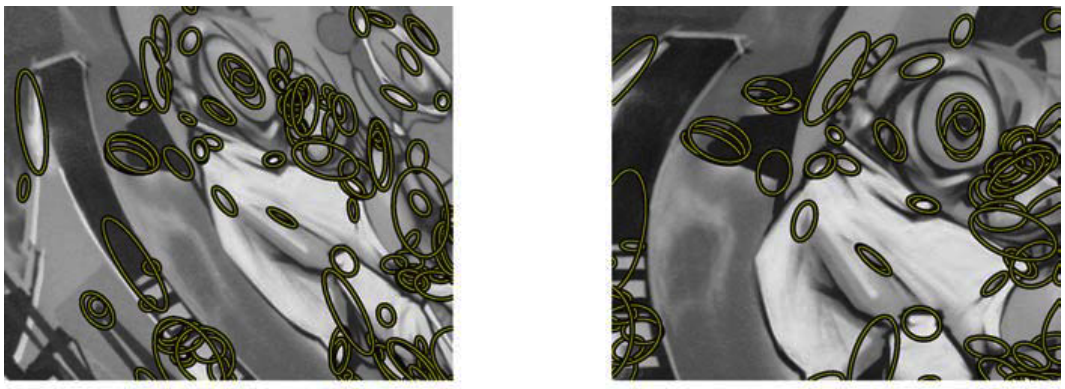
Finally, we introduce a *relative scale-sensitive distance* to x, which we define from the scale matrix $\mathbf{\Sigma}_x$ as follows

$$d_x(\mathbf{x}') \stackrel{\text{def}}{=} (\mathbf{x}' - \mathbf{x})^T \mathbf{\Sigma}_x^{-1} (\mathbf{x}' - \mathbf{x}). \tag{4.2}$$

This distance, which is also termed as *Mahalanobis distance*, will be useful in our geometry consistency formulation. Because the $\mathbf{\Sigma}_x$ is a positive definite matrix, level sets of d_x take the form of ellipses. We illustrate an anisotropic scale-sensitive distance d_x in Figure 4.4.

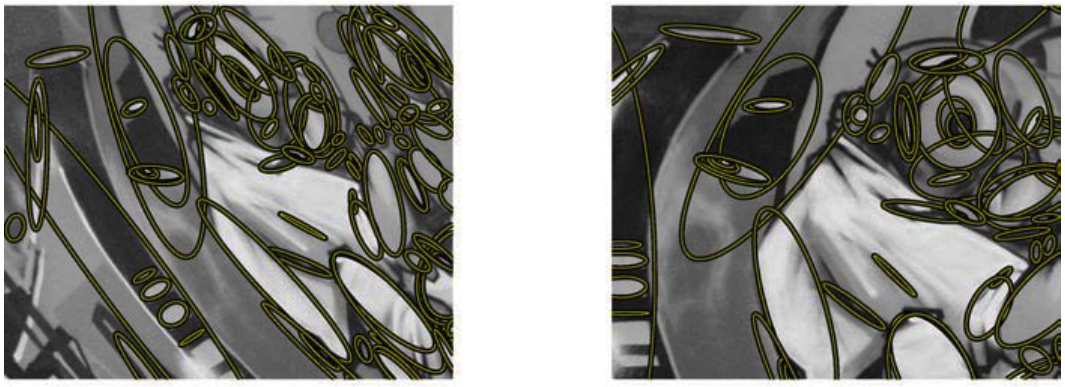
4.2 First Level of Consistency: Photometric Consistency

We now elaborate on the first level of geometry consistency, which is the photometric consistency at the level of a single feature match. More specifically, when (x, y) is a good



(a) **Harris-Affine**

(b) **Hessian-Affine**



(c) **MSER**

Figure 4.2: Different elliptic features. Credits: MIKOLAJCZYK et al. (2005).

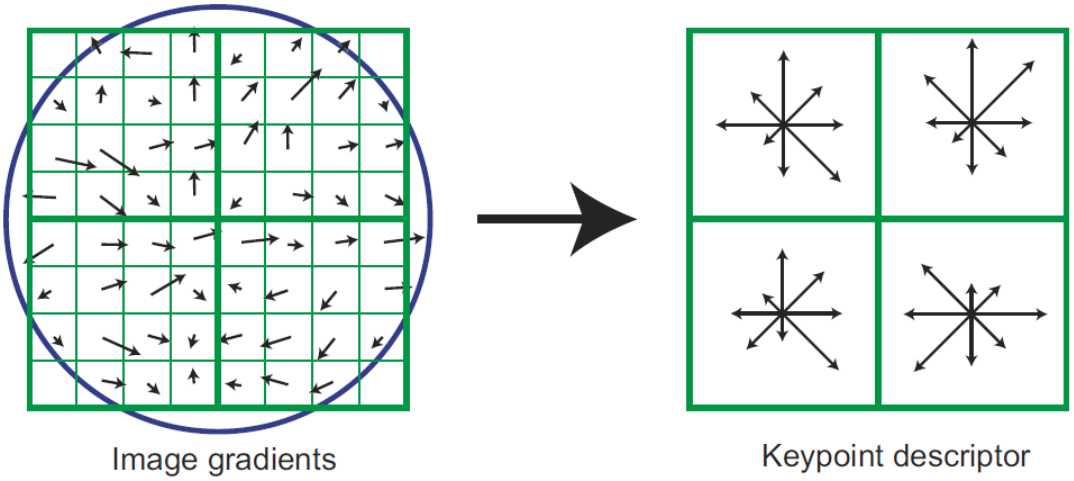


Figure 4.3: The SIFT descriptor. Credits: LOWE (2004).

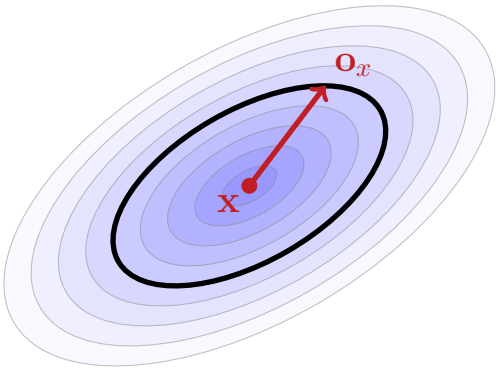


Figure 4.4: We illustrate here a feature point x, its orientation $\mathbf{o}_x$ and its level sets defined by d_x, which are ellipses because the scale matrix $\mathbf{\Sigma}_x$ is a positive definite matrix. The level set $\left\{\mathbf{x}' \in \mathbb{R}^2 : d_x(\mathbf{x}') = 1\right\}$ is represented by the black thick ellipse, while other elliptic level sets are in gray color.

feature match, the image around feature x should be similar to the image around feature y. This photometric criterion translates into features having "close enough" descriptors. As image patches around feature x and y have different scales and orientations and may be seen under different viewpoint changes, *normalizing* the image patches is necessary to accurately compare them. It then makes sense to have "close enough" descriptors.

The normalizing transform $\mathbf{T}_x$ of feature x is the affinity that maps the oriented shape $(\mathcal{S}_x, \mathbf{o}_x)$ to a zero-centered unit disk. The affinity $\mathbf{T}_x$ is of crucial importance to robustly describe the photometric appearance of shape $\mathcal{S}_x$ in an affine-invariant manner. Typically, $\mathbf{T}_x$ is determined from the joint detection and description of feature x. Indeed, only the

shape $\mathcal{S}_x$ of feature x is known at the detection step. Then, $\mathbf{o}_x$ and $\mathbf{v}_x$ are determined at the photometric description step, which follows the detection step.

Staying on the level of ideas, we recall that the description of feature x in a locally affine invariant manner with, say, SIFT, is roughly processed as follows.

- The area around feature x is convolved with an appropriate *anisotropic* Gaussian kernel, based on the scale matrix $\mathbf{\Sigma}_x$. There are two effects that results from a such convolution. On the one hand, fine details of the patch are suppressed in an *anisotropic* manner. On the other hand, the feature shape $\mathcal{S}_x$ is made *circular*. Indeed, these effects are desirable for a proper affine-invariant feature comparison.

- Histograms of local gradients are computed from the the convolved patch (with scale $\mathbf{\Sigma}_x$ and are then used to determine several dominant gradient orientations (see for example LOWE (2004) for details). Note that, when more than one dominant gradient orientations are found, the feature x is duplicated and are oriented differently according to these different gradient orientations.

- The convolved patch is rescaled into a zero-centered unit disk. The zero-centered unit disk is described with the SIFT descriptor $\mathbf{v}_x$ in a rotation invariant manner, using the dominant gradient orientation. Finally, the ellipse orientation $\mathbf{o}_x$ is deduced from the dominant gradient orientation.

The normalizing transform $\mathbf{T}_x$ can be computed explicitly from the scale matrix $\mathbf{\Sigma}_x$ and the orientation $\mathbf{o}_x$ as detailed in Appendix B. See in particular Algorithm B.1, which summarizes the computation of $\mathbf{T}_x$.

Note that MIKOLAJCZYK and SCHMID (2004) present an algorithm which adapts iteratively the shape $\mathbf{\Sigma}_x$ of feature x at position $\mathbf{x}$ to the image in a locally affine invariant manner. The convergence of the shape adaptation process yields a good estimate of shape $\mathcal{S}_x$ in the end. As a result, such process improves the robustness of descriptor to viewpoint changes. It ensures that if feature x corresponds to y, then descriptor $\mathbf{v}_x$ should be very close to $\mathbf{v}_y$, as illustrated in Figure 4.5.

Furthermore, if ϕ is a mapping that geometrically relates image 1 to image 2, a correct match (x, y) provides an affine approximation of ϕ around the corresponding positions $(\mathbf{x}, \mathbf{y})$, i.e., for all $\mathbf{x}'$ close to $\mathbf{x}$,

$$\phi(\mathbf{x}') \approx \mathbf{T}_y \mathbf{T}_x^{-1}(\mathbf{x}') \tag{4.3}$$

Nevertheless, let us remark that corresponding circular shapes from DoG+SIFT matches will give very coarse approximation of ϕ around their corresponding position, because they provide only similarity-based approximation around their corresponding positions.

4.3 Local Geometric Consistency Under Affinity Constraint

Standard matching procedures usually only rely on descriptor agreement, which is a local, point-wise property. Global consistency checks, if any, are deferred to the consumer

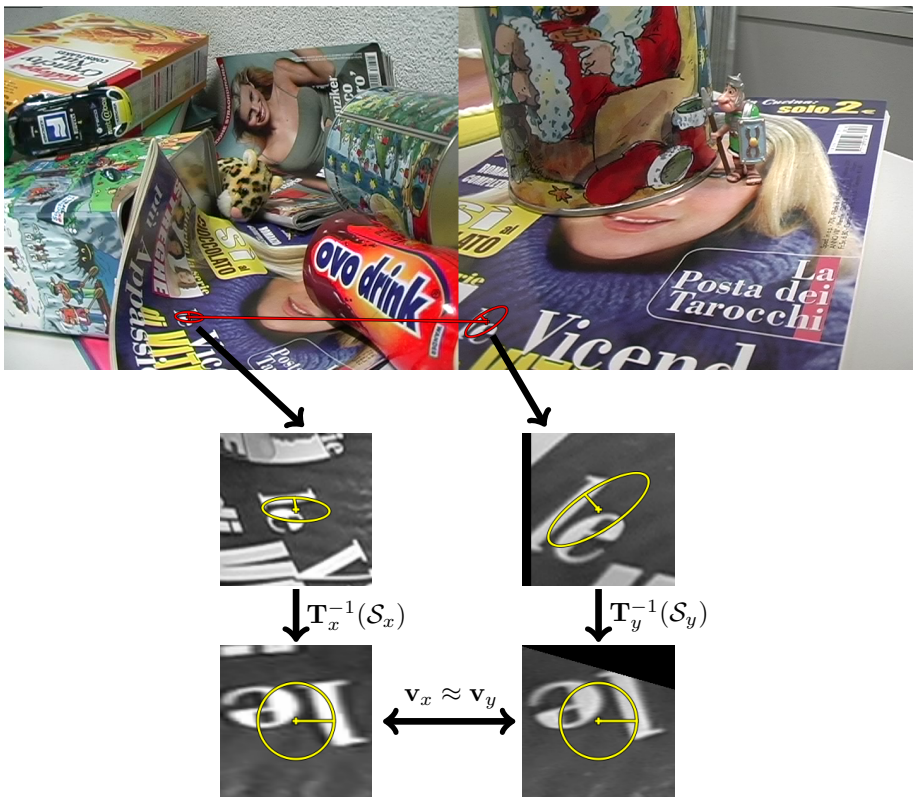


Figure 4.5: Affine invariant photometric comparison with normalizing transform. Here we show the least ambiguous match in this image pair. Looking closely, we notice that the "l" letter is slightly deformed in the left normalized patch with respect to the one in the right normalized patch, because, here, the local transform around the patch is actually projective than affine and the magazine is slightly curved. Best viewed using magnification.

of the matches. However, although most match consumers can cope with a certain proportion of outliers, many cannot properly deal with a large amount of ambiguous matches. As a result, ambiguous matches are often systematically discarded, which, in some circumstances, may greatly impoverish the set of remaining matches. Furthermore, even if we suppress ambiguous matches, the remaining unambiguous matches cannot be guaranteed to be inliers anyway.

It must be noted that such a coarse reduction of the number of matches does not smoothly degrade the quality of the overall process that exploits matches. Section 6.1 presents an example in 3D calibration where, in this case, only half of 60 cameras can be calibrated, whereas all 60 are calibrated based on the output of our algorithm.

Yet, feature information, besides descriptors, provides the ground for assessing the geometric consistency of a set of features. If x and y match, and if ϕ is a local affinity relating image 1 around x to image 2 then:

- the position $\phi(\mathbf{x})$ should be close to $\mathbf{y}$, taking scale into account;
- shape $\phi(\mathcal{S}_x)$ should be close to shape $\mathcal{S}_y$;
- and orientation $\phi(\mathbf{o}_x)$ should be close to orientation $\mathbf{o}_y$.

Symmetrically, this should also be true of (y, x) for the inverse affinity ϕ^{-1}. Note that "being close" hinges on the specific characteristics of the kind of feature as we will see next.

This section also aims at identifying the most discriminative information carried by local features and to what extent they can efficiently weed out outliers in the assessment of geometric consistency.

For the rest of the thesis, since we build upon Section 4.1, we formally define a feature and its transform by a geometric mapping ϕ as follows.

Definition 1 (Feature definition and transformed feature)**.**

- *A feature x is defined by the quadruple $(\mathbf{x}, \mathcal{S}_x, \mathbf{o}_x, \mathbf{v}_x)$, which is respectively composed of the geometric center $\mathbf{x}$, feature shape $\mathcal{S}_x$, orientation $\mathbf{o}_x$ and descriptor $\mathbf{v}_x$ (see Section 4.1).*
- *The transform of a feature x by a geometric mapping $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ is the* feature *denoted as $\phi(x)$ and defined by*

$$\phi(x) \stackrel{def}{=} (\phi(\mathbf{x}), \phi(\mathcal{S}_x), \phi(\mathbf{o}_x), \mathbf{v}_x) \tag{4.4}$$

Note that the descriptor $\mathbf{v}_x$ is assumed invariant by the transform ϕ in Equation (4.4). Notice that we also omit the feature kind f in the definition because, as stated in Section 4.1, we only consider matches (x, y) where features x and y are of the same kind f.

4.3.1 Expected Repeatability Scores of Feature Detector

We assume that each detector for a feature kind f comes with its *associated repeatability expectations*, that depend, e.g., on the detector precision and parameters or on the maximum expected change in images (viewpoint, illumination, etc.).

Based on this knowledge, we can test position, shape and orientation consistency between two features $\phi(x)$ and y. Let $\mathbb{P}_f$ be the predicate that tests such local consistency.

Informally, $(\mathbf{x}_1, \mathcal{S}_1, \mathbf{o}_1, \mathbf{v}_1)$ and $(\mathbf{x}_2, \mathcal{S}_2, \mathbf{o}_2, \mathbf{v}_2)$ are consistent according to $\mathbb{P}_f$ iff shape $\mathcal{S}_1$ at position $\mathbf{x}_1$ and shape $\mathcal{S}_2$ at position $\mathbf{x}_2$ are considered to coincide enough to possibly correspond to the same f-feature, and so are orientations $\mathbf{o}_1$ and $\mathbf{o}_2$.

Standard definitions for this include a threshold on the distance between the positions, possibly taking scale into account based on $\phi(\mathcal{S}_x)$ and/or $\mathcal{S}_y$. Joint with a threshold on the Jaccard distance between shapes $\phi(\mathcal{S}_x)$ and $\mathcal{S}_y$, it is used to estimate detector repeatability (MIKOLAJCZYK and SCHMID 2002). It can be combined with an orientation angle difference threshold.

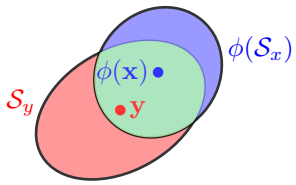


Figure 4.6: Overlapping ellipses $\phi(\mathcal{S}_x)$ and $\mathcal{S}_y$.

Recall that the Jaccard distance, denoted as $\mathcal{J}$, is a metric which measures the dissimilarity between two sets in terms of similarity and diversity. Here, in particular, given two shapes $\mathcal{S}, \mathcal{S}' \subset \mathbb{R}^2$, the Jaccard distance $\mathcal{J}$ is defined by

$$\mathcal{J}(\mathcal{S}, \mathcal{S}') = 1 - \frac{\text{area}(\mathcal{S} \cap \mathcal{S}')}{\text{area}(\mathcal{S} \cup \mathcal{S}')} \tag{4.5}$$

In MIKOLAJCZYK et al. (2005)'s work, the Jaccard distance is used to evaluate detector's repeatability and accuracy in planar scenes. Namely, when a match $m = (x, y)$ is considered to be an inlier with respect to some ground truth homography ϕ, i.e. $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \leq 1.5$ pixels, then shapes $\phi(\mathcal{S}_x)$ and $\mathcal{S}_y$ should normally differ very little in terms of Jaccard distance. Instead, we apply the Jaccard distance to any match $m = (x, y)$. $\phi(\mathcal{S}_x)$ and $\mathcal{S}_y$ are not required to have almost similar centers.

Figure 4.6 illustrates the Jaccard distance between shapes $\phi(\mathcal{S}_x)$ and $\mathcal{S}_y$. Given that shapes are here elliptic, the intersection area of overlapping shapes can be computed in analytic form. The computation is delicate and detailed in Appendix A.

A typical threshold value for the Jaccard distance is 0.4 as defined in MIKOLAJCZYK et al. (2005). However, because local affine approximation are not very accurate, this threshold can be loosened, e.g., to 0.6. This prevents the unwanted, premature rejection of possible matches.

Definition 2 (Repeatability-Precision Predicate $\mathbb{P}_f$ of Feature Kind f)**.**
Let f be a kind of feature. We assume that the following information are known:

- *the expected precision of position δ_p,*
- *the expected precision of shape δ_s and*
- *the expected precision of orientation angle δ_o.*

Then predicate $\mathbb{P}_f$ is a function which tests, for any pair of features (x_1, x_2) of kind f, position, shape and orientation consistency simultaneously and it is defined by:

$$\begin{aligned} \mathbb{P}_f(x_1, x_2) \overset{def}{=} & \quad (\ \max(d_{x_1}(\mathbf{x}_2), d_{x_2}(\mathbf{x}_1)) < \delta_p\ ) \\ & \wedge\ (\ \mathcal{J}(\mathcal{S}_{x_1}, \mathcal{S}_{x_2}) < \delta_s\ ) \\ & \wedge (\ \mathbf{o}_{x_1}.\mathbf{o}_{x_2} > \cos(\delta_o)\ ) \end{aligned} \tag{4.6}$$

4.3.2 Determining Repeatability Predicates for Each Feature Kind

We propose to empirically determine predicates $\mathbb{P}_f$ for each feature kind f. First, we discuss the relevance of the training datasets and how we should set predicates, given the possible weaknesses of such datasets. Second, we detail our experimental settings and the evaluation of detectors for different feature kinds $f \in \mathcal{F}$ where

$$\mathcal{F} \stackrel{\text{def}}{=} \{\text{DoG+SIFT}, \text{Harris-Affine+SIFT}, \text{Hessian-Affine+SIFT}, \text{MSER+SIFT}\}$$

Third, we discuss the expected repeatability and precision of detectors $\mathbb{P}_f$ and how we choose to set $\mathbb{P}_f$.

Choice of training datasets

We propose to learn predicate $\mathbb{P}_f$ from MIKOLAJCZYK et al. (2005)'s datasets for each considered feature kind f. These datasets are standard in computer vision and are constantly reused to evaluate the repeatability and precision of detectors and descriptors against a wide range of settings.

Note however that MIKOLAJCZYK et al. (2005)'s datasets only provides a repeatability/accuracy framework that relies on rigid projective transforms: images are related by a single homography. The learnt predicate $\mathbb{P}_f$, as described below, can be fairly reliable as long as the correspondence problem involves a rigid object or at least partially rigid object.

Regarding the matching of an object that has undergone very non-rigid deformation in an other image, it is hard to deem if the learnt $\mathbb{P}_f$ is sufficiently tolerant for such deformations. For this reason, it is actually reasonable to set more permissive thresholds δ_p, δ_s and δ_o than those estimated from MIKOLAJCZYK et al. (2005)'s datasets.

In the following, MIKOLAJCZYK et al. (2005)'s datasets are denoted by

$$\mathcal{D} \stackrel{\text{def}}{=} \{\textit{Bark}, \textit{Boat}, \textit{Graffiti}, \textit{Wall}, \textit{Trees}, \textit{Bikes}, \textit{Leuven}, \textit{UBC}\}.$$

Each of them consists of 6 images. For each dataset $d \in \mathcal{D}$ and for each image index

$$p \in \mathcal{P} \stackrel{\text{def}}{=} \{2, 3, \dots 6\},$$

a ground truth homography $\phi_{d,p}$ is provided for the image pair $(1, p)$ of dataset d for $p \in \mathcal{P}$.

The datasets evaluate the robustness of detectors and descriptors with respect to

1. increasing rotation and scale changes (*Bark* and *Boat*),
2. increasing viewpoint changes (*Graffiti* and *Wall*),
3. increasing blur (*Trees* and *Bikes*),
4. increasing illumination changes (*Leuven*) and
5. increasing JPEG compression (*UBC*).

Experimental settings

For each dataset of MIKOLAJCZYK et al. (2005), and each considered feature kind f, we extract all possible features with the default parameters of the detectors. All features are described with the SIFT descriptor.

It must be emphasized that the default parameters of detectors are usually such that detected features are as numerous as possible. Some works on feature correspondence tune empirically detectors in order to get results that favor their approach whereas we do not do so.

For each considered feature kind f, for each dataset $d \in \mathcal{D}$, for each image pair $p \in \mathcal{P}$, we match a set of initial feature matches $\mathcal{M}_{f,d,p}$ by matching SIFT descriptors using our *extended version* of Lowe's criterion (LOWE 2004), defined as a *distrust score* in Section 5.2 of Chapter 5. Specifically, we collect all matches such that their distrust score is less than 1.2. Such a value enables to consider ambiguous matches whereas MIKOLAJCZYK et al. (2005) discards them all by using much lower threshold values in $\{0.6, 0.8\}$. We refer the reader to Section 5.2 for details regarding the distrust score. We compute subsets of correspondences $\mathcal{M}_{f,d,p,i} \subset \mathcal{M}_{f,d,p}$ defined by

$$\mathcal{M}_{f,d,p,i} \stackrel{\text{def}}{=} \{(x,y) \in \mathcal{M}_{f,d,p} \mid \|\phi_{d,p}(\mathbf{x}) - \mathbf{y}\|_2 \in [\lambda_i, \lambda_{i+1}]\}\,,$$

where

$$\lambda_i \in \Lambda \stackrel{\text{def}}{=} \{0, 1.5, 5, 10\} \qquad \text{(values are in pixels.)}$$

Note that matches in $\mathcal{M}_{f,d,p,1}$ are considered inliers in MIKOLAJCZYK et al. (2005).

For each correspondence $(x,y) \in \mathcal{M}_{f,d,p,i}$, we compute the projected shape $\phi(\mathcal{S}_x)$ and orientation $\phi(\mathbf{o}_x)$ as detailed in Appendix C where $\phi = \phi_{d,p}$. We then compute the Jaccard distance $\mathcal{J}(\phi(\mathcal{S}_x), \mathcal{S}_y)$ and the cosine $\phi(\mathbf{o}_x).\mathbf{o}_y$ to get statistics about detector-descriptor accuracy. Namely, for each subset of correspondences $\mathcal{M}_{f,d,p,i}$, we compute:

- the minimum Jaccard distance/angle difference value,
- the maximum Jaccard distance/angle difference value,
- the mean Jaccard distance/angle difference value,
- the median Jaccard distance/angle difference value,
- the standard deviation value of the Jaccard distance/angle difference.

Analysis of the repeatability and accuracy of feature detector

All our reported experiments involve feature points with known scales (DoG, Harris-Affine, Hessian-Affine and MSER, all described with SIFT).

Experiments are summarized in Table 4.1. However, Table 4.1 actually carries too poor information as we will see that learning P_f is more complex than it actually seems.

Detailed experiments are reported in Appendix D. From the results, the median and mean value appears to be the most exploitable indicators and are often very similar.

Feature kind	Mean overlap precision δ_S	Mean angle precision δ_o
DoG + SIFT	0.48	59°
Harris-Affine + SIFT	0.34	12°
Hessian-Affine + SIFT	0.21	10°
MSER + SIFT	0.19	11.5°

Table 4.1: Mean overlap precision error and orientation precision error evaluated from likely correct matches in Mikolajczyk's datasets.

However, we preferably use the median value as it localizes well the half of "good" collected values of Jaccard distance or angle differences. Let us respectively denote the median Jaccard distance and the median angle difference by

$$\widetilde{\mathcal{J}}(f,d,p,i) \stackrel{\text{def}}{=} \underset{(x,y)\in\mathcal{M}_{f,d,p,i}}{\text{median}} \mathcal{J}(\phi(\mathcal{S}_x),\mathcal{S}_y)$$
$$\widetilde{\mathcal{A}}(f,d,p,i) \stackrel{\text{def}}{=} \underset{(x,y)\in\mathcal{M}_{f,d,p,i}}{\text{median}} \arccos(\phi(\mathbf{o}_x).\mathbf{o}_y).$$

Looking at the results, we observe that the most discriminative feature information is clearly the feature shape. In the following, let us fix a feature kind f and a dataset d. First, we see that the curve of the functions $p \mapsto \widetilde{\mathcal{J}}(f,d,p,i)$ are either coarsely constant or linear for $i \in \{1,\dots|\Lambda|\}$. This means the Jaccard distance are relatively stable across image pairs $((1,p))_{p\in\mathcal{P}}$ and for each interval $[\lambda_i,\lambda_{i+1}]$. Note that an increasing index p means that the matching setting (e.g., viewpoint changes, zoom+rotation, etc.) becomes increasingly difficult. We see that the Jaccard distance degrades consistently linearly for viewpoint changes or remains coarsely constant. Second, if $j \in \{2,\dots,|\Lambda|-1\}$, the difference function $p \mapsto \widetilde{\mathcal{J}}(f,d,p,j) - \widetilde{\mathcal{J}}(f,d,p,1)$ becomes dramatically larger and is quite constant. In each dataset d and each image pair p, as index i increases, the median Jaccard distance consistently becomes dramatically larger, which confirms that the discriminative power of feature shape.

In Appendix D, using DoG+SIFT features, the set $\mathcal{M}_{f,d,p,1}$ can be reliably distinguished from the sets $(\mathcal{M}_{f,d,p,i})_{i\neq 1}$, which are considered outliers in MIKOLAJCZYK et al. (2005) as the difference $\widetilde{\mathcal{J}}(f,d,p,i) - \widetilde{\mathcal{J}}(f,d,p,1)$ is positive and significantly large for $i \neq 1$.

However, with affine-covariant features, it is consistently hard to distinguish $\mathcal{M}_{f,d,p,1}$ and $\mathcal{M}_{f,d,p,2}$ as the difference $\widetilde{\mathcal{J}}(f,d,p,2) - \widetilde{\mathcal{J}}(f,d,p,1)$ is often very small. Fortunately, these two sets can be discriminated much more easily from the sets $(M_{f,d,p,i})_{i\geq 3}$.

Conversely, we do not observe such results with the median angle difference and they turn out to be unhelpful in the assessment of geometry consistency. In Appendix D, except for DoG+SIFT features whose orientations are confirmed to be quite inaccurate, the angle differences $\widetilde{\mathcal{A}}(f,d,p,j) - \widetilde{\mathcal{A}}(f,d,p,1)$ are often very low for $j \in \{2,\dots,|\Lambda|-1\}$.

Therefore, from this perspective, these results seems to clearly favor DoG+SIFT features over affine-covariant features for matching in camera calibration task. Indeed, the gap $\widetilde{\mathcal{J}}_{f,d,p,2} - \widetilde{\mathcal{J}}_{\text{f,d,p,1}}$ is sufficiently large for $j \geq 2$ for only $f =$ DoG+SIFT. It is then easy to set a threshold on the Jaccard distance to reject matches $m \notin \mathcal{M}_{\text{DoG+SIFT},d,p,1}$

during the assessment of geometry consistency. Note that if we are only concerned in object recognition, the confusion between $\mathcal{M}_{f,d,p,1}$ and $\mathcal{M}_{f,d,p,2}$ is not a serious problem if we are to use affine-covariant features.

To have a better overview, we factor the results in Appendix D by computing the averaged median over all datasets $d \in \mathcal{D}$. Specifically, we compute

$$\overline{\widetilde{\mathcal{J}}}\,(f,p,i) = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} \widetilde{\mathcal{J}}(f,d,p,i)$$

Finally, for each considered feature kind f, we plot the functions $\overline{\widetilde{\mathcal{J}}}\,(f,p,i)$ for $i \in \{1,\ldots,|\Lambda|-1\}$ in function of the image pairs $p \in \mathcal{P}$. The plotted functions as shown in Figures 4.7 and 4.8 confirm our analysis made above and they will help to trade off between accuracy and recall.

Choice of Parameters

Figures 4.7 and 4.8 show functions $\overline{\widetilde{\mathcal{J}}}\,(f,p,i)$ for considered feature kind f and index $i \in \{1,\ldots,|\Lambda|-1\}$. More importantly, they are relevant as they are quite stable and well-separated even though the image pair $(1,p)$ increases, i.e., when the viewpoint changes, illuminations changes, zoom and rotation and so becomes increasingly difficult. Unfortunately this is not the case for $\overline{\widetilde{\mathcal{A}}}\,(f,p,i)$.

Now, regarding shape and orientation consistency, we choose for simplicity to use constants $\delta_{\text{s}} = 0.4$ and $\delta_{\text{o}} = 60^\circ$ in all our experiments for all four kinds of features. Indeed, such permissive thresholds are required for DoG+SIFT, whose shape and orientation are not very precise, and given that $\mathbb{P}_f$ should be able to cope with nonrigid object deformation.

These constants are not the most optimal ones according to Figures 4.7 and 4.8 but they discriminate well set $\mathcal{M}_{\text{DoG+SIFT},d,p,1}$ w.r.t. $(\mathcal{M}_{\text{DoG+SIFT},d,p,i})_{i\neq 1}$. As for feature kinds f other than DoG+SIFT, they make a good separation between sets $(\mathcal{M}_{f,d,p,i})_{1\leq i\leq 2}$ and $(\mathcal{M}_{f,d,p,i})_{i>2}$. Furthermore, they are shown to be practically sufficient in camera calibration, deformable matching and pattern localization as we will see later in Chapters 6 and 7.

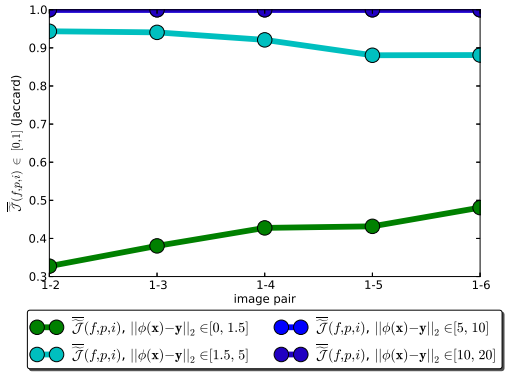
Alternatively, better constants can be chosen for each specific feature kind f. For example, given the empirical stability of $\overline{\widetilde{\mathcal{J}}}\,(f,p,i)$, it makes sense to choose

$$\delta_S = \frac{1}{2|\mathcal{P}|}\sum_{p\in\mathcal{P},i\in\{1,2\}} \overline{\widetilde{\mathcal{J}}}\,(f,p,i)$$

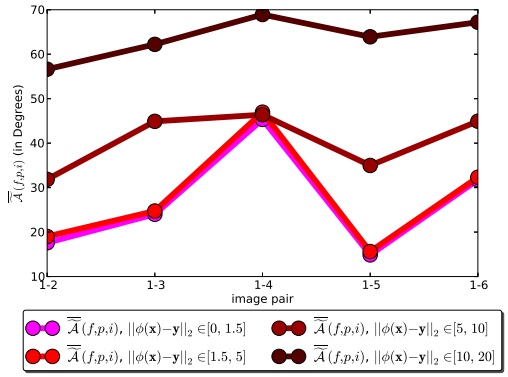
$$\delta_o = \frac{1}{2|\mathcal{P}|}\sum_{p\in\mathcal{P},i\in\{1,2\}} \overline{\widetilde{\mathcal{A}}}\,(f,p,i)$$

4.3.3 Position, Shape and Orientation Consistency

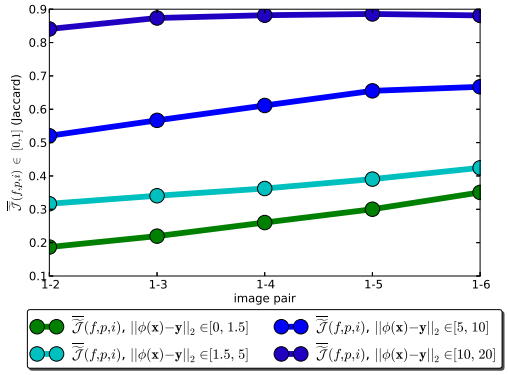
Building upon predicate $\mathbb{P}_f$ as defined in Definition 2, we introduce a predicate $\mathbb{P}_\phi$ which checks if an f-match (x,y) is geometrically consistent in terms of position, shape and



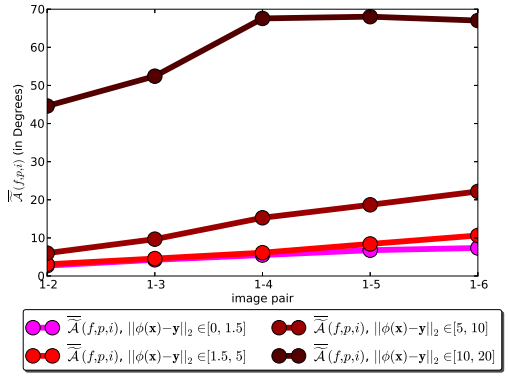
(a) Mean of median Jaccard distance $p \mapsto \overline{\widetilde{\mathcal{J}}}(f,p,i)$ for DoG+SIFT matches



(b) Mean of median angle difference $p \mapsto \overline{\widetilde{\mathcal{A}}}(f,p,i)$ for DoG+SIFT matches

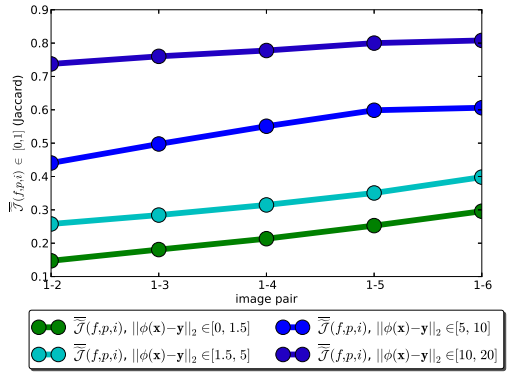


(c) Mean of median Jaccard distance $p \mapsto \overline{\widetilde{\mathcal{J}}}(f,p,i)$ for Harris-Affine+SIFT matches

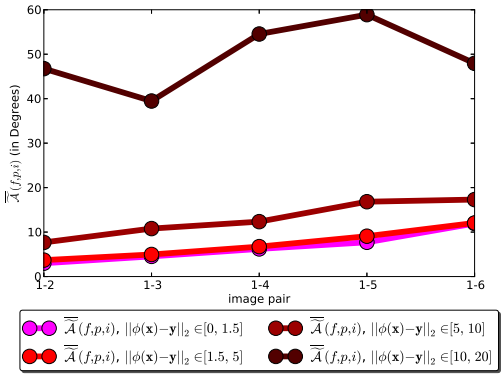


(d) Mean of median angle difference $p \mapsto \overline{\widetilde{\mathcal{A}}}(f,p,i)$ for Harris-Affine+SIFT matches

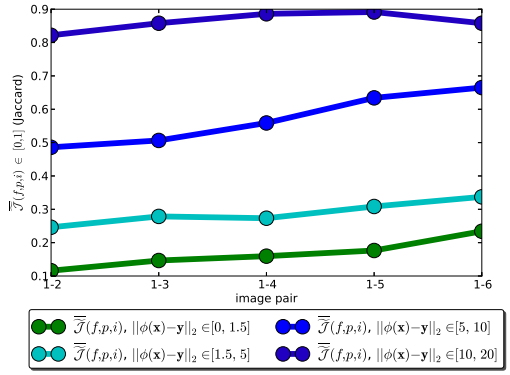
Figure 4.7: Plots of functions $p \mapsto \overline{\widetilde{\mathcal{J}}}(f,p,i)$ and $p \mapsto \overline{\widetilde{\mathcal{A}}}(f,p,i)$ are shown for feature kind $f \in \{\text{DoG+SIFT}, \text{Harris-Affine+SIFT}\}$.



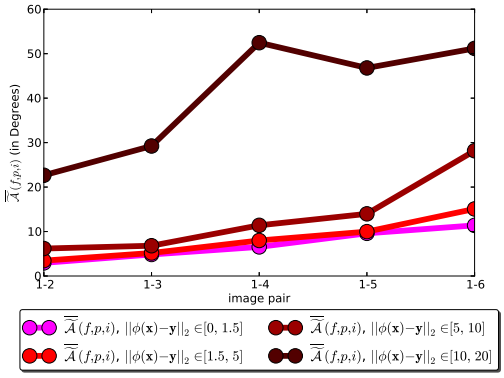
(a) Mean of median Jaccard distance $p \mapsto \overline{\widetilde{\mathcal{J}}}(f,p,i)$ for Hessian-Affine+SIFT matches



(b) Mean of median angle difference $p \mapsto \overline{\widetilde{\mathcal{A}}}(f,p,i)$ for Hessian-Affine+SIFT matches



(c) Mean of median Jaccard distance $p \mapsto \overline{\widetilde{\mathcal{J}}}(f,p,i)$ for MSER+SIFT matches



(d) Mean of median angle difference $p \mapsto \overline{\widetilde{\mathcal{A}}}(f,p,i)$ for MSER+SIFT matches

Figure 4.8: Plots of functions $p \mapsto \overline{\widetilde{\mathcal{J}}}(f,p,i)$ and $p \mapsto \overline{\widetilde{\mathcal{A}}}(f,p,i)$ are shown for feature kind $f \in \{\text{Hessian-Affine+SIFT}, \text{MSER+SIFT}\}$.

orientation with respect to a local affinity ϕ. As it is built upon predicate $\mathbb{P}_f$, once again, we emphasize that predicate $\mathbb{P}_\phi$ takes into account the expected repeatability and precision of the feature detector-descriptor f.

Definition 3 (Predicate $\mathbb{P}_\phi$ for Position, Shape and Orientation Consistency)**.**
Predicate $\mathbb{P}_\phi$ that assesses the consistency of a match $m = (x, y)$ under an affinity constraint ϕ is defined as

$$\mathbb{P}_\phi(x, y) \stackrel{def}{=} \mathbb{P}_f(\phi(x), y)) \wedge \mathbb{P}_f(x, \phi^{-1}(y)). \tag{4.7}$$

In words, the predicate stating that a given f-match $m = (x, y)$ is *position-, shape- and orientation- consistent w.r.t. affinity* ϕ holds iff $\mathbb{P}_f$ holds both for $(\phi(x), y)$ and $(x, \phi^{-1}(y))$.

4.4 Definitions for Higher Levels of Geometry Consistency

This section introduces several definitions and a terminology for both our geometry consistent formulation and our match propagation method presented in Chapter 5. These definitions gravitates around two fundamental components on which our geometry consistent formulation is firmly grounded. Namely, we define precisely (1) the notion of neighboring matches and (2) the notion of region boundary. They will also play a key role in our match propagation method.

4.4.1 Pairwise Scale-Sensitive Consistency Score Function.

Let us first introduce a relative scale-sensitive consistency score as follows. Given a match $m = (x, y)$ in $\mathcal{M}$, the consistency score of a match $m' = (x', y') \neq m$ with respect to m is the quantity defined by

$$\rho_m(m') \stackrel{\text{def}}{=} \frac{\min(d_x(\mathbf{x}'), d_y(\mathbf{y}'))}{\max(d_x(\mathbf{x}'), d_y(\mathbf{y}'))} \in [0, 1] \tag{4.8}$$

This score is motivated by the following observation. If two matches $m = (x, y)$ and $m = (x', y')$ are inliers, then it is expected that their scale-sensitive distance $d_x(\mathbf{x}')$ and $d_y(\mathbf{y}')$ are very similar. In particular, such similarity is efficiently translated by the fact that the ratio between these distances should be close to 1, which is what ρ_m expresses. Because the $\min$ and $\max$ functions are involved in the definition, the score ρ_m is bounded in $[0, 1]$ and it increases as the geometric scale-sensitive consistency gets better. Once again, note that consistency score for DoG features cannot be expected to be very high, especially in significant viewpoint changes, because their circular shapes can only account for isotropic scale changes as opposed to affine-covariant features.

Since the scale-distance consistency score ρ_m is defined relatively to match m, we introduce a symmetrised scale-sensitive score which additionally is more robust. It is defined as follows

$$\forall (m, m') \in \mathcal{M} \times \mathcal{M}, \quad \rho(m, m') \stackrel{\text{def}}{=} \min\left(\rho_m(m'), \rho_{m'}(m)\right) \in [0, 1] \tag{4.9}$$

4.4.2 Match Neighborhood Function

Before defining the notion of match neighborhood, let us first introduce the nearest neighbor function defined on a set of features.

Definition 4 (K Nearest Neighbor Function)**.**
Let $\mathcal{X}$ be a set of features and $K > 0$ be a positive integer. The neighbor function $\mathcal{N}_K^{\mathcal{X}} : \mathcal{X} \to 2^{\mathcal{X}}$ maps any $x \in \mathcal{X}$ to the set $\mathcal{N}_K^{\mathcal{X}}(x)$ of features $x' \in \mathcal{X} \setminus \{x\}$ such that position $\mathbf{x}'$ is among the K nearest neighbors of position $\mathbf{x}$ with respect to the Euclidean distance.

Now, based on these previous definitions, we introduce the match neighborhood function as follows.

Definition 5 (Match Neighborhood Function)**.**
Let K be a positive integer and ρ_0 be a real positive $\in [0, 1[$ [a]. The match neighborhood function $\mathcal{N}_{K,\rho_0} : \mathcal{M} \to 2^{\mathcal{M}}$ maps any potential match $m = (x, y) \in \mathcal{M}$ to the set of potential matches $m' = (x', y') \in \mathcal{M}$ defined as follows

$$\mathcal{N}_{K,\rho_0}(m) = \left\{ m' \in \mathcal{M} \setminus \{m\} \;\middle|\; \begin{array}{c} \left(x' \in \mathcal{N}_K^{\mathcal{X}}(x) \vee y' \in \mathcal{N}_K^{\mathcal{Y}}(y) \right) \wedge \\ \rho(m, m') \geq \rho_0 \end{array} \right\} \tag{4.10}$$

[a] ρ_0 may also depend on the type of feature f used.

Note that $\mathcal{N}_{K,\rho_0}$ is defined on feature pairs, not on single features. Put into words, a match $m' = (x', y')$ is a neighbor of $m = (x, y)$ if and only if either x' is among the K nearest neighbors of x or y' is among the K nearest neighbors of y, and the pair of matches (m, m') is sufficiently consistent.

The match neighborhood $\mathcal{N}_{K,\rho_0}(m)$ is controlled by two parameters K and ρ_0. These parameters play an important role in the match propagation, which will be presented later. If K is too small and/or ρ_0 is close to 1, $\mathcal{N}_{K,\rho_0}(m)$ will have a small cardinality, causing a premature termination of the match propagation. Conversely, if K is too large and/or ρ_0 is too small, the match neighborhood $\mathcal{N}_{K,\rho_0}(m)$ of m does not have a geometrically sound meaning anymore.

Because the match neighborhood function $\mathcal{N}_{K,\rho_0}$ is not symmetric, we construct a symmetrised version as follows

$$\widehat{\mathcal{N}}_{K,\rho_0}(m) \stackrel{\text{def}}{=} \mathcal{N}_{K,\rho_0}(m) \cup \left\{ m' \mid m \in \mathcal{N}_{K,\rho_0}(m') \right\} \tag{4.11}$$

As a result, this ensures that

$$\forall (m, m') \in \mathcal{M},\; m \in \widehat{\mathcal{N}}_{K,\rho_0}(m) \iff m' \in \widehat{\mathcal{N}}_{K,\rho_0}(m') \tag{4.12}$$

However, this is mostly of theoretical interest. As illustrated in our experiments in Subsection 5.8.1 in Chapter 5, using $\mathcal{N}_{K,\rho_0}$ as opposed to $\widehat{\mathcal{N}}_{K,\rho_0}$ yields almost as good results while being significantly faster.

Let us conclude this subsection with an important remark. Note that our K nearest neighbor function in Definition 4 does use the *Euclidean* distance instead of the scale-sensitive distance d_x. The main good reason that advocates such choice is the following.

A too strong anisotropy of distance d_x would force a d_x-based neighborhood to be arbitrarily *thin and long*, which does not make sense to evaluate a local affinity around x from a such neighborhood. Because our propagation method will be based on the estimation of the local affinity around x from match triples in $\mathcal{N}_{K,\rho_0}(x)$, these triples must be "nondegenerate", i.e., basically not aligned (we specify this notion later in Section 5.3 of Chapter 5). For this, it is better to use the Euclidean distance than an anisotropic distance. More specifically, if the anisotropy of distance d_x is very strong, then the set $\mathcal{N}_K^{\mathcal{X}}(x)$ of features are such that positions $\mathbf{x}'$ are almost aligned. As a result, this will harm the quality of the local affinity we try to estimate around x. Such anisotropy does arise commonly with MSER features.

However, the scale-sensitive distance is used to define a pairwise scale-sensitive consistency which is expressed by score ρ and used in the definition of $\mathcal{N}_{K,\rho_0}$.

Finally, note that the scale-sensitive distance d_x depends of feature x, which also raises nontrivial implementation issues regarding efficient point query search.

4.5 Second Level of Consistency: A Fourth Order Constraint for Neighborhood Affine-Consistency

We recall that the feature correspondence problem relies on two kinds of assumptions.

1. If (x, y) is a good match, the image around x should be similar to the image around y. This photometric criterion translates into features having "close enough" descriptors.

2. Given a set of good matches $(x_i, y_i)_{1\leq i\leq n}$, the relative position of feature x_i w.r.t. other features $(x_j)_{j\neq i}$ is expected to be similar to the relative position of y_i w.r.t. other features $(y_j)_{j\neq i}$. This criterion is mainly geometric, i.e., based on the relative coordinates of the features. But it also has an indirect, photometric flavor as the feature shapes and orientations also have to agree when relating x_i and y_i in the context of $(x_j, y_j)_{j\neq i}$, not just the position.

We have elaborated on the first assumption in Section 4.2. The second assumption only holds locally. In particular, by letting $m_i = (x_i, y_i)$ for $1 \leq i \leq n$, the relative agreement of m_i should hold at least for some of the matches

$$(m_j)_{j\neq i} = \widehat{\mathcal{N}}_{K,\rho_0}(m_i)$$

defined in Equation (4.11) (see also Definition 5).

Furthermore, in the second assumption, a set of consistent matches $(x_i, y_i)_{1\leq i\leq n}$ can be defined as a set of matches locally related by a local homography (MIKOLAJCZYK and SCHMID 2002) to relate image neighborhoods. In practice, if image neighborhoods are related by a locally smooth mapping ϕ, it is sufficient to approximate it with an affinity.

This makes sense because an affine approximation is a first-order Taylor approximation of ϕ. In the end, such an approximation is not only computationally efficient but also little affects the quality of the results.

A way to construct a good estimate of the local affinity around a correspondence m_i is to pick a triple of correct matches in the potential match neighborhood $\widehat{N}_{K,\rho_0}(m_i)$. The estimate of the local affinity is then determined from the corresponding position $(\mathbf{x}_i, \mathbf{y}_i)_{1\leq i\leq 3}$. We also stress that searching in the match neighborhood $\widehat{N}_{K,\rho_0}(m_i)$ will encourage choosing *close* matches $(m_i)_{1\leq i\leq 3}$ to m in the triple construction so that the estimated local affinity remains physically meaningful.

Here, we make an important assumption: a correct match m_i is consistent w.r.t. at least one local affinity defined from a triple of correct matches $(m_{a(i)}, m_{b(i)}, m_{c(i)})$ in $\widehat{\mathcal{N}}_{K,\rho_0}(m_i)^3$. As the phrasing is rather tedious and long, we will shorten it by just saying: "m_i is consistent w.r.t. at least one triple $(m_{a(i)}, m_{b(i)}, m_{c(i)})$".

Note that a single match m_i can be consistent w.r.t. many almost similar local affinities. This is the case when the feature mapping is projective and also when the matching involves curved surfaces. This means that a match m_i can be potentially consistent w.r.t. many triples of matches $\big(m_{a(i)}, m_{b(i)}, m_{c(i)}\big)$.

The goal of this section is to formalize what it means for a match m_i to be consistent w.r.t. to a triple $\big(m_{a(i)}, m_{b(i)}, m_{c(i)}\big)$.

4.5.1 Local Affine-Consistency within a Match Neighborhood

We first introduce the following notation. Let $(m_i)_{1\leq i\leq 3} = (x_i, y_i)_{1\leq i\leq 3}$ be a triple of matches. We define the associated affinity $A((m_i)_{1\leq i\leq 3})$ by the unique affinity ϕ that maps $\mathbf{x}_i$ in image 1 to $\mathbf{y}_i$ in image 2 for $i = 1, 2, 3$, i.e.,

$$\phi = A\left((m_i)_{1\leq i\leq 3}\right) \iff \forall i \in [\![1, 3]\!],\ \phi(\mathbf{x}_i) = \mathbf{y}_i \tag{4.13}$$

Note that the affinity $A((m_i)_{1\leq i\leq 3})$ associated to a triple of matches $(m_i)_{1\leq i\leq 3}$ only makes sense if the positions $(\mathbf{x}_i, \mathbf{y}_i)$ are not *degenerate*, i.e., if the points are not aligned, and more generally if the triangles corresponding to the feature triples in both images do not have too sharp angles (we specify minimum angle values later in Section 5.3 of Chapter 5). As stated earlier, triples are searched in some match neighborhood $\widehat{N}_{K,\rho_0}(m)$ and this is why we argued that using the Euclidean distance instead of a relative scale-sensitive distance d_x in Definition 4 does a better job to avoid choosing nondegenerate triples (see end of Subsection 4.4.2). Note also that, although we use the term triple, the match order is not relevant. A triple is actually a set with 3 elements, hence with no repetition (3 different matches).

Now, we define the consistency of a given match m w.r.t. a triple $(m_i)_{1\leq i\leq 3} \in \widehat{\mathcal{N}}_{K,\rho_0}(m)^3$ as follows.

Definition 6 (Affine Consistency of a Match with respect to a Match Triple)**.**
We say that m is affine-consistent *with matches $(m_i)_{1\leq i\leq 3}$ iff $(m_i)_{1\leq i\leq 3}$ is not degenerate and $\mathbb{P}_\phi(m)$ holds for $\phi = A((m_i)_{1\leq i\leq 3})$.*

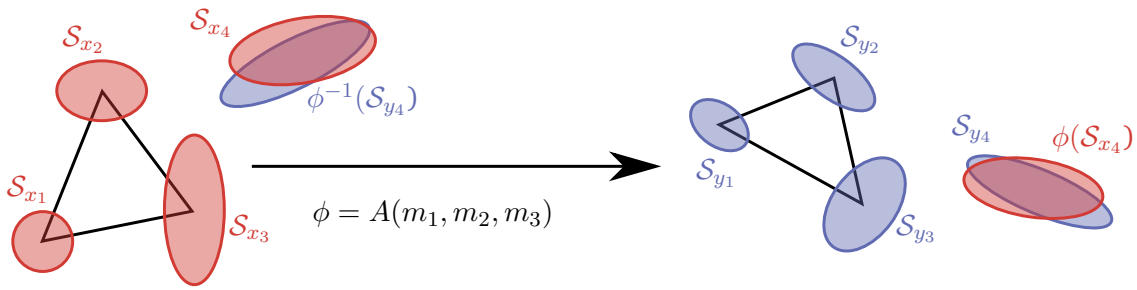


Figure 4.9: Affine consistency of match m_4 w.r.t. $\phi = A(m_1, m_2, m_3)$. The agreement between orientations is also taken into account but is not represented here for readability.

Figure 4.9 illustrates this concept.

As stated previously, we will define a consistent region not by a single local affinity but possibly by many local affinities. This particular setting provides a valuable flexibility allowing a region to adapt to substantial non-affine transformations. Later, our results will show that our approach is also applicable to deformable object matching (cf. Section 6.2).

4.5.2 Local Affine-Consistent Quadruples of Matches

In this subsection, we move to the central notion that firmly grounds our geometry consistent formulation as well as our match propagation method. Namely, we define an affine-consistent quadruple of matches as follows.

Definition 7 (Affine-Consistent Quadruple)**.**
A quadruple of matches $q = (m_i)_{1 \leq i \leq 4}$ *is* affine-consistent *iff*

- $\exists i \in 1 \leq i \leq 4,\ \forall j \in \{1, \ldots, 4\},\ j \neq i \Rightarrow\ m_j \in \mathcal{N}_{K,\rho_0}(m_i)$.
- *for all* $1 \leq i \leq 4$, m_i *is affine-consistent with* $(m_j)_{1 \leq j \leq 4, j \neq i}$.

For brevity, let us use the following notations:

- *we denote by* $\mathrm{Aff}\,(q)$ *the predicate that checks if the quadruple* q *of matches satisfies such property.*
- *Letting* m *be a match and* t *be a triple of matches, we write equivalently*

$$\mathrm{Aff}\,(q) = \mathrm{Aff}\,(m, t)\,,$$

where we define the quadruple q *as* $q = \{m\} \cup t$.

As for triples, the match order in a quadruple is not relevant. It is actually a set of cardinality 4.

We stress the importance of the two conditions in the local consistency. The first condition ensures that the geometric consistency makes sense only locally, i.e., within a

small neighborhood of matches. Note that we use $\mathcal{N}_{K,\rho_0}$ instead of $\widehat{\mathcal{N}}_{K,\rho_0}$ in Definition 7. This is mainly because $\mathcal{N}_{K,\rho_0}$ is more practical and $\widehat{\mathcal{N}}_{K,\rho_0}$ is mainly of theoretical interest. In any case, if the first condition applies, then, by construction of $\widehat{\mathcal{N}}_{K,\rho_0}$, we have

$$\forall (i,j) \in \{1,\dots,4\},\ j \neq i \Rightarrow\ m_j \in \widehat{\mathcal{N}}_{K,\rho_0}(m_i).$$

The second condition enforces the robustness of the geometric consistency.

Affine-consistent quadruples provide a solid ground to define what we call a consistent region in the next section. We will see that they provide good flexibility and robustness so that consistent regions are allowed to have a moderate amount of nonrigid deformation e.g., in object matching: when we want to match different views of some curved magazine which has undergone a moderate folding, a single region should cover the largest surface of the deformed magazine across different views, not just small almost rigid fragments.

4.6 Third Level of Consistency: Region Affine-Consistency

Three geometric ideas motivate our formulation of the region affine consistency. They are as follows.

1. A region R is geometrically consistent if the locally affine consistency holds for any match $m \in R$, i.e., for any match $m \in R$, there exists a triple of matches t in $\widehat{\mathcal{N}}_{K,\rho_0}(m) \cap R$ for which $\mathrm{Aff}\,(m,t)$ holds.

2. But this is just a sufficient condition. The union of two independent but geometry-consistent regions would then also be geometry-consistent. We are actually interested in "homogeneous" consistency. More precisely, a region R is viewed as a *single connected component* of matches, i.e., for any different pair of matches $(m,m') \in R^2$, there exists a chain $(m_{1,i}, m_{2,i}, m_{3,i}, m_{4,i})_{1\leq i\leq n}$ of affine-consistent quadruples from m to m' such that $m = m_{1,1}$, $m_{4,n} = m'$ and $m_{4,i} = m_{1,i+1}$ for $1 \leq i < n$.

3. In most 3D scenes, feature displacements are not homogeneous but related to the specific depths (and reflexion properties) of the observed objects. This is also the case when objects move. Geometric consistency is thus only expected in independent image regions, i.e., separate sets of features, not on a single region covering the whole image.

To begin with, let us bring a few more notations and definitions. Define the set of explanatory triples of a match m as

$$\mathcal{T}(m) \overset{\text{def}}{=} \left\{ t \in \widehat{\mathcal{N}}_{K,\rho_0}(m)^3 \mid \mathrm{Aff}\,(m,t) \right\}. \tag{4.14}$$

Then, we extend the definition to a region $R \subseteq \mathcal{M}$ as follows

$$\mathcal{T}(R) \overset{\text{def}}{=} \bigcup_{m\in R} \mathcal{T}(m) \tag{4.15}$$

Going back to affine-consistent quadruples, we define several sets of affine-consistent quadruples to m by reusing these definitions:

$$\mathcal{Q}(m) \stackrel{\text{def}}{=} \{\{m\} \cup t \mid t \in \mathcal{T}(m)\} \tag{4.16}$$

$$\mathcal{Q}(X) \stackrel{\text{def}}{=} \bigcup_{m \in X} \mathcal{Q}(m) \tag{4.17}$$

We now state some trivial observations that will be useful in later proofs.

Lemma 1. *If X and Y be two sets of matches such that $X \subseteq Y$, then*

$$\mathcal{T}(X) \subseteq \mathcal{T}(Y) \tag{4.18}$$

$$\bigcup_{t \in \mathcal{T}(X)} t \subseteq \bigcup_{t \in \mathcal{T}(Y)} t \tag{4.19}$$

The proofs are easy and left to the reader.

4.6.1 Region Affine-Consistency and Explanatory Networks

We now formalize the region affine-consistency with the definitions introduced in the last section. Given a match m, we can retrieve explanatory triples $t \in \mathcal{T}(m)$ of m. Then, the set of explanatory triples of a triple $t = (m_1, m_2, m_3)$ are the set of triples that explains at least one of the match m_i in the triple t, i.e.,

$$\mathcal{T}(t) = \mathcal{T}(m_1) \cup \mathcal{T}(m_2) \cup \mathcal{T}(m_3)$$

Now, going further by induction, we can somehow recover a "tree", more precisely, a network of ancestor triples that indirectly explain match m. Namely, for that purpose, we introduce the set of "ancestor triples" at depth n that directly or indirectly explain a given match m.

$$\mathcal{T}^1(m) \stackrel{\text{def}}{=} \mathcal{T}(m) \tag{4.20}$$

$$\forall n \in \mathbb{N}, \mathcal{T}^{n+1}(m) \stackrel{\text{def}}{=} \mathcal{T}\Big(\bigcup_{t \in \mathcal{T}^n(m)} t\Big) \tag{4.21}$$

where $\bigcup_{t \in \mathcal{T}^n(m)} t$ is a region, i.e., the set of matches used in $\mathcal{T}^n(m)$. We also observe that

$$\forall (i,j) \in \mathbb{N}^{\star 2},\ i < j \Rightarrow\ \mathcal{T}^i(m) \subseteq \mathcal{T}^j(m) \tag{4.22}$$

The rationale behind such inductive definition is that it lets us catch a glimpse of the region growing idea. Thus, we define the explanatory network of a match m as

$$\mathcal{E}(m) \stackrel{\text{def}}{=} \bigcup_{n \in \mathbb{N}^\star} \mathcal{T}^n(m). \tag{4.23}$$

According to the ideas sketched up previously, we finally define an affine-consistent region as follows.

Definition 8 (Affine-Consistent Region)**.**
A region R is affine-consistent iff we have the generative property:

$$\left(\bigcap_{m\in R}\mathcal{E}(m)\right)\cap R^3\neq\emptyset \tag{4.24}$$

For convenience, we denote the predicate checking for a given region R that Equation (4.24) holds by $\mathrm{Aff}(R)$.

Equation (4.24) means that the restricted explanatory networks $(\mathcal{E}(m)\cap R^3)_{m\in R}$ all have in common one triple t at least. If we consider such a triple $t=(m_1,m_2,m_3)$, t can be used directly or indirectly explains any $m\in R$, i.e., there always exists a chain of affine-consistent quadruples that joins m_1 to any $m\in R$. This is why region R can be *grown* from such a triple t of matches by successive chaining of affine-consistent quadruples. Finally, let us also observe that $|R|\geq 4$.

4.6.2 Maximal Affine-Consistent Region and Region Growing

In Definition 8, a region R is affine-consistent if it can be generated from an affine-consistent quadruple q. With the last property of the region affine-consistency, some region growing procedure can be indeed retrieved.

Namely, the match propagation process will operate on the *boundary* ∂R of a consistent region R. It is defined as follows.

Definition 9 (Region Boundary)**.**
The boundary ∂R of a region R is the set of neighbors of matches in R, excluding R itself, i.e.,

$$\partial R \overset{def}{=} \left\{ m'\in\mathcal{M}\setminus R \;\middle|\; \exists m\in R,\ m'\in\widehat{\mathcal{N}}_{K,\rho_0}(m)\right\} \tag{4.25}$$

$$\overset{def}{=} \left(\bigcup_{m\in R}\widehat{\mathcal{N}}_{K,\rho_0}(m)\right)\setminus R \tag{4.26}$$

The rationale behind this definition is that our match propagation seeks to grow the current region R iteratively by adding "consistent" matches in the boundary ∂R. Eventually, the region R becomes "maximal" when our match propagation method cannot add anymore "consistent" match in the boundary ∂R. We will specify the notions in quotes in the next sections.

Definition 10 (Region Growing)**.**
Define the affine-consistent boundary $\partial_{\text{Aff}}R$ of R as the subset of matches m of ∂R such that there exists a triple of matches $t \in R^3$ that explains m, i.e.,

$$\begin{aligned}\partial_{\text{Aff}}R &= \left\{m \in \partial R \mid \mathcal{T}(m) \cap R^3 \neq \emptyset\right\} && (4.27)\\ &= \{m \in \partial R \mid \exists t \in \mathcal{T}(m),\ t \subseteq R\} && (4.28)\\ &= \{m \in \partial R \mid \exists q \in \mathcal{Q}(m),\ q \setminus m \subset R\} && (4.29)\end{aligned}$$

Then, the region growing is a mapping G defined

$$\begin{aligned}G : 2^{\mathcal{M}} &\to 2^{\mathcal{M}}\\ R &\mapsto R \cup \partial_{\text{Aff}}R \qquad (4.30)\end{aligned}$$

Let us also state an important property.

Proposition 1. *If R be an affine-consistent region, then $G(R)$ is also affine-consistent.*

Proof. Let R be an affine-consistent region. We assume that $\partial_{\text{Aff}}R \neq \emptyset$, otherwise region $G(R) = R$ is affine-consistent.

Let us show that.

$$\left(\bigcap_{m\in G(R)} \mathcal{E}(m)\right) \cap G(R)^3 \neq \emptyset$$

We have the following hypotheses

$$\left(\bigcap_{m\in R} \mathcal{E}(m)\right) \cap R^3 \neq \emptyset \qquad (R \text{ is affine-consistent}) \qquad (4.31)$$

$$G(R) = R \cup \partial_{\text{Aff}}R \qquad (\text{by definition of function } G) \qquad (4.32)$$

Let t_0 be a triple in $\left(\bigcap_{m\in R} \mathcal{E}(m)\right) \cap R^3$. Thus, $t_0 \in G(R)^3$ because $R \subseteq G(R)$ (cf. Equation (4.32)). Now, fix $m \in \partial_{\text{Aff}}R$. It remains to show that $t_0 \in \mathcal{E}(m)$. Therefore, $\mathcal{T}(m) \cap R^3 \neq \emptyset$ and let t be a triple in $\mathcal{T}(m) \cap R^3$. As t is also a set of three distinct matches m' in R. Now we observe that

$$\forall m' \in t,\ \mathcal{T}(m') \subseteq \mathcal{T}^2(m). \qquad (4.33)$$

Thus, by induction, it is easy to show that

$$\forall n \in \mathbb{N}^\star, \mathcal{T}^n(m') \subseteq \mathcal{T}^{n+1}(m) \qquad \textit{(we prove this at the end.)} \qquad (4.34)$$

Consequently, $\mathcal{E}(m') \subseteq \mathcal{E}(m)$, because of the definition of the network of explanatory triples $\mathcal{E}$. But by definition of t_0, we observe that $t_0 \in \mathcal{E}(m')$. This holds for any $m' \in t$ and therefore $t_0 \in \mathcal{E}(m)$. That means $t_0 \in \mathcal{E}(m)$ for both any $m \in \partial_{\text{Aff}}R$ and for any $m \in R$, by hypothesis, which concludes the proof.

To fully terminate the proof, we prove Proposition (4.34) by induction. First, let us notice that Proposition (4.34) holds for $n = 1$, since it corresponds to Equation (4.33). Suppose now that Proposition (4.34) holds for some $n \in \mathbb{N}^\star$, let us show it still holds for $n + 1$. For this, denote

$$X = \mathcal{T}^n(m') \text{ and } Y = \mathcal{T}^{n+1}(m).$$

Proposition (4.34) holds for n, which means $X \subseteq Y$. Then, with Property (4.19), we observe that

$$\mathcal{T}(\bigcup_{t \in X} t) = \mathcal{T}(\bigcup_{t \in Y} t)$$

which is, by definition

$$\mathcal{T}^{n+1}(m') = \mathcal{T}^{n+2}(m)$$

This concludes the proof by induction. □

Now, using the notation $\underbrace{G \circ G \circ \ldots G}_{n \text{ times}} = G^n$ for successive function composition and letting $G^0 = \mathrm{Id}$, we have for any affine consistent region R

$$\forall i, j \in \mathbb{N}^*, i < j \Rightarrow G^i(R) \subseteq G^j(R) \tag{4.35}$$

$$G^\infty(R) \overset{\text{def}}{=} \lim_{n \to \infty} G^n(R) = \bigcup_{n=0}^{\infty} G^n(R) \subseteq \mathcal{M} \tag{4.36}$$

$$\forall n \in \mathbb{N}, |G^0(R)| + \sum_{i=1}^{n-1} |\partial_{\text{Aff}} G^i(R)| = |G^n(R)| \tag{4.37}$$

4.6.3 Maximal Affine-Consistent Region and Properties

As the feature correspondence problem seeks to find the maximum number of matches, it is natural to look for maximal affine-consistent regions, which we formally define as follows.

Definition 11 (Maximal Affine-Consistent Region)**.**
We say that an affine-consistent region R is maximal if and only if

$$\forall R' \subset \mathcal{M}, \text{Aff}(R') \wedge (R \subset R') \Rightarrow R = R' \tag{4.38}$$

Necessary condition. The following proposition states a practical necessary condition in the maximality of an affine-consistent region R.

Proposition 2. *For any affine-consistent region $R \subset \mathcal{M}$, if R is maximal then*

$$\partial_{\text{Aff}} R = \emptyset. \tag{4.39}$$

Proof. Let R be a maximal affine-consistent region. Let us suppose that $\partial_{\text{Aff}} R \neq \emptyset$. Since $\partial_{\text{Aff}} R \subset \mathcal{M} \setminus R$, R is strictly included in $G(R)$, which is also an affine-consistent region because of Proposition 1. However, this is impossible because it contradicts the maximality of R. □

Let us also observe that an empty affine-consistent boundary is equivalent to the following standpoints.

$$\partial_{\text{Aff}} R = \emptyset \tag{4.40}$$
$$\iff R = G(R) \tag{4.41}$$
$$\iff \forall m \in \partial R, \forall t \in \mathcal{T}(m) \cap R^3, \neg\text{Aff}\,(m, t) \tag{4.42}$$

The equivalences are easily checked from the definitions of the affine-consistent boundary and of the function G.

Besides, the necessary condition (4.39) naturally leads to try constructing maximal affine consistent region by growing an affine-consistent region $R = G^\infty(t)$ from a seed triple t (cf. Chapter 5.). Empirically, ensuring that $\partial_{\text{Aff}} R = \emptyset$ produces a quasi-maximal affine-consistent region R as our experiments will show in Section 5.8.1 of Chapter 5 and in Chapter 6.

Sufficient condition? Actually, the necessary condition (4.39) is not sufficient to ensure maximality of the affine-consistent region. To see that, let R_1 and R_2 be two affine-consistent regions. Let us suppose that R_1 and R_2 are such that

- R_1 and R_2 are quadruples,
- the affine-consistent boundaries $\partial_{\text{Aff}} R_1$ and $\partial_{\text{Aff}} R_2$ are empty,
- their intersection $R_1 \cap R_2$ is non-empty and is a match singleton $R_1 \cap R_2 = \{m\}$.

We will show that R_1 and R_2 are not maximal by showing that the union $R_1 \cup R_2$ is also an affine-consistent region. Let us write for $i \in \{1, 2\}$,

$$R_i = \{a_i, b_i, c_i, m\}.$$

Let us show $t_1 = \{a_1, b_1, c_1\} \in \mathcal{E}(m)$ for all $m' \in R_1 \cup R_2$. Indeed, we observe the following facts.

- $t_1 \in \mathcal{T}(m) \subseteq \mathcal{E}(m)$ because R_1 is an affine-consistent quadruple.
- For any $m' \in R_1 \setminus \{m\}$, $t_1 \in \mathcal{T}^2(m') \subseteq \mathcal{E}(m')$. Indeed, let us just show this, for example when $m' = a_1$. Then the triple

  $$t_{a_1} \overset{\text{def}}{=} \{b_1, c_1, m\} \in \mathcal{T}(a_1).$$

  Therefore, as $t_1 \in \mathcal{T}(m)$,

  $$t_1 \in \mathcal{T}^2(a_1) \subseteq \mathcal{E}(a_1),$$

  by using Equation (4.21) and observing that $m \in t_{a_1}$.

- For any $m' \in R_2 \setminus \{m\}$, $t_1 \in \mathcal{T}^2(m') \subseteq \mathcal{E}(m')$. We reuse the same technique as above. Let us show this, for example when $m' = a_2$. Then the triple

$$t_{a_2} \stackrel{\text{def}}{=} \{b_2, c_2, m\} \in \mathcal{T}(a_2).$$

  Therefore, as $t_1 \in \mathcal{T}(m)$,

$$t_1 \in \mathcal{T}^2(a_2) \subseteq \mathcal{E}(a_2).$$

- Finally, $t_1 \in R_1^3 \subseteq (R_1 \cup R_2)^3$. Therefore, the following holds

$$\left(\bigcap_{m' \in R_1 \cup R_2} \mathcal{E}(m') \right) \cap (R_1 \cup R_2)^3 \neq \emptyset.$$

  $R_1 \cap R_2$ is thus affine-consistent by definition.

As a result, we have shown that $R_1 \cup R_2$ is affine-consistent.

4.7 Problem Formulation

The confidence in the matches of a maximal affine-consistent region is related to the region cardinality. The larger the cardinality, the more likely correct the region is. Generally, a maximal affine-consistent region with large cardinality also has a large spatial extent. However, correct matches can be in maximal affine-consistent regions of small size, e.g., because feature displacements are not homogeneous but related to the specific depths, or they are occluded image parts. But we found it to be quite rare in our experiments. Therefore, we choose to eliminate small regions by using an absolute threshold on their cardinality.

We can now state our feature matching problem (in its first variant) as follows:

Problem Formulation:
Find the maximum number of matches in $\mathcal{M}$ that are in affine-consistent regions of sufficient size.

This comes down to finding maximal affine-consistent regions of sufficient size. And algorithmically, we have seen that maximal affine-consistent regions can be generated from seed triples.

Ambiguity Freedom. Different tasks have different requirements regarding match ambiguity. For instance, whereas repeated pattern detection overtly calls for ambiguous matches, scene tracks used for estimating camera calibration parameters require unambiguous matches. We can define other variants of our feature matching problem that additionally require ambiguity-freedom.

In particular, in matching for camera calibration, a second variant consists in finding the largest number of maximally consistent regions that are ambiguity-free, because the

goal is to resolve the ambiguity globally, i.e., a feature x in image 1 can only be matched to one y in image 2. On the contrary, for pattern matching, a third variant of the problem only requires that each region of this set of regions be ambiguity-free, without imposing global uniqueness. The problem formulation is well-suited for pattern detection, e.g. window detection tasks.

More formally,

- a match (x, y) is *unambiguous in* M iff for all $(x', y') \in M \setminus \{(x, y)\}$, $x \neq x'$ and $y \neq y'$;
- a region R is *ambiguity-free* iff for any match $m \in R$, m is unambiguous in $R \setminus \{m\}$, i.e., equivalently, iff for any two matches (x, y) and (x', y') in R, then $x \neq x'$ and $y \neq y'$;
- and a set of regions $\mathcal{R}$ is *ambiguity-free* iff R is ambiguity-free for all $R \in \mathcal{R}$.

Chapter 5

Sequential Fourth Order Match Propagation

Due to the highly combinatorial nature of the optimization problem described in Chapter 4, we propose an approximate algorithm and a set of heuristics that efficiently finds a large number of matches in $\mathcal{M}$ that are in maximal affine-consistent regions of sufficient size, possibly ambiguity-free. Although consistent regions grown with our approximate algorithm cannot be guaranteed to be maximal, our experiments show that our algorithm still yields nearly maximal affine-consistent regions. (cf. Chapter 6). We first describe the general structure of the algorithm, and then develop pruning heuristics.

Contents

5.1 Single Match Propagation Algorithm

The algorithm follows a region growing scheme. Given an initial region consisting of a triple of potential matches, we iteratively add more matches into the region provided they are geometrically consistent with some triple of matches already in the region. When no

Algorithm 5.1 Region growing from a seed match m_1.

1: **procedure** GROWREGION(m_1, K, ρ_0)
2: $t \leftarrow$ ConstructTriple(m_1, K, ρ_0) *// cf. algo 5.3*
3: $R \leftarrow t$ *// Initialize R with seed*
4: $\partial R \leftarrow \bigcup_{m \in t} \left(\widehat{\mathcal{N}}_{K,\rho_0}(m) \setminus R \right)$ *// Initialize region boundary*
5: keep ∂R sorted by increasing distrust score
6: **while** $\partial R \neq \emptyset$ **do** *// Loop while the region boundary ∂R is not empty*
7: $\partial_{\mathrm{Aff}} R \leftarrow \emptyset$
8: *// Lines 8-11 is detailed in Algorithm 5.2*
9: **for each** $(m, t) \in \partial R \times R^3$ **do**
10: **if** $\mathrm{Aff}(m, t)$ **then**
11: $\partial_{\mathrm{Aff}} R \leftarrow \partial_{\mathrm{Aff}} R \cup \{m\}$
12: **end if**
13: **end for**
14: *// Check if region R terminated its growth*
15: **if** $\partial_{\mathrm{Aff}} R = \emptyset$ **then**
16: **break** *// Leave the loop to return the region R*
17: **end if**
18: *// Grow R*
19: $R \leftarrow R \cup \partial_{\mathrm{Aff}} R$
20: *// Update the region boundary ∂R*
21: **for** $m \in \partial_{\mathrm{Aff}} R$ **do**
22: $\partial R \leftarrow \partial R \setminus \{m\}$
23: $\partial R \leftarrow \partial R \cup \left(\widehat{\mathcal{N}}_{K,\rho_0}(m) \setminus R \right)$
24: **end for**
25: **end while**
26: **return** R *// R is maximal affine-consistent*
27: **end procedure**

more match can be added, the region is considered as valid iff it is large enough. More regions can be grown by re-running the algorithm on the remaining potential matches. See algorithm 5.1 for details.

Besides, if unambiguity is required, any match (x, y) is checked for ambiguity before being added to a growing region R (line 19 of Algorithm 5.1). If there already is a match (x, y') or (x', y) in R, then (x, y) is removed from the remaining potential matches and associated to R, but without contributing to $|R|$.

The key ingredients of the algorithm are additional heuristics for growing the regions, that prevent a combinatorial explosion and only explore a limited number of pertinent cases, most likely matches being tried first. They enable a selective evaluation of concistency checks, in particular the shape consistency which can be computationally intensive. They are presented in the following.

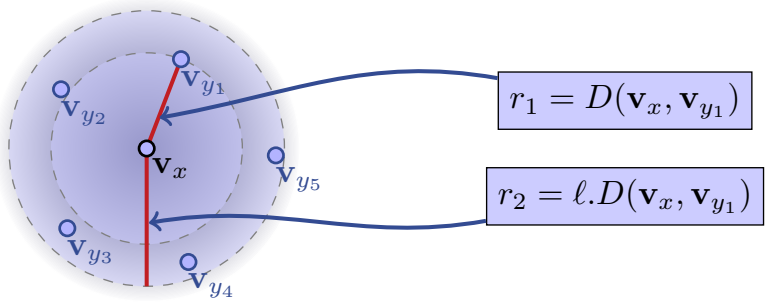


Figure 5.1: Illustration of the distrust score L. Nearest neighbors $\{\mathbf{v}_{y_i}\}_{i=2}^{5}$ in $\mathcal{Y}$ to $\mathbf{v}_x$ lie in the annulus with center $\mathbf{v}_x$, inner radius $r_1 = D(\mathbf{v}_x, \mathbf{v}_{y_1})$ and outer radius $r_2 = \ell.D(\mathbf{v}_x, \mathbf{v}_{y_1})$. As a reminder: $L(x, y_1) \leq 1$ and $1 \leq L(x, y_i) \leq \ell$ for $i \geq 2$.

5.2 Ordering and Limiting Potential Matches.

Matches (x, y) are ordered by increasing distrust score, defined as follows. Let D be a distance in the descriptor space, e.g., Euclidean distance for SIFT. For a descriptor $\mathbf{v}_x \in \mathcal{V}_\mathcal{X}$, let $\mathbf{v}_{y_1}, \mathbf{v}_{y_2} \in \mathcal{V}_\mathcal{Y}$ be respectively its nearest neighbor (1-NN) and its second nearest neighbor (2-NN). The distrust score (or Lowe score (LOWE 2004)) of match $m = (x, y)$ is defined as

$$L_{\mathcal{X}\to\mathcal{Y}}(x, y) = \frac{D(\mathbf{v}_x, \mathbf{v}_{y_1})}{D(\mathbf{v}_x, \mathbf{v}_y^2)} \quad \leq 1 \tag{5.1}$$

The smaller the score $L_{\mathcal{X}\to\mathcal{Y}}(m)$ is, the less ambiguous match m is. Usually, a set of reliable matches is obtained with matches m such that $L_{\mathcal{X}\to\mathcal{Y}}(m) \leq \ell$. Typically, ℓ ranges in $[0.6; 0.8]$. However, doing so discards ambiguous matches. To avoid it, the distrust score is extended as follows:

$$L_{\mathcal{X}\to\mathcal{Y}}(x, y) = \begin{cases} \frac{D(\mathbf{v}_x, \mathbf{v}_y)}{D(\mathbf{v}_x, \mathbf{v}_{y_2})} \leq 1 & \text{if } \mathbf{v}_y = \mathbf{v}_{y_1} \\ \frac{D(\mathbf{v}_x, \mathbf{v}_y)}{D(\mathbf{v}_x, \mathbf{v}_{y_1})} \geq 1 & \text{if } \mathbf{v}_y \neq \mathbf{v}_{y_1} \end{cases} \tag{5.2}$$

It quantifies an ambiguous match (x, y) by the relative proximity of $\mathbf{v}_y$ with respect to its 1-NN. We actually use the symmetric distrust score defined as

$$L(m) = \min\Big(L_{\mathcal{X}\to\mathcal{Y}}(m)\ , L_{\mathcal{Y}\to\mathcal{X}}(m)\Big). \tag{5.3}$$

Note that using $\max$ rather than $\min$ would delay too much the analysis of 1-to-many ambiguities. In our work, $\mathcal{M}$ is the set of matches m such that $L(m) \leq \ell$, where ℓ can be greater than 1. Consequently, $\mathcal{M}$ is much more ubiquitous than with the usual Lowe criterion, for a better support of repetitive patterns.

The distrust score is illustrated in fig. 5.1.

With this score, ambiguous matches tend to be ordered after unambiguous matches. Moreover, the search may be efficiently pruned by putting an upper bound on distrust.

Algorithm 5.2 Exhaustive Local Search for Region Growing.

```
1:  procedure LocalSearch(R, ∂R)
2:      keep ∂R sorted by increasing distrust score
3:      for match m = (x, y) ∈ ∂R do
4:          for triple t ∈ (N̂_{K,ρ0}(m) ∩ R)^3 do
5:              if t satifies Equations (5.4) and (5.5) and Aff(m, t) then
6:                  return (m, t)
7:              end if
8:          end for
9:      end for
10:     return ∅
11: end procedure
```

Ambiguous matches with larger distrust are excluded right from the start and are never considered for seeding or growing a region. They thus do not appear in the final selection of matches $\mathcal{R}$. The resulting match ordering and filtering is used to always select the best match candidates first, either for constructing a region seed, i.e., a match quadruple, or for growing a region. (Also, because of the greedy strategy, we only consider matches that are not currently assigned to a region.)

5.3 Local Search for Region Growing

When trying to grow a region R with a match $m = (x, y) \in \partial R$ (line 8 of Algorithm 5.1), we look for specific triple of matches $(m', m'', m''') \in (\hat{\mathcal{N}}_{K,\rho_0}(m) \cap R)^3$ such that the triangles $(\mathbf{x}', \mathbf{x}'', \mathbf{x}''')$ and $(\mathbf{y}', \mathbf{y}'', \mathbf{y}''')$ are such that

$$\begin{cases} x' \neq x'' \neq x''' \neq x' \\ y' \neq y'' \neq y''' \neq y' \end{cases}, \tag{5.4}$$

and that their two most acute angles (α, β) satisfy the following conditions:

$$\begin{cases} \alpha < \beta, \\ \alpha > \theta_1 \ , \ \beta > \theta_2 \end{cases} \tag{5.5}$$

The nondegeneracy conditions in Equations (5.4) and (5.5) ensures that the local affinity around match m can be properly estimated. We call such match triple (m', m'', m''') nondegenerate. We empirically set in all our experiments $\theta_1 > 15°$ and $\theta_2 > 25°$. Specifically, line 7 of algorithm 5.1 actually calls algorithm 5.2 to iterate over all triples of matches that provides affine consistency to candidate match m.

5.4 Sidedness Constraint

Optionally, we can also introduce a sidedness constraint that, experimentally, is very efficient in pruning the search and more efficient than the one in FERRARI et al. (2004).

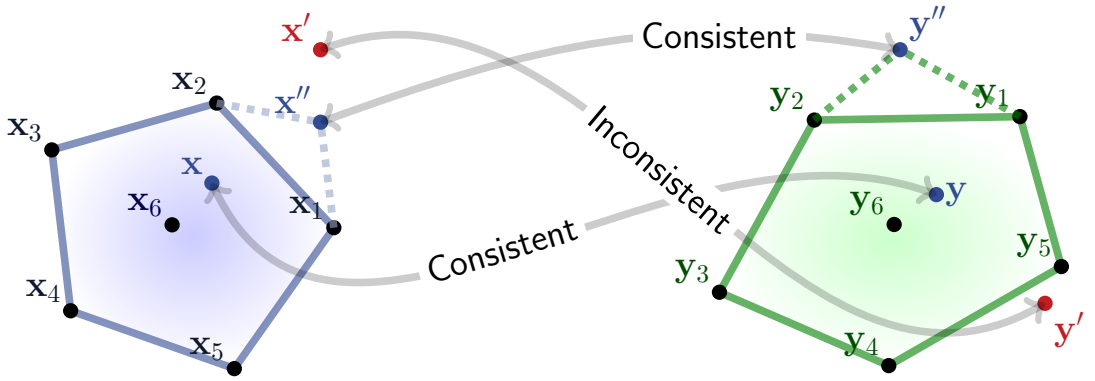


Figure 5.2: Geometric consistency with respect to sidedness constraints.

The general idea is that if $m_1 = (x_1, y_1)$ and $m_2 = (x_2, y_2)$ are good matches, then the directed lines $\overrightarrow{\mathbf{x}_1\mathbf{x}_2}$ and $\overrightarrow{\mathbf{y}_1\mathbf{y}_2}$ should define corresponding half spaces. More formally, given two points $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$, the half space on the left of $\overrightarrow{\mathbf{uv}}$ is $E(\mathbf{u}, \mathbf{v}) = \{\mathbf{w} \in \mathbb{R}^2 \mid \det(\mathbf{v} - \mathbf{u}, \mathbf{w} - \mathbf{u}) > 0\}$. A match (x, y) is *side-consistent* w.r.t. matches $(x_1, y_1), (x_2, y_2)$ iff $\mathbf{x} \in E(\mathbf{x}_1, \mathbf{x}_2) \Leftrightarrow \mathbf{y} \in E(\mathbf{y}_1, \mathbf{y}_2)$. When evaluating a match candidate m for growing a region R, m can be excluded if there are $m_1, m_2 \in R$ such that m is not side-consistent w.r.t. matches m_1, m_2.

For robustness, the sidedness consistency applies only to matches (x, y) such that $\mathbf{x}$ (resp. $\mathbf{y}$) is not to close to line $\overrightarrow{\mathbf{x}_1\mathbf{x}_2}$ (resp. $\overrightarrow{\mathbf{y}_1\mathbf{y}_2}$). This prevents spurious match rejections caused by non-affine transformations or due to the imprecision of feature localization. For efficiency, we limit consistency checks for a region $R = (x_i, y_i)_{1 \leq i \leq n}$ to the contour edges of the convex hulls associated respectively to $(\mathbf{x}_i)_{1 \leq i \leq n}$ and $(\mathbf{y}_i)_{1 \leq i \leq n}$. We also impose that at any step of the region growing, the contour vertices of the convex hull of the points already matched in $\mathcal{X}$ should correspond to the contour vertices of the convex hull of the points already matched in $\mathcal{Y}$. Fig. 5.2 illustrates these two points.

The sidedness-checking procedure in FERRARI et al. (2004) operates over all pairs of matches in a given region R, and thus performs $O(|R|^2)$ line checks. Our sidedness check operates only on the perimeter of R, rather than the whole area. The number of line checks is thus linear in the number of vertices on the contour of the convex hull, which is in practice $O(\sqrt{|R|})$.

5.5 Multiple Match Propagations Run Sequentially

Until now we have focused on the growth of a single region, we now describe our strategy to find all the maximal affine-consistent regions. First, to avoid a costly combinatorial search for the search of initial match triple, we propose to use Algorithm 5.3 to build

Algorithm 5.3 Triple construction from seed match m_1.

procedure CONSTRUCTTRIPLE(m_1, K, ρ_0)
  $t \leftarrow \emptyset$
  pick match $m_2 \in \widehat{\mathcal{N}}_{K,\rho_0}(m_1) \setminus \{m_1\}$ with the best distrust score.
  $C \leftarrow \left(\widehat{\mathcal{N}}_{K,\rho_0}(m_1) \cup \widehat{\mathcal{N}}_{K,\rho_0}(m_2)\right) \setminus \{m_1, m_2\}$
  sort C by increasing distrust score
  for $m_3 \in C$ **do**
    $t \leftarrow (m_1, m_2, m_3)$
    if triple t satisfies Equations (5.4) and (5.5) **then**
      return t
    end if
  end for
  return $\emptyset$
end procedure

Algorithm 5.4 Multiple region growing.

procedure GROWMULTIPLEREGIONS($\mathcal{M}, N, \tau, K, \rho_0$)
  $\mathcal{R} \leftarrow \emptyset$ *// Initialize set of maximal affine-consistent regions*
  for $n = 1, \ldots, N$ **do**
    Select the next best match $m \in \mathcal{M}$ such that $m \notin \bigcup_{R \in \mathcal{R}} R$.
    $R \leftarrow$ GROWREGION(m, K, ρ_0) *// cf. Algorithm 5.1*
    if $|R| > \tau$ **then** *// Add region R if it has reached critical cardinal size.*
      $\mathcal{R} \leftarrow \mathcal{R} \cup R$
    end if
  end for
  return $\mathcal{R}$
end procedure

an affine-consistent triple that is likely to be a good seed as follows. Given a match m_1, matches m_2 and m_3 are in the neighborhood of m_1 such that they have the lowest distrust score and the triple (m_1, m_2, m_3) is nondegenerate. This simple strategy turns out to be a very powerful heuristics in practice. In particular, even when dealing with challenging viewpoint change, this heuristic search still finds very good triples from which a maximal affine-consistent region can be grown.

Then we propose Algorithm 5.4 to grow multiple regions. Essentially, Algorithm 5.4 sequentially tries to grow sequentially from most reliable seed matches. If a region R has grown successfully, then we filter out all matches in the region R so that they are not used as potential seed matches. Hence, we avoid re-growing the same region R and try to grow other regions instead.

5.6 Efficient Approximate Algorithms

In the previous sections, we have presented algorithms that implements our match propagation procedure. However these are not efficient because both (1) the exhaustive local search of affine-consistent quadruple in Algorithm 5.2, and (2) the match neighborhood function $\widehat{\mathcal{N}}_{K,\rho_0}$ are computationally expensive.

As a result, let us see that Algorithm 5.3 becomes Algorithm 5.6, which basically just replaces the computationally expensive symmetric match neighborhood $\widehat{\mathcal{N}}_{K,\rho_0}$ by the cheaper nonsymmetric one $\mathcal{N}_{K,\rho_0}$.

Next, we will thus propose two robust and efficient approximating algorithms in the following.

5.6.1 Approximate Local Search

The LocalSearch in Algorithm 5.2 is in the worst case $O(K^3)$ because a brute-force search is used to find an affine-consistent quadruple. In practice, it is computationally costly. Its computational burden can be reduced significantly by approximating Algorithm 5.2 by Algorithm 5.7 with little performance degradation. In particular, for a given $m \in \partial R$, Algorithm 5.7 does not enumerate all possible valid triple $t \in \mathcal{T}(m) \cap R^3$, it only picks one nondegenerate triple $t \in (\widehat{\mathcal{N}}_{K,\rho_0}(m) \cap R)^3$ and we check if the predicate $\mathrm{Aff}(m,t)$ holds for this triple t only, otherwise the local search of triple is stopped. It is still $O(K^3)$ in the worst case but much more efficient in practice than the brute-force search of Algorithm 5.2.

Another consequence of using an approximate search in Algorithm 5.7 is that the maximality of the grown region will not be guaranteed anymore with Algorithm 5.1.

Besides, grown regions using Algorithm 5.1 can then overlap and we have to merge regions in such a case. Specifically, let R and R' be two affine-consistent regions obtained with Algorithm 5.5. If they have a common a triple $t \in \mathcal{T}(m)$ for some $m \in R \Delta R'$ such that

$$t \in (R \cap R')^3, \tag{5.6}$$

then R and R' are merged.

To accomodate region overlaps, Algorithm 5.1 and Algorithm 5.4 are replaced by Algorithm 5.5 and Algorithm 5.9. In the following, we give a brief algorithmic idea to take into account region overlaps. Let us consider a set of grown regions $(R_i)_{1 \leq i \leq N}$ and then a region R_{N+1} is being grown. As described in Algorithm 5.5, whenever R_{N+1} overlaps with a R_i during its growth in the sense of Equation (5.6), then we only add matches $m \in \partial R \setminus \left(\bigcup_{i=1}^{I} R_i \right)$ (I is the set of indices i such that R overlaps with R_i) when the region boundary update and the set of overlapping indices I is updated with Algorithm 5.8.

When R_{N+1} has terminated its growth, it is merged with the concerned $(R_i)_{i \in I}$ as described in Algorithm 5.9.

In summary, the algorithms presented previously are still implementable but practically inefficient and we consequently use Algorithms 5.5, 5.6, 5.8, 5.7 and 5.9 instead.

We also observe that Algorithms 5.5 is partially parallelizable, leading to significant computational speed-up in practice.

5.6.2 Choice of Match Neighborhood Function

We will now see that the match neighborhood function $\widehat{\mathcal{N}}_{K,\rho_0}$ remains the main computational bottleneck in the algorithm although the computation of $\widehat{\mathcal{N}}_{K,\rho_0}(m)$ is performed once only when we update the region boundary ∂R (see Algorithm 5.5).

By construction, the match neighborhood function $\widehat{\mathcal{N}}_{K,\rho_0}$ enjoys the property of being symmetric, i.e. $m \in \widehat{\mathcal{N}}_{K,\rho_0}(m') \iff m' \in \widehat{\mathcal{N}}_{K,\rho_0}(m)$, contrary to the nonsymmetric match neighborhood function $\mathcal{N}_{K,\rho_0}$. However, $\widehat{\mathcal{N}}_{K,\rho_0}$ requires that every $\mathcal{N}_{K,\rho_0}(m)$ are computed first before we try to grow regions using Algorithm 5.4. In practice, the match neighborhood function $\widehat{\mathcal{N}}_{K,\rho_0}$ is found to be computationally prohibitive, when the set of matches $\mathcal{M}$ becomes very large and ambiguous. Indeed, it is more advantageous to avoid computing match neighborhood $\mathcal{N}_{K,\rho_0}(m)$ for spurious matches $m \in \mathcal{M}$ as they will not be used in the match propagation anyway. Therefore, computing $\mathcal{N}_{K,\rho}(m)$ on the fly and memorizing it in a cache is practically more efficient and it is observed that $\mathcal{M}$ is very contaminated and ambiguous in practice.

We will see experimentally, that using the match neighborhood function $\mathcal{N}_{K,\rho_0}$ instead of $\widehat{\mathcal{N}}_{K,\rho_0}$ does not degrade the matching performance.

5.7 Implementation

Data structures To make the match propagation simple, we adopt the following implementation. A region R and a region boundary ∂R are basically ordered sets of match indices based on a red-black tree data structure. This makes insertion, removal and searching efficient. These operations have a logarithmic cost $O(|R|)$ with respect to the region size $|R|$ or boundary size $|\partial R|$.

Match Neighborhood Query Implementation The computation of match neighborhood $\mathcal{N}_{K,\rho_0}$ relies on fast point query search using 2D-trees, which are respectively built from the point location of sets of features $\mathcal{X}$ in image 1 and sets of features $\mathcal{Y}$ in image 2. We maintain two tables of matches $T_{\mathcal{X}}$ and $T_{\mathcal{Y}}$. They are respectively tables for which we retrieve for a given position $\mathbf{x}$ or $\mathbf{y}$ all corresponding matches $(x,.)$ and $(.,y)$ in constant time. The construction of such tables is respectively done in $O(|\mathcal{X}|\log(|\mathcal{X}|))$ and $O(|\mathcal{Y}|\log(|\mathcal{Y}|))$, using a quick sort.

In the following, let us analyze the complexity with respect to matches $(x,.)$ only as the reasoning is symmetric. The computation of $\mathcal{N}_K^{\mathcal{X}}(x)$ (cf. Definition 4) are respectively $O(K|\mathcal{X}|\log|\mathcal{X}|)$. For each feature $x' \in \mathcal{N}_K^{\mathcal{X}}(x)$, we collect matches $(x',.)$ which are at most $K \times B$ where B denotes the maximum degree of matching ambiguity. The query costs $O(K(|\mathcal{X}|\log|\mathcal{X}| + B))$. Taking also into consideration matches $(.,y)$, we have the query costs in total $O(\mathcal{N}_{K,\rho_0}(m)) = O(K(|\mathcal{X}|\log|\mathcal{X}| + |\mathcal{Y}|\log|\mathcal{Y}| + B))$

Introducing

$$F \stackrel{\text{def}}{=} \max(|\mathcal{X}|, |\mathcal{Y}|),$$

we have

$$O(\mathcal{N}_{K,\rho_0}(m)) = O(K(F \log F + B))$$

Note that we can estimate $B = O(|\mathcal{M}|/F)$.

We deliberately not give a complexity analysis of Algorithm 5.5 because the worst case scales very badly with respect to $\mathcal{M}$ and F, but corresponds to a situation that never happens in practice. We prefer to provide experimental evidence on the scalability of our method.

5.8 Experimental Validation

5.8.1 Choice of Match Neighborhood

We choose to validate the choice of match neighborhood on only one dataset of MIKOLAJCZYK et al. (2005) which is *Wall* for all features

$$f \in \mathcal{F} = \{\text{DoG+SIFT}, \text{Harris-Affine+SIFT}, \text{Hessian-Affine+SIFT}, \text{MSER+SIFT}\}.$$

This dataset is representative enough of the results we obtained on the other datasets and tests the matching against increasing viewpoint changes. In addition, the dataset has a lot of repeated structures, which adds up to the challenge.

Computation of Initial Matches

As in Section 4.3 in Chapter 4, we detected all possible features without tuning detectors. However, for each image pair $(1, p)$ we choose to compute smaller sets of matches $\mathcal{M}_p$ defined by

$$\mathcal{M}_p \stackrel{\text{def}}{=} \{m \mid L(m) \leq 1\}.$$

whereas Section 4.3 collects all matches m such that $L(m) \leq 1.2$. Then we identify sets of inliers

$$\mathcal{I}_{p,i} = \{(x, y) \mid \|\mathbf{H}_p\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon_i\},$$

where $\mathbf{H}_p$ is the ground truth homography for image pair $(1, p)$ and $\varepsilon_i \in \{1.5, 5\}$ (in pixels) for $i \in \{1, 2\}$.

Evaluation of the Method with Multiple Neighborhoods

We run our method by attempting $N = 1000$ region growing and a region R is considered valid if $|R| \geq 7$. Denoting the set of regions returned by our method by $\mathcal{R}$, the matches m found by our method are those that satisfy $\exists R \in \mathcal{R}, m \in R$.

As we evaluate our method for different kinds of match neighborhood, We denote such set of matches found by our method $\mathcal{I}'_{\mathcal{W},K,\rho_0}$ where $\mathcal{W}$ is a neighborhood function

choosen between $\{\mathcal{N}_{.,.}(.), \widehat{\mathcal{N}}_{.,.}(.)\}$ (nonsymmetric and symmetric neighborhood functions) and (K, ρ_0) is a pair of parameter choosen in the list $\{(80, 0.5), (200, 0.3)\}$.

We evaluate the performance of our match propagation in terms of precision rate and recall rate for each image pair $(1, p)$ for $p \in \{2, \dots 6\}$. We recall that:

- the precision rate $P_{p,i,\mathcal{W},K,\rho_0}$ is the percentage of inliers found by our method with respect to the the total number of matches found by our method, i.e.,

$$P_{p,i,\mathcal{W},K,\rho_0} = \frac{|\mathcal{I}'_{\mathcal{W},K,\rho_0} \cap \mathcal{I}_{p,i}|}{|\mathcal{I}'_{\mathcal{W},K,\rho_0}|}$$

- the recall rate $R_{p,i,\mathcal{W},K,\rho_0}$ is the percentage of inliers returned by our method with respect to the total number of inliers, i.e.,

$$R_{p,i,\mathcal{W},K,\rho_0} = \frac{|\mathcal{I}'_{\mathcal{W},K,\rho_0} \cap \mathcal{I}_{p,i}|}{|\mathcal{I}_{\mathcal{W},K,\rho_0}|}$$

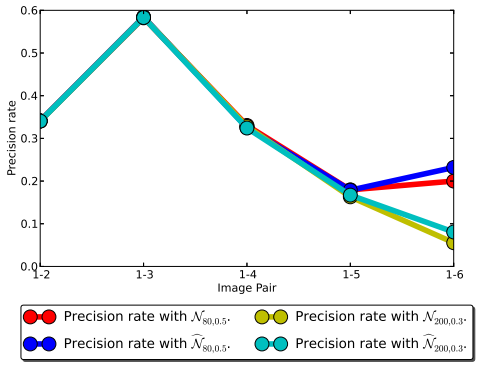
We show in Figures 5.3, 5.4, 5.5 and 5.6 plots of precision rate function $p \mapsto P_{p,i,\mathcal{W},K,\rho_0}$ and recall rate function $p \mapsto R_{p,i,\mathcal{W},K,\rho_0}$ which are function of the image pair $(1, p)$. We recall that an increasing index p indicates the difficulty of the viewpoint change.

Analysis of the results

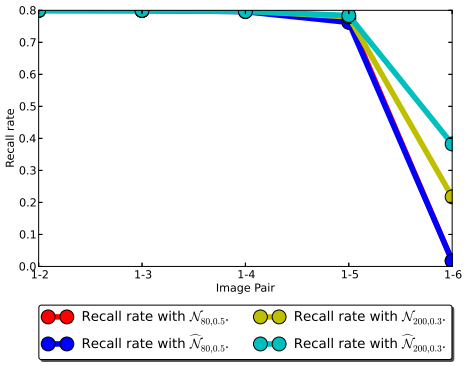
We can draw the following conclusions.

Robustness and efficiency of match neighborhoods computed on-the-fly. Figures 5.3, 5.4, 5.5 and 5.6 clearly show that there are practically no performance difference between $\mathcal{N}_{K,\rho_0}$ and $\widehat{\mathcal{N}}_{K,\rho_0}$ for all chosen pair (K, ρ_0) and for all image pairs $(1, p)$. Thus, the practical choice of neighborhood $\mathcal{N}_{K,\rho_0}$ is legitimately justified, which enables to compute on-the-fly match neighborhoods.

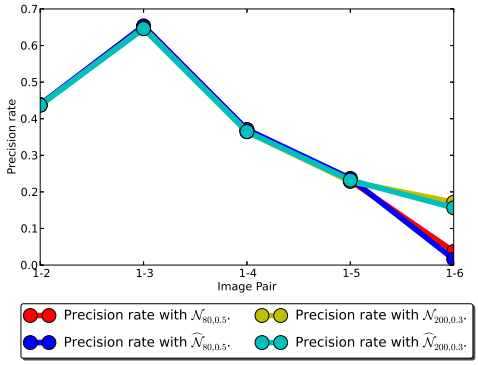
Smaller and more consistent neighborhoods is generally better. The results shows a smaller and more consistent neighborhoods, here $\mathcal{N}_{80,0.5}$ and $\widehat{\mathcal{N}}_{80,0.5}$ has practically as good performance as much larger and less consistent neighborhoods, here $\mathcal{N}_{200,0.3}$ and $\widehat{\mathcal{N}}_{200,0.3}$, in terms of precision and recall at least for image pairs $((1, p))_{2 \leq p \leq 4}$. Notice that the pair $(1, 4)$ corresponds to the median level of difficulty in the matching setting. Larger and less consistent neighborhoods always improve the recall rate especially on the most difficult image pairs $((1, p))_{5 \leq p \leq 6}$. However, the performance in terms of precision is less stable as it either degrades in half of the cases or improves in half of the cases. In practice, we choose $(K, \rho_0) = (80, 0.5)$. Indeed, first, the associated precision rate is better than with the pair $(K, \rho_0) = (200, 0.3)$ for all image pairs $(1, p)$ except the most difficult one $(1, 6)$. Second, the recall rate is lower but the difference of recall rate is not significant for all image pair $(1, p)$ except the most difficult one $(1, 6)$. To conclude, such choice is in practice benefic as smaller and more consistent neighborhoods are significantly faster to compute, which in turn significantly speeds up the match propagation.



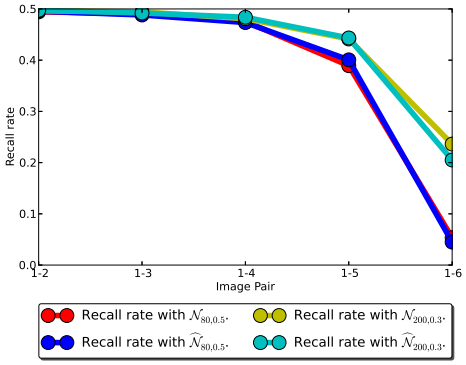
(a) Precision rate with DoG+SIFT matches.



(b) Recall rate with DoG+SIFT matches.



(c) Precision rate with Harris-Affine+SIFT matches.



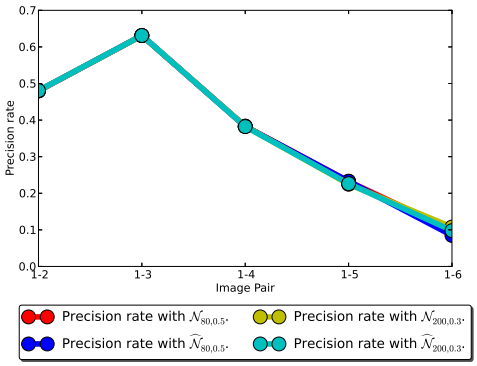
(d) Recall rate with Harris-Affine+SIFT matches.

Figure 5.3: Precision rate and recall rate with all match neighborhoods and $(K, \rho_0) \in \{(80, 0.5), (200, 0.3)\}$ w.r.t. $\mathcal{I}_{p,1} = \{(x, y) \mid \|\mathbf{H}_p\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon_1 = 1.5\}$ on the *Wall* dataset.

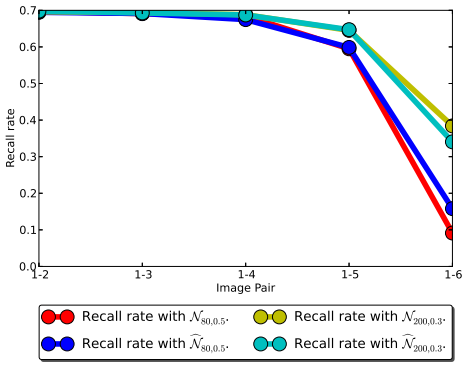
The match propagation is precise. Third, if we compare the precision rate in Figures 5.3, 5.4 to precision rates in in Figures 5.3, 5.4. We see that precision rates $P_{p,2,\mathcal{W},K,\rho_0}$ are much higher than $P_{p,1,\mathcal{W},K,\rho_0}$. This means that many matches (x, y) found by our method are such that their reprojection errors $\|\mathcal{H}\mathbf{x} - \mathbf{y}\|$ ranges within 5 pixels. This confirms that the quality of estimation of local affinities is rather precise. On the other hand, it can be argued that the image pairs are not really related by a homography since the wall is not perfectly planar in the dataset.

5.8.2 Triple of Matches vs Single Match

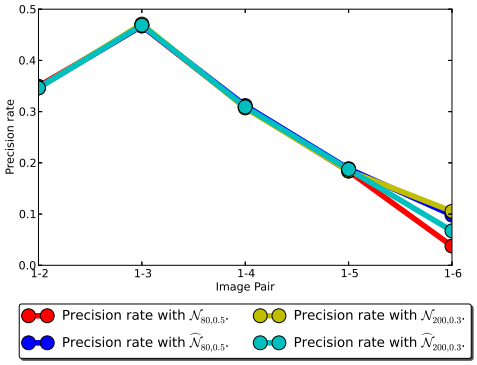
Our match propagation method does not rely very much on the quality of the affine shape adaptation proposed by MIKOLAJCZYK and SCHMID (2004) as shape adaptation is not as precise as expected. Indeed, we evaluate the interest of match triples (m, m', m'') to construct accurate and robust affinities vs resorting to single matches $m = (x, y)$,



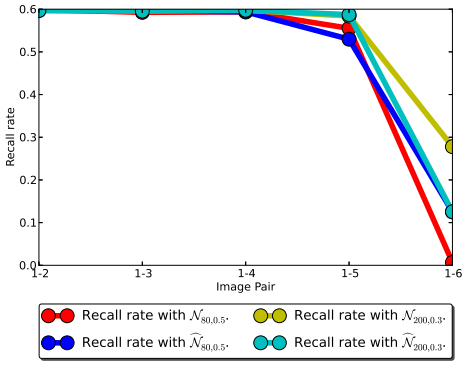
(a) Precision rate with Hessian-Affine+SIFT matches.



(b) Recall rate with Hessian-Affine+SIFT matches.



(c) Precision rate with MSER+SIFT matches.



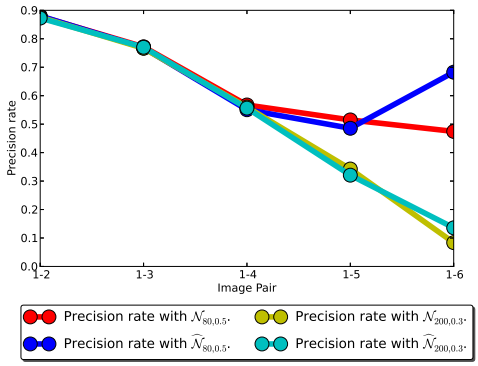
(d) Recall rate with MSER+SIFT matches.

Figure 5.4: Precision rate and recall rate with all match neighborhoods and $(K, \rho_0) \in \{(80, 0.5), (200, 0.3)\}$ w.r.t. $\mathcal{I}_{p,1} = \{(x, y) \mid \|\mathbf{H}_p\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon_1 = 1.5\}$ on the *Wall* dataset.

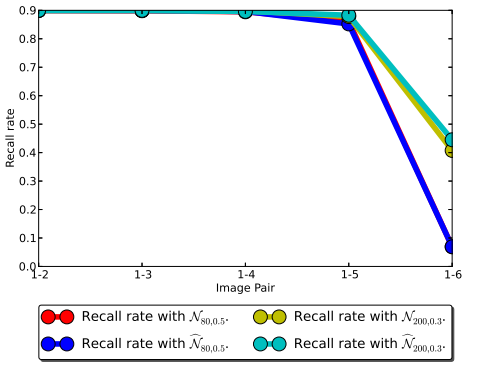
using the shapes $(\mathcal{S}_x, \mathcal{S}_y)$ and orientation $(\mathbf{o}_x, \mathbf{o}_y)$. Recall that only the affine transform estimated from a single match $(\mathbf{x}, \mathbf{y})$ is computed with Equation (4.3).

Experiments with MIKOLAJCZYK et al. (2005)'s dataset demonstrate that our region growing process performs consistently and significantly better when affinities ϕ are estimated with match triples. Each dataset consists of 6 images. For each dataset and for a given kind of feature f, we extract all feature points of type f. We match image 1 to images 2–6. Initial f-matches are obtained and ranked with Lowe's criterion. The distrust threshold is set to $\ell = 1$. On average, our region growing deals with $7,000$ to $28,000$ f-matches with an outlier proportion of at least 75%. We compare the performance of our region growing in terms of precision and recall for both variants: triples and single matches (cf. Figure 5.7).

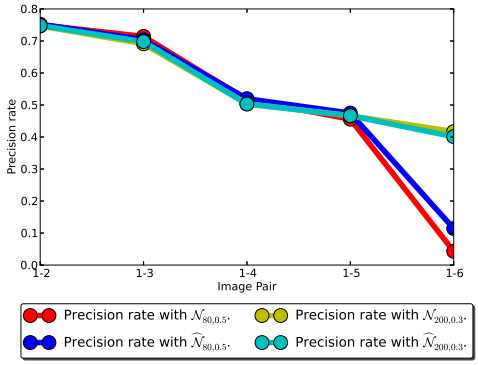
In Figure 5.7, precision rates clearly benefits from affinities computed with match



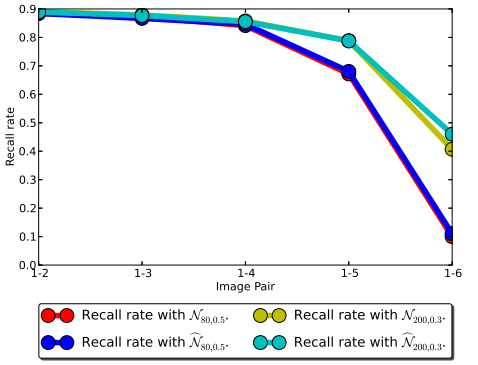
(a) Precision rate with DoG+SIFT matches.



(b) Recall rate with DoG+SIFT matches.



(c) Precision rate with Harris-Affine+SIFT matches.



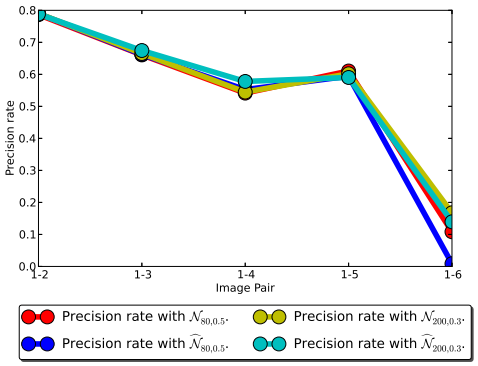
(d) Recall rate with Harris-Affine+SIFT matches.

Figure 5.5: Precision rate and recall rate with all match neighborhoods and $(K, \rho_0) \in \{(80, 0.5), (200, 0.3)\}$ w.r.t. $\mathcal{I}_{p,2} = \{(x, y) \mid \|\mathbf{H}_p\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon_2 = 5\}$ on the *Wall* dataset.

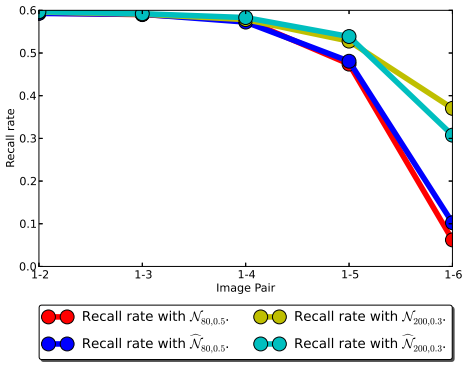
triples, as opposed to single matches. Besides, this is not at the expense of recall rates, that are comparable in the two approaches. We give two explanations for why precision rates for triples are consistently better. First, orientation estimation is often unstable; it remains sensitive to illuminations changes, blurring and compression. Second, local affinities estimated from the shape of DoG features are unsurprisingly inaccurate and consequently produces worse precision rates in general. Even when elliptic features are used, affinities estimated from triples still produce much better results in many cases.

5.8.3 Empirical Evidence of Scalability

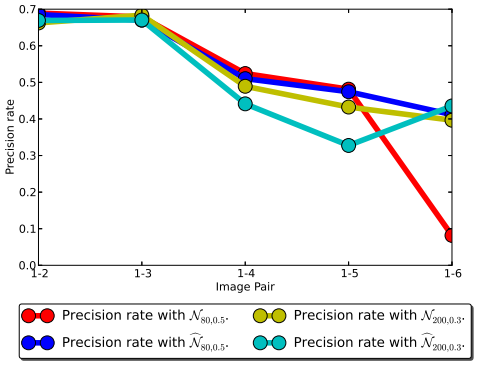
Our experiments show that our empirical complexity analysis is less than quadratic in N in practice, as illustrated in Table 5.1 on MIKOLAJCZYK et al. (2005)'s dataset. It is better, e.g., than DUCHENNE et al. (2011)'s tensor-based matching, which would be here $O(N^3 \log N)$ or $O(N^4 \log N)$, or CHO et al. (2009)'s agglomerative clustering, which is at



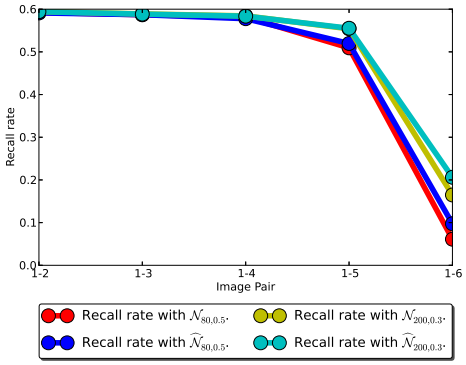
(a) Precision rate with Hessian-Affine+SIFT matches.



(b) Recall rate with Hessian-Affine+SIFT matches.



(c) Precision rate with MSER+SIFT matches.



(d) Recall rate with MSER+SIFT matches.

Figure 5.6: Precision rate and recall rate with all match neighborhoods and $(K, \rho_0) \in \{(80, 0.5), (200, 0.3)\}$ w.r.t. $\mathcal{I}_{p,2} = \{(x, y) \mid \|\mathbf{H}_p\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon_2 = 5\}$ on the *Wall* dataset.

least $O(|\mathcal{M}|^2)$.

A few words on the used heuristics.

Our algorithm is based on three heuristics:

- Considering most likely matches first. This is common to many computer vision algorithms. What is specific here is our original extension of the Lowe score to accommodate ambiguous matches.

- Considering only close matches (K nearest, to construct an affine-consistent quadruple). This strategy is also frequent for pruning unlikely configurations. It is used, e.g., in [9]. Others use a fixed-size neighborhood [14].

$\lvert\mathcal{M}\rvert$	3,000	10,000	30,000	100,000
DoG	2,676 0.21 s	5,342 0.42 s	7,027 0.70 s	7,027 1.36 s
MSER	1,585 0.84 s	2,283 1.11 s	2,283 1.46 s	2,283 1.83 s
Hessian-Affine	2,190 1.71 s	5,054 3.02 s	5,922 3.35 s	5,922 3.99 s
Harris-Affine	2,178 1.59 s	6,250 3.62 s	10,273 3.58 s	10,623 4.01 s

Table 5.1: For a given number $|\mathcal{M}|$ of potential matches, number N of corresponding features and average running time, on all image pairs of MIKOLAJCZYK et al. (2005)'s dataset.

- Assuming little local transformation (sidedness constraint to prune region growing). This is standard too. The specific topological filter we use here (with improvements) is also used in FERRARI et al. (2004).

Note that considering for growing a region R only matches that are among the K nearest of some match in R is not really a heuristic but a consistency constraint to ensure that growing a region is local, otherwise it makes little sense. While relying on heuristics is indeed undesirable, we consider that our use is moderate (compare, e.g., to FERRARI et al. (2004)) and not uncommon.

Algorithm 5.5 Practical region growing from a seed match m_1 using nonsymmetric neighborhood $\mathcal{N}_{K,\rho_0}$, with overlapping test using Equation (5.6).

procedure GROWREGION($m_1, K, \rho_0, \mathcal{R} = (R_i)_{1 \le i \le N}$) *// $\mathcal{R}$ is a set of affine-consistent regions*
  $I \leftarrow \emptyset$ *// Indices $i \in I$ such that R overlaps with $R_i \in \mathcal{R}$*
  $t \leftarrow$ CONSTRUCTTRIPLE(m_1, K, ρ_0) *// cf. Algorithm 5.6*
  $R \leftarrow t$ *// Initialize R with seed*
  // $\mathcal{N}_{K,\rho_0}(m)$ are computed on-the-fly and memorized in a cache
  $\partial R \leftarrow \bigcup_{m \in t} (\mathcal{N}_{K,\rho_0}(m) \setminus R)$
  keep ∂R sorted by increasing distrust score
  while $\partial R \neq \emptyset$ **do**
    $\partial_{\mathrm{Aff}} R \leftarrow \emptyset$
    // We denote $(m_i)_{1 \le i \le |\partial R|} \overset{def}{=} \partial R$ and let $(t_i)_{1 \le i | \partial R|}$ be a set of triples
    for $i \in \{1, \dots, |\partial R|\}$ **do**
      $t_i \leftarrow$ FINDNONDEGENERATETRIPLE($m, R, \partial R$) *// cf Algorithm 5.7*
    end for
    // Check if quadruple (m_i, t_i) is affine-consistent $i \in \{1, \dots, |\partial R|\}$
    for $i \in \{1, \dots, |\partial R|\}$ **do**
      if $(t_i \neq \emptyset) \wedge \mathrm{Aff}(m_i, t_i)$ **then**
        $\partial_{\mathrm{Aff}} R \leftarrow \partial_{\mathrm{Aff}} R \cup \{m_i\}$
        $I \leftarrow I \cup$ FINDOVERLAP($(m_i, t_i), \mathcal{R}$) *// cf Algorithm 5.8*
      end if
    end for
    // Update region R
    $R \leftarrow R \cup \partial_{\mathrm{Aff}} R$
    // Update region boundary ∂R and ensure that matches in overlapping
    // regions R_i are excluded
    for $m \in \partial_{\mathrm{Aff}} R$ **do**
      $\partial R \leftarrow \partial R \setminus \{m\}$
      $\partial R \leftarrow \partial R \cup \left(\mathcal{N}_{K,\rho_0}(m) \setminus \left(R \cup \bigcup_{i \in I} R_i \right) \right)$
    end for
  end while
  return (R, I)
end procedure

Algorithm 5.6 Triple construction from seed match m_1 using nonsymmetric neighborhood $\mathcal{N}_{K,\rho_0}$.

procedure ConstructTriple(m_1, K, ρ_0)
  $t \leftarrow \emptyset$
  Pick match $m_2 \in \mathcal{N}_{K,\rho_0}(m_1) \setminus \{m_1\}$ with the best distrust score.
  $C \leftarrow (\mathcal{N}_{K,\rho_0}(m_1) \cup \mathcal{N}_{K,\rho_0}(m_2)) \setminus \{m_1, m_2\}$
  sort C by increasing distrust score
  for $m_3 \in C$ **do**
    $t \leftarrow (m_1, m_2, m_3)$
    if triple t satisfies Equations (5.4) and (5.5) **then**
      return t
    end if
  end for
  return $\emptyset$
end procedure

Algorithm 5.7 Approximate local search of triples using nonsymmetric Neighborhood $\mathcal{N}_{K,\rho_0}$.

procedure FindNonDegenerateTriple($m, R, \partial R$)
  for match $m = (x, y) \in \partial R$ **do**
    for triple $t \in (\mathcal{N}_{K,\rho_0}(m) \cap R)^3$ **do**
      if t satifies Equation (5.4) and (5.5) **then**
        return t
      end if
    end for
  end for
  return $\emptyset$
end procedure

Algorithm 5.8 Check if quadruple q overlaps with set of affine-consistent regions $(R_i)_{1 \le i \le N}$

procedure FindOverlap($q, (R_i)_{1 \le i \le N}$)
  $I \leftarrow \emptyset$
  for $i \in \{1, \dots, N\}$ **do** *// Check for overlap with existing R_i*
    if $\exists t \in q^3 \cap R_i$ such that t satisfies Equation (5.6) **then**
      $I \leftarrow I \cup \{i\}$
    end if
  end for
  return I
end procedure

Algorithm 5.9 Multiple region growing using nonsymmetric neighborhood $\mathcal{N}_{K,\rho_0}$ with region merging.

procedure GROWMULTIPLEREGIONS($\mathcal{M}, N, \tau, K, \rho_0$)
  $\mathcal{R} \leftarrow \emptyset$
  for $n = 1, \ldots, N$ **do**
    Select the next best match $m \in \mathcal{M}$ such that $m \notin \bigcup_{R \in \mathcal{R}} R$.
    $(R, I) \leftarrow$ GROWREGION($m, K, \rho_0, \mathcal{R}$) *// cf. Algorithm 5.5*
    if $I \neq \emptyset$ **then**
      Merge all regions $(R_i)_{i \in I}$ that overlaps with R *// cf. Equation (5.6)*
    else
      if $|R| > \tau$ **then**
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$
      end if
    end if
  end for
  return $\mathcal{R}$
end procedure

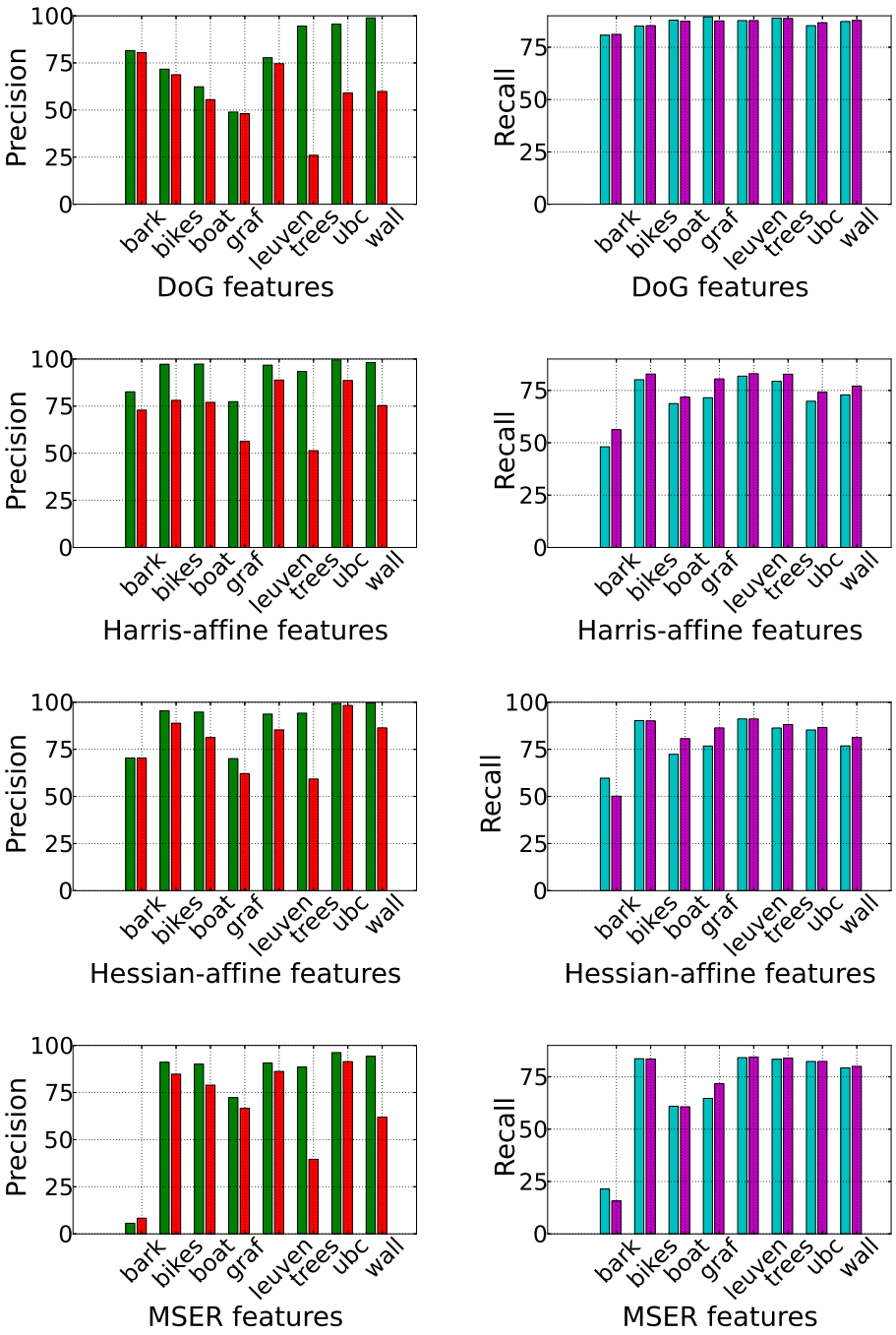


Figure 5.7: Precision (%) and recall (%) of region growing on MIKOLAJCZYK et al. (2005)'s dataset (pair 1-3). *Green*: precision rates with match triples, *Red*: precision rates with single matches. *Cyan*: recall rates with match triples, *Magenta*: recall rates with single matches.

Chapter 6

Matching Results with Fourth Order Match Propagation

In this chapter, we evaluate our method on several vision tasks and compare favorably w.r.t. the state-of-the-art. We use the same parameters for *all* our experiments, which indicates the stability of our method. The region growing parameters defined in Chapter 4 are set as $K = 80$ and $\rho_0 = 0.5$. The thresholds values set in predicate $\mathbb{P}_f$ can be found in Subsection 4.3.2 of Chapter 4. A region R is deemed valid iff $|R| \geq 7$. In the reported experiments, we processed on average $N = 5000$ points per image (sometimes tens of thousands) and 15 matches per point, i.e., $|\mathcal{M}| = 75000$ on average. The number of matches per point, up to 650 in our examples, depends on the ambiguity of the descriptor value. A complete region-growing trial can take up to 2 seconds, for a very large and dense region. For deformable object matching and calibration, we performed 1000 attempts to grow regions; for pattern detection, all possible seeds were explored.

Other experiments are described in Chapter 7.

Contents

6.1 Camera calibration

We tested a calibration task using Bundler (SNAVELY et al. 2008) as a black-box calibration system taking as input a set of matches. We used two pathological datasets described below: *Books* (31 images) and *Mars* (60 images). These sets of images are difficult to calibrate due to numerous ambiguities arising from repeated patterns.

We report (1) the number of calibrated cameras, (2) the mean squared reprojection error (MSRE) of 3D points in images (in pixels), (3) the number of consistent scene

Methods	#Cameras	Matching time
Ours	**20**/31	60 mn
LOWE (2004)	5/31	**5** mn
CHO et al. (2009)	7/31	2880 mn
FERRARI et al. (2004)	2/31	540 mn

Table 6.1: Calibration of the *Books* dataset with Harris-Affine+SIFT features.

Methods	#Cameras	MSRE	#Tracks
Ours	**30**/31	2.92×10^{-1}	4,875
LOWE (2004)	$5 - 20$/31	2.46×10^{-1}	2,574
CHO et al. (2009)	N/A	N/A	N/A
FERRARI et al. (2004)	N/A	N/A	N/A

Table 6.2: Calibration of the *Books* dataset with DoG+SIFT features. Concurrent methods (CHO et al. 2009; FERRARI et al. 2004) are not applicable here because they need elliptic features whereas DoG+SIFT features are circular.

tracks used for the estimation of camera parameters. (A scene track is a connected set of matching keypoints across multiple images and is deemed inconsistent if it contains more than one keypoint in the same image.)

6.1.1 Books Dataset

In the *Books* dataset shown in Figure 6.1, matching ambiguities arises from the uniform background and the chair, as well as the repeated letters on the covers. Calibration results with Harris-Affine+SIFT features show that we calibrate many more cameras than FERRARI et al. (2004)'s method, CHO et al. (2009)'s method, and a baseline consisting in a Lowe criterion (LOWE 2004).

We first use Harris-Affine features. The best Lowe threshold (here $\ell = 0.8$) leads to the calibration of only 5 cameras out of 31. Our method matches all possible image pairs (465) in one hour without parallelization. FERRARI et al. (2004)'s method takes over 9 hours and only matches 60 image pairs. CHO et al. (2009)'s method takes more than two days to complete (because it is MATLAB-based whereas other method are implemented in C++). The results are shown in Tables 6.1. Unlike FERRARI et al. (2004)'s method and CHO et al. (2009)'s, our matching algorithm provide a significant improvement over the baseline with Harris-Affine+SIFT features. Next, we used DoG+SIFT features, for which FERRARI et al. (2004)'s method and CHO et al. (2009)'s are not applicable because they need elliptic features whereas DoG+SIFT features are circular. As shown in Table 6.2, filtering correspondences with RANSAC calibrates 5 to 20 cameras. Our method performs significantly better as our method calibrates 30 cameras over 31.

Figure 6.2, 6.3a and 6.3b respectively shows the resulting 3D point cloud obtained with Bundler (SNAVELY et al. 2008), the 3D mesh reconstruction and textured reconstruction obtained from VU et al. (2012)'s 3D reconstruction pipeline. Note that the calibration and the 3D reconstruction shown in these figures are obtained from DoG+SIFT features.

Figure 6.1: The 31 images of the *Books* dataset.

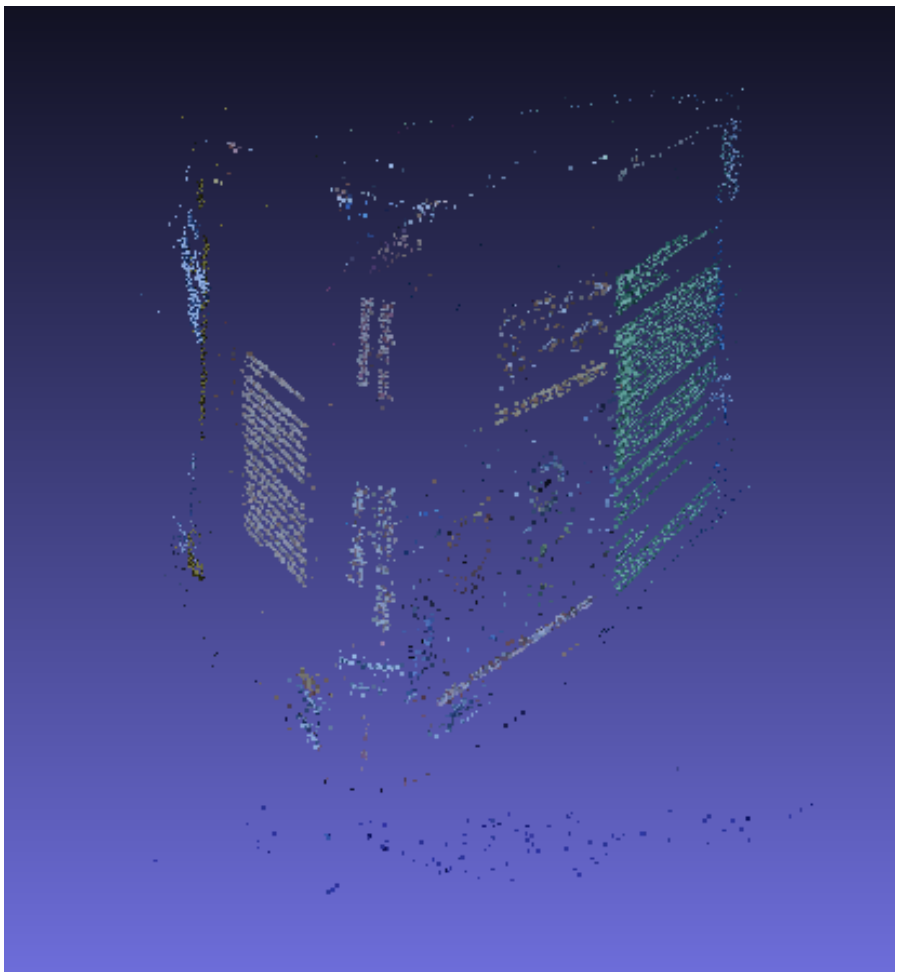
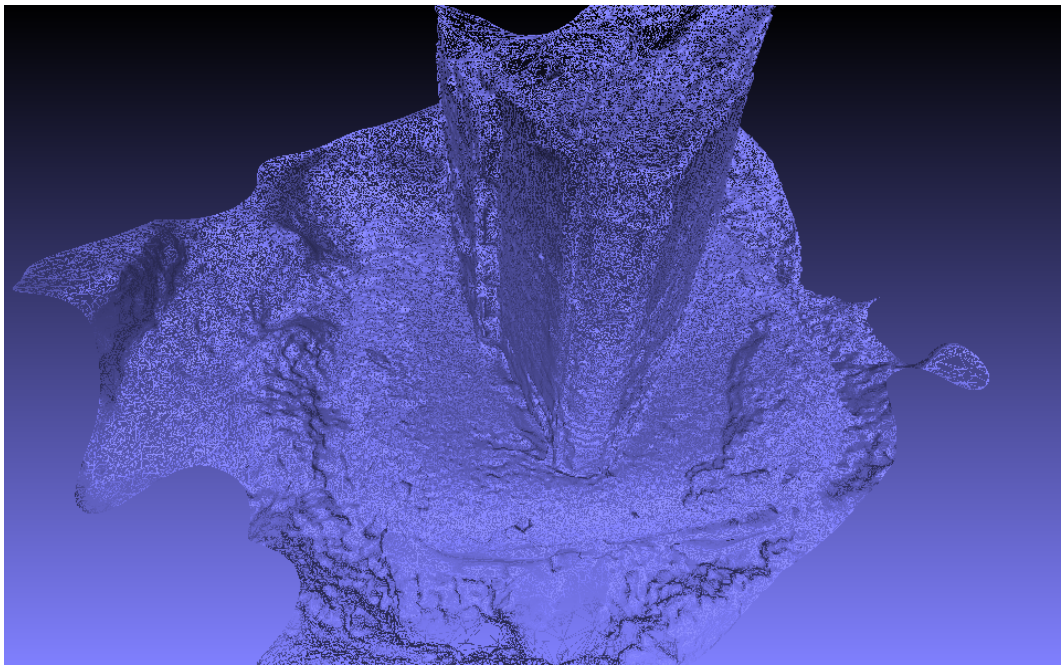
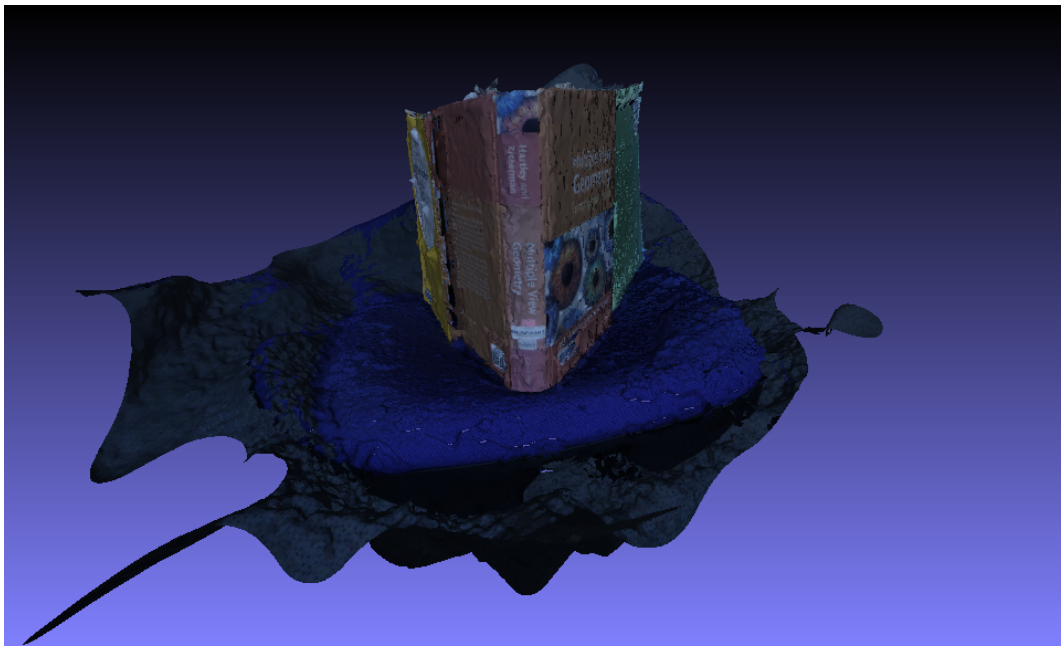


Figure 6.2: 3D point cloud resulting from the calibration of the *Books* dataset with DoG+SIFT features.



(a) 3D mesh reconstruction of the *Books* dataset after matching with our method.



(b) 3D textured reconstruction of the *Books* dataset after matching with our method.

Figure 6.3: 3D reconstruction of the *Books* dataset obtained from Vu et al. (2012)'s 3D reconstruction pipeline.

Methods	# Cameras	MSRE	# Tracks
Ours	**60/60**	5.00×10^{-2}	**75,966**
LOWE (2004) ($\ell = 0.3$)	22/60	$\mathbf{2.00 \times 10^{-2}}$	3,266
LOWE (2004) ($\ell = 0.4$)	30/60	3.13×10^{-2}	5,598
LOWE (2004) ($\ell = 0.5$)	**33/60**	47.50×10^{-2}	1,131
LOWE (2004) ($\ell = 0.6$)	28/60	5.68×10^{-2}	6,378
LOWE (2004) ($\ell = 0.7$)	28/60	6.47×10^{-2}	6,533
LOWE (2004) ($\ell = 0.8$)	28/60	8.27×10^{-2}	**6,667**
LOWE (2004) ($\ell = 0.9$)	28/60	8.84×10^{-2}	6,564

Table 6.3: Some images of the *Mars* dataset and calibration results.

6.1.2 Mars Dataset

The 60 images in the *Mars* dataset picture the Martian landscape. They were acquired by the rovers *Spirit* and *Opportunity*[1]. Some are shown in Figure 6.4.

With *Mars* (cf. Table 6.3), the landscape is very flat and the numerous rocks create ambiguous matches. Yet all 60 cameras are calibrated successfully with our method (with DoG+SIFT features), contrary to Lowe's criterion (LOWE 2004), which only leads to the calibration of half of the cameras. The mean squared reprojection error (MSRE, in pixels) of 3D points in images and the number of consistent scene tracks used for the estimation of camera parameters also compare favorably.

As reported in Table 6.3, all 60 cameras were calibrated successfully with our method. This is a significant improvement over the standard use of Bundler, which is only able to calibrate about half of the scene.

We show in Figure 6.5a and 6.5b the point clouds respectively obtained from (1) matches obtained with Lowe's criterion (LOWE 2004) and RANSAC (FISCHLER and BOLLES 1981) and (2) matches obtained with our method. Note that with the matching with Lowe's criterion and RANSAC, the calibration is unable to reconstruct the left part of the landscape whereas our method is able to reconstruct it completely. Moreover, for calibrated cameras in both cases, the right side of the loop is more complete with our method and the point cloud is much more dense.

Our implementation has actually been used in the calibration an 3D reconstruction chain of the winners of the *PRoVisG Mars 3D Challenge 2011*, from which this dataset is extracted.

For this dataset, all 1770 possible image pairs are considered in 3.5 hours using parallelization on a 8-core CPU Xeon 2.8GHz machine.

Figures 6.5b, 6.6 and 6.7 respectively shows the resulting 3D point cloud obtained with our feature matches and Bundler (SNAVELY et al. 2008), the 3D mesh reconstruction and textured reconstruction obtained from VU et al. (2012)'s 3D reconstruction pipeline. Note that the calibration and 3D reconstruction shown in these figures are obtained from

[1]PRoVisG Mars 3D Challenge, http://cmp.felk.cvut.cz/mars/

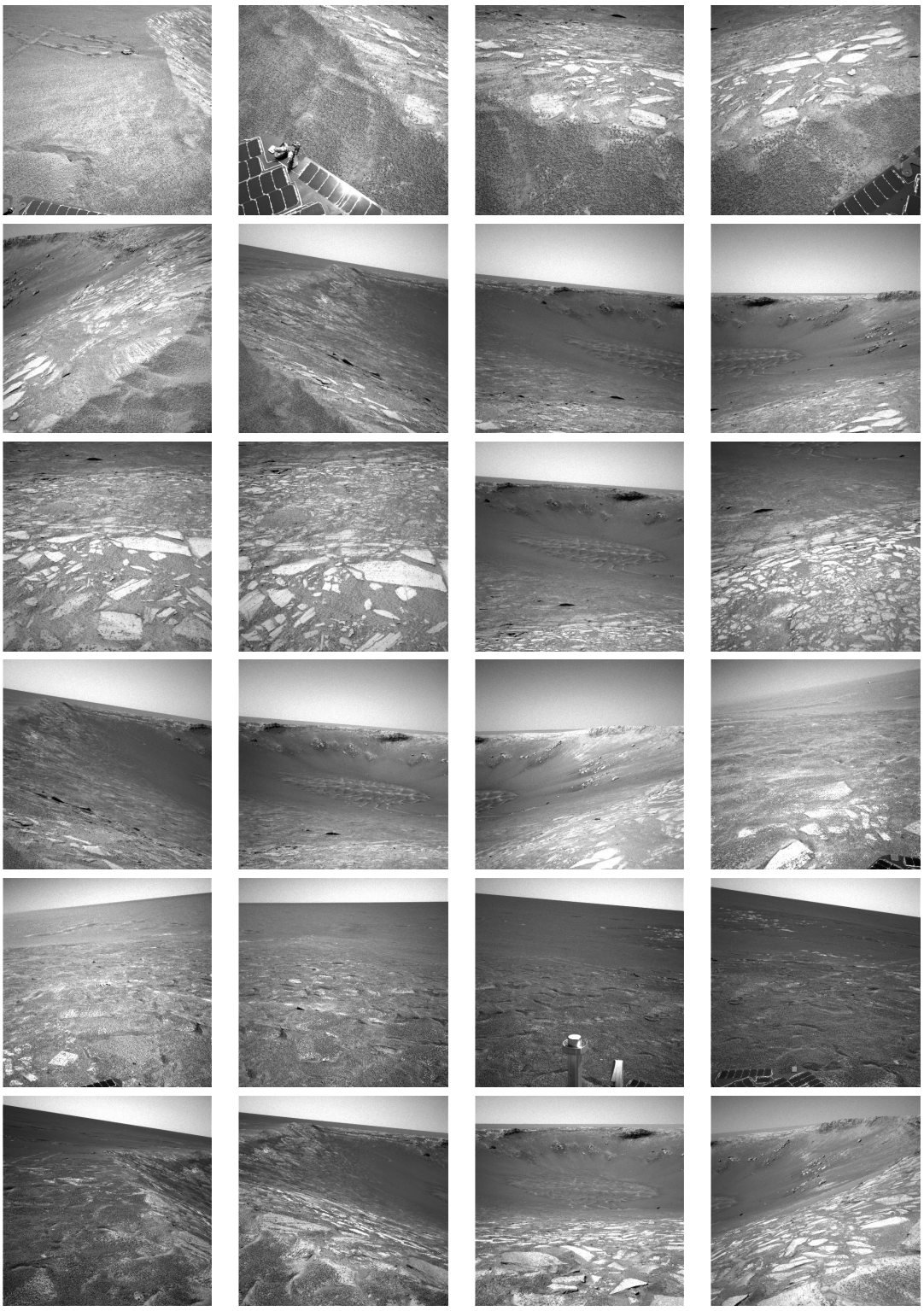
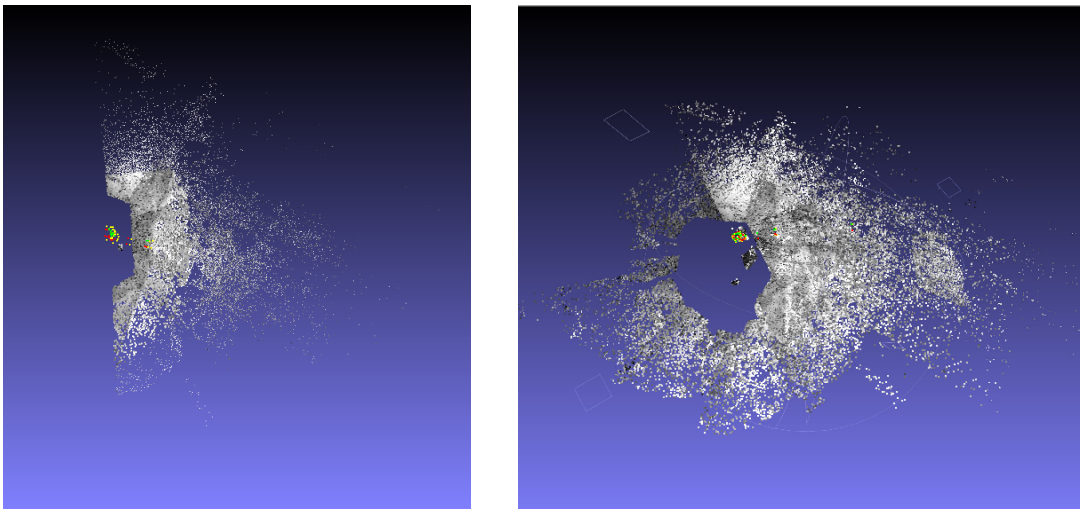


Figure 6.4: Some images of the *Mars* dataset.



(a) With Lowe's criterion (LOWE 2004). (b) With our method.

Figure 6.5: 3D point cloud resulting from the calibration of the *Mars* dataset with DoG features.

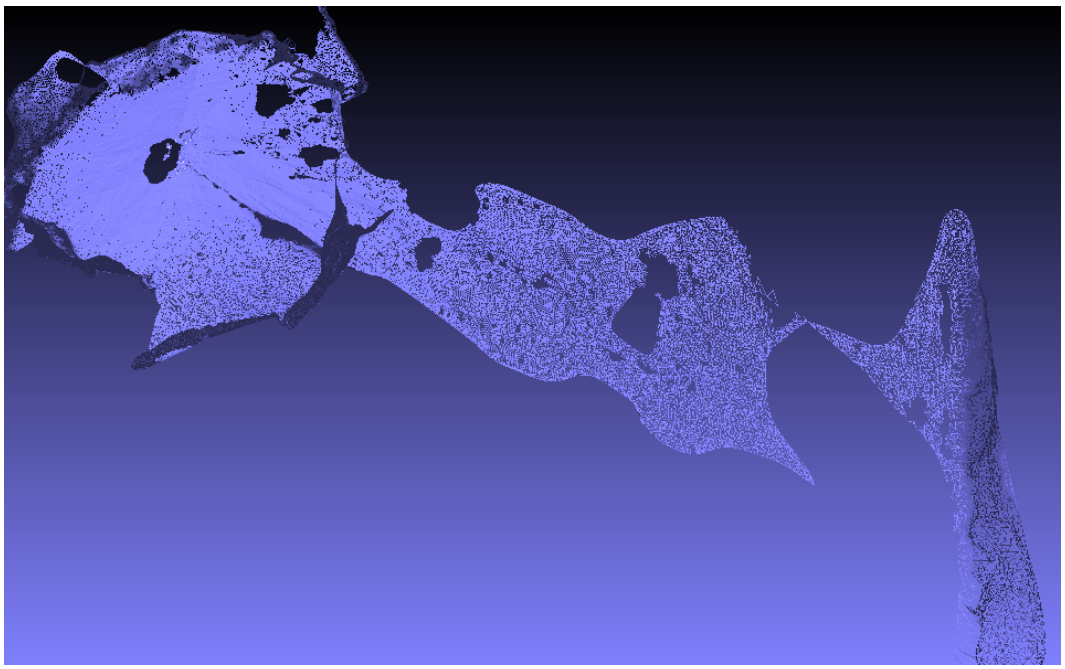


Figure 6.6: 3D mesh reconstruction of the *Mars* dataset after matching with our method and using VU et al. (2012)'s 3D reconstruction pipeline.

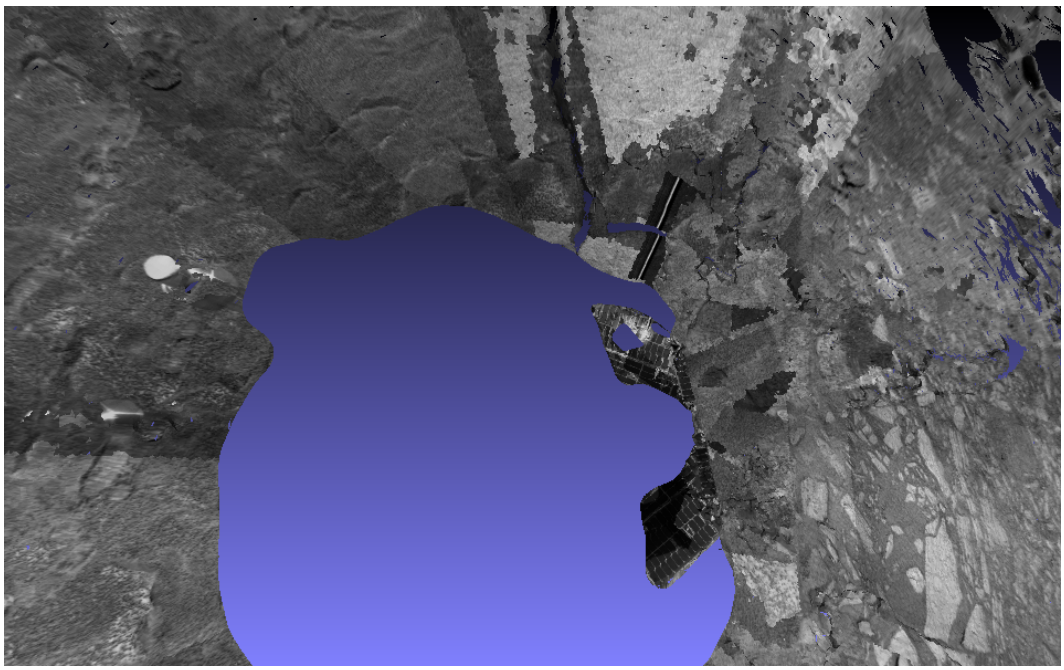


Figure 6.7: Close up on the 3D textured reconstruction of the *Mars* dataset after matching with our method and using VU et al. (2012)'s 3D reconstruction pipeline.

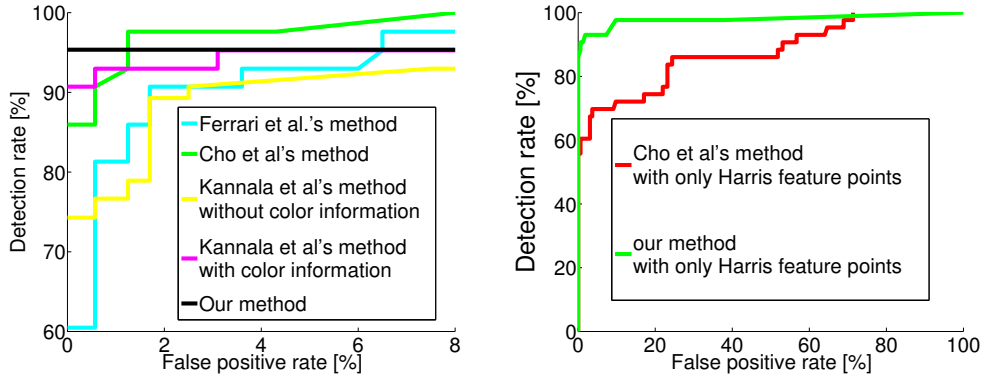
DoG features.

6.2 Deformable Object Matching

We evaluated our method on deformable object matching using the ETHZ Toys dataset (40 images of 9 models, and 23 test images), testing each model image against each test image. We compared with FERRARI et al. (2004), KANNALA et al. (2008) and CHO et al. (2009), as reported in their papers. For a fair comparison, we used MSER+SIFT and Harris-Affine+SIFT features, like CHO et al. (2009). Note that, in addition to these affine-covariant features, KANNALA et al. (2008) and FERRARI et al. (2004) use color information and dense photometric information, which we do not use.

Performance is reported in the ROC curve in Figure 6.8a, which depicts the detection rate versus false positive rate, letting a detection threshold vary. (An object is considered as detected if the number of produced matches, summed over all its model views, exceeds this threshold.) Our method outperforms others, except for high false positive rate. This makes our method attractive for object matching tasks that tolerate only few wrong detections.

We performed a second experiment with the same dataset and the same parameters as CHO et al. (2009), but only considering Harris-Affine+SIFT features, which are reported to be among the most ambiguous affine-covariant features (MIKOLAJCZYK et al. 2005). Figure 6.8b confirms that our method is less prone to false detection, as it



(a) With Harris-Affine+SIFT and MSER+SIFT features.

(b) With Harris-Affine+SIFT features only.

Figure 6.8: ROC curves on the ETHZ Toys dataset.

outperforms Cho et al.'s method both for low and high false positive rates.

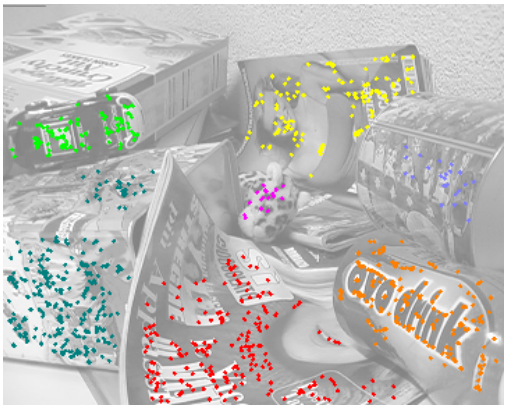
The ETHZ Toys dataset can be found and downloaded from the CALVIN RESEARCH GROUP (2004)'s homepage. We show in Figures 6.9, 6.10, 6.11 and 6.12 the results that we obtained on the 23 test images, which contain 43 objects in total. The names in italic such as *All* refer to the image file names in the dataset. In the illustrations, we use different colors to reference each model object, as summarized in Table 6.4.

Color	Model object	Color	Model object
Red	Blonde	Green	Car
Blue	Guard	Magenta	Leo
Yellow	Michelle	Orange	Ovo
Dark Green	Suchard	Lavender Blue	Xmas

Table 6.4: Reference color for each model object.

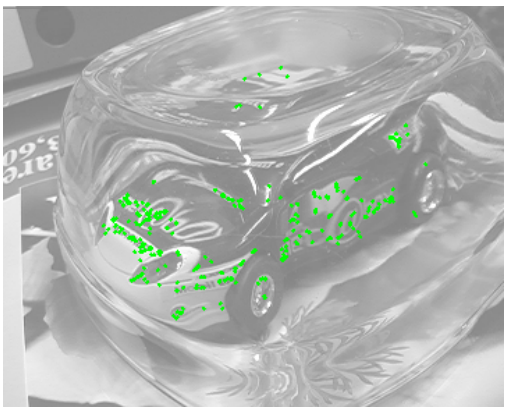
Our results are shown for Harris-Affine and MSER interest points. For each image pair, we obtained matches such that their distrust score is less than $\ell = 1.1$, whereas CHO et al. (2009) use a much more restricted set of matches, i.e., such that their distrust score is less than $\ell = 0.9$.

We can see that the results are visually very clean. The consistency of matches actually goes beyond the mere recognition of objects. As already pointed out in the paper, region affine-consistency does not assume a single affinity but many. And this network of locally similar affinity is flexible enough to adapt to substantial non-affine transformations, as occurs with deformable objects.



(a) *All*

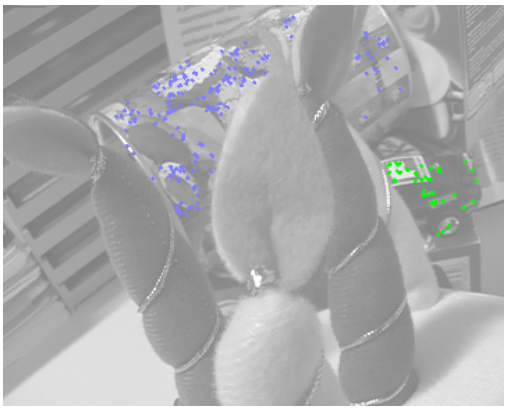
(b) *CarBooks*



(c) *CarGlas*

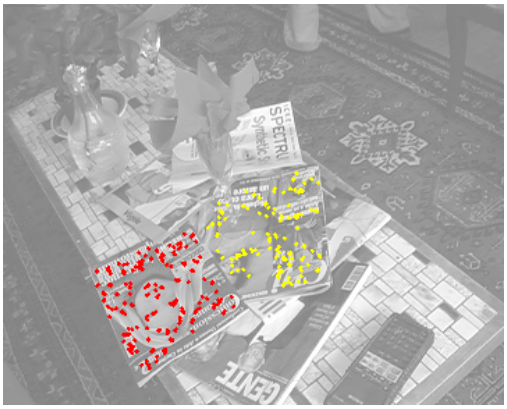
(d) *CarViewpoint*

(e) *CarXmasA*

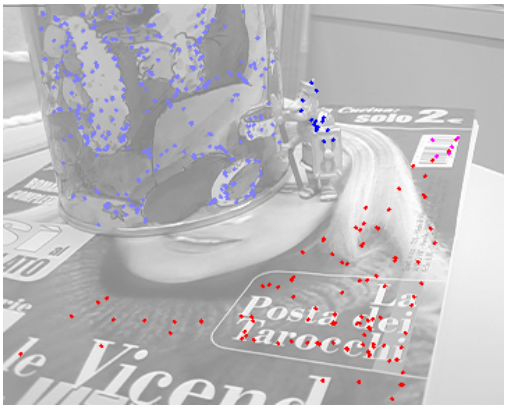


(f) *CarXmasB*

Figure 6.9: Detection in ETHZ test images (1/4). Colors are defined in Table 6.4.

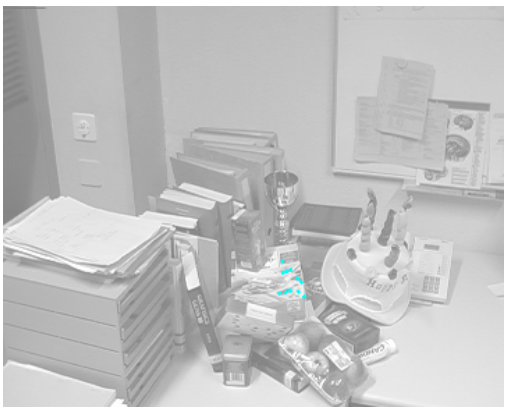


(a) *Magazines*



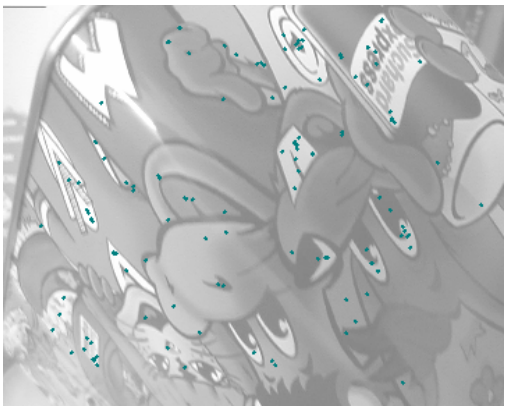
(b) *GuardOnBlonde*

(c) *GuardOnMichelle*



(d) *KellogsClutter*

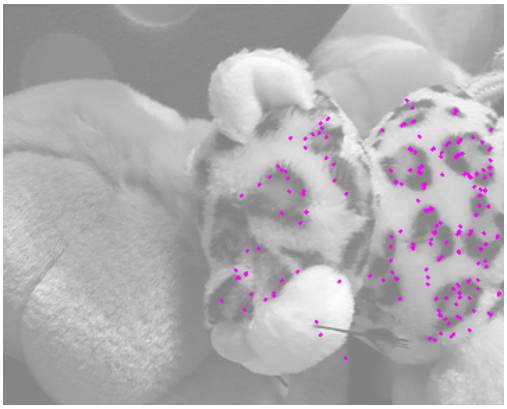
(e) *KitchenA*



(f) *KitchenB*

Figure 6.10: Detection in ETHZ test images (2/4). Colors are defined in Table 6.4.

(a) *LeoHidden*



(b) *LeoSleeps*

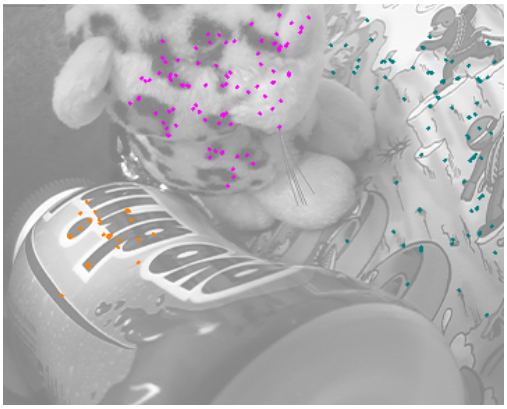
(c) *MichelleBentA*

(d) *MichelleBentB*

(e) *MichelleBentC*

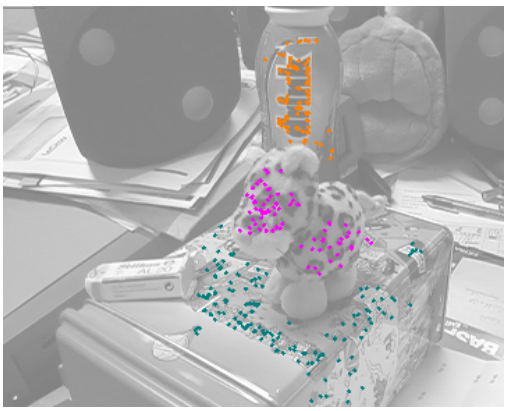
(f) *MichelleBentD*

Figure 6.11: Detection in ETHZ test images (3/4). Colors are defined in Table 6.4.

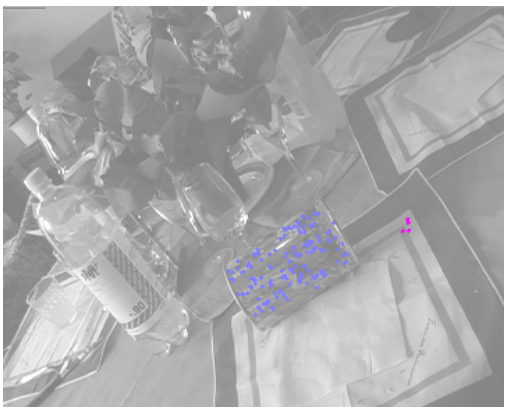


(a) *OvoLeoSuchardA*

(b) *OvoLeoSuchardB*



(c) *OvoLeoSuchardC*



(d) *TableA*

(e) *TableB*

Figure 6.12: Detection in ETHZ test images (4/4). Colors are defined in Table 6.4.

Methods	Accuracy
TM (DUCHENNE et al. 2011)	$a \leq 0.80$ for $N_f \leq 20$ $a \leq 0.05$ for $N_f \geq 30$
SIFT (LOWE 2004) + TM (DUCHENNE et al. 2011)	$0.75 \leq a \leq 0.85$ for $N_f \leq 200$
Our method	$0.95 \leq a$ for $N_f \leq 200$

Table 6.5: Compared accurracy a with 3rd-order hypergraph (TM).

6.3 Comparison with Hypergraph Matching Methods

We also compared with a tensor-based, 3rd-order hypergraph matching method (TM) (DUCHENNE et al. 2011), with image 1 and 4 of the *graffiti* dataset used in MIKOLAJCZYK et al. (2005), where the ground truth homography $\mathbf{H}$ is known. DoG features were detected and described with the SIFT descriptor. We evaluated the accuracy, i.e., the proportion of actually correct matches among produced ones, as a function of the number N_f of features to match. Because TM does not handle very well outliers, we only experimented various feature sets such that: $|\mathcal{X}| = |\mathcal{Y}| = N_f$ and there is a bijection between $\mathcal{X}$ and $\mathcal{Y}$ such that for each $x \in \mathcal{X}$, there is a unique $y \in \mathcal{Y}$ satisfying $\|\mathbf{H}\mathbf{x} - \mathbf{y}\| \leq 5$ pixels, and likewise when permuting $\mathcal{X}$ and $\mathcal{Y}$. Results are presented in Table 6.5.

The combined use of 1st-order SIFT descriptors and 3rd-order affinities (SIFT+TM) improves the poor result of TM, but our method performs much better.

To conclude this section, let us also note that the experiments with DUCHENNE et al. (2011)'s tensor-based matching cannot be very large scale, because it does not scale well in space and computation time. Indeed, its complexity is $O(N^3 \log N)$ or $O(N^4 \log N)$, which is well reflected in practice. When $N_f > 500$, using DUCHENNE et al. (2011)'s MATLAB implementation, the needed affinity tensor already occupies several gigabytes of memory and is computed in order of several minutes before even starting the global optimization process. On the contrary, our method is much more memory friendly and faster, especially our method only needs to compute on-the-fly match neighborhoods.

PART II

Pattern Detection for Robust Façade Analysis

Chapter 7

Repetitive Pattern Search

Contents

7.1 Recursive Breadth-First Search Algorithm

Our feature matching algorithm can easily be turned into a pattern matcher. Given a object model M_0 defined by a geometric region I_0 in some input image I, the goal is to retrieve all objects that are similar to M_0 in some image J (possibly equal to I), i.e., to find image regions in J that are similar to I_0. We consider the case where I_0 is defined as the interior of a polygon P_0.

For this, we define $\mathcal{X}_0$ as the set of features inside polygon P_0 in I and $\mathcal{Y}$ as the set of features in J not in $\mathcal{X}_0$ (in case $J = I$). We then grow regions of $\mathcal{M} \subset \mathcal{X}_0 \times \mathcal{Y}$ as described above, allowing ambiguity on $\mathcal{X}_0$, which we formalize in Section 4.7 of Chapter 4. The resulting set of regions $\mathcal{R} = (R_i)_{1 \leq i \leq n}$ corresponds to as many discovered pattern instances. The image region in J corresponding to a set of matches R_i can be retrieved by assuming local affinity transformations from I to J. More formally, given a vertex $\mathbf{u} \in \mathbb{R}^2$ of polygon P_0 in I and assuming that R_i is of sufficient cardinality, let x_1, x_2, x_3 be the geometrically closest 3 features (also nonaligned) in I such that there are matches $(m_j)_{1 \leq j \leq 3} = (x_j, y_j)_{1 \leq j \leq 3} \in R_i$. Then the corresponding polygon vertex in image J is $A(m_1, m_2, m_3)(\mathbf{u})$. The polygon P_i formed by its vertices defines an image region J_i of J that delineates the matched object instance M_i. This is detailed in Algorithm 7.1.

Algorithm 7.1 Generic Pattern Search

Require:

- $N_{\min}$: minimum number of matches between the pattern and an instance
- $p_{\min}$: minimum percentage of matches between the pattern and an instance

procedure FINDSIMILARPATTERNS(P_0, I, J)
  $\mathcal{P} \leftarrow \emptyset$ *// Initialize result*
  $I_0 \leftarrow$ subimage of I delineated by polygon P_0
  $\mathcal{X} \leftarrow$ features detected in I_0
  $\mathcal{Y} \leftarrow$ features detected in J
  $\mathcal{M} \leftarrow$ feature matches between images I_0 and J
  Sort $\mathcal{M}$ by decreasing confidence
  while $\mathcal{M} \neq \emptyset$ **do**
    Get the best match m from $\mathcal{M}$ *// Get good seed*
    $R \leftarrow$ GROWREGION(m, K, ρ_0) *// Cf. Algorithm 5.5*
    if $|R| > \max(N_{\min}, p_{\min}.|\mathcal{X}|)$ **then** *// If region R is large enough*
      Estimate transform T from matches in R *// T is a homography or affinity*
      $P \leftarrow T(P_0)$ *// Estimate new pattern P*
      if $\forall P' \in \mathcal{P}, P \cap P' = \emptyset$ **then** *// If there is no overlap with a previous pattern*
        $\mathcal{P} \leftarrow \mathcal{P} \cup \{P\}$ *// Remember P in result*
      end if
      $\mathcal{M} \leftarrow \mathcal{M} \setminus R$ *// Stop exploring matches in R, keeping $\mathcal{M}$ sorted*
    end if
  end while
  return $\mathcal{P}$
end procedure

When working on a single image, i.e., when $J = I$, Algorithm 7.1 is to be run with arguments $(P_0, I, I \setminus I_0)$ where I_0 is the subimage of I delineated by polygon P_0.

More pattern instances can be found by removing features in $\mathcal{R}$ from $\mathcal{Y}$ and reusing recursively image regions $(J_i)_{1 \leq i \leq n}$ as new input patterns, until no new pattern instance is found. To reduce the risk of pattern drifting, recursive pattern search has to be performed in a breadth-first search as detailed in Algorithm 7.2.

7.2 Accurate Pattern Localization: Window Detection

7.2.1 Related Work and Challenges

We experimented with pattern detection, looking for windows in building facades. Although this problem has already been attacked in LEE and NEVATIA (2004), ALI et al. (2007), HAUGEARD et al. (2009) and RECKY and LEBERL (2010), *accurate* localization has not been addressed adequately: they do not really quantify the performance of their methods regarding the localization or segmentation in terms of pixellic precision.

Algorithm 7.2 Generic Breadth-First Pattern Search

Require:

- $\mathcal{P}$: set of initial polygons that delineate patterns of interest
- $N_{\min}$: minimum number of matches between the pattern and an instance
- $p_{\min}$: minimum percentage of matches between the pattern and an instance

procedure BREADTHFIRSTPATTERNSEARCH($\mathcal{P}, I$)
  Enqueue all polygons P in $\mathcal{P}$ in a queue $\mathcal{Q}$
  while $\mathcal{Q} \neq \emptyset$ **do**
    Dequeue the first polygon P from $\mathcal{Q}$
    $\mathcal{P}' \leftarrow$ FINDSIMILARPATTERNS(P, I).
    Remove all polygons in $\mathcal{P}'$ that overlap with any polygon in $\mathcal{P}$.
    $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}'$
    Enqueue all polygons of $\mathcal{P}'$ in $\mathcal{Q}$
  end while
  Return $\mathcal{P}$
end procedure

Note that pattern detection has been extensively studied and we refer the reader to LIU et al. (2010)'s exhaustive survey for details. In pattern detection, structural regularity or symmetry are commonly assumed (LIU et al. 2010). However, they are not always appropriate because windows are not necessary laid out according a grid-like structure. For example, buildings often have their windows laid out according to a more complex structure than a grid. We can find such examples of buildings in the *eTRIMS* dataset (KORČ and FÖRSTNER 2009). For this reason, this makes our method complementary to most of the methods surveyed in (LIU et al. 2010).

To circumvent the challenges posed by the existence of complex structures, one can just resort to object detection, in particular based on learning techniques, such as the cascade classifier (VIOLA and JONES 2004). However, these methods are not perfect and still make errors, i.e., miss objects, hallucinate objects, or do not localize them properly. One or several parameters may usually be tuned to favor precision or recall. Moreover, they are trained to recognize a wide variety of object instances. They do not exploit the fact that, in some circumstances, only similar object instances appear in the image, like windows on a given single facade. A hypothesis of these methods is that all detected objects are independent one from another (as long as they do not overlap).

Here, we propose to use such a detector only to find few but very likely object occurrences, tuning the detector for precision. We then use these accurate detections as problem-specific models and rely on a robust pattern search procedure to look for similar instances of these models in the image. For more robustness, to improve recall, this procedure is repeated recursively on the new detections (resulting from the pattern search) until there are no more detections.

Eventually, window localization poses more challenges because of the wide range

of appearance variety, the lack of texture, and the illumination variations. Unrectified images adds up to these challenges. Windows are then related by homographies or affinities: they may vary in size and shape, and it is difficult to detect small windows with almost no texture. Still, in rectified images, windows on the same facade often have two or three different widths, depending on the size or use of the corresponding rooms. However, although stretched horizontally, all windows on the facade "look alike". Also for older structures, including Haussmanian buildings, bottom floors have higher ceilings and higher windows than top floors, to compensate for the lesser illumination. But here again the window appearance is only stretched, vertically. Our pattern search procedure will prove well suited to accomodate these different situations.

In the end, selecting just a few best candidates of windows instances in a facade and then searching for their repeated occurrences yields more accurate detections than looking for many instances of any kind of windows. The search is then indeed more robust as well as specialized for a specific kind of window that is pertinent for the given facade.

7.2.2 Method

To apply our repetitive pattern search, DoG, Harris-Affine and MSER features are extracted in each image and described by the SIFT descriptor. We only keep matches whose distrust score (cf. Section 5.2) is less than 1.2, i.e., matches within 20% of the best match (descriptor-wise). Note that in our experiments, the outlier contamination rate, with such a distrust score, can reach 98% of about 500,000 feature matches. Note that it is normally be beyond what most RANSAC variants can handle. Several transformations are to be sought here, one for each instance of the model. RANSAC thus has to be iterated after a first instance is found, to find other ones, or a variant of RANSAC looking simultaneously for several models has to be used (see, for example, ZULIANI et al. (2005)).

A few window are indicated in the images by means of bounding boxes or polygons. The dimensions of the bounding box of the pattern windows are dilated by 50% before search, to include a little environmental information, and shrunk back when instances are found to estimate the window region accurately. Indeed, including environmental information is important mostly because glass material in windows hardly has any meaningful textural information. And very often, enviromental information often plays an essential role in the feature correspondence task as illustrated in Figure 7.1. Thus, we actually use Algorithm 7.3 instead of Algorithm 7.1. Indeed, because of the dilation, adjacent dilated bounded boxes can overlap with each other contrary to original bounding boxes, which can cause rejection to many correctly detected windows.

We actually consider two task variants, as presented below.

Semi-Automatic Window Localization Task. The first task is the semi-automatic window localization task, in which a human operator indicate a few windows (on the order of 10%). It has the advantage of being adapted for both rectified and unrectified images containing building façades.

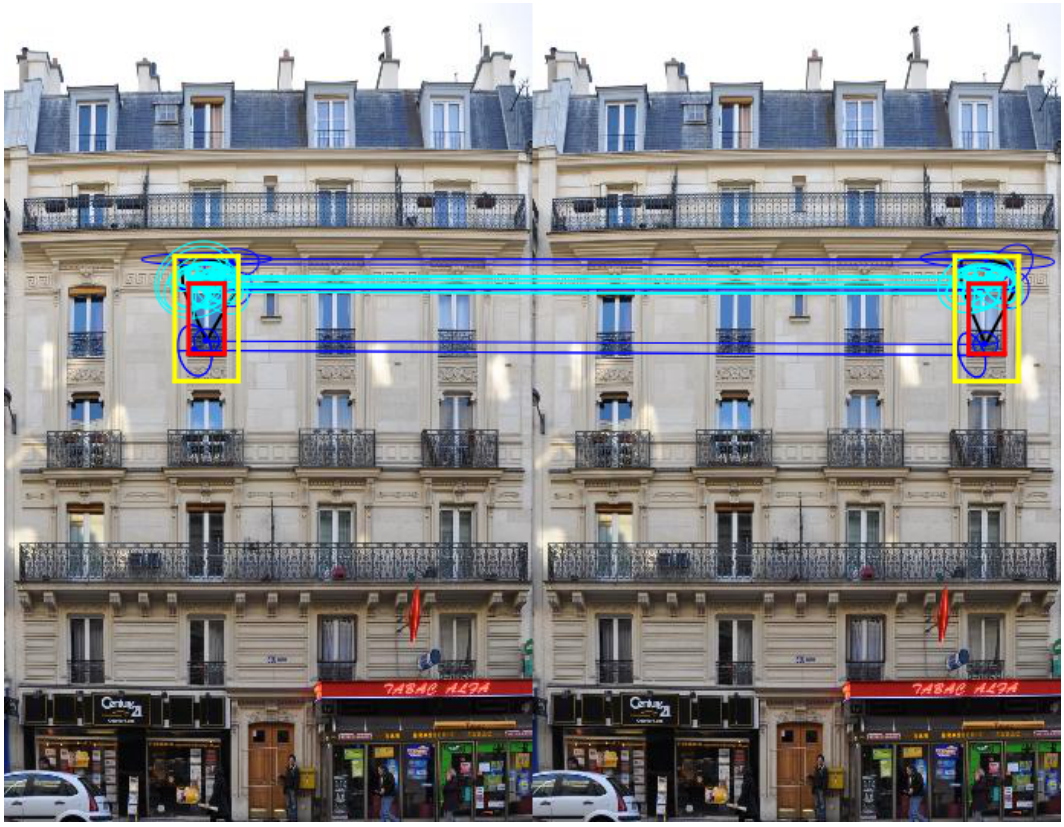


Figure 7.1: We illustrate the importance of including environmental information in window detection. The *red* bounding box is the polygon delinating the window model and the *yellow* one is the dilated bounding polygon. Here, a matched window is found indeed mostly because a lot of features on the ornemental elements at the top of the windows were matched successfully. Matched features (in *blue* color) on the balcony at the bottom of the window would not have been sufficient to reliably detect such a window.

Automatic Window Localization Task. For the second, fully automatic task, we trained a cascade classifier (*CC*) (VIOLA and JONES 2004; ALI et al. 2007) to recognize windows. Its best detections are used as seeds for our pattern search (*RG*) as in task 1. To get the best compromise between false positive rate (FPR) and true positive rate (TPR) while training, we empirically set the minimum hit rate threshold to 0.9 and the maximum false alarm rate threshold to 0.1. Moreover, when performing as a detector, *CC* is tuned by a detection parameter threshold τ_{CC} that balances TPR and FPR: as τ_{CC} increases, TPR and FPR decrease (cf. table 7.1). More details about CC's training step can be found in Section E.1 of Appendix E.

Algorithm 7.3 Window Pattern Search

Require:

- $N_{\min}$: minimum number of matches between the pattern and an instance
- $p_{\min}$: minimum percentage of matches between the pattern and an instance

procedure FINDSIMILARWINDOWS(P_0, I)
  $\mathcal{P} \leftarrow \emptyset$
  Dilate P_0 by 50% *// Include environmental information for windows*
  $I_0 \leftarrow$ subimage of I delineated by polygon P_0
  $\mathcal{F} \leftarrow$ features detected in I
  $\mathcal{M} \leftarrow$ matches in $\mathcal{F}^2$ between I_0 and $(I \setminus I_0)$
  Sort $\mathcal{M}$ by decreasing confidence
  while $\mathcal{M} \neq \emptyset$ **do**
    Get the best match m from $\mathcal{M}$
    $R \leftarrow$ GROWREGION(m, K, ρ_0) *// Cf. Algorithm 5.5*
    if $|R| > N_{\min}$ **then**
      Estimate transform T from matches in R
      $P \leftarrow T(P_0)$
      Shrink $\mathcal{P}$ by 50% *// Get the polygon correctly delineating the window*
      if $\forall P' \in \mathcal{P}, P \cap P' = \emptyset$ **then** *// If there is no overlap with a previous pattern*
        $\mathcal{P} \leftarrow \mathcal{P} \cup \{P\}$ *// Remember P in result*
      end if
      $\mathcal{M} \leftarrow \mathcal{M} \setminus R$ *// Stop exploring matches in R, keeping $\mathcal{M}$ sorted*
    end if
  end while
  return $\mathcal{P}$
end procedure

7.2.3 Experimental Settings and Performance Measurement

Datasets

We used for evaluation the *ECP CVPR 2010* dataset (20 training images and 10 test images) and *ECP Benchmark 2011* datasets (104 test images) (TEBOUL 2010), that picture rectified buildings, as well as the *eTRIMS* dataset (60 test images) (KORČ and FÖRSTNER 2009). Whereas the ECP datasets only picture rectified images of Haussmannian buildings, the *eTRIMS* dataset contains nonrectified images of very different architectural and building styles. Out of the 60 images in the *eTRIMS* dataset, we only considered those having at least 6 windows, which corresponds to 45 images. As ground truth windows were sometimes erroneous or not delineated similarly (i.e., with the same definition) in all datasets, we first corrected and normalized the annotations, more particularly in the *eTRIMS* dataset. We also constructed a rectified version of the *eTRIMS* dataset for comparison purposes. This image rectification was performed by hand (and eye).

Finally, we manually indicated seed windows for the first task, delineating 1 to 4 windows depending on the appearance variability.

Evaluation Method

For each method and for each image, we compute the

$$\text{confusion matrix} = \begin{pmatrix} \text{TPR} & \text{FNR} \\ \text{FPR} & \text{TNR} \end{pmatrix}.$$

where the true positive rate (TPR), false negative rate (FNR), false positive rate (FPR) and the true negative rate (TNR) are defined as follows:

- TPR is the percentage of *window* pixels correctly labeled as *window*,
- FNR is the percentage of *window* pixels incorrectly labeled as *non-window*,
- FPR is the percentage of *non-window* pixels incorrectly labeled as *window*,
- TNR is the percentage of *non-window* pixels correctly labeled as *non-window*.

For each dataset, methods are then compared in terms of mean true positive rate ($\overline{\text{TPR}}$), mean false negative rate ($\overline{\text{FNR}}$), mean false negative rate ($\overline{\text{FPR}}$), mean true negative rate ($\overline{\text{TNR}}$) over all images. This corresponds to the

$$\text{mean confusion matrix} = \begin{pmatrix} \overline{\text{TPR}} & \overline{\text{FNR}} \\ \overline{\text{FPR}} & \overline{\text{TNR}} \end{pmatrix}.$$

Concurrent Methods

We compare our method (*RG*) with two others. The first one (*RL*) is TEBOUL et al. (2011)'s grammar-based method (cf. Chapter 8), that we ran on images of the ECP datasets using appropriate grammars (cf. Table 7.1) as provided by the authors. *RL* only works with rectified images. The second method is the cascade classifier (*CC*) (VIOLA and JONES 2004), for various detection thresholds τ_{CC}. Methods are compared in terms of mean confusion matrix. A method is deemed good if the confusion matrix is close to the identity.

7.2.4 Results

Rectified Case

In the rectified case, windows are simply related by translation. Tables 7.1 and 7.2 summarize our results. Detection examples are shown in Figure 7.2. More detection results on the ECP datasets can be found in Appendix E. In all datasets, for $\overline{\text{FPR}}$ less than 10%, *RG* outperforms other methods in terms of $\overline{\text{TPR}}$ and significantly improves the initial $\overline{\text{TPR}}$ of *CC*, by 12 points at least. *RG* also hardly increases $\overline{\text{FPR}}$ from *CC*, at most by 3 points. Unsurprisingly, looking for windows that are *specific* to the facade is better than

Methods	$\overline{\text{TPR}}$	$\overline{\text{FNR}}$	$\overline{\text{FPR}}$	$\overline{\text{TNR}}$
manual + RG	**80**%	**20**%	**3.5**%	**96.5**%
CC ($\tau_{CC} = 20$) + RG	76%	26%	6%	94%
CC ($\tau_{CC} = 30$) + RG	71%	29%	5%	95%
CC ($\tau_{CC} = 5$)	77%	23%	14.5%	85.5%
CC ($\tau_{CC} = 10$)	70%	30%	9%	91%
CC ($\tau_{CC} = 20$)	64%	36%	6%	94%
CC ($\tau_{CC} = 30$)	56.5%	43.5%	**4.5**%	95.5%
RL (bin-hue)	68.5%	31.5%	12.5%	87.5%
RL (bin-rf)	51.5%	43.5%	35.5%	64.5%
RL (4-color-rf)	25.5%	75.5%	12%	88%
RL (haussm-rf)	67%	33%	6.5%	93.5%

Table 7.1: Results averaging the performance on *ECP CVPR2010* and *ECP Benchmark 2011* datasets. *manual+RG* and *CC+RG* denote our method run using bounding boxes provided respectively by hand and by *CC*. For *RL*, we used 3 shape grammars: *binary* (bin), 4-*color*, and *Haussmannian* (haussm). *hue* and *rf* are different probability priors for façade segmentation when parsing with the shape grammars.

looking for many instances of *any kind* of windows. Finally, *RG* does even better if initial bounding boxes are provided manually.

Note that, because texture are generally lacking in *eTrims* images, windows have generally very few detected features in them. Still, *RG* obtains a good $\overline{\text{TPR}}$ especially for the semi-automatic window detection task. In any case, Tables 7.1 and 7.2 show that *RG* always maintains a very low $\overline{\text{FPR}}$ for the *eTrims* dataset.

Regarding concurrent methods, *CC* achieves a good $\overline{\text{TPR}}$ of 77% with a threshold $\tau_{CC} = 5\%$, but at the cost of a high $\overline{\text{FPR}}$ of 14.5% in the ECP datasets. *RL* only achieves a $\overline{\text{TPR}}$ of 67% but has a good $\overline{\text{FPR}}$ of 6.5% by using a *Haussmannian* grammar (*haussm*) and a random forest-based reward (*rf*). However, the performance of all these concurrent methods dramatically drops on the *eTrims* dataset.

Non-rectified Case

In the non-rectified case, windows are related by homographies or affinities. This is more challenging as windows may vary in size and shape, and it is difficult to detect small windows with almost no texture. *RL* is not applicable in this setting, and *CC* provides too coarse detections. We thus only consider bounding quadrilaterals that are provided manually.

Results are reported in Table 7.3. $\overline{\text{TPR}}$ loses 4 points w.r.t. the rectified case. This slight degradation is chiefly due to estimation errors of the geometric transformation

Methods	$\overline{\text{TPR}}$	$\overline{\text{FNR}}$	$\overline{\text{FPR}}$	$\overline{\text{TNR}}$
manual + RG	**75**%	**25**%	**4**%	**96**%
CC ($\tau_{CC} = 5$) + RG	60%	40%	7%	93%
CC ($\tau_{CC} = 5$)	46%	54%	4%	96%
RL (bin-hue)	27%	73%	11%	89%
RL (bin-rf)	27%	73%	23%	77%
RL (4-color-rf)	29%	71%	13%	87%

Table 7.2: Results on the *eTRIMS* dataset with (manually) rectified images. See caption of Table 7.1 for details about *RL*.

Methods	Dice score	$\overline{\text{TPR}}$	$\overline{\text{FNR}}$	$\overline{\text{FPR}}$	$\overline{\text{TNR}}$
manual+RG	0.77	71%	29%	2%	98%

Table 7.3: Results on the *eTRIMS* dataset with non-rectifed images. Bounding quadrilaterals are provided manually.

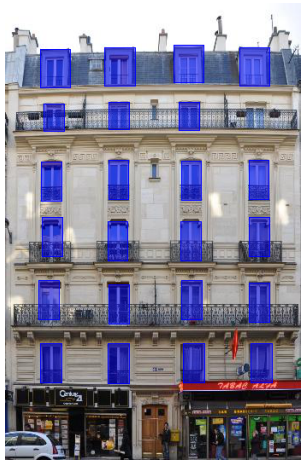
between the matched patterns. Shift and size errors between the geometric region of the detected pattern and the estimated image region also accumulates. Still, our method achieves a very low $\overline{\text{FPR}}$ of 2%.

Excerpts of our results on the *eTRIMS* dataset (KORČ and FÖRSTNER 2009) are shown in Figures 7.3 and 7.4. As we can see in Figures 7.3 and 7.4, the robustness of our pattern detection in the non-rectified case shows that a preliminary rectification (often defined manually and to be repeated for each different facade plane in a given image) is not absolutely necessary.

(a) initial monge_40

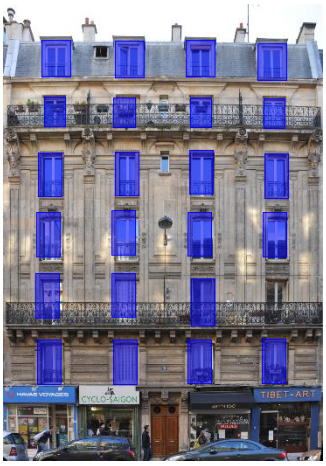
(b) initial monge_45

(c) initial monge_50



(d) result monge_40

(e) result monge_45



(f) result monge_50

$$\begin{pmatrix} \mathbf{86} & 14 \\ 4 & \mathbf{96} \end{pmatrix}$$

(g) monge_40

$$\begin{pmatrix} \mathbf{77} & 23 \\ 6 & \mathbf{96} \end{pmatrix}$$

(h) monge_45

$$\begin{pmatrix} \mathbf{81} & 19 \\ 3 & \mathbf{97} \end{pmatrix}$$

(i) monge_50

Figure 7.2: Three window detection results. Figures 7.2a, 7.2b and 7.2c show the window seeds and Figures 7.2d, 7.2e and 7.2f show the results with our feature matching approach. Figures 7.2g, 7.2h and 7.2i are the resulting confusion matrices.

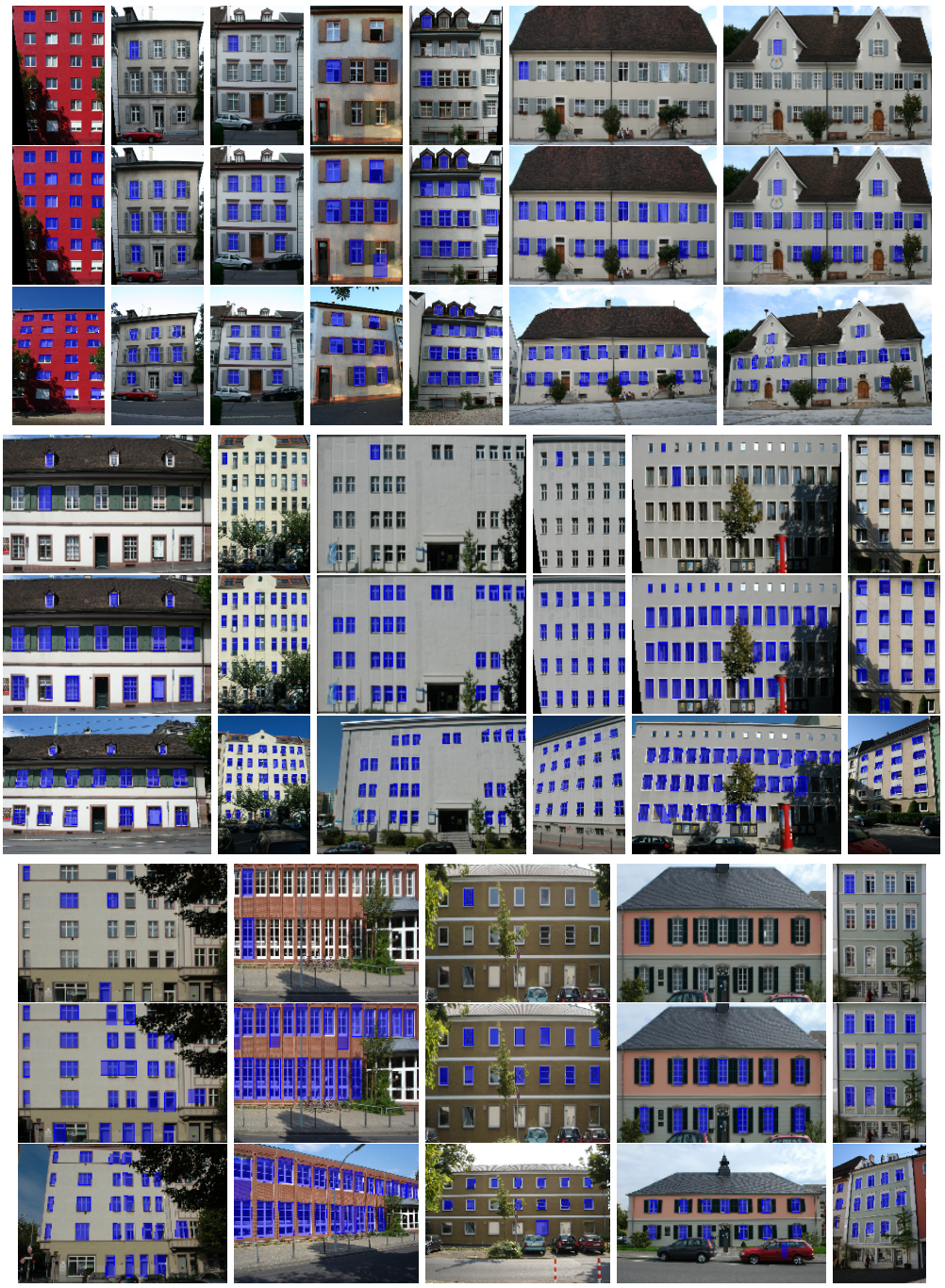


Figure 7.3: Window detection results on the eTRIMS dataset (1/2): input quadrilaterals, detections on rectified images, and detections on non-rectified images.

Figure 7.4: Window detection results on the eTRIMS dataset (2/2): input quadrilaterals, detections on rectified images, and detections on non-rectified images.

Chapter 8

High-Level Bottom-Up Cues for Fast and Robust Façade Parsing

In this last chapter, we address the problem of better parsing images of building facades. The goal is to segment images, assigning to the resulting regions semantic labels that correspond to the basic architectural elements. We assume a top-down parsing framework based on a 2D shape grammar that encodes a prior knowledge on the possible composition of facades. The algorithm explores the space of feasible solutions by generating possible configurations of the facade and comparing them to the input data by means of a local, pixel- or patch-based classifier. We propose new bottom-up cues for the algorithm, both for the evaluation of a candidate parse and for guiding the exploration of the space of feasible solutions. The method that we propose benefits from detection-based information and leverages on the similar appearance of elements that repeat in a given facade. Experiments performed on standard datasets show that this use of more discriminative bottom-up cues improves the convergence in comparison to state-of-the-art algorithms, and giving better results in terms of precision and recall, as well as computation time and performance deviation.

Contents

8.1 Introduction

Image segmentation remains a generic, and in a large part, unsolved problem in computer vision. For many instances of this problem we are not restricted to using the information contained in the image alone. We may also resort to prior knowledge of the likely compositions of objects present in the scenes. Shape grammars are a concept that encodes such prior knowledge and is used for image segmentation. In the framework of shape grammars, segmentation amounts to assigning semantic labels to image regions and is known under the name of *image parsing* (ZHU and MUMFORD 2006).

Applications involving images of highly structured scenes are likely to benefit most from the development of image grammars. One such field, which has been drawing increasing attention recently, is facade parsing (ALEGRE and DELLAERT 2004; MÜLLER et al. 2006; MÜLLER et al. 2007; BARINOVA et al. 2010; TEBOUL et al. 2010; TEBOUL et al. 2011). The goal of facade parsing is to automatically provide a hierarchical decomposition of a building facade into its constituent elements, given an image of the facade. Facade parsing has a variety of applications, including urban planning, thermal performance assessment of existing structures and reconstruction of models of existing buildings for games and simulators.

Amongst the most successful solutions to the problem of facade parsing are the top-down parsers proposed by Teboul et al. (TEBOUL et al. 2010; TEBOUL et al. 2011). They draw from the idea of split grammars for architectural modeling (MÜLLER et al. 2006), where a variety of building models can be generated from a single grammar. The process is analogous to string or sentence derivation in formal and natural language processing. The goal of parsing here is to perform the derivation in such a manner that the resulting model corresponds to the input image. The top-down parsers have proven to be efficient in this task.

However, the robustness of this approach is dependent on the quality of the bottom-up information used for comparing the candidate models with the input image. Such information can be degraded because of challenging lighting conditions, facade appearance variation, or occlusions. This sensitivity is partly due to the fact that the underlying merit function is based on low-level, pixel- or patch-based information, as proposed in TEBOUL et al. (2011). Besides, because of the high complexity of the problem space and due to the randomized nature of the approach, a good data-driven exploration of the solution space is crucial to simultaneously limit significant performance deviation in the parsing, and achieve fast convergence rate. But again, exploration presented in TEBOUL et al. (2011) is only based on pixels or patches.

To address the above issues, we propose a modified algorithm for top-down facade parsing that benefits from higher level and more robust abstractions, as provided by object detectors and geometric primitives. Namely, we integrate an object detector into TEBOUL et al. (2011)'s existing framework, in this case a window detector. For this, we use our robust pattern search method described in Chapter 7, exploiting the fact that architectural elements present in a particular facade frequently share similar appearance. Additionally, to improve the convergence properties of the method, we guide the parser

with the window detections and line segment cues. As a result, the parser not only better locates facade elements but also prunes the solution space much more selectively.

8.2 Related Work

The idea of representing the image contents in a hierarchical and semantized manner can be traced back to OHTA et al. (1978)'s and OHTA et al. (1979)'s work. However, the practical applications of grammars to image interpretation or segmentation are attributed to more recent works (see, for example, HAN and ZHU (2009), WANG et al. (2006), JIN and GEMAN (2006) and AHUJA and TODOROVIC (2008)).

In the litterature, the hierarchical and regular structure of man-made objects is explored to mainly improve segmentation or detection results (WANG et al. 2006; JIN and GEMAN 2006; AHUJA and TODOROVIC 2008). We focus on flexible grammars that allow the user to encode specific knowledge of the domain in the form of production rules that constrain the space of feasible solutions. The grammar-based image interpretation paradigm is thoroughly reviewed in ZHU and MUMFORD (2006)'s survey. A good example of this approach is HAN and ZHU (2009)'s rectangle-based grammar, in which the prior knowledge is represented by means of an and/or graph. The terminal symbols are rectangles and the production rules combine them into rows, columns or grids, and allow for rectangle nesting. This case illustrates one of the difficulty of the problem: the number of terminals in the solution is unknown. The greedy algorithm presented in the chapter copes well with this difficulty. However, since there is no semantic interpretation associated with the rectangles, there is no sensible way of deciding which of any two candidate parse trees is better.

The use of grammar-based facade parsing has been inspired by the successful application of split grammars for generating virtual urban environments (see, for example, (MÜLLER et al. 2006)). The key to success is to encode in the grammar basic constraints on the generated objects: the principles of adjacency, non-overlap and snaplines. A number of research work have been aimed at applying the grammar principles for retrieving building models from images (TEBOUL et al. 2010; TEBOUL et al. 2011; MATHIAS et al. 2011; RIEMENSCHNEIDER et al. 2012; SIMON et al. 2011). In their work, Teboul et al. present an application of a 2D binary split grammar for parsing rectified facade images (TEBOUL et al. 2010; TEBOUL et al. 2011; SIMON et al. 2011). The method can accommodate several classes of terminal symbols and has been shown to be robust to partial occlusions and relatively flexible to variable facade appearance (TEBOUL et al. 2011). However, the algorithm suffers from a number of shortcomings. For example, they rely on bottom-up gradient cues, which can hardly cope with common challenges in urban photographs, such as noise, occlusion, illumination changes YANG et al. (2012)'s work focuses on the application of rank-1 matrix approximation for facade parsing. A binary classifier of window color is applied to the facade image. The image is divided into rectangular regions. The algorithm attempts to fit an irregular grid of windows to each of these regions. This is performed by approximating the output of the classifier by a rank-1 matrix. The main drawback of the algorithm is the constraint of two-class (window

and wall) facades and the lack of flexibility in defining the grammar. MATHIAS et al. (2011) propose to use MÜLLER et al. (2006)'s grammar and generate the building while estimating the attributes of the applied grammar rules from the input images and a 3D point cloud. While the general idea seems attractive, the algorithm has not been shown to perform well with more than two classes of terminal symbols and accommodates only a small subset of rules of the original grammar (MATHIAS et al. 2011).

8.3 Grammar-Based Parsing

A shape grammar (STINY 1975; ZHU and MUMFORD 2006) is a formalism to represent a structured collection of shapes. The symbols of the grammar are basic shapes, and the production rules transform one configuration of basic shapes into another. The split grammars (TEBOUL et al. 2010; MÜLLER et al. 2006) are context-free shape grammars, where the production rules split the non-terminal basic shape on the left-hand side of the production rule along one dimension at a time. This simplification decreases the dimensionality of the space of parameters of a single production rule while preserving the expressive power of the grammar. This makes split grammars particularly suitable for modeling building facades.

The following part of this section gives a brief overview of 2D split grammars for facade modeling. The reader is referred to TEBOUL et al. (2010) and TEBOUL et al. (2011) for more details.

8.3.1 Split Grammars

The grammar dealt with in this chapter operates on rectangles as basic shapes. Each production rule splits a rectangle along the horizontal or vertical dimension into a number of new rectangles. In the case of building grammars, the terminal basic shapes represent architectural elements, like windows and wall tiles, and the production rules encode the possible spatial compositions of these elements. Generating from the grammar, one obtains schematic images of building facades (TEBOUL et al. 2010; MÜLLER et al. 2006).

Formally, a 2D split grammar $\mathcal{G}$ is a context-free grammar $(\mathcal{N}, \mathcal{T}, \mathcal{R}, S)$ where $\mathcal{N}$ is a finite set of non-terminal basic shapes $\{N_1, \ldots, N_m\}$, $\mathcal{T}$ is a finite set of terminal basic shapes $\{t_1, \ldots, t_n\}$, $\mathcal{R}$ is a finite set of rules $\{r_1, \ldots, r_l\}$ and $S \in \mathcal{N}$ is the starting shape (axiom).

Terminal and non-terminal symbols of the grammar are called *basic shapes*. They have a semantic type from a finite subset $\mathcal{C}$, (e.g., window, balcony or floor), and a bounding box. The vector of attributes of a basic shape of type c at position (x, y) with width w and height h is denoted as (c, x, y, w, h). (Note that x and y denote here pixel coordinates in the image, not features.)

A rule $r : A \to B_1 B_2 \ldots B_k$ splits a single non-terminal basic shape A along a selected dimension into a sequence of basic shapes $(B_i)_{1 \leq i \leq k}$. For example, a vertical split rule decomposes a basic shape into multiple chunks of basic shapes along the y axis. The grammar can be transformed into Chomsky Normal Form, so that each rule applies at

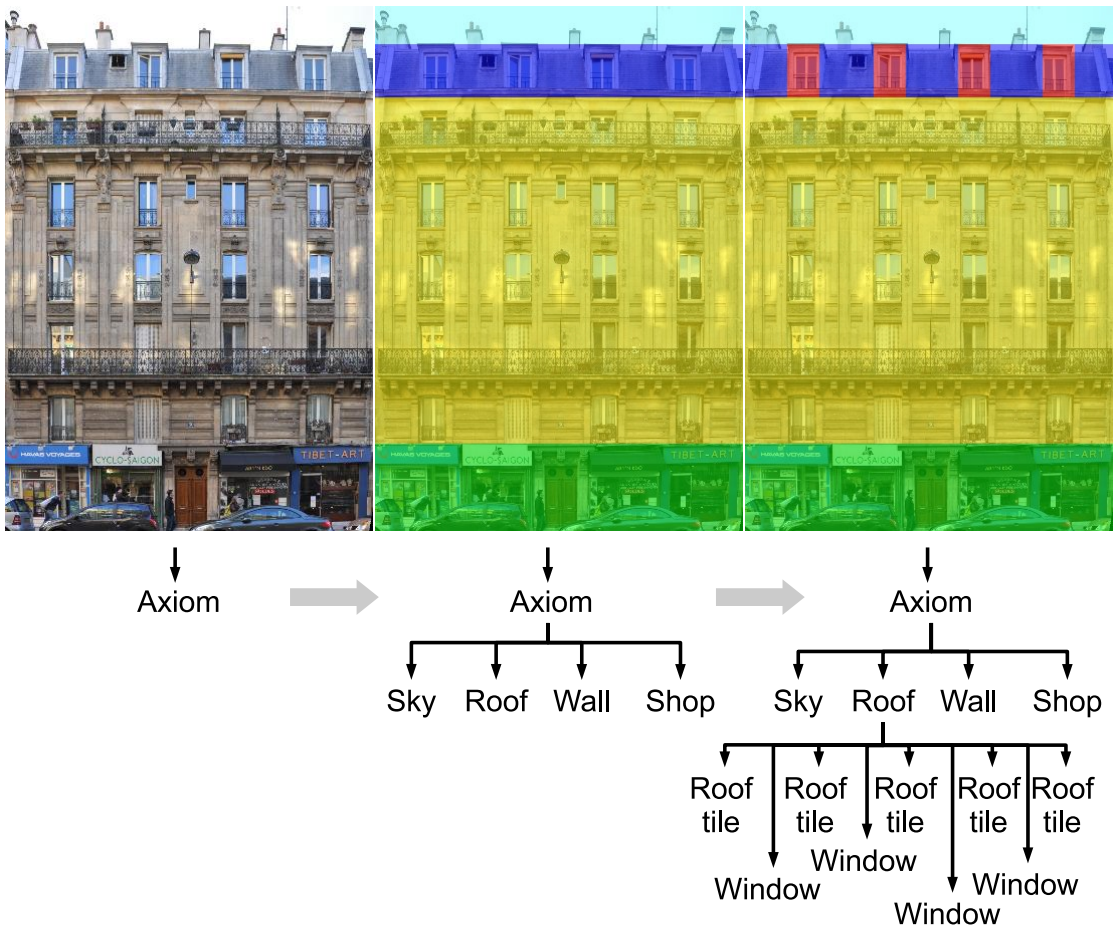


Figure 8.1: Top-down construction of a derivation tree. Top: the input image with the overlaid symbols. Bottom: the derivation tree under construction.

most one split to the processed basic shape. This reduces the number of continuous attributes of a production rule to one.

The generation process starts by applying a production rule to the axiom and continues applying production rules to the non-terminal basic shapes until there are only terminals in the derived configuration. The application of a rule requires the selection of the rule and the determination of its attributes, i.e., the number and positions of the splits. Generation is thus a sequence of decisions. It constructs a *derivation tree*. The root of the tree is the axiom S and all the nodes correspond to basic shapes, with terminal basic shapes at the leaves. An operation on a non-terminal node is performed by attaching to it the children nodes. Figure 8.1 illustrates the first two steps of a derivation process for an exemplary split grammar.

The idea of parsing is to construct a derivation tree corresponding to a given configu-

ration of terminal shapes. It can be performed in a top-down or bottom-up fashion. In the first case, the derivation tree is constructed starting from the axiom; in the second case, the leaves of the tree are instantiated first and combined recursively.

In many practical facade parsing applications, the input data consists of a rectified facade image. The goal is to segment the image into a configuration of semantically meaningful regions that is allowed by the grammar. The grammar presented in TEBOUL et al. (2011) represents the Haussmannian architecture of the XIXth century buildings in Paris. The set of terminals includes sky, roof, shop, door, window, wall and balcony areas. As in TEBOUL et al. (2011), choosing a relevant split grammar is an issue we do not address. We assume the split grammar is known and written beforehand.

8.3.2 Reinforcement Learning for Top-Down Parsing

To assess the quality of a derivation tree, a pixel-wise merit function is computed first. The merit function $m(x, y, c) \in [0, 1]$ estimates the likelihood that a pixel at (x, y) is of semantic type c. In TEBOUL et al. (2011), a random forest (RF) classifier is used to estimate m. The parser's goal is to find a derivation tree T that maximizes the *cumulated reward*, defined over all the terminals and the input image as: $\sum_t M(t)$. $M(t)$ is the reward for a single terminal $t = (c, x, y, w, h)$ and cumulates the reward over all pixels covered by the terminal:

$$M(t) = \sum_{x'=x}^{x+w} \sum_{y'=y}^{y+h} m(x', y', c) \; . \tag{8.1}$$

The task is difficult because the effect of decisions taken on the non-terminal nodes is not known until the terminal nodes are instantiated. In TEBOUL et al. (2011), the problem is formulated in terms of a Markov Decision Process. The parser acts as an agent constructing a derivation tree. Such tree has a state $s = (T, N)$ which consists of the tree T under construction and the next non-terminal node N to be processed.

Speaking in terms of reinforcement learning language, the parser learns a policy function $\pi(s, a)$, which is the probability of "choosing action a" in state s. Using the terminology of grammar parsing, "action a" merely corresponds to "rule attribute a", e.g., if the next non-terminal shape/node

$$N = (c, x, y, w, h)$$

has to be processed with a horizontal split rule, then the rule attribute a is nothing more than a parameter

$$0 \leq a \leq w$$

that splits the shape N into two child shapes/nodes

$$N_1 = (c, x, y, a, h) \text{ and } N_2 = (c, x + \Delta_x, y, w - a, h).$$

By repeatedly simulating derivation trees with the current policy function $\pi(s, a)$, the parser updates at each iteration $\pi(s, a)$ according to the history of cumulated rewards,

which are determined when derivation trees are complete. Denoting the current best optimal attribute by a^* and the prior distribution of rule attributes by $P(a|s)$, the policy function takes the form

$$\pi(s,a) = (1-\varepsilon)\delta(a,a^*) + \varepsilon P(a|s), \tag{8.2}$$

where δ is the Kronecker delta and ε is a parameter of the algorithm. Specifically, the policy function either chooses with probability $1-\varepsilon$ the best rule attribute a^*, learnt "from experience", or draws another rule attribute a from the prior distribution $P(a|s)$ with probability ε.

After a number of iterations, $\pi(s,a)$ converges to the optimal policy function, which in turn can generate a derivation tree yielding the highest reward. In order to speed up the convergence, a data-driven version of the algorithm is used, where $P(a|s)$ is determined from the bottom-up cues. The prior distribution $P(a|s)$ for split locations is generated by marginalizing the horizontal and vertical gradient magnitudes along the y and x directions of the image.

8.3.3 Improved Bottom-up Cues for Façade Parsing

The algorithm proposed by TEBOUL et al. (2011) only utilizes local, pixel- or patch-based information. In particular, this is the case of the merit function, which lacks robustness. Such a limitation is significant in the case of buildings, because of possibly high variations of facade color and lighting conditions of image acquisition. Within the framework, it is also not possible to benefit from the fact that in a single image, elements of the same type may share similar appearance; e.g., the pixel classifier in TEBOUL et al. (2011) looks for any kind of window at any position. It does not reinforce the specific similarity of windows and window surroundings in a given façade. Besides, the random nature of the Q-learning algorithm used in TEBOUL et al. (2011) results in significant result variations from run to run. In the following part of this chapter we propose modifications to the algorithm to address these drawbacks.

Instead of constraining the bottom-up information to a local, low-level merit function, we propose to also use an object detector and a robust pattern search method (cf. Chapter 7). Our algorithm may thus exploit the repetition of specific instances of architectural elements within the facade. To better guide the parsing, we also design discriminative distributions of parsing actions using object detection and line segment cues. As a result, the parser not only better locates elements but also prunes the solution space much more selectively.

In the rest of the chapter, we describe how we construct our improved merit function, the new distribution for split positions from the results of our repetitive pattern search obtained in Chapter 7. Then we provide and discuss experimental results.

8.4 Enhanced Merit Function

We propose a new, more robust and more accurate merit function, which combines the local, low-level (pixel- or patch-based) information with standalone, high-level object

detection. We interpret a detector for a class d as a pixel classifier $w_d(x, y)$, the value of which is 1 if (x, y) is a pixel belonging to the detected object and 0 otherwise. Not all semantic categories can have a sensible detector in practice. For instance, a classifier can be trained to detect windows or doors, but it is much harder to practically and reliably detect walls or roofs. Moreover, although the semantic types of terminals have no intersection in this kind of grammar, actual detectors can locate objects that overlap several semantic types of the grammar. For example, a general window model for detection can encompass in the grammar both window-only areas and cast iron balconies in front of windows. We thus make a difference between the semantic classes C of the grammar and the semantic classes D of the detectors, and define $c(d)$ as the classes of $\mathcal{C}$ that have an intersection with class $d \in \mathcal{D}$. In case we have several detectors, for classes in D, we define

$$c(D) \stackrel{\text{def}}{=} \bigcup_{d \in D} c(d). \tag{8.3}$$

The improved merit function m_+ gives confidence to the high-level detectors over the underlying, low-level merit: in case of a detection at a given pixel, it zeroes the merit of undetected classes, and the merit is renormalized. More formally, let

$$D_{x,y} \stackrel{\text{def}}{=} \{d \in D \mid w_d(x, y) = 1\} \tag{8.4}$$

be the set of detected classes at pixel (x, y). We define

$$m_+(x, y, c) \stackrel{\text{def}}{=} m(x, y, c) \text{ if } D_{x,y} = \emptyset, \tag{8.5}$$

i.e., it is unchanged where there are no detection. Otherwise, if $D_{x,y} \neq \emptyset$, then $m_+(x, y, c)$ is defined as

$$m_+(x, y, c) \stackrel{\text{def}}{=} \begin{cases} \dfrac{m(x, y, c)}{\displaystyle\sum_{c' \in c(D_{x,y})} m(x, y, c')} & \text{if } c \in c(D_{x,y}) \\ 0 & \text{otherwise.} \end{cases} \tag{8.6}$$

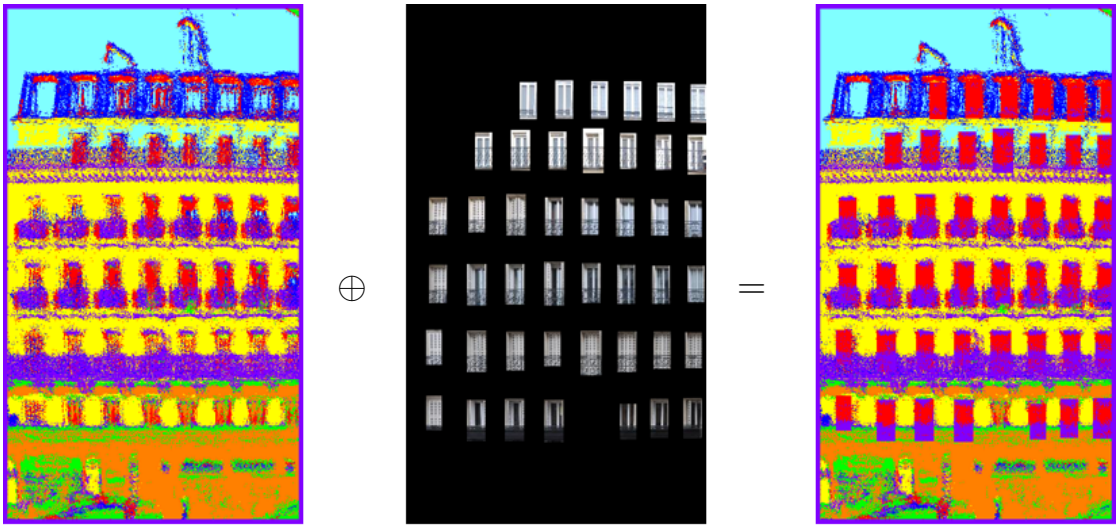
In our experiments we trained a general window detector that also localizes windows with a cast iron balcony in the foreground. We thus have $D = \{\text{whole-window}\}$ and

$$c(D) = \{\text{window}, \text{balcony}\}.$$

Figure 8.2 illustrates the improved merit function. We display

$$m^*(x, y, c) \stackrel{\text{def}}{=} \arg\max_c m(x, y, c)$$

with different colors for different classes (and likewise for m^*_+) as well as an image illustrating $w_{\text{whole-window}}$ with patches of the original image in places where whole windows have been detected.



$m^*(x,y)$ $\quad \oplus \quad$ $w_{\text{whole-window}}(x,y)$ $\quad = \quad$ $m^*_+(x,y)$

Figure 8.2: Classification based on the local merit function (left) vs the higher-level merit function (right). The result of the window detection is presented in the middle.

8.5 Enhanced Distribution of Split Positions

Our last contribution is the design of more discriminative distributions of parsing actions $P(a|s)$ for the policy function $\pi(s,a)$. The most critical parsing action is the choice of the split position that decomposes a basic shape in the optimal manner. We consider two distributions for the split positions: one for horizontal splits and one for vertical splits. In TEBOUL et al. (2011), these distributions are obtained by accumulating gradients in the image along the x and y axes. However, these marginal distributions are noisy because of the harmful accumulation of gradients not corresponding to objects of interest, but resulting from shadows, texture or small architectural details. We propose another approach, based on marginalizing the distribution of line segments detected in the image. As illustrated by our experiments, these higher-level abstractions are better split indicators.

We first detect line segments L in the image. (In our experiments we use GROMPONE VON GIOI et al. (2010)'s line segment detector.) Let $v(y)$ be the distribution of vertical split positions. We denote by $[a_l, b_l]$ the projection of a segment $l \in L$ on the vertical axis, and by θ_l its angle with respect to horizontality. The value of the distribution at height y is computed as follows:

$$v(y) = C \sum_{l \in L} \mathbb{1}_{y \in [a_l, b_l]} \exp(-\frac{\tan^2 \theta_l}{2\sigma^2}) \,, \tag{8.7}$$

where σ is a parameter of the distribution and C is a normalization constant. The

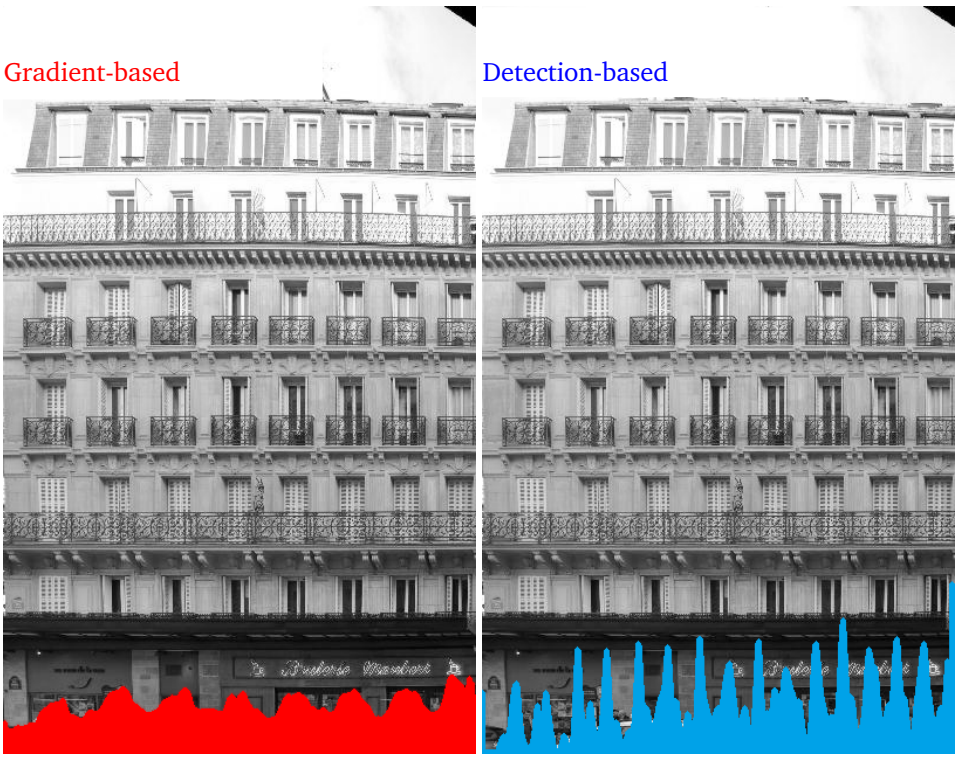


Figure 8.3: The gradient-based vs the detection-based distribution of split positions.

definition is symmetrical for horizontal splits. For our experiments we set $\sigma = 0.06$, which roughly leads to a segment contribution of $\frac{1}{3}$ for a segment with a $5°$ angle, whereas a perfectly axis-aligned segment contributes for 1. To reduce computation time, line segments with an angle beyond a threshold (around $10°$ for the given σ value) can be discarded right after detection.

In the same manner, we build a normalized histogram of the contours of the detected objects. The two distributions are summed and the resulting histogram is normalized, yielding the final distribution of applicable split positions. The major benefit of this approach is that the exploration of the solution space is significantly pruned and the splits are attracted to optimal positions. The parser avoids being stuck in local minima and the temporal standard deviation of the energy decreases over time faster than for the original algorithm (see Figures 8.3 and 8.4).

8.6 Experimental Validation on Façade Parsing

Our experimental validation is based on the *ECP Benchmark 2011* datasets (TEBOUL 2010), that picture rectified Haussmanian buildings annoted with 7 semantic classes: sky, roof, wall, window, balcony, shop, door. We have shown in Chapter 7 that our window

pattern search obtained good window localization results on the *ECP CVPR 2010* and *ECP Benchmark 2011* datasets (TEBOUL 2010).

To evaluate our approach, we ran the modified shape grammar parser on the test set of the ECP Benchmark 2011 dataset[1] (104 images). The window detector we used is CC($\tau = 20$)+PS, that experimentally performs best (see Table 7.1). We compare this parser against the original one, presented in TEBOUL et al. (2011). In each case we run the parsers once. The results are evaluated with use of the ground truth annotations accompanying the dataset.

We present the results of the comparison in the form of the confusion matrices. The detection rate of building elements corresponds to the diagonal entry of the matrix (see TEBOUL et al. (2011) for details). Table 8.1 shows the efficiency of our two contributions separately. Consistent improvement of the results over the whole range of classes is visible already even for the partial contributions of the refined merit function and the new distribution of split positions separately. The improvement is amplified when we combine the two modifications into our final algorithm (bottom-right matrix). In particular, the window detection improves from 60% to 85% while most other rates are improved or preserved. Our algorithm also shows better convergence properties than the original one. In Figure 8.4 we show that the proposed algorithm converges faster, attains better values of the reward function and is less prone to deviate from the optimal solution. A few actual results are illustrated in Figure 8.5. More results can be found in Appendix F, which compares TEBOUL et al. (2011) and our approach.

[1] This dataset must *not* be confused with *the ECP CVPR 2010 dataset* which consists of 10 test images only. Hence the numbers differ from what is reported in TEBOUL et al. (2011).

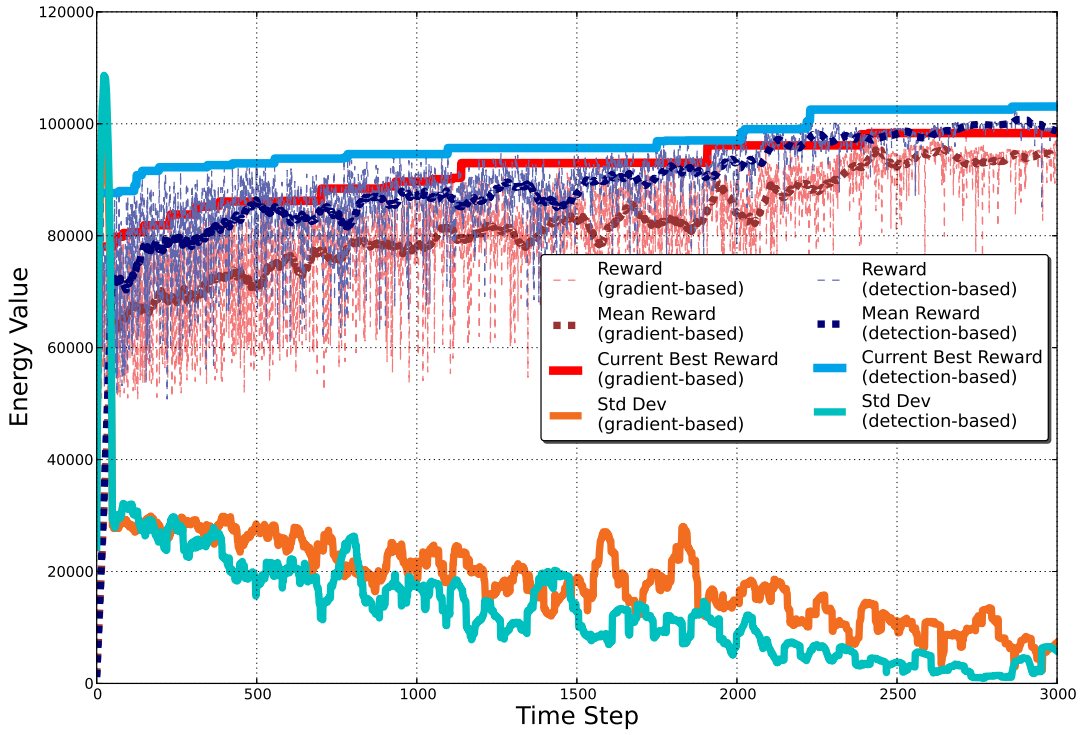


Figure 8.4: The convergence of our algorithm with respect to TEBOUL et al. (2011). The plot shows the evolution of the reward, the current best reward and the standard deviation of the reward over time for a single run of the parser. The 'mean reward' is the plot of the reward function smoothed over time, to eliminate the high-frequency variation.

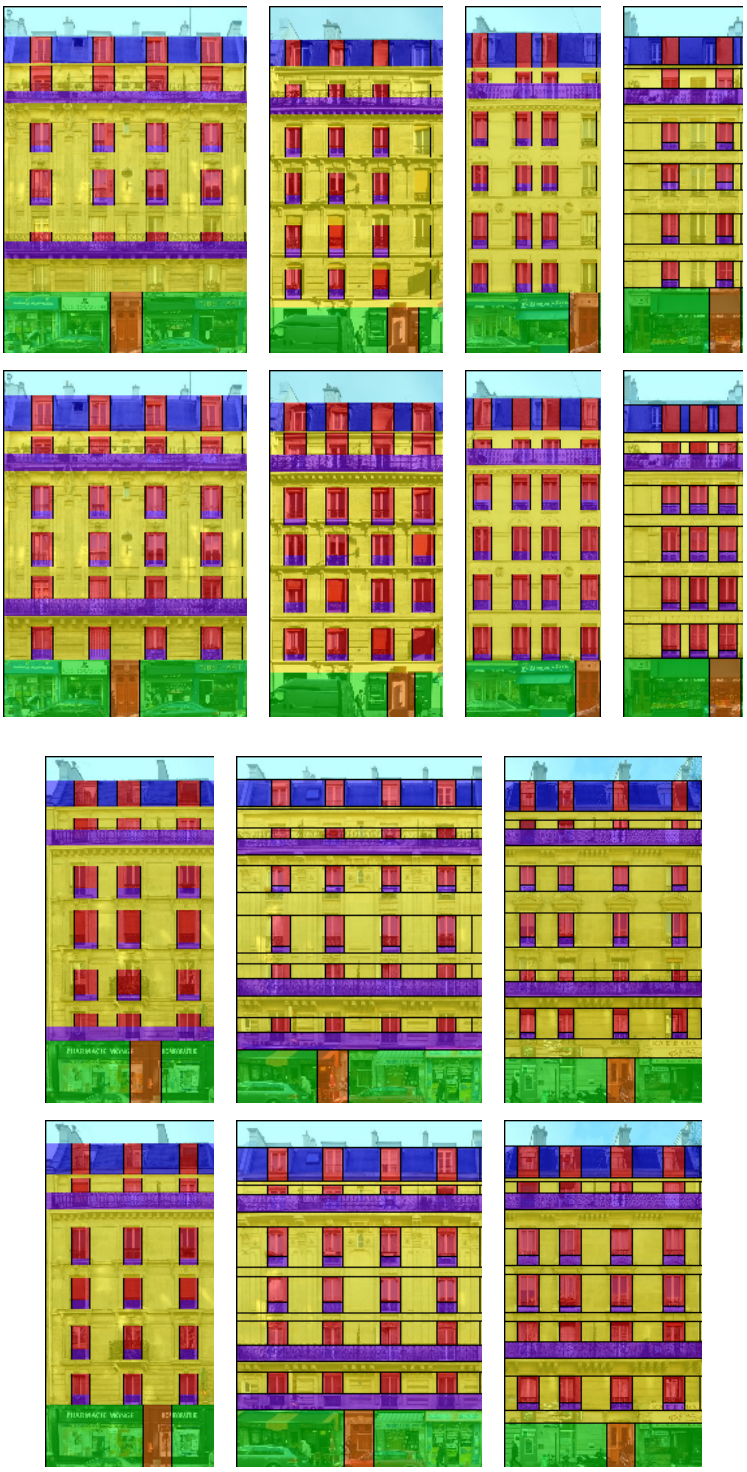


Figure 8.5: Examples of images for which our parser outperforms the original one (best viewed using magnification). Odd rows: results of original parser (TEBOUL et al. 2011). Even rows: our modified parser.

	Gradient-Based Action Distribution								*Detection-Based Action Distribution*								*Type*
Plain RF merit	**60**	30	4	0	4	2	0		**72**	20	3	0	2	2	0	+12	*window*
	6	**81**	10	0	2	0	1		5	**84**	8	0	2	0	1	+3	*wall*
	16	31	**50**	0	2	0	1		18	26	**53**	0	2	0	1	+3	*balcony*
	0	1	1	**51**	0	0	48		0	2	0	**45**	0	0	53	−6	*door*
	13	3	0	0	**74**	10	0		8	2	0	0	**81**	9	0	+7	*roof*
	3	0	0	0	6	**91**	0		3	0	0	0	5	**92**	0	+1	*sky*
	0	7	3	8	0	0	**82**		0	8	2	8	0	0	**81**	−1	*shop*
Detection-enhanced merit	**71**	16	8	0	2	2	0	+11	**85**	8	5	0	1	2	0	+25	*window*
	9	**73**	15	0	2	0	0	−8	8	**76**	13	0	2	0	0	−5	*wall*
	15	27	**56**	0	2	0	1	+6	19	19	**60**	0	1	0	0	+10	*balcony*
	0	2	0	**55**	0	0	43	+4	0	2	0	**52**	0	0	45	+1	*door*
	16	2	0	0	**71**	10	0	−3	12	2	1	0	**77**	7	0	+3	*roof*
	4	0	0	0	6	**90**	0	−1	4	0	0	0	5	**91**	0	0	*sky*
	0	7	4	7	0	0	**81**	−1	0	7	3	7	0	0	**82**	0	*shop*

Table 8.1: The average confusion matrices. Top-left: the original algorithm. Top-right: the algorithm with the new action distribution. Bottom-left: the algorithm based on the modified merit function. Bottom-right: our final algorithm with both modifications combined.

Chapter 9

Conclusion and Perspectives

In this chapter, we would like to summarize the main contributions of the thesis.

In Chapter 4, we have proposed a mathematical formalization of the geometry consistency which is enforced at two levels, namely, at the level of the vicinity of a feature match, and at the level of a whole geometry-consistent region. Our 4th-order constraint constitutes the key stone in our geometry consistent formulation. Indeed, this constraint is used to the notion of affine consistent region and rigorously linked with a match propagation scheme. In addition, we provide an extensive study of the feature detector-descriptor repeatability to know how to assess the geometry consistency at the local scale of a feature.

In Chapter 5, from our geometric formulation, we have then derived a match propagation method that enforces photometric and geometric consistency. We validated the choice of parameters that controls our match propagation. Our method has been evaluated in terms of precision and empirical evidence of scalability are provided. As demonstrated on a wide range of experiments in Chapter 6, it is efficient, scalable, accurate and robust, even in the presence of high ambiguity, improving over other existing methods.

Besides, our algorithm is well suited for repeated pattern detection as it can be used as a standalone algorithm for detecting multiple object instances in images (cf. Chapter 7). Our pattern detector is shown to significantly outperforms existing methods in accurate window localization in a façade. In such situation, the algorithm can be viewed as an adaptive detector that adjusts to the specific appearance of the repeated object.

In turn, the pattern detection results prove to be extremely valuable cues for façade image parsing. We show in Chapter 8 how to efficiently exploit such high-level bottom-up cues to enhance top-down facade parsing. It is based on the parser presented in TEBOUL et al. (2011) and carries two significant contributions with respect to the original version: the use of robust and adaptive object detectors to better estimate the merit function used by the parser, and the use of detected objects as well as line segments to better determine split positions, which effectively guides the reinforcement learning algorithm and speeds up its convergence. The significant performance improvements that we observe experimentally demonstrate the importance of high-level bottom-up cues in

top-down parsing.

Let us conclude this thesis with perspectives. In particular, there is still room for improvements in our feature correspondence method, which actually constitutes the central piece of the thesis. This mainly concerns our 4th-order constraint, which formulates a position-, shape- and orientation- consistency under a local affinity constraint.

- While being extremely robust, it is still challenging to get round the enumeration issue involved in our match propagation method. In our implementation, we do not try to enumerate all possible quadruples. Finding one good quadruple is sufficient to propagate matches. This is what our implementation does, but it does not guarantee that it always finds a good quadruple when there is one. It would be interesting to design a principled and efficient method which selects good quadruples without resorting to brute-force enumeration.

- Finally, our 4th-order constraint depends on learnt thresholds. These thresholds varies from one specific feature detector to another, each feature detector coming with specific precision and repeatability. The relevance of the learnt thresholds are very dependent of the training images. We recall that we used MIKOLAJCZYK et al. (2005)'s datasets to learn these thresholds. However, as explained in Chapter 4, these datasets evaluate the feature detector on rigid scenes only. This is actually a very limited baseline and we actually use more permissive threshold values than those found by learning.

  Besides, these thresholds are fixed once for all. Fortunately, this does not prevent from obtaining good results in deformable object matching. Perhaps thresholds should also depend on the strength of local deformation for better matching deformed objects.

- To conclude, it would be interesting to investigate an equivalent energy formulation which is practical to solve.

Chapter 10

Conclusion et Perspectives (French)

Afin de conclure, nous récapitulons encore une fois les contributions de cette thèse.

Dans le Chapitre 4, nous avons proposé une formalisation mathématique de la cohérence géométrique à deux niveaux d'échelle, à savoir, d'une part au niveau du voisinage d'une correspondance locale et d'autre part au niveau d'une région géométriquement cohérente, notre contrainte d'ordre 4 constituant la base d'une telle formalisation mathématique. En effet, cette contrainte est utilisée pour definir la notion de région affine-cohérente et rigoureusement liée à un processus de croissance de régions. De plus, nous fournissons une étude détaillée sur la répétabilité de détecteurs-descripteurs afin de savoir comment évaluer la cohérence géométrique à l'échelle locale d'une caractéristique visuelle.

Dans le Chapitre 5, à partir de la formulation de cohérence géométrique, nous en avons dérivé une méthode de propagation de correspondances qui assurent à la fois cohérence photométrique et géométrique. Notre méthode a été évaluée en terme de précision et nous avons mis en évidence sa capacité à passer à l'échelle de manière empirique. Comme démontré dans un large éventail d'expériences dans le Chapitre 6, notre méthode s'avère algorithmiquement efficace, adaptée à des problèmes de grande dimension, précise et robuste pour des ensembles de correspondances très contaminées et massivement ambiguës, améliorant ainsi l'état de l'art.

Par ailleurs, notre algorithme peut être réadapté comme un algorithme de détection d'éléments répétés dans les images (cf. Chapter 7). Nous démontrons que notre détecteur d'éléments répétés fait mieux que les méthodes existantes dans la localisation précise de fenêtres sur une façade. D'une certaine manière, l'algorithme peut être vu comme un détecteur adaptatif qui s'ajuste à l'apparence spécifique de l'élément répété.

Ensuite, les résultats de détection de modèles visuels répétés s'avèrent être des informations particulièrement précieuses pour l'analyse d'images de façades. Nous montrons dans le Chapitre 8 comment exploiter ces informations pour l'analyse de façades. En se basant sur le parseur de TEBOUL et al. (2011), nous proposons deux contributions: l'utilisation de détecteurs robustes et adaptatifs pour produire une meilleure information *a priori*, qui servira ensuite à évaluer la représentation proposée par le parseur; la

combinaison des résultats de détections d'objets répétés et des lignes pour analyser mieux et plus vite la structure des façades pendant la phase d'optimisation. Des améliorations très nettes ont été obtenues dans l'analyse des façades, ce qui souligne bien l'importance des informations haut-niveau.

Enfin concluons cette thèse par des perspectives. En particulier, notre méthode de mise en correspondance, qui constitue de fait la principale contribution de cette thèse, peut encore bénéficier d'améliorations notables. Notamment, nous portons notre attention sur notre contrainte d'ordre, qui exprime la cohérence entre les positions, les formes et les orientations correspondantes sous contrainte d'une transformation localement affine.

- Bien qu'elle s'avère très robuste, notre contrainte d'ordre 4 ainsi conçue reste un ingrédient coûteux au niveau de l'efficacité algorithmique. Contourner le problème d'énumération de quadruplets reste un problème ouvert. Dans notre implémentation, nous n'essayons pas d'énumérer tous les quadruplets possibles. Nous nous contentons d'en trouver un bon mais notre algorithme ne garantit pas de trouver systématiquement un bon quadruplet s'il en existe un. Il serait intéressant de concevoir une méthode rigoureuse et efficace pour sélection des bons triplets sans avoir recours à une énumération par force brute.

- Enfin, notre contrainte d'ordre 4 dépend de seuils appris. Ces seuils varient d'un détecteur de caractéristiques visuelles à un autre, chaque détecteur possédant leur répétabilité et précision qui leur est propre. La pertinence des seuils appris dépendent toutefois du choix des images d'entraînement. Rappelons que nous avons utilisé les jeux de données de MIKOLAJCZYK et al. (2005) pour apprendre ces seuils. Toutefois, comme expliqué dans le Chapitre 4, ces jeux de données évaluent le détecteur sur des scènes rigides seulement. De fait, la pertinence des seuils en est amoindrie et nous avons fixé des seuils beaucoup plus permissifs que ceux appris à la phase d'apprentissage.

  Bien que ces seuils sont fixés une fois pour toutes, nous avons pu malgré tout obtenir de très bons résultats pour mettre en correspondance des objets déformés. Mais peut-être les seuils doivent également dépendre de la force de déformation locale pour mieux mettre en correspondance les objets déformés.

- Enfin, trouver une formulation énergétique pratique et équivalente à notre formulation reste une piste de recherche intéressante.

PART III

Appendix

Chapter A

Ellipse Intersections

EBERLY (2008) provides a comprehensive study on the computation of ellipses intersection, namely the computation of its area and its intersection points. This is a non-trivial geometric problem. We complement EBERLY (2008)'s study with some additional technical details about the area computation of two intersecting ellipses.

A.1 Origin-Centered Axis-Aligned Ellipses

Let $\mathcal{E}$ be an ellipse with semi-major axis a and semi-minor axis b. Let us first suppose that $\mathcal{E}$ is centered at the origin and is axis-aligned oriented, i.e., such that the axis a is along the x-axis and the axis b along the y-axis. Then the equation of ellipse $\mathcal{E}$ is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{A.1}$$

A.1.1 Ellipse Area

By using symmetry property of the ellipse, the area of ellipse $\mathcal{E}$ is 4 times the upper quadrant area of the ellipse, i.e.

$$\text{area}(\mathcal{E}) = 4\int_0^a y(x)\,\mathrm{d}x = 4b\int_0^a \sqrt{1-\frac{x^2}{a^2}}\,\mathrm{d}x = \pi ab \tag{A.2}$$

The integral is the limit of the Riemann sum as illustrated in Figure A.1.

Let us detail the computation. We use the $\mathcal{C}^1$-diffeomorphism change of variable $\frac{x}{a} = \sin\theta$ which is valid for $[0,a] \to [0,\pi/2]$. (A $\mathcal{C}^1$-diffeomorphism is an invertible differentiable function with continuous derivative.)

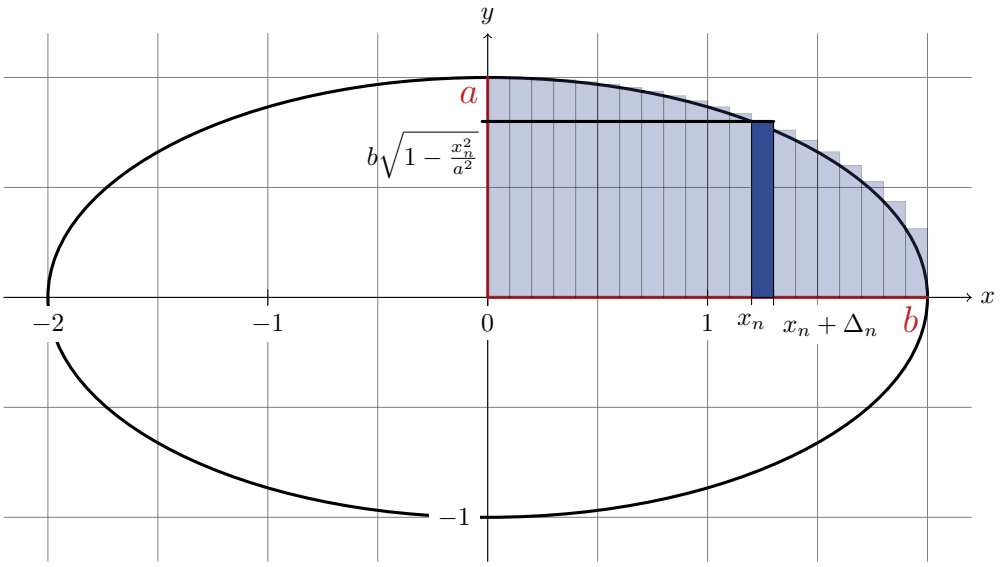


Figure A.1: Riemann sum approximating the upper quadrant area of the ellipse.

Differentiating, $\mathrm{d}x = a\cos(\theta)\,\mathrm{d}\theta$, and hence,

$$\begin{aligned}\mathrm{area}(\mathcal{E}) &= 4ab\int_0^{\pi/2}\cos^2(\theta)\,\mathrm{d}\theta \\ &= 4ab\int_0^{\pi/2}\frac{1+\cos(2\theta)}{2}\,\mathrm{d}\theta \\ &= 4ab\left[\frac{x}{2}+\frac{\sin(2\theta)}{4}\right]_0^{\pi/2} \\ &= \pi ab.\end{aligned}$$

A.1.2 Area of an Elliptical Sector

In this part, we review the computation of the area of an ellipse sector. It has already been covered in EBERLY (2008) but computation details are omitted in EBERLY (2008).

The elliptic sector area is delimited in polar coordinates by $[\theta_1, \theta_2]$ (with $\theta_1 < \theta_2$) as illustrated in Figure A.2. Using polar coordinates, it equals to the following nonnegative integral

$$A(\theta_1, \theta_2) = \frac{1}{2}\int_{\theta_1}^{\theta_2} r^2\,\mathrm{d}\theta\,. \tag{A.3}$$

The change of variable in polar coordinates is $x = r\cos\theta$ and $y = r\sin\theta$ and, thus with Equation (A.1), $\dfrac{r^2\cos^2(\theta)}{a^2} + \dfrac{r^2\sin^2(\theta)}{b^2} = 1$, therefore $r^2 = \dfrac{a^2b^2}{b^2\cos^2(\theta) + a^2\sin^2(\theta)}$.

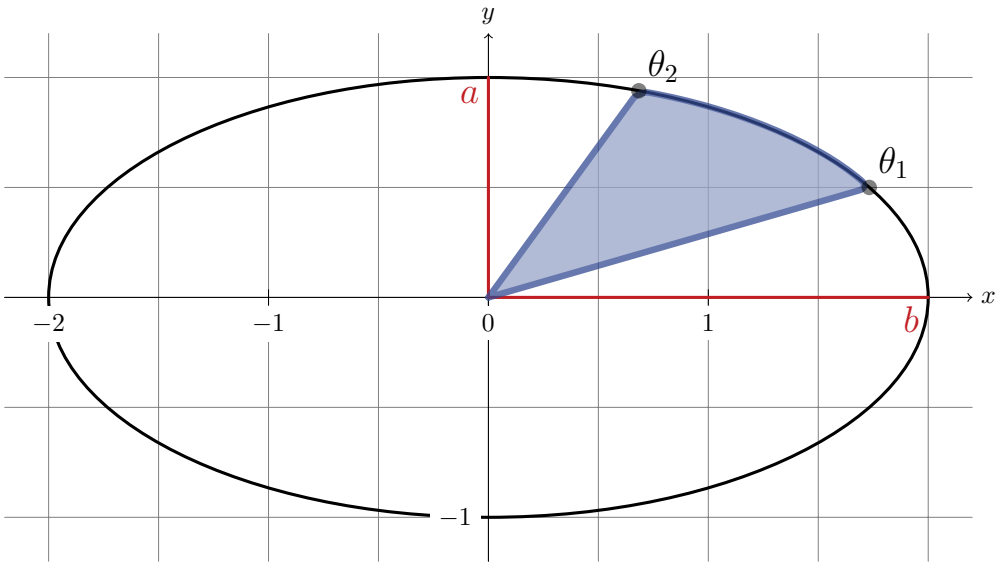


Figure A.2: The ellipse sector delimited by the polar angles (θ_1, θ_2) is colored in blue

Plugging the formula of r in the integral,

$$A(\theta_1, \theta_2) = \frac{a^2 b^2}{2} \int_{\theta_0}^{\theta_1} \frac{\mathrm{d}\theta}{b^2 \cos^2(\theta) + a^2 \sin^2(\theta)} \tag{A.4}$$

Now, the integrand $\frac{\mathrm{d}\theta}{b^2 \cos^2(\theta) + a^2 \sin^2(\theta)}$ is invariant by the transformation $\theta \mapsto \theta + \pi$, i.e.,

$$\frac{\mathrm{d}\theta}{b^2 \cos^2(\theta) + a^2 \sin^2(\theta)} = \frac{\mathrm{d}(\theta + \pi)}{b^2 \cos^2(\theta + \pi) + a^2 \sin^2(\theta + \pi)}.$$

According to Bioche's rule, a relevant change of variable is the $\mathcal{C}^1$-diffeomorphism change of variable $t = \tan(\theta)$ which is valid for $]-\pi/2, \pi/2[\to]-\infty, \infty[$. Let us first rewrite

$$\begin{aligned} A(\theta_1, \theta_2) &= \frac{a^2 b^2}{2} \int_{\theta_1}^{\theta_2} \frac{\mathrm{d}\theta}{b^2 \cos^2(\theta) + a^2 \sin^2(\theta)} \\ &= \frac{a^2 b^2}{2} \int_{\theta_1}^{\theta_2} \frac{\frac{\mathrm{d}\theta}{\cos^2(\theta)}}{b^2 + a^2 \tan^2(\theta)} \\ &= \frac{\cancel{a^2} b^2}{2} \int_{\theta_1}^{\theta_2} \frac{\frac{\mathrm{d}\theta}{\cos^2(\theta)}}{\cancel{a^2}(b/a)^2 + \tan^2(\theta))} \end{aligned}$$

Differentiating $t = \tan\theta$, $\mathrm{d}t = \frac{\mathrm{d}\theta}{\cos^2(\theta)}$, thus

$$\begin{aligned} A(\theta_1, \theta_2) &= \frac{b^2}{2} \int_{\tan\theta_1}^{\tan\theta_2} \frac{\mathrm{d}t}{(b/a)^2 + t^2} \\ &= \frac{b\cancel{^2}}{2} \left[\frac{a}{\cancel{b}} \arctan\left(\frac{a}{b}t\right)\right]_{\tan\theta_1}^{\tan\theta_2} \\ &= \frac{ab}{2} \left[\arctan\left(\frac{a}{b}t\right)\right]_{\tan\theta_1}^{\tan\theta_2} \\ &= \frac{ab}{2} \left(\arctan\left(\frac{a}{b}\tan\theta_2\right) - \arctan\left(\frac{a}{b}\tan\theta_1\right)\right) \end{aligned}$$

Hence,

$$A(\theta_1, \theta_2) = \frac{ab}{2} \left(\arctan\left(\frac{a}{b}\tan\theta_2\right) - \arctan\left(\frac{a}{b}\tan\theta_1\right)\right) \tag{A.5}$$

Warning: The integral is properly defined for $(\theta_1, \theta_2) \in]-\pi/2, \pi/2[$. But, using symmetry properties of the ellipse, we can easily retrieve the elliptical sector for any $(\theta_1, \theta_2) \in]-\pi, \pi[$.

Alternatively, EBERLY (2008) provides a more convenient antiderivative because it is defined in $]-\pi, \pi]$ as follows

$$F(\theta) = \frac{ab}{2} \left[\theta - \arctan\left(\frac{(b-a)\sin 2\theta}{(b+a) + (b-a)\cos 2\theta}\right)\right]. \tag{A.6}$$

Hence, the elliptic sector area equals to the following *nonnegative* quantity

$$\forall(\theta_1, \theta_2) \in]-\pi, \pi],\ A(\theta_1, \theta_2) = |F(\theta_2) - F(\theta_1)|\,. \tag{A.7}$$

A.1.3 Area Bounded by a Line Segment and an Elliptical Arc

We are interested in computing the elliptic portion by a line segment and the elliptical arc (θ_1, θ_2) such that

$$|\theta_2 - \theta_1| \leq \pi$$

This condition *is* important as a such elliptic portion always corresponds to the blue elliptic portion in Figure A.3. Let us denote the area of such portion by $B(\theta_1, \theta_2)$. Geometrically, we see that, if $|\theta_2 - \theta_1| \leq \pi$, then

$$\begin{aligned} B(\theta_1, \theta_2) &= \mathrm{area}(\mathrm{sector}(\theta_1, \theta_2)) - \mathrm{area}(\mathrm{triangle}(\theta_1, \theta_2)) \\ &= A(\theta_1, \theta_2) - \frac{1}{2}|x_2 y_1 - x_1 y_2| \end{aligned}$$

where $(x_i, y_i) = (r_i \cos\theta_i, r_i \sin\theta_i)$ and $r_i = \dfrac{ab}{\sqrt{b^2\cos^2(\theta_i) + a^2\sin^2(\theta_i)}}$ for $i = \{1, 2\}$.

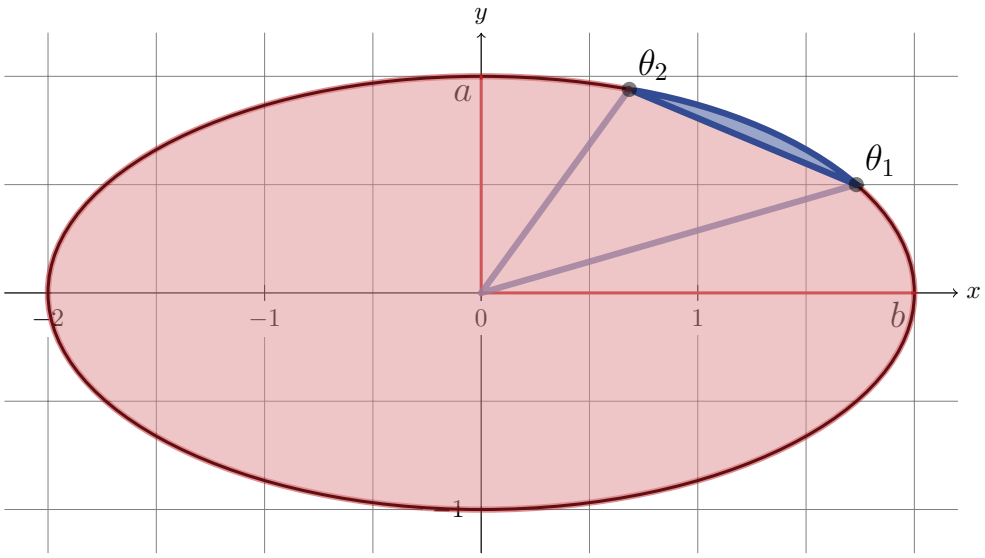


Figure A.3: The ellipse sector bounded by a line segment and the elliptical arc (θ_1, θ_2) is colored in blue.

Note that the other portion corresponding to the red one in Figure A.3 has an area which equals to $\pi ab - B(\theta_1, \theta_2) \geq B(\theta_1, \theta_2)$ if $|\theta_2 - \theta_1| \leq \pi$.

To summarize, our portion of interest, illustrated by the blue elliptic portion in Figure A.3, has an area which equals to

$$\forall (\theta_1, \theta_2) \in]-\pi, \pi], \; B(\theta_1, \theta_2) = \begin{cases} A(\theta_1, \theta_2) - \frac{1}{2}|x_2 y_1 - x_1 y_2| & \text{if } |\theta_2 - \theta_1| \leq \pi \\ \pi ab - A(\theta_1, \theta_2) + \frac{1}{2}|x_2 y_1 - x_1 y_2| & \text{otherwise} \end{cases}. \tag{A.8}$$

A.2 General Ellipse Parameterization

The previous sections has provided the basis for area of intersecting ellipses. However, ellipses are neither centered at the origin nor aligned with the axes of the reference frame in general. Therefore, an ellipse $\mathcal{E}$ is entirely defined by the following geometric information

- a center $\mathbf{x}_\mathcal{E}$,
- axis radii $(a_\mathcal{E}, b_\mathcal{E})$,
- an orientation $\theta_\mathcal{E}$, i.e., the oriented angle between the x-axis and the axis of radius $a_\mathcal{E}$.

or more concisely by the pair $(\mathbf{x}_\mathcal{E}, \mathbf{\Sigma}_\mathcal{E})$ where the positive definite matrix $\mathbf{\Sigma}_\mathcal{E} \in \mathcal{S}_2^{++}$ is such that

$$\mathbf{\Sigma}_\mathcal{E} = \mathbf{R}_\mathcal{E}\mathbf{D}_\mathcal{E}\mathbf{R}_\mathcal{E}^T \tag{A.9}$$

where $\mathbf{R}_\mathcal{E}$ is a rotation matrix defined as

$$\mathbf{R}_\mathcal{E} \overset{\text{def}}{=} \begin{bmatrix} \cos\theta_\mathcal{E} & -\sin\theta_\mathcal{E} \\ \sin\theta_\mathcal{E} & \cos\theta_\mathcal{E} \end{bmatrix}$$

and $\mathbf{D}_\mathcal{E}$ is the diagonal matrix defined as

$$\mathbf{D}_\mathcal{E} \overset{\text{def}}{=} \begin{bmatrix} 1/a_\mathcal{E}^2 & 0 \\ 0 & 1/b_\mathcal{E}^2 \end{bmatrix}$$

Note that Equation (A.9) is the singular value decomposition of $\mathbf{\Sigma}_\mathcal{E}$ if the axis radii satisfy $a_\mathcal{E} < b_\mathcal{E}$. Using these information, ellipse $\mathcal{E}$ can be parameterized by the following equation:

$$(\mathbf{x} - \mathbf{x}_\mathcal{E})^T \mathbf{\Sigma}_\mathcal{E} (\mathbf{x} - \mathbf{x}_\mathcal{E}) = 1 \tag{A.10}$$

Or

$$\mathbf{x}^T \mathbf{A}_\mathcal{E} \mathbf{x} + \mathbf{b}_\mathcal{E}^T \mathbf{x} + c_\mathcal{E} = 0$$

with $\mathbf{A}_\mathcal{E} = \mathbf{\Sigma}_\mathcal{E}$, $\mathbf{b}_\mathcal{E} = 2\mathbf{\Sigma}_\mathcal{E}\mathbf{x}_\mathcal{E}$ and $c_\mathcal{E} = \mathbf{x}_\mathcal{E}^T\mathbf{\Sigma}_\mathcal{E}\mathbf{x}_\mathcal{E} - 1$. Denoting $\mathbf{x}^T = [x, y]$, ellipse $\mathcal{E}$ can be defined algebraically as

$$E(x, y) = e_1 x^2 + e_2 xy + e_3 y^2 + e_4 x + e_5 y + e_6 = 0, \tag{A.11}$$

where $\mathbf{A}_\mathcal{E} = \begin{bmatrix} e_1 & e_2/2 \\ e_2/2 & e_3 \end{bmatrix}$, $\mathbf{b}_\mathcal{E}^T = [e_4, e_5]$ and $c_\mathcal{E} = e_6$. This algebraic form is the convenient one that we will use in order to compute the intersection points of two intersecting ellipses.

A.3 Intersection Points of Two Ellipses

In this section, we sketch the computation of the intersection points. Our presentation slightly differs from EBERLY (2008). First, let $(\mathcal{E}_i)_{1\leq i\leq 2}$ be two ellipses defined as

$$(x, y) \in \mathcal{E}_i \iff E_i(x, y) = e_{i1} x^2 + e_{i2} xy + e_{i3} y^2 + e_{i4} x + e_{i5} y + e_{i6} = 0 \tag{A.12}$$

The intersection points of ellipses $(\mathcal{E}_i)_{1\leq i\leq 2}$ satisfy Equation (A.12) for $i \in \{1, 2\}$, i.e., the following equation system holds for intersection points

$$\begin{cases} E_1(x, y) = 0 \\ E_2(x, y) = 0 \end{cases} \tag{A.13}$$

Now, let us rewrite $E_i(x, y)$ as a quadratic polynomial in x, i.e.

$$E_i(x, y) = e_{i1} x^2 + (e_{i2} y + e_{i4}) x + (e_{i3} y^2 + e_{i5} y + e_{i6}) = 0 \tag{A.14}$$

For convenience, we define

$$p_0(y) = e_{13}y^2 + e_{15}y + e_{16} \qquad q_0(y) = e_{23}y^2 + e_{25}y + e_{26} \tag{A.15}$$
$$p_1(y) = e_{12}y + e_{14} \qquad q_1(y) = e_{22}y + e_{24} \tag{A.16}$$
$$p_2(y) = e_{11} \qquad q_2(y) = e_{21} \tag{A.17}$$

Using the notations above, we observe that x can be computed as follows

$$\begin{aligned}(A.13) &\iff \begin{cases} p_2(y)x^2 + p_1(y)x + p_0(y) = 0 \\ q_2(y)x^2 + q_1(y)x + q_0(y) = 0 \end{cases} \\ &\implies \begin{cases} q_2(y) \times \big(p_2(y)x^2 + p_1(y)x + p_0(y)\big) = 0 \times q_2(y) \\ p_2(y) \times \big(q_2(y)x^2 + q_1(y)x + q_0(y)\big) = 0 \times p_2(y) \end{cases}\end{aligned}$$

Then subtracting the first equation from the second equation, we get

$$x = \frac{p_0(y)q_2(y) - p_2(y)q_0(y)}{p_1(y)q_2(y) - p_2(y)q_1(y)}. \tag{A.18}$$

Furthermore, Equation System (A.13) is equivalent to the following augmented equation system

$$\begin{cases} E_1(x,y) &= 0 \\ x \times E_1(x,y) &= 0 \\ E_2(x,y) &= 0 \\ x \times E_2(x,y) &= 0 \end{cases}, \tag{A.19}$$

which is equivalent to

$$\underbrace{\begin{bmatrix} p_0(y) & p_1(y) & p_2(y) & 0 \\ 0 & p_0(y) & p_1(y) & p_2(y) \\ q_0(y) & q_1(y) & q_2(y) & 0 \\ 0 & q_0(y) & q_1(y) & q_2(y) \end{bmatrix}}_{\mathbf{B}(y)} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.20}$$

We recognize a linear system in the vector $[1, x, x^2, x^3]^T$. More particularly, $[1, x, x^2, x^3]^T$ is in the nullspace of $\mathbf{B}(y)$, which then must have a zero determinant. Note that all the equations systems are *equivalent*, so Equation System (A.13) holds if and only if the determinant of $\mathbf{B}(y)$ is zero. Letting the resultant be

$$R = (p_0q_2 - p_2q_0)^2 - (p_0q_1 - p_1q_0)(p_1q_2 - p_2q_1), \tag{A.21}$$

Equation System (A.13) is equivalent to the following quartic equation in y.

$$\det(\mathbf{B}(y)) = R(y) = 0, \tag{A.22}$$

This quartic equation can be solved either by SVD from the characteristic polynomial of the companion matrix. The SVD is computed either from a direct method or from

Jacobi's iterative and numerically stable method. Instead we compute the roots with Ferrari's method. While it is a tedious method, it has the advantage of being direct. Also, we experimentally observe Ferrari's method can sometimes be numerically inaccurate in particular situations, e.g., one of the ellipse is quasi-degenerate. Therefore, some tuning may be required for numerical accuracy.

Using any polynomial solver, we get the 4 roots $(y_i)_{1\leq i\leq 4}$ of the quartic polynomial R and only keep those that are real. Finally $(x_i)_{1\leq i\leq 4}$ are deduced from Equation (A.18).

A.4 Intersection Area of Two Ellipses

Our presentation is different from EBERLY (2008) and details are added. In the rest of the section, we consider two ellipses $(\mathcal{E}_i)_{1\leq i\leq 2}$ and we respectively denote

- the axes of ellipse $\mathcal{E}_i$ by (a_i, b_i), the ellipse center by $\mathbf{x}_i$, the orientation by θ_i, and the direction vectors of axis a_i and b_i by

$$\mathbf{u}_i \overset{\text{def}}{=} \begin{bmatrix}\cos(\theta_i)\\ \sin(\theta_i)\end{bmatrix} \qquad \mathbf{v}_i \overset{\text{def}}{=} \begin{bmatrix}-\sin(\theta_i)\\ \cos(\theta_i)\end{bmatrix} \tag{A.23}$$

- the area of the elliptic portion bounded a line segment and an arc for ellipse $\mathcal{E}_i$ by B_i,
- the number of intersection points by L,
- the intersection points by $\mathbf{p}_i$ for $i \in [\![1, L]\!]$, sorted in a counter-clockwise order, i.e.

$$\forall i \in [\![1, L-1]\!], \quad \angle\left([1,0]^T, \mathbf{p}_i\right) \;<\; \angle\left([1,0]^T, \mathbf{p}_{i+1}\right) \tag{A.24}$$

  where $\angle(.,.)$ denotes the angle between two vectors in the plane $\mathbb{R}^2$.

- the polar angles of points $(\mathbf{p}_i)_{1\leq i\leq L}$ with respect to ellipses $\mathcal{E}_1$ and $\mathcal{E}_2$ by $(\phi_i)_{1\leq i\leq 2}$ and $(\psi_i)_{1\leq i\leq 2}$, i.e.

$$\forall i \in [\![1, L]\!], \phi_i \overset{\text{def}}{=} \angle(\mathbf{u}_1, \mathbf{p}_i - \mathbf{x}_1) \tag{A.25}$$

$$\forall i \in [\![1, L]\!], \psi_i \overset{\text{def}}{=} \angle(\mathbf{u}_2, \mathbf{p}_i - \mathbf{x}_2) \tag{A.26}$$

A.4.1 Retrieving the polar angles

To retrieve the polar angles, we need to place ourselves in the reference frame $(\mathbf{x}_i, \mathbf{u}_i, \mathbf{v}_i)$, where $\mathbf{x}_i$ is the origin of the reference frame and $\mathbf{u}_i$ and $\mathbf{v}_i$ are the direction vectors determining the ellipse orientation. Using the convenient $\mathrm{atan2}$ function giving values ranging in $]-\pi, \pi]$, we have

$$\phi_i = \mathrm{atan2}\left(\langle \mathbf{p}_i - \mathbf{x}_1, \mathbf{v}_1\rangle, \langle \mathbf{p}_i - \mathbf{x}_1, \mathbf{u}_1\rangle\right) \tag{A.27}$$

$$\psi_i = \mathrm{atan2}\left(\langle \mathbf{p}_i - \mathbf{x}_2, \mathbf{v}_2\rangle, \langle \mathbf{p}_i - \mathbf{x}_2, \mathbf{u}_2\rangle\right) \tag{A.28}$$

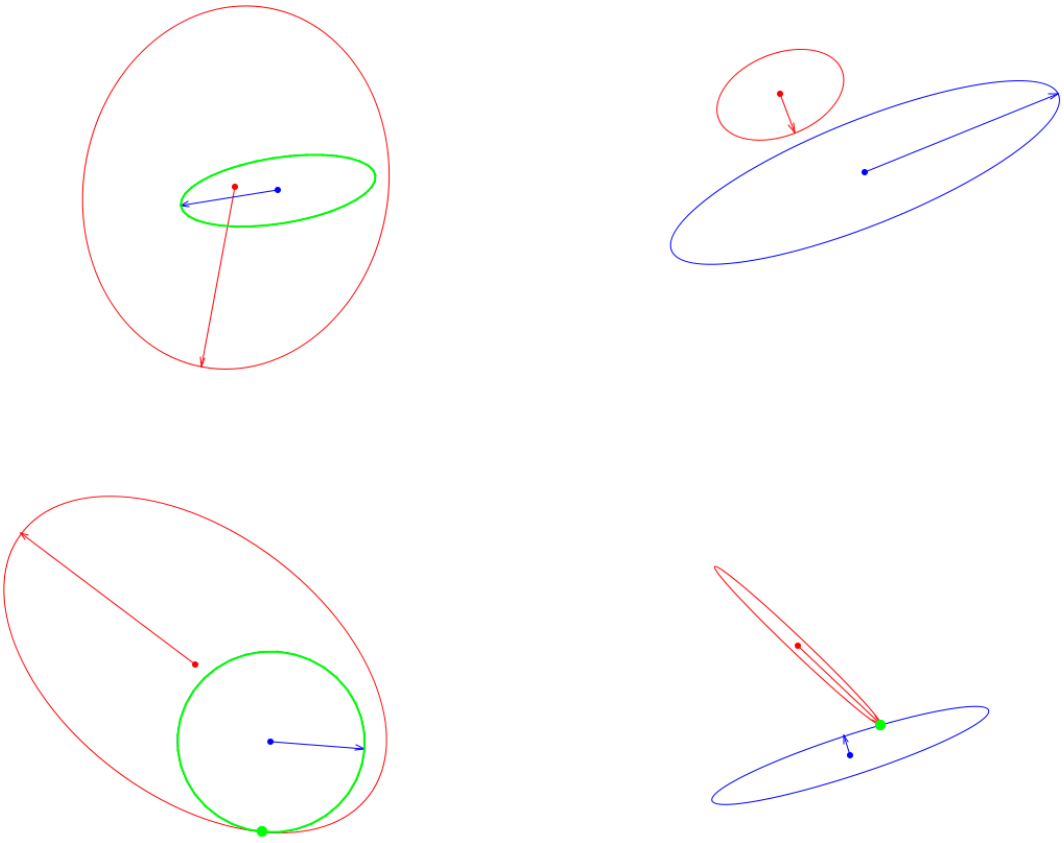


Figure A.4: Cases where there is zero or one intersection point.

A.4.2 0 or 1 intersection point

Either one ellipse is contained in the other or there are separated as illustrated in Figure A.4. An ellipse, say $\mathcal{E}_1$, is contained in the other $\mathcal{E}_2$ if and only if its center satisfies $E_2(\mathbf{x}_1) < 0$. In that case, the area of the intersection is just the area of ellipse $\mathcal{E}_1$. Otherwise, if there is no containment, the intersection area is zero. In summary,

$$\text{area}(\mathcal{E}_1 \cap \mathcal{E}_2) = \begin{cases} \pi a_1 b_1 & \text{if } E_2(\mathbf{x}_1) < 0 \\ \pi a_2 b_2 & \text{if } E_1(\mathbf{x}_2) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.29}$$

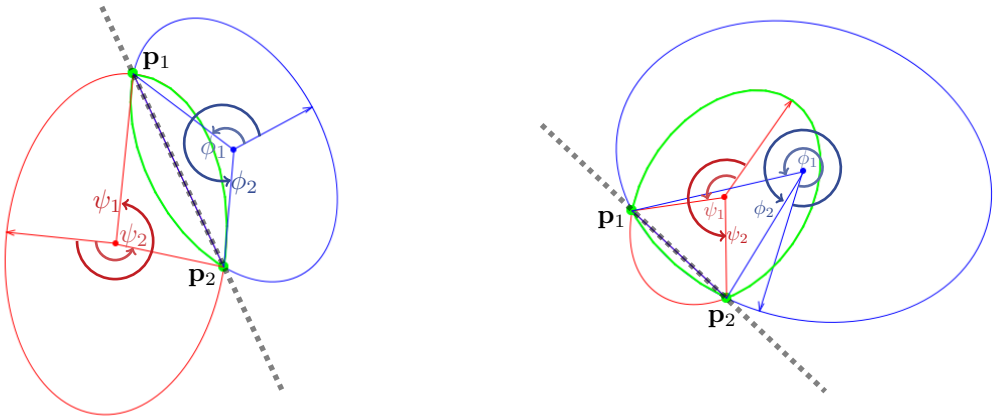


Figure A.5: Cases where there are two intersection points.

A.4.3 2 intersection points

We will not detail the case when Polynomial (A.22) have 2 roots with multiplicity 2. This still corresponds to the case where there are two intersection points. But because of the root multiplicities, one ellipse is contained in the other one and then Formula (A.29) gives the correct intersection area.

Otherwise, we have to consider two cases as illustrated in Figure A.5, which EBERLY (2008) apparently forgot to consider. Namely, the cases correspond to whether the center of ellipses $\mathcal{E}_1$ and $\mathcal{E}_2$ are on the same side or on opposite side with respect to the line $(\mathbf{p}_1, \mathbf{p}_2)$.

Denoting a unit normal of the line going across the intersection points $(\mathbf{p}_1, \mathbf{p}_2)$ by $\mathbf{n}$ (cf. Figure A.5). If the ellipse centers $\mathbf{x}_1$ and $\mathbf{x}_2$ are on opposite side with respect to the line $(\mathbf{p}_1, \mathbf{p}_2)$, i.e., $\langle \mathbf{n}, \mathbf{x}_1 - \mathbf{p}_1 \rangle \times \langle \mathbf{n}, \mathbf{x}_2 - \mathbf{p}_1 \rangle < 0$, then

$$\text{area}(\mathcal{E}_1 \cap \mathcal{E}_2) = B_1(\phi_1, \phi_2) + B_2(\psi_1, \psi_2) \tag{A.30}$$

If they are on the same side with respect to the line $(\mathbf{p}_1, \mathbf{p}_2)$, i.e., $\langle \mathbf{n}, \mathbf{x}_1 - \mathbf{p}_1 \rangle \times \langle \mathbf{n}, \mathbf{x}_2 - \mathbf{p}_1 \rangle > 0$, then

$$\text{area}(\mathcal{E}_1 \cap \mathcal{E}_2) = \begin{cases} (\pi a_1 b_1 - B_1(\phi_1, \phi_2)) + B_2(\psi_1, \psi_2) & \text{if } |\langle \mathbf{n}, \mathbf{x}_1 - \mathbf{p}_1 \rangle| \leq |\langle \mathbf{n}, \mathbf{x}_2 - \mathbf{p}_1 \rangle| \\ B_1(\phi_1, \phi_2) + (\pi a_2 b_2 - B_2(\psi_1, \psi_2)) & \text{otherwise.} \end{cases} \tag{A.31}$$

Note that the condition $|\langle \mathbf{n}, \mathbf{x}_1 - \mathbf{p}_1 \rangle| \leq |\langle \mathbf{n}, \mathbf{x}_2 - \mathbf{p}_1 \rangle|$ in Equation (A.31) just expresses the fact that the distance of ellipse center $\mathbf{x}_1$ to the line $(\mathbf{p}_1, \mathbf{p}_2)$ is smaller than the distance of ellipse center $\mathbf{x}_2$ to the line $(\mathbf{p}_1, \mathbf{p}_2)$.

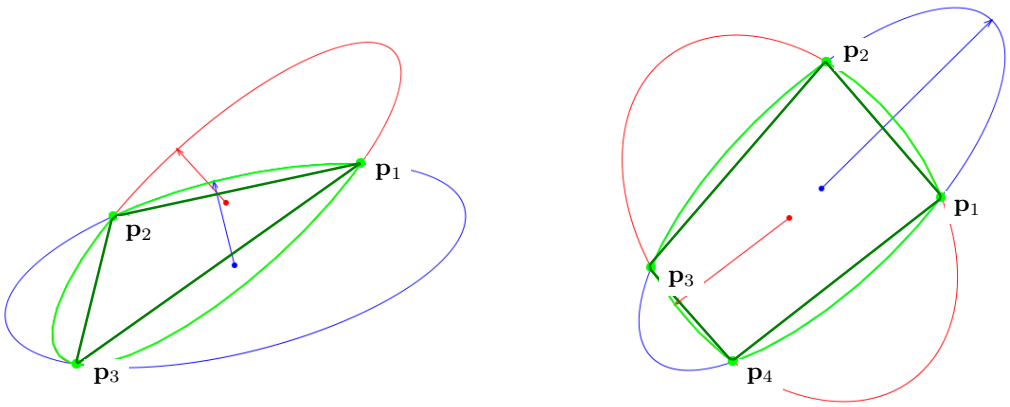


Figure A.6: Cases where there are three of four intersection points.

A.4.4 3 and 4 intersection points

These cases are rather easy to handle. Indeed, we see geometrically from Figure A.6,

$$\text{area}(\mathcal{E}_1 \cap \mathcal{E}_2) = \sum_{i=1}^{L} \underbrace{\min\left(B_1(\phi_i, \phi_{i+1}), B_2(\psi_i, \psi_{i+1})\right)}_{\text{smallest of elliptic portion area}} + \underbrace{\frac{1}{2}\sum_{i=1}^{L} \left|\det\left(\mathbf{p}_i, \mathbf{p}_{i+1}\right)\right|}_{\text{area of polygon } (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_L)} \tag{A.32}$$

with $\phi_{L+1} = \phi_1$, $\psi_{L+1} = \psi_1$ and $\mathbf{p}_{L+1} = \mathbf{p}_1$.

Chapter B

Normalizing Transform of a Feature

Let us remark the following proposition which relates the normalizing transform $\mathbf{T}_x$ to the feature shape $\mathbf{\Sigma}_x$.

Proposition 3. *Let L be an invertible linear transformation in $\mathbb{R}^2$ whose matrix is denoted by $\mathbf{L}$. For any point $\mathbf{x}$ in the zero-centered unit circle in $\mathbb{R}^2$, its transformed point by L is in the ellipse defined by $\left\{\mathbf{z} \in \mathbb{R}^2 | \mathbf{z}^T (\mathbf{L}^T)^{-1} \mathbf{L}^{-1} \mathbf{z} = 1\right\}$*

Proof. Fix a point $\begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}$ of the unit circle in $\mathbb{R}^2$. We write its transformed point by L as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{L} \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}.$$

Since $\mathbf{L}$ is invertible

$$\mathbf{L}^{-1} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}$$

The squared Euclidean norm of the equality yields

$$\begin{bmatrix} u & v \end{bmatrix} (\mathbf{L}^{-1})^T \mathbf{L}^{-1} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos(t) & \sin(t) \end{bmatrix} \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix} = 1$$

We recognize the equation of an ellipse, which concludes the proof of proposition 3. □

Consider the shape matrix $\mathbf{\Sigma}_x$. Recall that $\mathbf{\Sigma}_x$ defines the elliptic shape $\mathcal{S}_x$. We want to retrieve the transformation L_x that satisfies

$$\mathbf{\Sigma}_x = (\mathbf{L}_x^{-1})^T \mathbf{L}_x^{-1}. \tag{B.1}$$

Observe from the QR factorization $\mathbf{L}_x = \mathbf{QR}$ that L_x can be decomposed uniquely in two specific transformations $\mathbf{Q}$ and $\mathbf{R}$, which have the following geometric interpretations. The upper triangular matrix $\mathbf{R}$ expresses a combination of shear and scaling

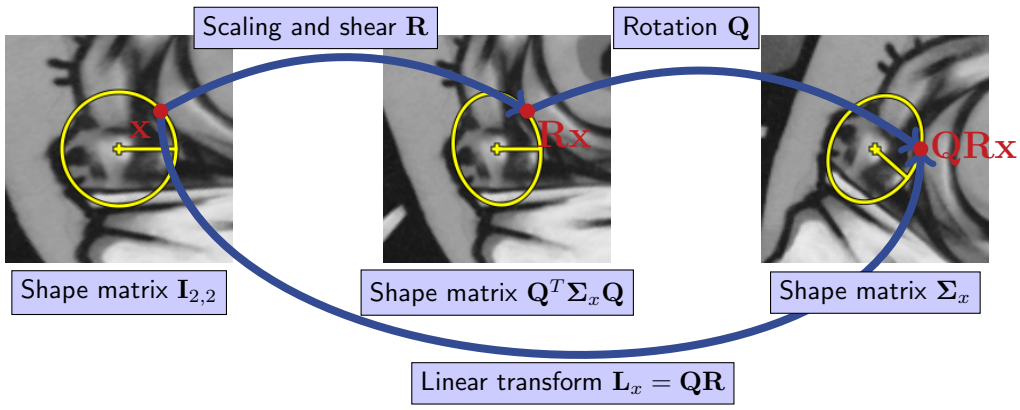


Figure B.1: Geometric interpretation of the QR factorization of linear transform matrix $\mathbf{L}_x$.

transforms. The orthonormal matrix $\mathbf{Q}$ expresses a rotation. This geometric interpretation is illustrated in Figure B.1.

Unless L_x involves no rotation, $\mathbf{L}_x$ is an upper triangular matrix. Then, because Equation (B.1) is a Cholesky decomposition, $\mathbf{L}_x$ can be identified by unicity of the Cholesky decomposition.

In general, $\mathbf{L}_x$ is not upper triangular. Orientations $\mathbf{o}_x$ of elliptic shape $\mathbf{\Sigma}_x$ are provided from feature detectors. Recall that, as far as the SIFT descriptor is concerned, $\mathbf{o}_x$ corresponds to a dominant local gradient orientation.

Thus, introducing $\theta_x \stackrel{\text{def}}{=} \angle\left(\begin{bmatrix}1\\0\end{bmatrix}, \mathbf{o}_x\right)$, we have $\mathbf{Q} = \begin{bmatrix}\cos(\theta_x) & -\sin(\theta_x)\\ \sin(\theta_x) & \cos(\theta_x)\end{bmatrix}$ and expanding Equation (B.1) yields

$$\begin{aligned}\mathbf{\Sigma}_x &= (\mathbf{L}_x^{-1})^T\mathbf{L}_x^{-1}\\ &= \mathbf{Q}(\mathbf{R}^{-1})^T\mathbf{R}^{-1}\mathbf{Q}^T \quad \text{since } \mathbf{Q}^T = \mathbf{Q}^{-1}\\ \mathbf{Q}^T\mathbf{\Sigma}_x\mathbf{Q} &= (\mathbf{R}^{-1})^T\mathbf{R}^{-1}\end{aligned}$$

We recognize the Cholesky decomposition of matrix $\mathbf{Q}^T\mathbf{\Sigma}_x\mathbf{Q}$ which is the rotated ellipse as shown in Figure B.1, in which case $\mathbf{L}_x$ can be determined completely.

Finally, the affinity that maps the zero-centered unit circle to ellipse $\mathcal{S}_x$ is of the form, in homogeneous coordinates

$$\mathbf{T}_x = \begin{bmatrix}\mathbf{L}_x & \mathbf{x}\\ \mathbf{0}_2^T & 1\end{bmatrix}. \tag{B.2}$$

Algorithm B.1 summarizes how to compute $\mathbf{T}_x$.

Algorithm B.1 Computation of the normalizing transform $\mathbf{T}_x$ of feature x

procedure COMPUTENORMALIZEDTRANSFORMOF(x)

$\theta_x := \operatorname{atan2}\left(\left\langle \mathbf{o}_x, \begin{bmatrix}0\\1\end{bmatrix}\right\rangle, \left\langle \mathbf{o}_x, \begin{bmatrix}1\\0\end{bmatrix}\right\rangle\right)$

$\mathbf{Q} := \begin{bmatrix}\cos(\theta_x) & -\sin(\theta_x)\\ \sin(\theta_x) & \cos(\theta_x)\end{bmatrix}$

$\mathbf{M} := \text{Cholesky}(\mathbf{Q}^T\boldsymbol{\Sigma}_x\mathbf{Q})$

// $\mathbf{M}$ is a lower triangular matrix such that $\mathbf{M}\mathbf{M}^T = \mathbf{Q}^T\boldsymbol{\Sigma}_x\mathbf{Q}$

$\mathbf{R} := (\mathbf{M}^T)^{-1}$

$\mathbf{L} := \mathbf{Q}\mathbf{R}$

return $\mathbf{T}_x := \begin{bmatrix}\mathbf{L} & \mathbf{x}\\ \mathbf{0}_2^T & 1\end{bmatrix}$

end procedure

Chapter C

Homography and Local Affine Approximation

C.1 Local Affine Approximation

In the projective space, a homography ϕ is expressed as a matrix in $\mathbb{R}^{3\times 3}$

$$\mathbf{H} \overset{\text{def}}{=} \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix}$$

where $\mathbf{h}_i = [h_{i,1}, h_{i,2}, h_{i,3}]^T$ for $i = \{1, 2, 3\}$.

For any point $\mathbf{x} \in \mathbb{R}^2$, we denote its normalized projective coordinates by $\mathbf{X}$, i.e. $\mathbf{X} = [\mathbf{x}, 1]^T \in \mathbb{R}^3$. Then, we have

$$\phi(\mathbf{x}) = \frac{1}{\mathbf{h}_3^T \mathbf{X}} \begin{bmatrix} \mathbf{h}_1^T \mathbf{X} \\ \mathbf{h}_2^T \mathbf{X} \end{bmatrix}.$$

Using a first-order Taylor expansion of ϕ around $\mathbf{x}_0$, we have

$$\phi(\mathbf{x}) \underset{\mathbf{x}\to\mathbf{x}_0}{=} \phi(\mathbf{x}_0) + \phi'(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) + o(||\mathbf{x} - \mathbf{x}_0||).$$

We recognize that the first-order Taylor approximation is the local affinity $\mathbf{A}_{\mathbf{x}_0}$ that approximates ϕ around $\mathbf{x}_0$

$$\mathbf{A}_{\mathbf{x}_0}(\mathbf{x}) = \phi(\mathbf{x}_0) + \phi'(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) \tag{C.1}$$

Note that $J_\phi(\mathbf{x}_0) = \phi'(\mathbf{x}_0)^T$ is the Jacobian matrix of ϕ about $\mathbf{x}_0$ and equals to

$$J_\phi(\mathbf{x}_0) = \phi'(\mathbf{x}_0)^T = \frac{1}{(\mathbf{h}_3^T \mathbf{X}_0)^2} \begin{bmatrix} \mathbf{h}_{1,1:2}^T(\mathbf{h}_3^T \mathbf{X}_0) - \mathbf{h}_{3,1:2}^T(\mathbf{h}_1^T \mathbf{X}_0) \\ \mathbf{h}_{2,1:2}^T(\mathbf{h}_3^T \mathbf{X}_0) - \mathbf{h}_{3,1:2}^T(\mathbf{h}_2^T \mathbf{X}_0) \end{bmatrix} \in \mathbb{R}^{2\times 2}$$

where we denote $\mathbf{h}_{i,1:2} = [h_{i,1}, h_{i,2}]^T$ for $i = 1, \ldots, 3$.

C.2 Projection of a Feature by a Homography

Let us remark that a homography ϕ does not transform an ellipse into an ellipse in general, while an affinity does. This is why we use the local affine approximation $\mathbf{A_x}$ of ϕ about $\mathbf{x}$ to approximate the geometric information of the projected feature $\phi(x)$. For any elliptic feature x, its projection $\phi(x)$ is approximated as follows.

- Its position $\phi(\mathbf{x})$ is expressed as in the above section.
- Its shape $\phi(\mathcal{S}_x)$ is *approximated* as

$$\phi(\mathcal{S}_x) \approx \left\{ \mathbf{x}' \in \mathbb{R}^2 \,\middle|\, (\mathbf{x}' - \phi(\mathbf{x}))^T \mathbf{\Sigma}_{\phi(x)} \left(\mathbf{x}' - \phi(\mathbf{x})\right)^T \leq 1 \right\} \tag{C.2}$$

  where

$$\mathbf{\Sigma}_{\phi(x)} = J_\phi(\mathbf{x})^{-1}\ \mathbf{\Sigma}_x\ (J_\phi(\mathbf{x})^{-1})^T \in \mathbb{R}^{2\times 2} \tag{C.3}$$

- Its orientation vector $\phi(\mathbf{o}_x)$ is *approximated* as

$$\phi(\mathbf{o}_x) \approx \frac{\mathbf{T_x}(\mathbf{x}+\mathbf{o}_x) - \phi(\mathbf{x})}{||\mathbf{T_x}(\mathbf{x}+\mathbf{o}_x) - \phi(\mathbf{x})||_2} \tag{C.4}$$

Chapter D

Repeatability Study of Feature Detector-Descriptor

D.1 Generating the results on repeatability and precision of detectors

In Section 4.3 of Chapter 4, we described and conducted an extensive study on repeatability of feature detector-descriptor.

D.1.1 Brief Reminder of the Used Datasets

We recall that MIKOLAJCZYK et al. (2005)'s datasets are denoted by

$$\mathcal{D} \stackrel{\text{def}}{=} \{\textit{Bark}, \textit{Boat}, \textit{Graffiti}, \textit{Wall}, \textit{Trees}, \textit{Bikes}, \textit{Leuven}, \textit{UBC}\}.$$

Each of them consists of 6 images. For each dataset $d \in \mathcal{D}$ and for each image index

$$p \in \mathcal{P} \stackrel{\text{def}}{=} \{2, 3, \dots 6\},$$

a ground truth homography $\phi_{d,p}$ is provided for the image pair $(1, p)$ of dataset d for $p \in \mathcal{P}$. The datasets evaluate the robustness of detectors and descriptors with respect to

1. increasing rotation and scale changes (*Bark* and *Boat*),
2. increasing viewpoint changes (*Graffiti* and *Wall*),
3. increasing blur (*Trees* and *Bikes*),
4. increasing illumination changes (*Leuven*) and
5. increasing JPEG compression (*UBC*).

The tested features f are

$$f \in \mathcal{F} \stackrel{\text{def}}{=} \{\text{DoG+SIFT}, \text{Harris-Affine+SIFT}, \text{Hessian-Affine+SIFT}, \text{MSER+SIFT}\}$$

D.1.2 Brief Reminder of the Generated Data

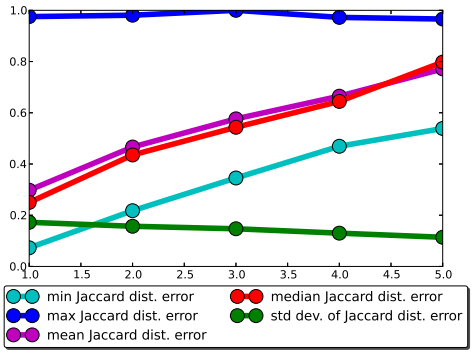
We recall that, for each feature kind $f \in \mathcal{F}$, for each dataset $d \in \mathcal{D}$, for each image pair $(1, p)$, we compute several sets of matches $(\mathcal{M}_{f,d,p,i})_{1 \leq i |\Lambda| - 1}$. For each set of matches $\mathcal{M}_{f,d,p,i}$, we compute the following statistics: minimum, maximum, mean, median, standard deviation values for the Jaccard distance and the angle difference.

Because the data is massive, we show in Figures D.1 and D.2 examples of data we get for only one representative dataset, namely, *Graffiti* which evaluates the detector-descriptor precision and repeatability against increasing viewpoint changes) and only for feature kind $f \in \{\text{DoG+SIFT}, \text{Harris-Affine+SIFT}\}$.

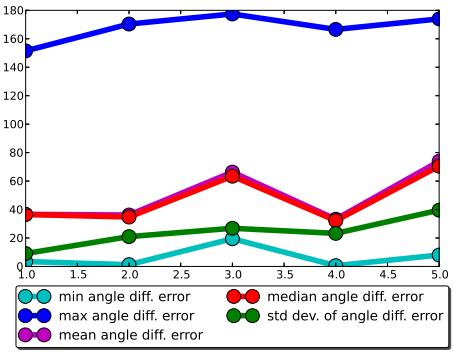
D.2 Factoring the results

As said in Section 4.3, we see from the results that the median and mean value appears to be the most exploitable indicators and are often very similar. However, we preferably use the median value as it localizes well the half of "good" collected values of Jaccard distance or angle differences.

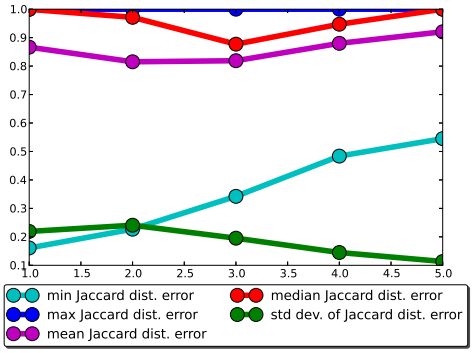
We show plots of functions $p \mapsto \widetilde{\mathcal{J}}(f, d, p, i)$ in Figures D.3, D.4, D.5, D.6, D.7, D.8, D.9, and D.10.



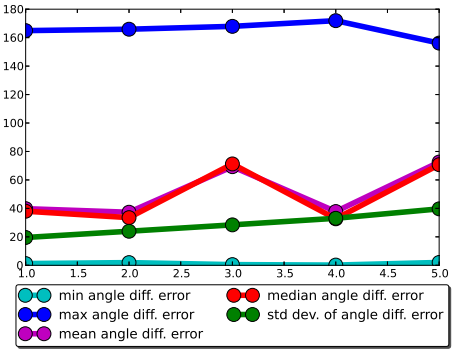
(a) Jaccard distance for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [0, 1.5]$ pixels



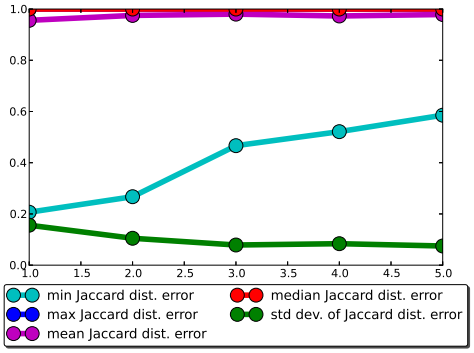
(b) Angle difference (degrees) for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [0, 1.5]$ pixels



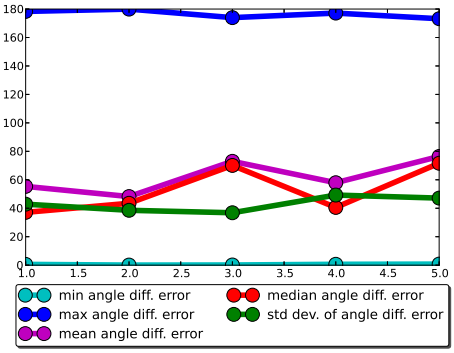
(c) Jaccard distance for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [1.5, 5]$ pixels



(d) Angle difference (degrees) for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [1.5, 5]$ pixels

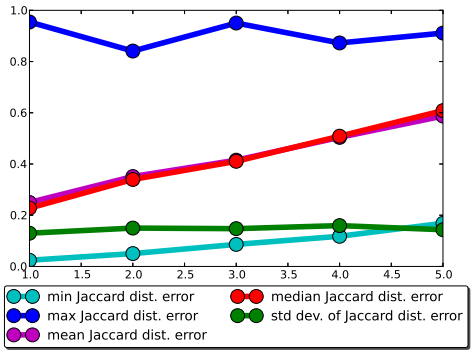


(e) Jaccard distance for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [5, 10]$ pixels

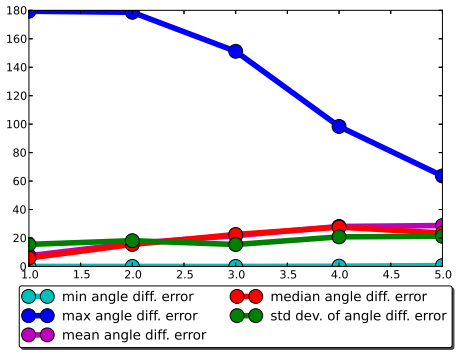


(f) Angle difference (degrees) for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [5, 10]$ pixels

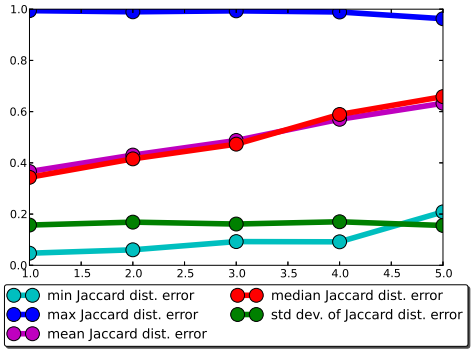
Figure D.1: Statistics of sets of matches $\mathcal{M}_{f,d,p,i}$ for *Graffiti* dataset with DoG+SIFT matches.



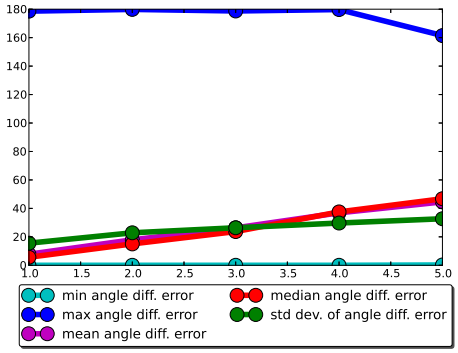
(a) Jaccard distance for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [0, 1.5]$ pixels



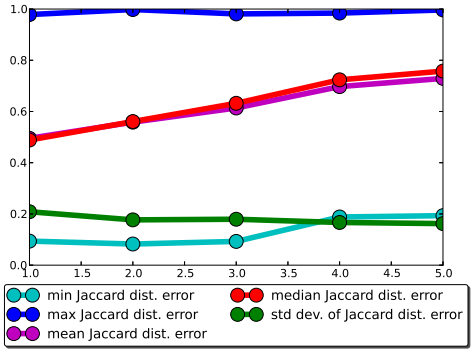
(b) Angle difference (degrees) for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [0, 1.5]$ pixels



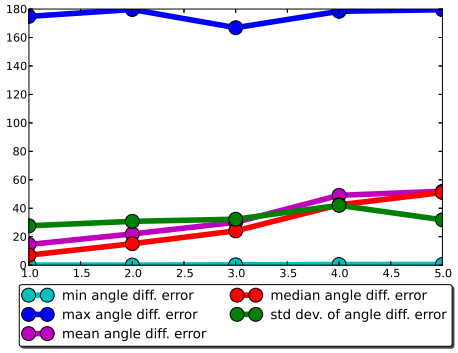
(c) Jaccard distance for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [1.5, 5]$ pixels



(d) Angle difference (degrees) for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [1.5, 5]$ pixels

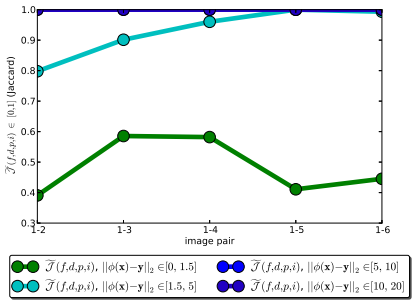


(e) Jaccard distance for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [5, 10]$ pixels

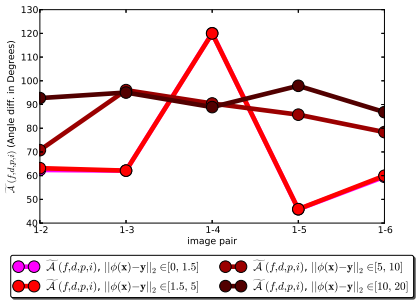


(f) Angle difference (degrees) for matches $m = (x, y)$ such that $\|\phi(\mathbf{x}) - \mathbf{y}\|_2 \in [5, 10]$ pixels

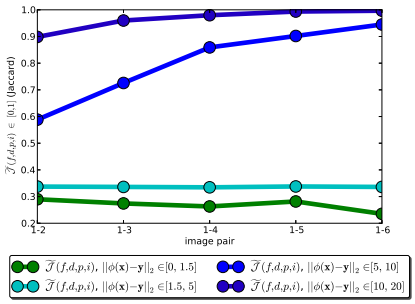
Figure D.2: Statistics of sets of matches $\mathcal{M}_{f,d,p,i}$ for *Graffiti* dataset with Harris-Affine+SIFT matches.



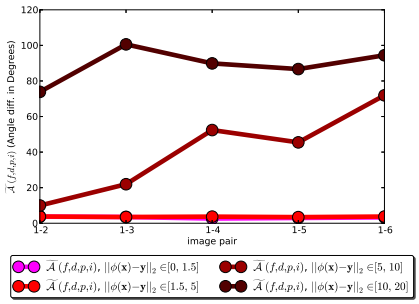
(a) Median Jaccard distance for DoG+SIFT matches.



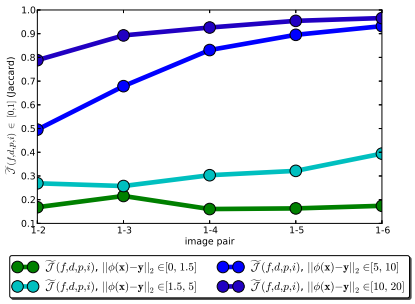
(b) Median angle differences (in degrees) for DoG+SIFT matches.



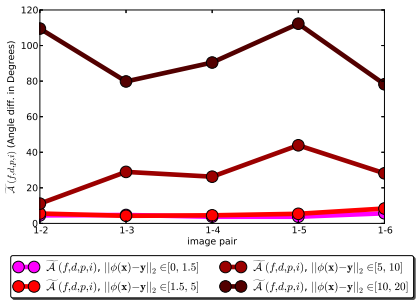
(c) Median Jaccard distance for Harris-Affine+SIFT matches.



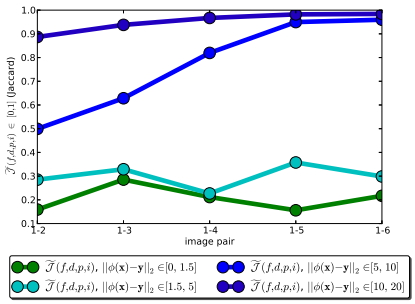
(d) Median angle differences (in degrees) for Harris-Affine+SIFT matches.



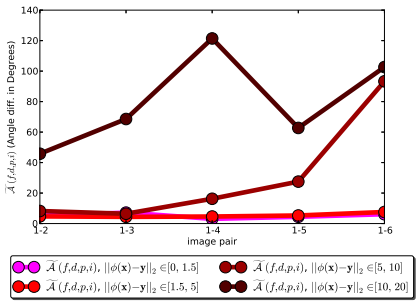
(e) Median Jaccard distance for Hessian-Affine+SIFT matches.



(f) Median angle differences (in degrees) for Hessian-Affine+SIFT matches.

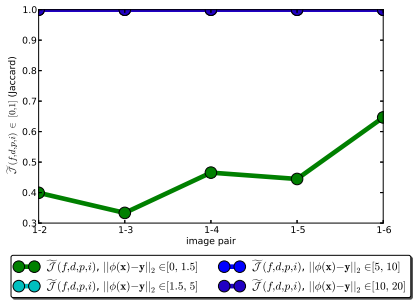


(g) Median Jaccard distance for MSER+SIFT matches.

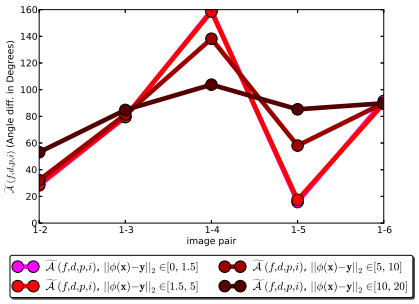


(h) Median angle differences (in degrees) for MSER+SIFT matches.

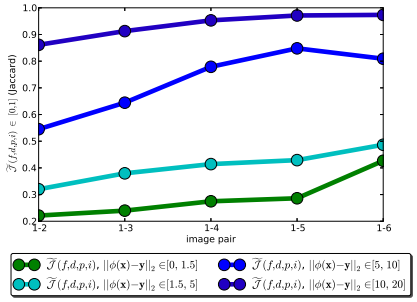
Figure D.3: Median values for *Bark* dataset.



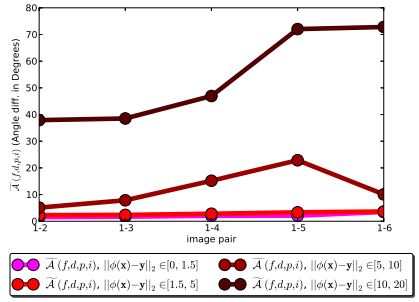
(a) Median Jaccard distance for DoG+SIFT matches.



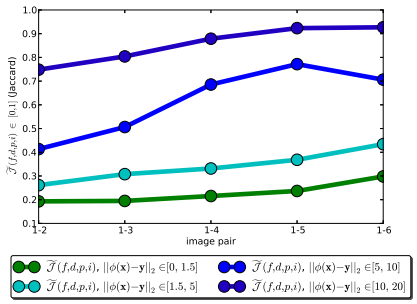
(b) Median angle differences (in degrees) for DoG+SIFT matches.



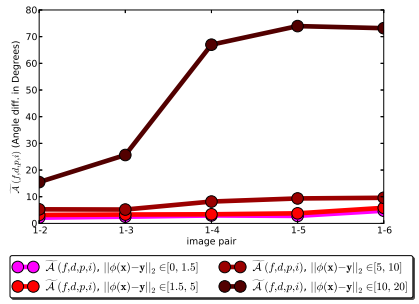
(c) Median Jaccard distance for Harris-Affine+SIFT matches.



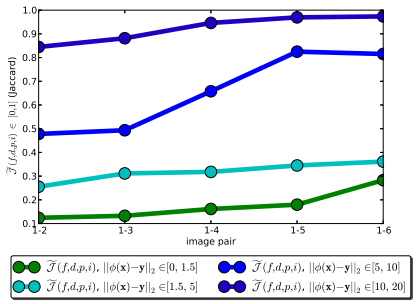
(d) Median angle differences (in degrees) for Harris-Affine+SIFT matches.



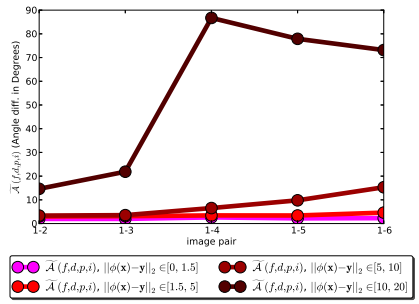
(e) Median Jaccard distance for Hessian-Affine+SIFT matches.



(f) Median angle differences (in degrees) for Hessian-Affine+SIFT matches.

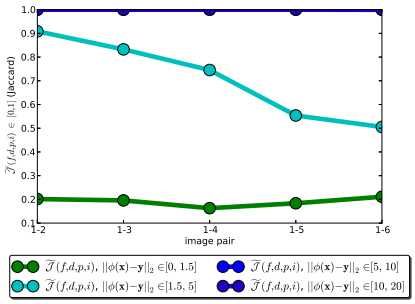


(g) Median Jaccard distance for MSER+SIFT matches.

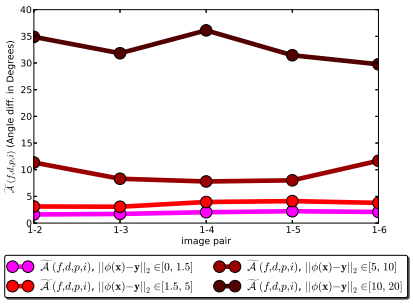


(h) Median angle differences (in degrees) for MSER+SIFT matches.

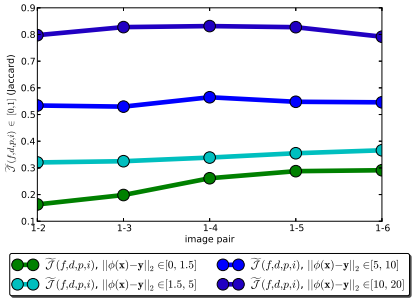
Figure D.4: Median values for *Boat* dataset.



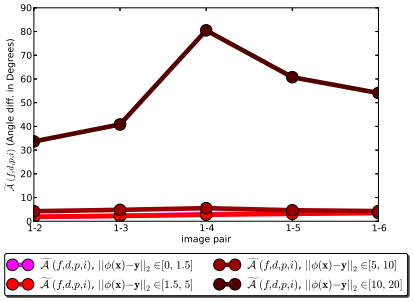
(a) Median Jaccard distance for DoG+SIFT matches.



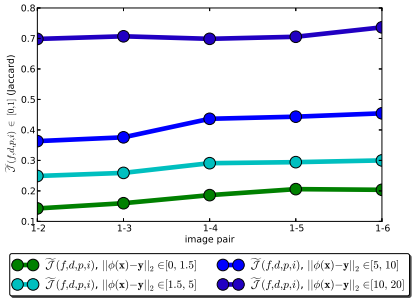
(b) Median angle differences (in degrees) for DoG+SIFT matches.



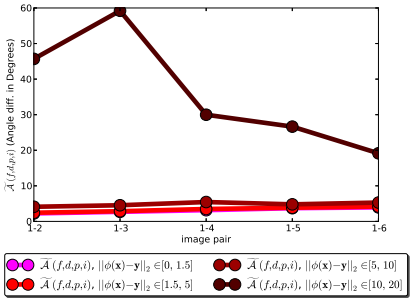
(c) Median Jaccard distance for Harris-Affine+SIFT matches.



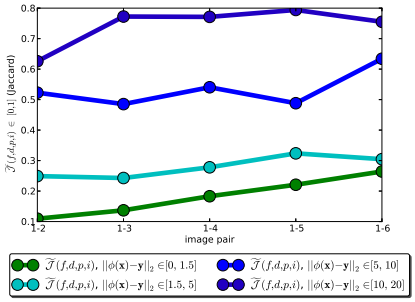
(d) Median angle differences (in degrees) for Harris-Affine+SIFT matches.



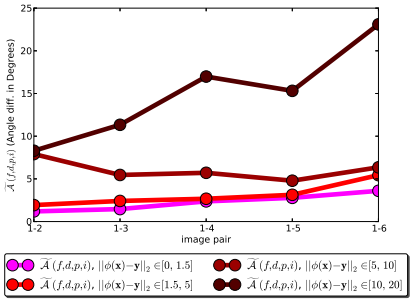
(e) Median Jaccard distance for Hessian-Affine+SIFT matches.



(f) Median angle differences (in degrees) for Hessian-Affine+SIFT matches.

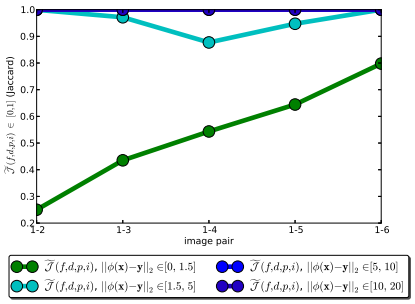


(g) Median Jaccard distance for MSER+SIFT matches.

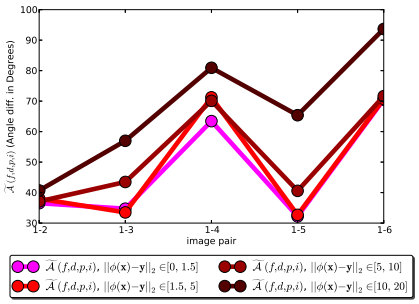


(h) Median angle differences (in degrees) for MSER+SIFT matches.

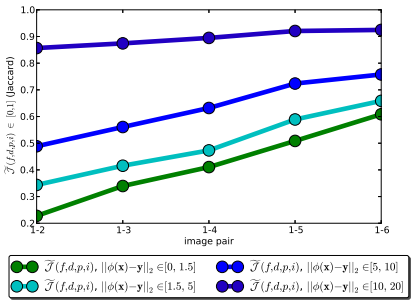
Figure D.5: Median values for *Bikes* dataset.



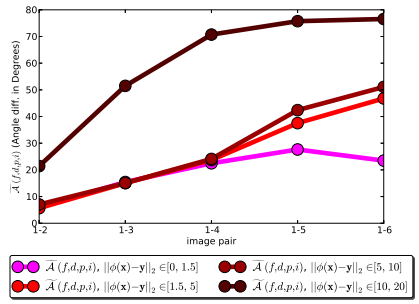
(a) Median Jaccard distance for DoG+SIFT matches.



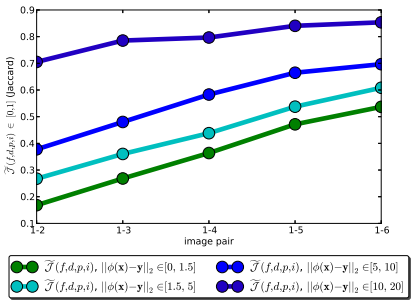
(b) Median angle differences (in degrees) for DoG+SIFT matches.



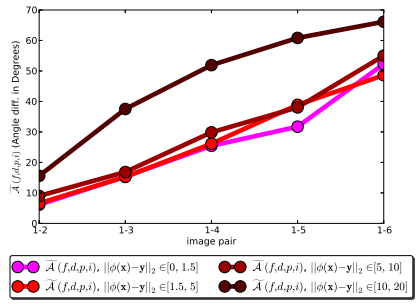
(c) Median Jaccard distance for Harris-Affine+SIFT matches.



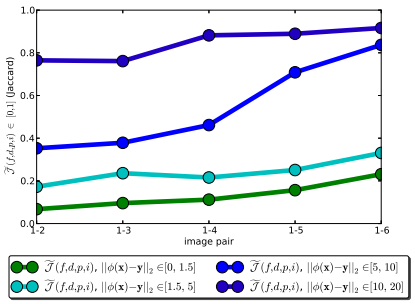
(d) Median angle differences (in degrees) for Harris-Affine+SIFT matches.



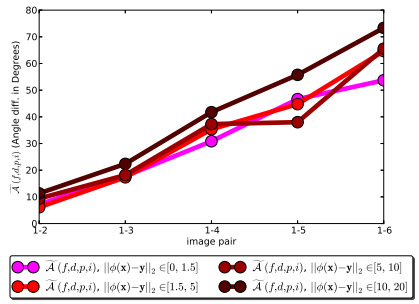
(e) Median Jaccard distance for Hessian-Affine+SIFT matches.



(f) Median angle differences (in degrees) for Hessian-Affine+SIFT matches.

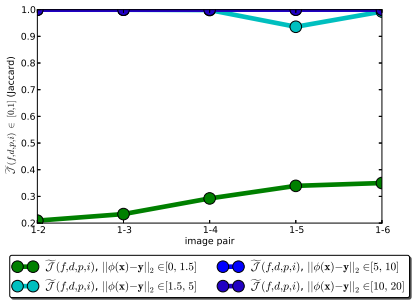


(g) Median Jaccard distance for MSER+SIFT matches.

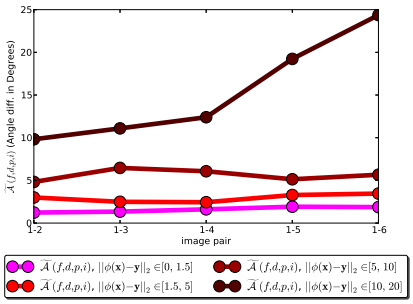


(h) Median angle differences (in degrees) for MSER+SIFT matches.

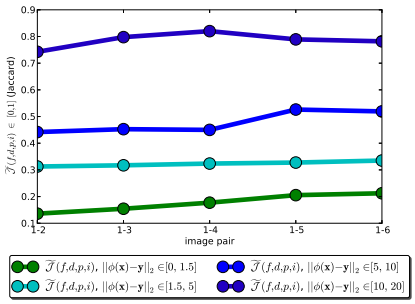
Figure D.6: Median values for *Graffiti* dataset.



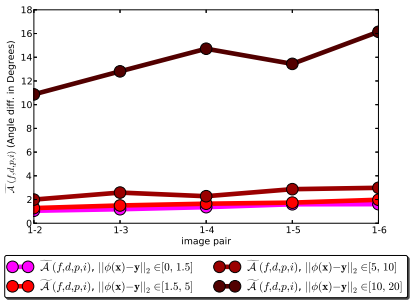
(a) Median Jaccard distance for DoG+SIFT matches.



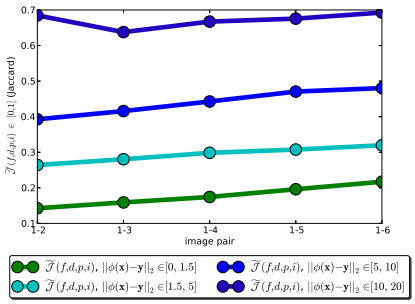
(b) Median angle differences (in degrees) for DoG+SIFT matches.



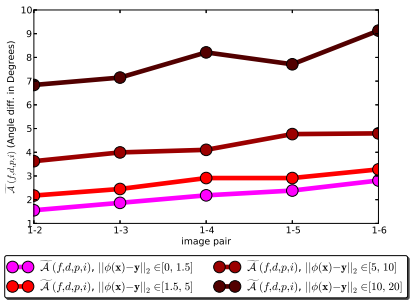
(c) Median Jaccard distance for Harris-Affine+SIFT matches.



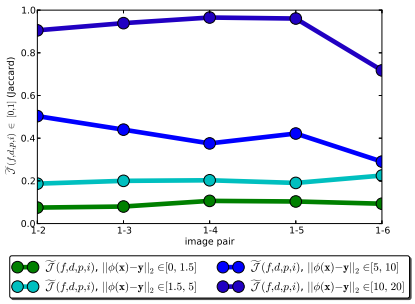
(d) Median angle differences (in degrees) for Harris-Affine+SIFT matches.



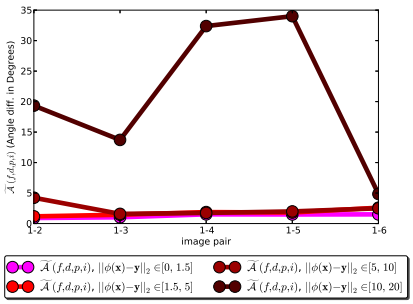
(e) Median Jaccard distance for Hessian-Affine+SIFT matches.



(f) Median angle differences (in degrees) for Hessian-Affine+SIFT matches.

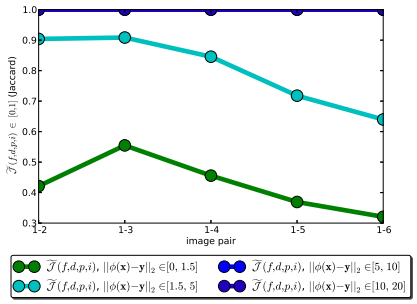


(g) Median Jaccard distance for MSER+SIFT matches.

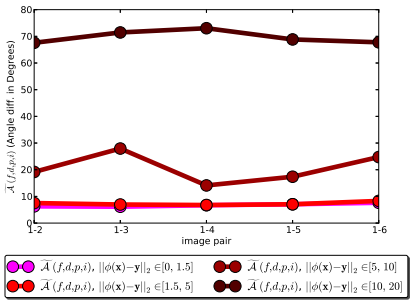


(h) Median angle differences (in degrees) for MSER+SIFT matches.

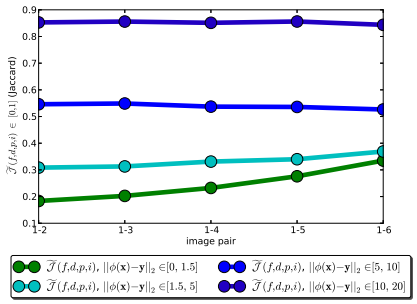
Figure D.7: Median values for *Leuven* dataset.



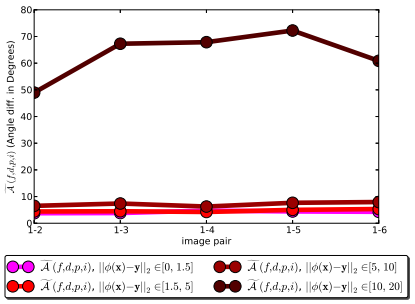
(a) Median Jaccard distance for DoG+SIFT matches.



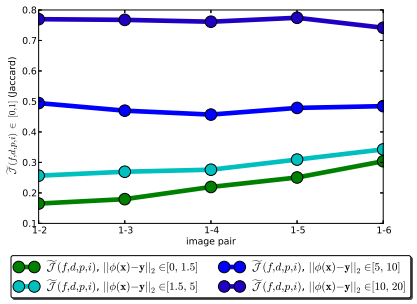
(b) Median angle differences (in degrees) for DoG+SIFT matches.



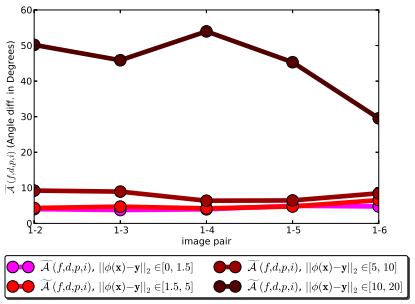
(c) Median Jaccard distance for Harris-Affine+SIFT matches.



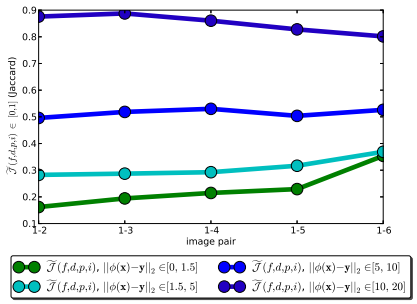
(d) Median angle differences (in degrees) for Harris-Affine+SIFT matches.



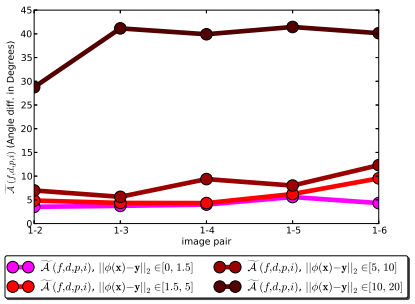
(e) Median Jaccard distance for Hessian-Affine+SIFT matches.



(f) Median angle differences (in degrees) for Hessian-Affine+SIFT matches.

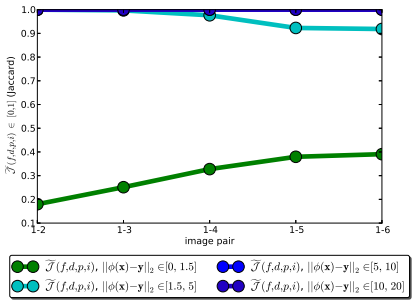


(g) Median Jaccard distance for MSER+SIFT matches.

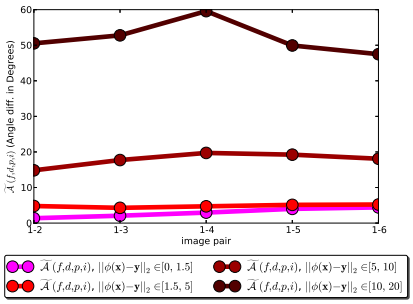


(h) Median angle differences (in degrees) for MSER+SIFT matches.

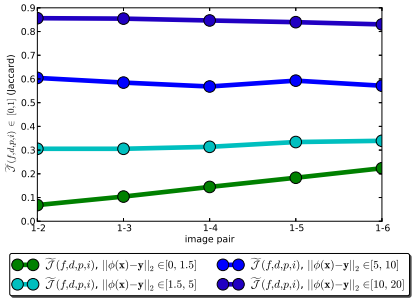
Figure D.8: Median values for *Trees* dataset.



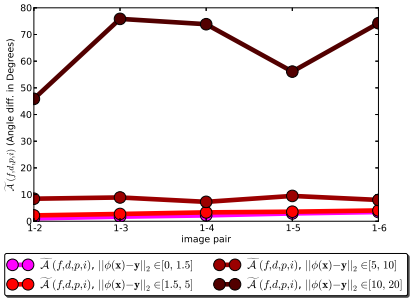
(a) Median Jaccard distance for DoG+SIFT matches.



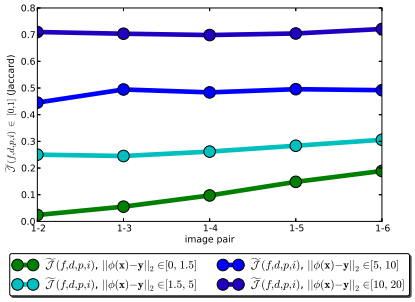
(b) Median angle differences (in degrees) for DoG+SIFT matches.



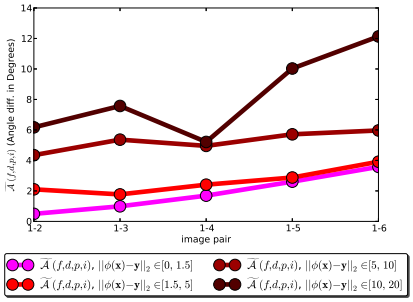
(c) Median Jaccard distance for Harris-Affine+SIFT matches.



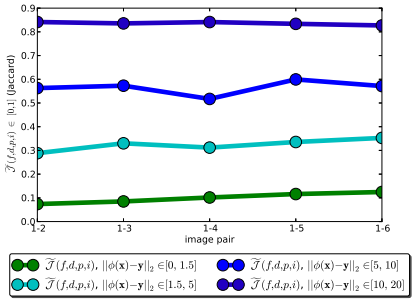
(d) Median angle differences (in degrees) for Harris-Affine+SIFT matches.



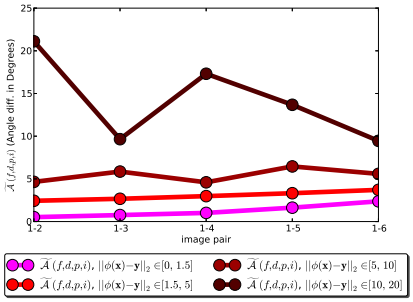
(e) Median Jaccard distance for Hessian-Affine+SIFT matches.



(f) Median angle differences (in degrees) for Hessian-Affine+SIFT matches.

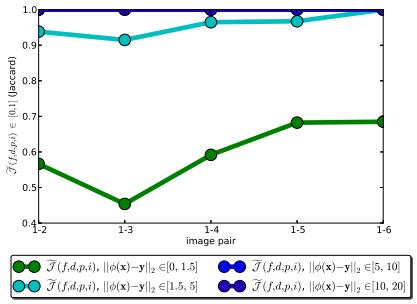


(g) Median Jaccard distance for MSER+SIFT matches.

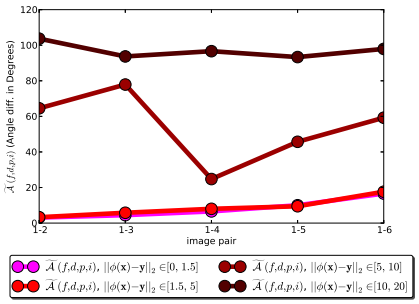


(h) Median angle differences (in degrees) for MSER+SIFT matches.

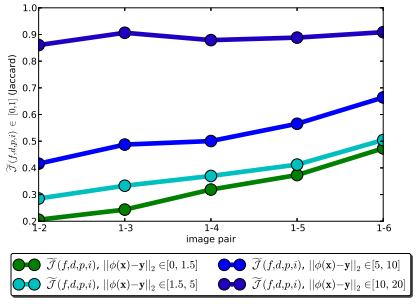
Figure D.9: Median values for *UBC* dataset.



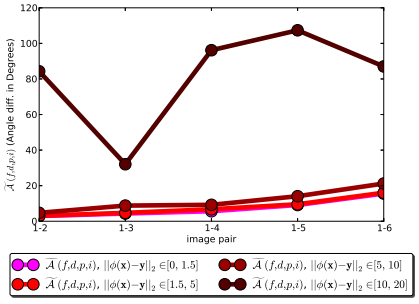
(a) Median Jaccard distance for DoG+SIFT matches.



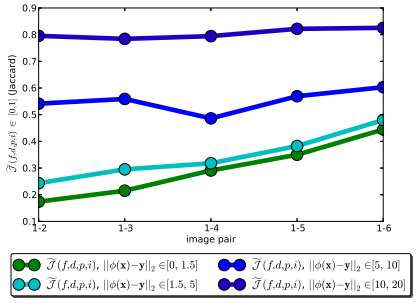
(b) Median angle differences (in degrees) for DoG+SIFT matches.



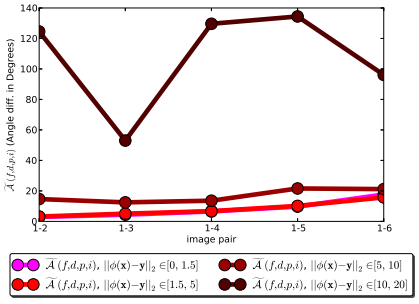
(c) Median Jaccard distance for Harris-Affine+SIFT matches.



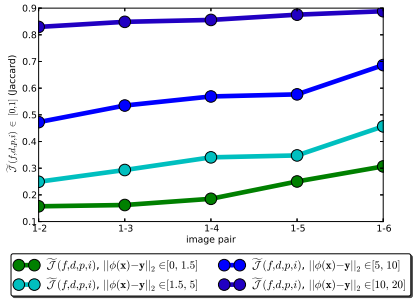
(d) Median angle differences (in degrees) for Harris-Affine+SIFT matches.



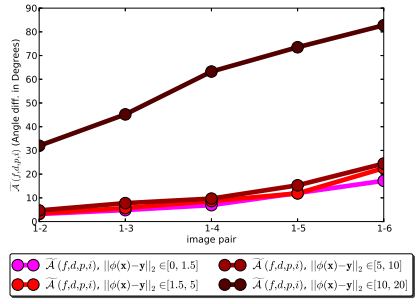
(e) Median Jaccard distance for Hessian-Affine+SIFT matches.



(f) Median angle differences (in degrees) for Hessian-Affine+SIFT matches.



(g) Median Jaccard distance for MSER+SIFT matches.



(h) Median angle differences (in degrees) for MSER+SIFT matches.

Figure D.10: Median values for *Wall* dataset.

Chapter E

Accurate Window Localization

E.1 Ecole Centrale Paris Datasets

The *École Centrale Paris* datasets (TEBOUL 2010) consist of rectified images of Haussman buildings, with pixel labelings. Specifically, there are two datasets. The first one is *ECP CVPR 2010*, which consists of 20 training images and 10 test images, and the second one *ECP Benchmark 2011*, consisting of 104 test images.

We observed that, in these datasets, window parts that are partly occluded by wrought iron balconies (more or less the lower half of the window area) were completely labeled as balconies. This was inconvenient for one of our applications, related to energy saving, that requires the automatic computation of the percentage of glass area. We thus procedurally relabeled all pixels of these parts as being both window *and* balcony pixels.

Besides, in the *ECP CVPR 2010* dataset, the ground truth is defined by hand, with an acceptable precision (with respect to what architects consider as a window). On the contrary, the ground truth of the *ECP Benchmark 2011* dataset is generated procedurally with a shape grammar. This resulted in reasonnably good pixel labelings for façades windows (i.e., windows surrounded by stone walls), whereas roof windows (i.e., windows surrounded by zinc plates) were inaccurately and unreliably located. The reason probably is that roof windows are often not in the plane of the façades. As a result, image rectification does not position them aligned with other façade windows, which the shape grammar did not properly handle. Because there were too many errors for roof windows in the ground truth, we decided to exclude them from our evaluation on the *ECP Benchmark 2011*; only façades windows were taken into account for this dataset.

We would like to emphasize that it is extremely important not to overlook the false positive rate for accurate window localization, especially in the perspective of accurately estimating the thermal performance of buildings. Table E.1 gives a few orders of magnitudes to help interpret the false positive rate.

The cascade classifier (*CC*) (VIOLA and JONES 2004) has been trained on the 20 training images of the *ECP CVPR 2010* dataset. Positive examples are image patches taken from the ground truth. Note that in case, there is a balcony part at the bottom of the

Object type	Typical size	Pixel area	Presence ratio w.r.t to image size
Image	300×550	$156,000$	100%
One window	25×50	$1,250$	0.8%
All windows	24 windows	$30,000$	19.2%
Everything but windows	.	$126,000$	80.8%
$\overline{\text{FPR}} = 3\%$	≈ 3 windows	$3,780$	2%
$\overline{\text{FPR}} = 10\%$	≈ 12.5 windows	$12,600$	8%
$\overline{\text{FPR}} = 20\%$	≈ 20 windows	$25,200$	16%
$\overline{\text{FPR}} = 30\%$	≈ 30 windows	$37,800$	24%

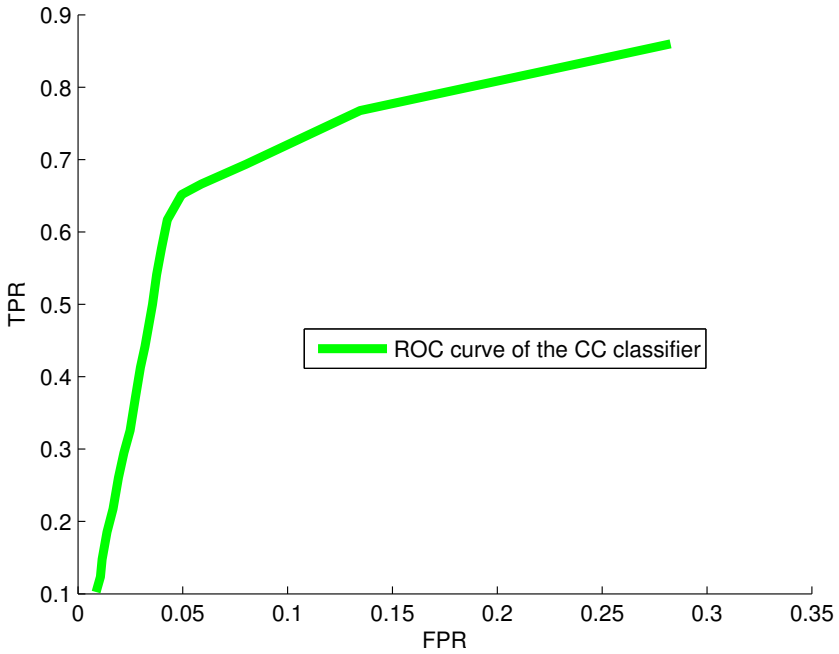
Table E.1: Approximate reference interpretation of figures for the *ECP* datasets.

window, the positive example also contains the balcony part. On the other hand, negative examples are randomly generated patches in training images such that they overlap little with true windows. We set the training parameters as follows. We recall that we only care about CC's precision rate and not its recall rate. Thus, the minimum hit rate threshold is set to 0.9, and the maximum false alarm rate threshold to 0.1 in order to achieve this goal.

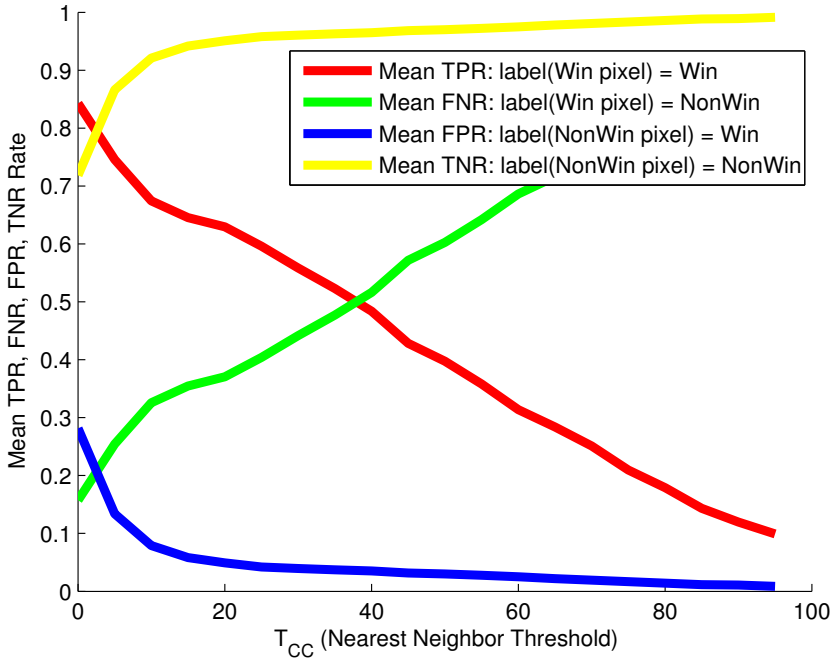
CC is then used on test images of the 2010 and 2011 datasets. Figures E.1a and E.1b shows the performance of the cascade classifier as a function of the detection threshold τ_{CC} as well as its ROC curve. We see that CC's performance is practically the same on these two datasets as they both contain only photographs of Haussman buildings in the Monge street in Paris, which were acquired with the same quality and size.

We separate the results obtained with the two datasets. They are detailed in tables E.2 and E.3.

Figures E.2, E.3, E.3 provide excerpts of window detection results obtained with our method. As we rely on robust feature points, we are relatively insensitive to illumination variations. We are also robust to partial occlusion with plants on balconies. Shutters, on the contrary, can sometimes cause a few missed detections.



(a) ROC curve of the cascade classifier (*CC*).



(b) Mean TPR, FNR, FPR, TNR curves as a function of the classifier threshold τ_{CC}.

Figure E.1: Performance of the cascade classifier (VIOLA and JONES 2004) on window detection.

Methods	$\overline{\text{TPR}}$	$\overline{\text{FNR}}$	$\overline{\text{FPR}}$	$\overline{\text{TNR}}$
manual + RG	**81**%	**19**%	**3**%	**97**%
CC $\tau_{CC} = 20$ + RG	75%	25%	6%	94%
CC $\tau_{CC} = 30$ + RG	72%	28%	5%	95%
CC $\tau_{CC} = 5$	75%	25%	13%	87%
CC $\tau_{CC} = 10$	67%	33%	8%	92%
CC $\tau_{CC} = 20$	63%	37%	5%	95%
CC $\tau_{CC} = 30$	56%	44%	**4**%	96%
RL (bin-hue)	72%	28%	10%	90%
RL (bin-rf)	47%	53%	38%	62%
RL (4-color-rf)	24%	76%	13%	87%
RL (haussm-rf)	70%	30%	7%	93%

Table E.2: Results summary on the *ECP CVPR 2010*. manual+RG and CC+RG denote our method run using bounding boxes provided respectively by hand and by CC. For RL, we used 3 shape grammars: binary (bin), 4-color, and Haussmannian (haussm). hue and rf are different probability priors for façade segmentation when parsing with the shape grammars.

Methods	$\overline{\text{TPR}}$	$\overline{\text{FNR}}$	$\overline{\text{FPR}}$	$\overline{\text{TNR}}$
manual+RG	**79**%	**21**%	**4**%	**96**%
CC $\tau_{CC} = 20$ + RG	73%	23%	6%	94%
CC $\tau_{CC} = 30$ + RG	70%	30%	5%	95%
CC $\tau_{CC} = 5$	79%	21%	16%	84%
CC $\tau_{CC} = 10$	73%	27%	10%	90%
CC $\tau_{CC} = 20$	65%	35%	7%	93%
CC $\tau_{CC} = 30$	57%	43%	5%	95%
RL (bin-hue)	65%	35%	15%	85%
RL (bin-rf)	38%	62%	33%	67%
RL (4-color-rf)	27%	73%	11%	89%
RL (hauss-rf)	64%	36%	6%	94%

Table E.3: Results summary on the *ECP Benchmark 2011*. See caption of Table E.2 for details.

Figure E.2: Window detection results on the *ECP CVPR 2010* dataset. The first row shows the input quadrilaterals and the second row shows the window detection results.

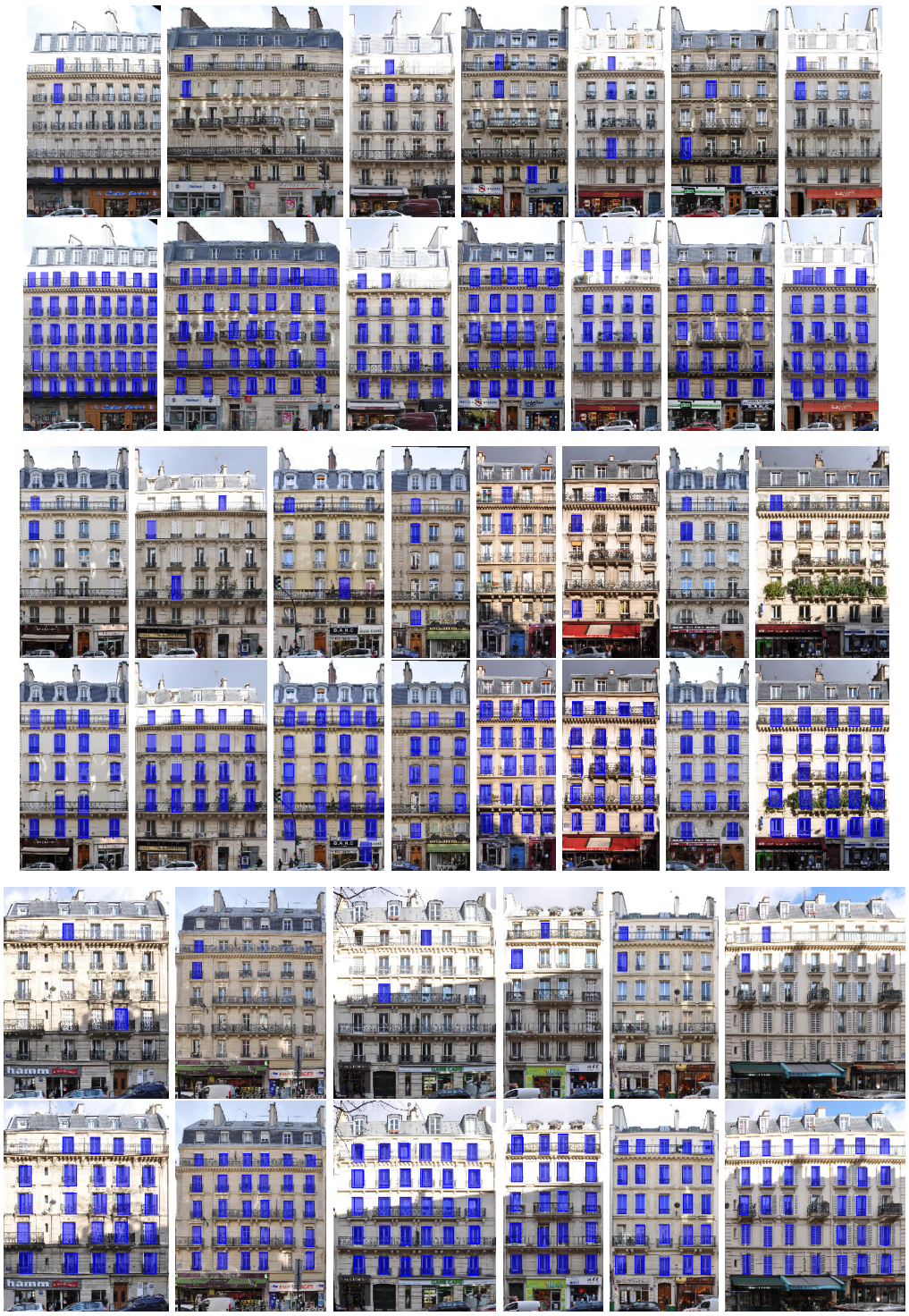


Figure E.3: Window detection results on the *ECP Benchmark 2011* dataset. The odd rows show the input quadrilaterals and the even rows show the window detection results. (1/2)

Figure E.4: Window detection results on the *ECP Benchmark 2011* dataset. The odd rows show the input quadrilaterals and the even rows show the window detection results. (2/2)

Chapter F

Facade Parsing Results

In this material we provide the results of the parsing experiments described in Section 8.6 of Chapter 8.

Figures F.1,F.2,...F.9 display the semantic segmentation produced our modified parser, overlaid on the input images. In each figure the first and the fourth rows present the original images, the second and the fifth rows present the results of the algorithm by Teboul et al. and the third and the sixth rows present the results of our algorithm. The color codes for the symbols are as follows:

color	semantics class
light blue	sky
dark blue	roof
yellow	wall
green	shop
orange	door
red	window
violet	balcony

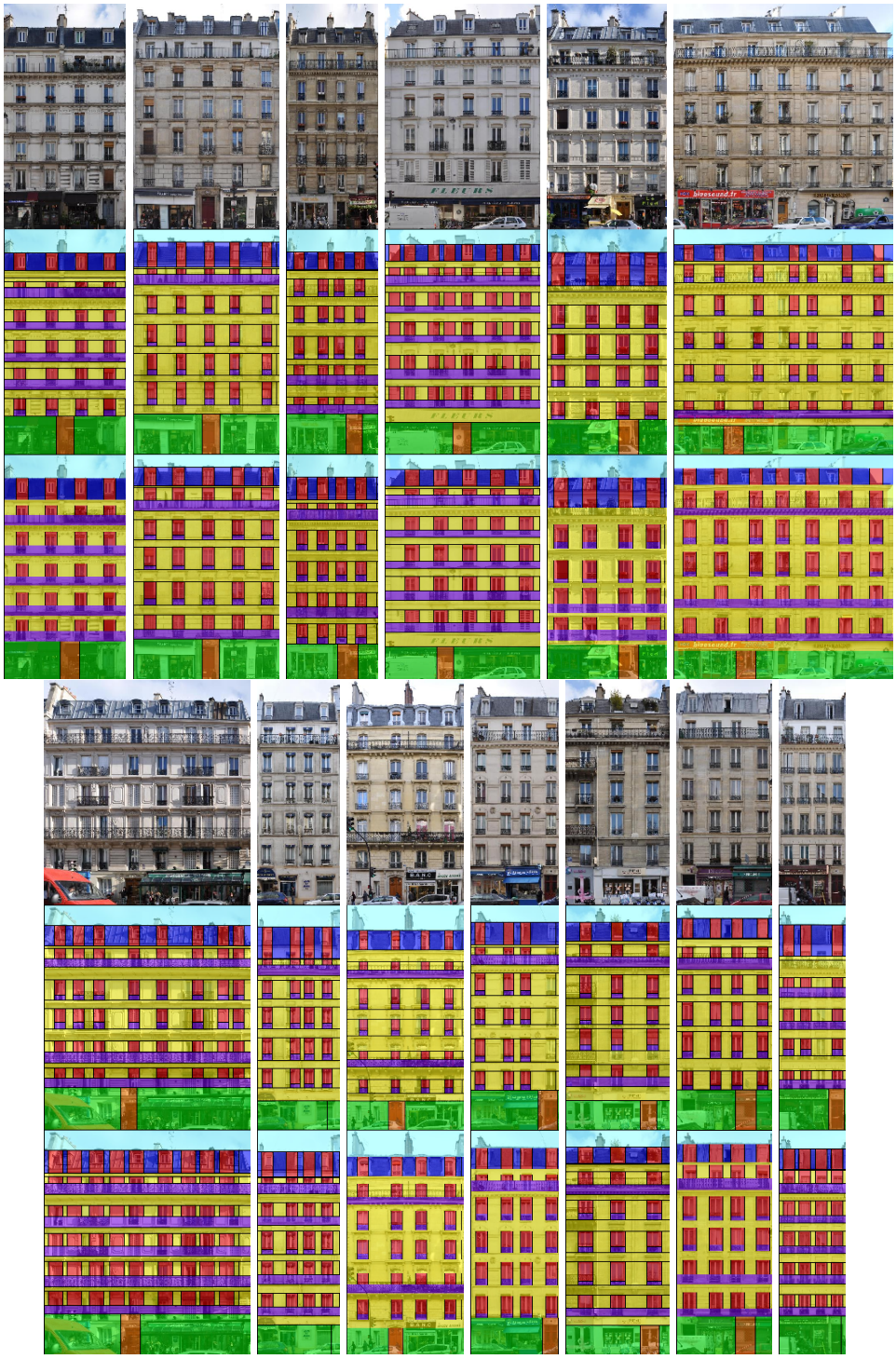


Figure F.1: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (1/9)

Figure F.2: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (2/9)

Figure F.3: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (3/9)

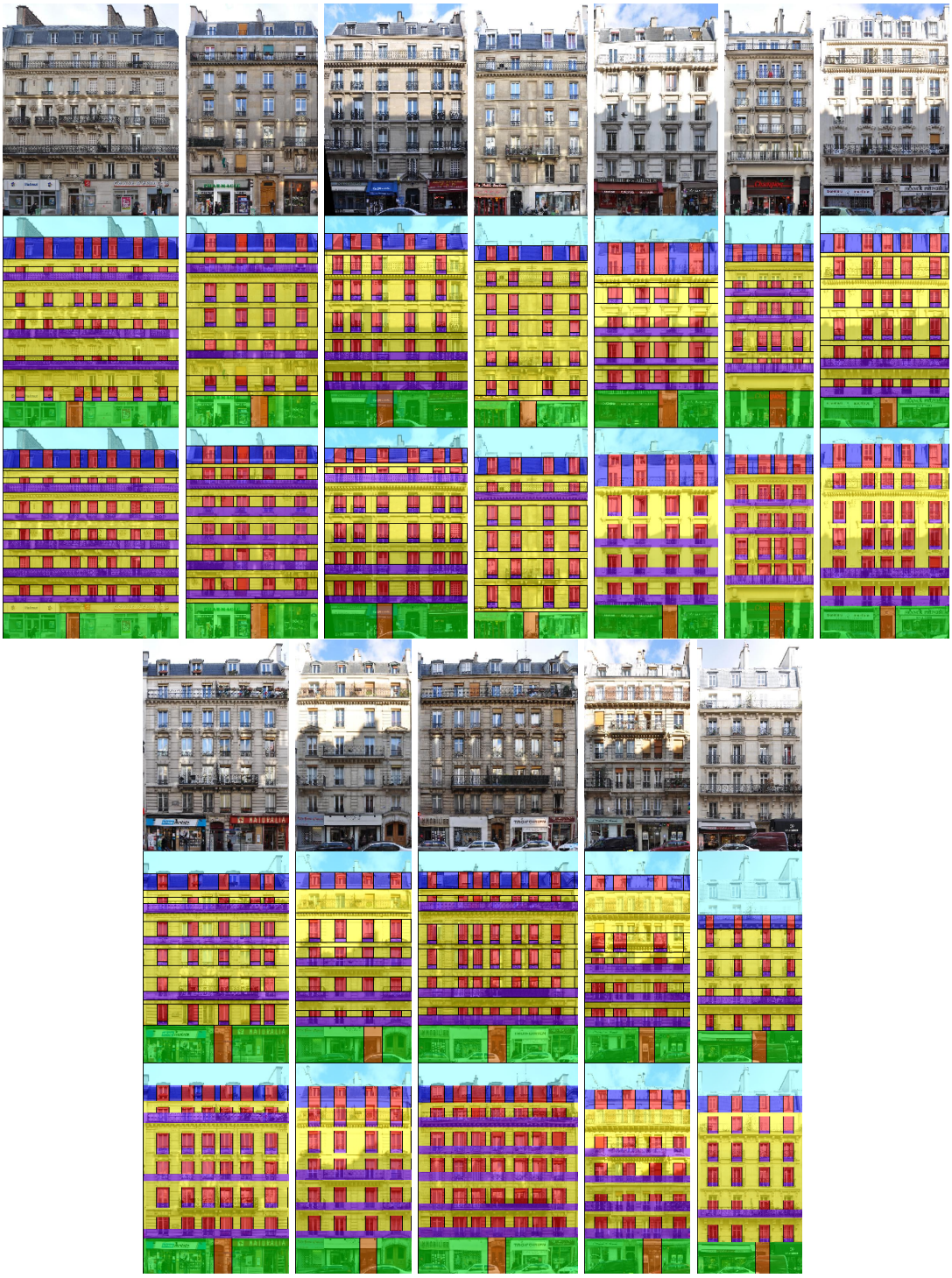


Figure F.4: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (4/9)

Figure F.5: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (5/9)

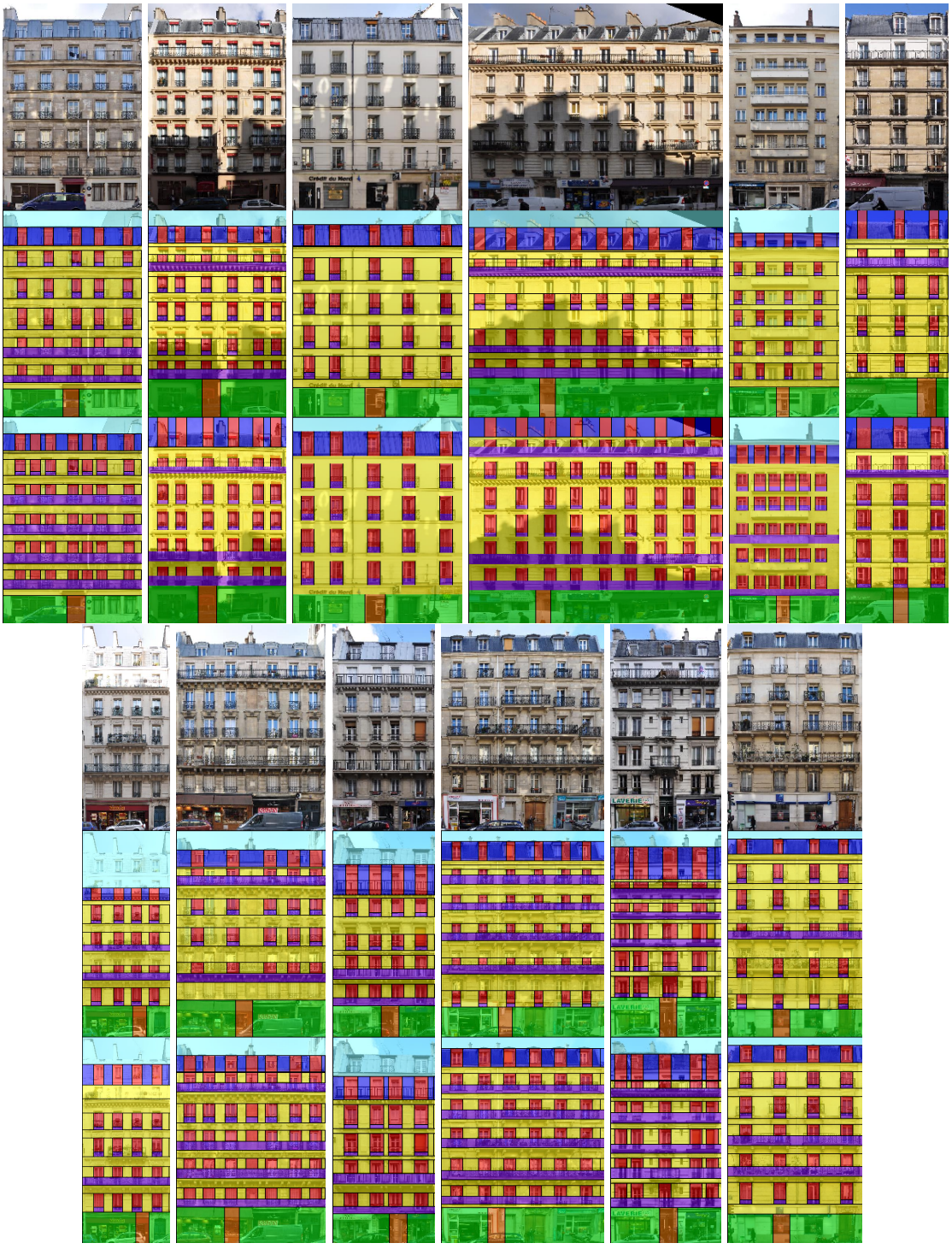


Figure F.6: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (6/9)

Figure F.7: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (7/9)

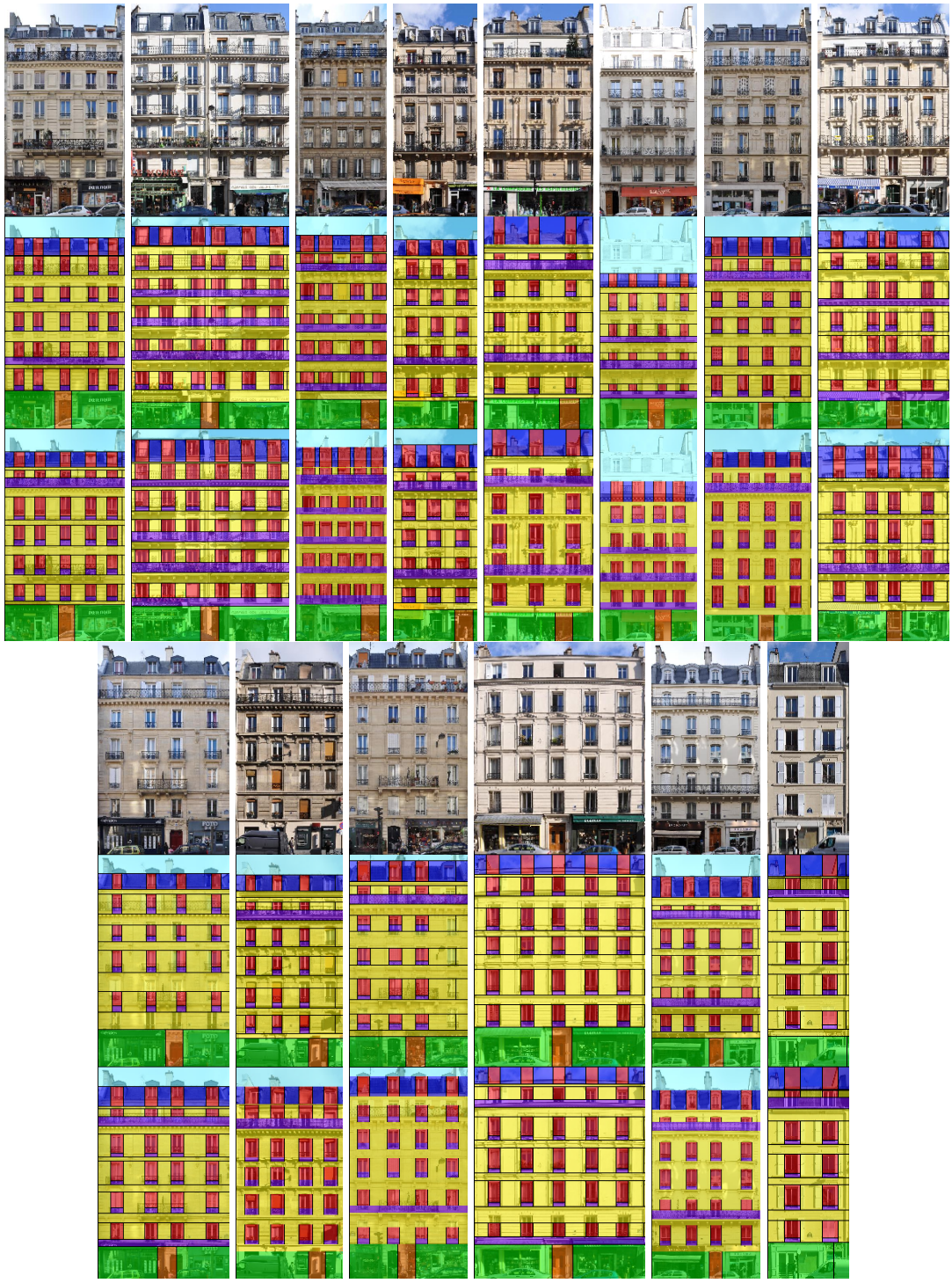


Figure F.8: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (8/9)

Figure F.9: The results of the parsing algorithm on the *ECP Benchmark 2011* dataset. (9/9)

Bibliography

AHUJA, NARENDRA and SINISA TODOROVIC (2008). Connected Segmentation Tree - A Joint Representation of Region Layout and Hierarchy. In: *Computer Vision and Pattern Recognition* (see p. 111)

ALEGRE, FERNANDO and FRANK DELLAERT (2004). *A Probabilistic Approach to the Semantic Interpretation of Building Facades - Georgia Tech's Institutional Repository*. Tech. rep. URL: http://smartech.gatech.edu/dspace/handle/1853/4483 (see p. 110)

ALI, HAIDER, CHRISTIN SEIFERT, NITIN JINDAL, LUCAS PALETTA, and GERHARD PAAR (2007). Window Detection in Facades. In: *ICIAP*, pp. 837–842 (see pp. 98, 101)

BARINOVA, OLGA, VICTOR LEMPITSKY, ELENA TRETIAK, and PUSHMEET KOHLI (2010). Geometric image parsing in man-made environments. In: *11th European Conference on Computer vision: Part II*. ECCV'10. Heraklion, Crete, Greece: Springer-Verlag, pp. 57–70. ISBN: 3-642-15551-0, 978-3-642-15551-2. URL: http://dl.acm.org/citation.cfm?id=1888028.1888034 (see p. 110)

BAY, HERBERT, ANDREAS ESS, TINNE TUYTELAARS, and LUC VAN GOOL (June 2008). Speeded-Up Robust Features (SURF). In: *Comput. Vis. Image Underst.*, **110**:3, pp. 346–359. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014. URL: http://dx.doi.org/10.1016/j.cviu.2007.09.014 (see pp. 4, 13)

BELONGIE, SERGE, JITENDRA MALIK, and JAN PUZICHA (2002). Shape Matching and Object Recognition Using Shape Contexts. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**:4, pp. 509–522 (see pp. 3, 4, 13, 14)

BERG, ALEXANDER C., TAMARA L. BERG, and JITENDRA MALIK (2005). Shape Matching and Object Recognition Using Low Distortion Correspondences. In: *CVPR (1)*, pp. 26–33 (see pp. 3, 12, 13, 27)

BIRCHFIELD, S. (2007). *KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker*. http://www.ces.clemson.edu/stb/klt/ (see pp. 2, 12)

BLASCHKO, MATTHEW and CHRISTOPH H. LAMPERT (2008). Learning to Localize Objects with Structured Output Regression. In: *ECCV (1)*, pp. 2–15 (see pp. 7, 16)

BREIMAN, LEO and E. SCHAPIRE (2001). Random forests. In: *Machine Learning*, pp. 5–32 (see pp. 6, 15)

BROWN, MATTHEW and DAVID G. LOWE (Aug. 2007). Automatic Panoramic Image Stitching using Invariant Features. In: *Int. J. Comput. Vision*, **74**:1, pp. 59–73. ISSN: 0920-5691. DOI: 10.1007/s11263-006-0002-3. URL: http://dx.doi.org/10.1007/s11263-006-0002-3 (see pp. 2, 12)

BROWN, MATTHEW and DAVID LOWE (2002). Invariant Features from Interest Point Groups. In: *BMVC*, pp. 656–665 (see pp. 22, 24, 25)

CALVIN RESEARCH GROUP (2004). *ETHZ Toys datasets*. http://www.vision.ee.ethz.ch/~calvin/datasets.html (see p. 88)

CANNY, J (June 1986). A Computational Approach to Edge Detection. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**:6, pp. 679–698. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1986.4767851. URL: http://dx.doi.org/10.1109/TPAMI.1986.4767851 (see pp. 3, 6, 13, 15)

CECH, JAN, JORDI SANCHEZ-RIERA, and RADU HORAUD (2011). Scene flow estimation by growing correspondence seeds. In: *Computer Vision and Pattern Recognition*, pp. 3129–3136 (see pp. 28, 29)

CHENG, BIN, GUANGCAN LIU, JINGDONG WANG, ZHONGYANG HUANG, and SHUICHENG YAN (2011). Multi-task low-rank affinity pursuit for image segmentation. In: *ICCV*, pp. 2439–2446 (see pp. 6, 15)

CHERTOK, MICHAEL and YOSI KELLER (2010). Efficient High Order Matching. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**: pp. 2205–2215. ISSN: 0162-8828. DOI: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.51 (see pp. 26, 29)

CHO, M., J.M. LEE, and K.M. LEE (2010). Reweighted Random Walks for Graph Matching. In: *ECCV10*, V: 492–505 (see pp. 27, 29)

CHO, MINSU and KYOUNG MU LEE (2012). Progressive Graph Matching: Making a Move of Graphs via Probabilistic Voting. In: *Computer Vision and Pattern Recognition* (see p. 27)

CHO, MINSU, JUNGMIN LEE, and KYOUNG MU LEE (2009). Feature Correspondence and Deformable Object Matching via Agglomerative Correspondence Clustering. In: *ICCV* (see pp. 27–29, 71, 80, 87, 88)

CHOI, OUK and IN SO KWEON (2009). Robust feature point matching by preserving local geometric consistency. In: *Comput. Vis. Image Underst.*, **113**:6, pp. 726–742. ISSN: 1077-3142. DOI: http://dx.doi.org/10.1016/j.cviu.2008.12.002 (see p. 29)

CHUI, HAILI and ANAND RANGARAJAN (2003). A new point matching algorithm for non-rigid registration. In: *Computer Vision and Image Understanding*, **89**:2-3, pp. 114–141 (see p. 25)

CHUM, ONDREJ and JIRI MATAS (2005). Matching with PROSAC - Progressive Sample Consensus. In: *CVPR (1)*, pp. 220–226 (see p. 23)

CHUM, ONDREJ, JIRI MATAS, and JOSEF KITTLER (2003). Locally Optimized RANSAC. In: *Pattern Recognition, 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003, Proceedings*. Ed. by BERND MICHAELIS and GERALD KRELL. Vol. 2781. Lecture Notes in Computer Science. Springer, pp. 236–243. ISBN: 3-540-40861-4. DOI: http://springerlink.metapress.com/openurl.asp?genre=article\&issn=0302-9743\&volume=2781\&spage=236 (see p. 23)

COOTES, TIMOTHY F., CHRISTOPHER J. TAYLOR, DAVID H. COOPER, and JIM GRAHAM (1995). Active Shape Models-Their Training and Application. In: *Computer Vision and Image Understanding*, **61**:1, pp. 38–59 (see pp. 6, 15)

DELONG, ANDREW, LENA GORELICK, OLGA VEKSLER, and YURI BOYKOV (2012). Minimizing Energies with Hierarchical Costs. In: *International Journal of Computer Vision*, **100**:1, pp. 38–58 (see pp. 7, 16)

DUCHENNE, O., F. BACH, I. KWEON, and J. PONCE (Aug. 2009). A tensor-based algorithm for high-order graph matching. In: *Computer Vision and Pattern Recognition*, pp. 1980–1987. URL: http://dx.doi.org/10.1109/CVPRW.2009.5206619 (see pp. 26, 27)

DUCHENNE, OLIVIER, FRANCIS BACH, IN-SO KWEON, and JEAN PONCE (Dec. 2011). A Tensor-Based Algorithm for High-Order Graph Matching. In: *IEEE Trans. PAMI*, **33**: (12), pp. 2383–2395. ISSN: 0162-8828. DOI: http://dx.doi.org/10.1109/TPAMI.2011.110. URL: http://dx.doi.org/10.1109/TPAMI.2011.110 (see pp. 27, 29, 71, 93)

DUDA, RICHARD O. and PETER E. HART (Jan. 1972). Use of the Hough transformation to detect lines and curves in pictures. In: *Commun. ACM*, **15**:1, pp. 11–15. ISSN: 0001-0782. DOI: 10.1145/361237.361242. URL: http://doi.acm.org/10.1145/361237.361242 (see pp. 22, 24)

EBERLY, DAVID (Sept. 2008). *The Area of Intersecting Ellipses*. Tech. rep. Geometric Tools, LLC. URL: http://www.geometrictools.com/Documentation/AreaIntersectingEllipses.pdf (see pp. 129, 130, 132, 134, 136, 138)

FERRARI, VITTORIO, TINNE TUYTELAARS, and LUC J. VAN GOOL (2004). Simultaneous Object Recognition and Segmentation by Image Exploration. In: *ECCV (1)*, pp. 40–54 (see pp. 27–29, 62, 63, 73, 80, 87)

FISCHLER, MARTIN A. and ROBERT C. BOLLES (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Commun. ACM*, **24**:6, pp. 381–395 (see pp. 22, 84)

FRAHM, JAN-MICHAEL and MARC POLLEFEYS (2006). RANSAC for (Quasi-)Degenerate data (QDEGSAC). In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*. IEEE Computer Society, pp. 453–460. ISBN: 0-7695-2597-0. DOI: http://doi.ieeecomputersociety.org/10.1109/CVPR.2006.235 (see p. 23)

GOLD, STEVEN and ANAND RANGARAJAN (Apr. 1996). A Graduated Assignment Algorithm for Graph Matching. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**:4, pp. 377–388. ISSN: 0162-8828. DOI: 10.1109/34.491619. URL: http://dx.doi.org/10.1109/34.491619 (see p. 27)

GRAUMAN, KRISTEN and TREVOR DARRELL (2004). Fast Contour Matching Using Approximate Earth Mover's Distance. In: *CVPR (1)*, pp. 220–227 (see pp. 3, 13)

GROMPONE VON GIOI, RAFAEL, JEREMIE JAKUBOWICZ, JEAN-MICHEL MOREL, and GREGORY RANDALL (Apr. 2010). LSD: A Fast Line Segment Detector with a False Detection Control. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**:4, pp. 722–732. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.300. URL: http://dx.doi.org/10.1109/TPAMI.2008.300 (see pp. 3, 6, 13, 15, 117)

HACOHEN, YOAV, ELI SHECHTMAN, DAN B. GOLDMAN, and DANI LISCHINSKI (2011). Non-rigid dense correspondence with applications for image enhancement. In: *SIGGRAPH*, **30**:4 (see pp. 28, 29)

HAN, FENG and SONG-CHUN ZHU (2009). Bottom-up/Top-down Image Parsing with Attribute Graph Grammar. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**:1, pp. 59–73 (see p. 111)

HARRIS, C. and M. STEPHENS (1988). A Combined Corner and Edge Detector. In: *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151 (see pp. 3, 13)

HAUGEARD, JEAN-EMMANUEL, SYLVIE PHILIPP-FOLIGUET, and FRÉDÉRIC PRECIOSO (2009). Windows and facades retrieval using similarity on graph of contours. In: *ICIP*, pp. 269–272 (see p. 98)

ILLINGWORTH, JOHN and JOSEF KITTLER (1987). The Adaptive Hough Transform. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **9**:5, pp. 690–698 (see p. 24)

JIN, YA and STUART GEMAN (2006). Context and Hierarchy in a Probabilistic Image Model. In: *CVPR (2)*, pp. 2145–2152 (see p. 111)

KANNALA, J.H., E. RAHTU, S.S. BRANDT, and J. HEIKKILA (2008). Object recognition and segmentation by non-rigid quasi-dense matching. In: *Computer Vision and Pattern Recognition*, pp. 1–8 (see pp. 27–29, 87)

KASS, MICHAEL, ANDREW WITKIN, and DEMETRI TERZOPOULOS (1988). Snakes: Active contour models. In: *International Journal of Computer Vision*, **1**:4, pp. 321–331 (see pp. 6, 15)

KORČ, F. and W. FÖRSTNER (Apr. 2009). *eTRIMS Image Database for Interpreting Images of Man-Made Scenes*. Tech. rep. TR-IGG-P-2009-01. University of Bonn. URL: {http://www.ipb.uni-bonn.de/projects/etrims_db/} (see pp. 8, 17, 99, 102, 105)

LAZEBNIK, SVETLANA, CORDELIA SCHMID, and JEAN PONCE (2006). Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: *IEEE Conference on Computer Vision & Pattern Recognition*. URL: http://lear.inrialpes.fr/pubs/2006/LSP06 (see pp. 3, 5, 12, 14)

LEE, SUNG CHUN and RAMAKANT NEVATIA (2004). Extraction and Integration of Window in a 3D Building Model from Ground View Image. In: *CVPR (2)*, pp. 113–120 (see p. 98)

LEORDEANU, MARIUS and MARTIAL HEBERT (2005). A Spectral Technique for Correspondence Problems Using Pairwise Constraints. In: *ICCV*, pp. 1482–1489 (see pp. 12, 26, 29)

LEORDEANU, MARIUS, MARTIAL HEBERT, and RAHUL SUKTHANKAR (2009). An Integer Projected Fixed Point Method for Graph Matching and MAP Inference. In: *NIPS*. Ed. by YOSHUA BENGIO, DALE SCHUURMANS, JOHN D. LAFFERTY, CHRISTOPHER K. I. WILLIAMS, and ARON CULOTTA. Curran Associates, Inc., pp. 1114–1122. ISBN: 9781615679119 (see p. 26)

LHUILLIER, MAXIME and LONG QUAN (2002). Match Propagation for Image-Based Modeling and Rendering. In: *TPAMI*, **24**:8, pp. 1140–1146 (see pp. 27–29)

LINDEBERG, T. (1991). Discrete Scale-Space Theory and the Scale-Space Primal Sketch. In: *ISRN KTH* (see pp. 3, 13)

LINDEBERG, TONY and JONAS GÅRDING (1997). Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. In: *Image Vision Comput.*, **15**:6, pp. 415–434 (see p. 33)

LIU, YANXI, HAGIT HEL-OR, CRAIG S. KAPLAN, and LUC J. VAN GOOL (2010). Computational Symmetry in Computer Vision and Computer Graphics. In: *Foundations and Trends in Computer Graphics and Vision*, **5**:1-2, pp. 1–195 (see p. 99)

LOWE, DAVID G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision*, **60**:2, pp. 91–110 (see pp. 3, 4, 6, 13, 15, 21, 26, 27, 32, 35, 36, 41, 61, 80, 84, 86, 93)

MACIEL, JOÃO and JOÃO COSTEIRA (2003). A Global Solution to Sparse Correspondence Problems. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**:2, pp. 187–199 (see p. 26)

MALIK, JITENDRA, SERGE BELONGIE, JIANBO SHI, and THOMAS LEUNG (1999). Textons, contours and regions: Cue integration in image segmentation. In: *International Conference on Computer Vision* (see pp. 6, 15)

MARR, D. and E. HILDRETH (1980). Theory of Edge Detection. In: *Proceedings of the Royal Society of London. Series B, Biological Sciences*, **207**:1167, pp. 187–217. ISSN: 00804649. DOI: 10.2307/35407. URL: http://dx.doi.org/10.2307/35407 (see pp. 3, 13)

MATAS, J., O. CHUM, M. URBAN, and T. PAJDLA (2002). Robust wide baseline stereo from maximally stable extremal regions. In: *BMVC*, pp. 384–393 (see pp. 3, 13, 32)

MATHIAS, MARKUS, ANDELO MARTINOVIC, JULIEN WEISSENBERG, and LUC VAN GOOL (2011). Procedural 3D Building Reconstruction Using Shape Grammars and Detectors. In: *3DIMPVT*. ISBN: 978-0-7695-4369-7 (see pp. 111, 112)

MÉLER, ANTOINE, MARION DECROUEZ, and JAMES CROWLEY (2010). BetaSAC: A New Conditional Sampling For RANSAC. In: *Proceedings of the British Machine Vision Conference*. doi:10.5244/C.24.42. BMVA Press, pp. 42.1–42.11. ISBN: 1-901725-40-5 (see p. 23)

MIKOLAJCZYK, K. and C. SCHMID (2002). An affine invariant interest point detector. In: *ECCV (I)*, pp. 128–142 (see pp. 32, 33, 38, 48)

— (2004). Scale & affine invariant interest point detectors. In: *International Journal of Computer Vision*, **60**:1, pp. 63–86 (see pp. 33, 36, 69)

MIKOLAJCZYK, K., T. TUYTELAARS, C. SCHMID, A. ZISSERMAN, J. MATAS, F. SCHAFFALITZKY, T. KADIR, and L. VAN GOOL (2005). A comparison of affine region detectors. In: *International Journal of Computer Vision*, **65**: p. 2005 (see pp. 3, 13, 27, 28, 33, 34, 39–42, 67, 70, 71, 73, 77, 87, 93, 124, 126, 147)

MITTAL, S., S. ANAND, and P. MEER (2011). Generalized projection based M-estimator: Theory and applications. In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '11. Washington, DC, USA: IEEE Computer Society, pp. 2689–2696. ISBN: 978-1-4577-0394-2. DOI: 10.1109/CVPR.2011.5995514. URL: http://dx.doi.org/10.1109/CVPR.2011.5995514 (see p. 23)

MOISAN, LIONEL and BÉRENGER STIVAL (May 2004). A Probabilistic Criterion to Detect Rigid Point Matches Between Two Images and Estimate the Fundamental Matrix. In: *Int. J. Comput. Vision*, **57**:3, pp. 201–218. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000013094.38752.54. URL: http://dx.doi.org/10.1023/B:VISI.0000013094.38752.54 (see p. 23)

MÜLLER, PASCAL, PETER WONKA, SIMON HAEGLER, ANDREAS ULMER, and LUC VAN GOOL (2006). Procedural modeling of buildings. In: *ACM Transations on Graphics*, **25**:3, pp. 614–623. ISSN: 0730-0301. DOI: http://doi.acm.org/10.1145/1141911.1141931 (see pp. 110–112)

MÜLLER, PASCAL, GANG ZENG, PETER WONKA, and LUC J. VAN GOOL (2007). Image-based procedural modeling of facades. In: *ACM Transations on Graphics*, **26**:3, p. 85 (see p. 110)

MYATT, D. R., PHILIP H. S. TORR, SLAWOMIR J. NASUTO, J. MARK BISHOP, and R. CRADDOCK (2002). NAPSAC: High Noise, High Dimensional Robust Estimation - it's in the Bag. In: *BMVC*. Ed. by PAUL L. ROSIN and A. DAVID MARSHALL. British Machine Vision Association. ISBN: 1-901725-19-7 (see p. 23)

NI, KAI, HAILIN JIN, and FRANK DELLAERT (2009). GroupSAC: Efficient consensus in the presence of groupings. In: *ICCV*. IEEE, pp. 2193–2200 (see p. 23)

OHTA, Y., TAKEO KANADE, and T. SAKAI (1978). An Analysis System for Scenes Containing objects with Substructures. In: *Proceedings of the Fourth International Joint Conference on Pattern Recognitions*, pp. 752–754 (see p. 111)

— (1979). A Production System for Region Analysis. In: *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pp. 684 –686 (see p. 111)

PRITCHETT, PHILIP, ANDREW ZISSERMAN, and ANDREW ZISSERMAN (1998). Wide Baseline Stereo Matching. In: *ICCV*, pp. 754–760 (see pp. 3, 12, 25)

RAGURAM, RAHUL and JAN-MICHAEL FRAHM (2011). RECON: Scale-adaptive robust estimation via Residual Consensus. In: *ICCV*. Ed. by DIMITRIS N. METAXAS, LONG QUAN, ALBERTO SANFELIU, and LUC J. VAN GOOL. IEEE, pp. 1299–1306. ISBN: 978-1-4577-1101-5 (see p. 23)

RECKY, MICHAL and FRANZ LEBERL (2010). Windows Detection Using K-means in CIE-Lab Color Space. In: *ICPR*, pp. 356–359 (see p. 98)

RIEMENSCHNEIDER, HAYKO, ULRICH KRISPEL, WOLFGANG THALLER, MICHAEL DONOSER, SVEN HAVEMANN, DIETER FELLNER, and HORST BISCHOF (2012). Irregular lattices for complex shape grammar facade parsing. In: *Computer Vision and Pattern Recognition* (see p. 111)

ROUSSEEUW, PETER J. (Dec. 1984). Least Median of Squares Regression. In: *Journal of the American Statistical Association*, **79**:388, pp. 871–880 (see p. 22)

SCHMID, CORDELIA and ROGER MOHR (1997). Local Grayvalue Invariants for Image Retrieval. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**:5, pp. 530–535 (see pp. 3, 12)

SHI, JIANBO and JITENDRA MALIK (2000). Normalized Cuts and Image Segmentation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, (see pp. 7, 16)

SHI, JIANBO and CARLO TOMASI (1994). Good Features to Track. In: *Computer Vision and Pattern Recognition*, pp. 593–600 (see pp. 2, 12)

SHOTTON, JAMIE, MATTHEW JOHNSON, and ROBERTO CIPOLLA (2008). Semantic texton forests for image categorization and segmentation. In: *Computer Vision and Pattern Recognition* (see pp. 6, 15)

SIMON, LOÏC, OLIVIER TEBOUL, PANAGIOTIS KOUTSOURAKIS, and NIKOS PARAGIOS (2011). Random Exploration of the Procedural Space for Single-View 3D Modeling of Buildings. In: *International Journal of Computer Vision*, **93**:2, pp. 253–271 (see p. 111)

SNAVELY, NOAH, STEVEN M. SEITZ, and RICHARD SZELISKI (2008). Modeling the World from Internet Photo Collections. In: *Int. J. Comput. Vision*, **80**:2, pp. 189–210. ISSN: 0920-5691. DOI: http://dx.doi.org/10.1007/s11263-007-0107-3 (see pp. 3, 12, 79, 80, 84)

STEWART, CHARLES V. (Oct. 1995). MINPRAN: A New Robust Estimator for Computer Vision. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**:10, pp. 925–938. ISSN: 0162-8828. DOI: 10.1109/34.464558. URL: http://dx.doi.org/10.1109/34.464558 (see p. 22)

STINY, GEORGE NICHOLAS (1975). *Pictorial and formal aspects of shape and shape grammars and aesthetic systems*. AAI7526993. PhD thesis. University of California, Los Angeles (see p. 112)

TEBOUL, OLIVIER (2010). *École Centrale Paris datasets*. http://vision.mas.ecp.fr/Personnel/teboul/data.php (see pp. 6, 8, 15, 17, 102, 118, 119, 159)

TEBOUL, OLIVIER, LOÏC SIMON, PANAGIOTIS KOUTSOURAKIS, and NIKOS PARAGIOS (2010). Segmentation of building facades using procedural shape priors. In: *Computer Vision and Pattern Recognition*, pp. 3105–3112 (see pp. 6, 15, 110–112)

TEBOUL, OLIVIER, IASONAS KOKKINOS, LOÏC SIMON, PANAGIOTIS KOUTSOURAKIS, and NIKOS PARAGIOS (2011). Shape grammar parsing via Reinforcement Learning. In: *Computer Vision and Pattern Recognition*, pp. 2273–2280 (see pp. 6, 15, 103, 110–112, 114, 115, 117, 119–121, 123, 125)

TOLA, ENGIN, VINCENT LEPETIT, and PASCAL FUA (2010). DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**:5, pp. 815–830 (see pp. 4, 6, 13, 15)

TORR, P. H. S. and A. ZISSERMAN (2000). MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. In: *Computer Vision and Image Understanding*, **78**: pp. 138–156 (see p. 23)

VIOLA, PAUL A. and MICHAEL J. JONES (2004). Robust Real-Time Face Detection. In: *International Journal of Computer Vision*, **57**:2, pp. 137–154 (see pp. 7, 16, 99, 101, 103, 159, 161)

VU, HOANG-HIEP, PATRICK LABATUT, JEAN-PHILIPPE PONS, and RENAUD KERIVEN (2012). High Accuracy and Visibility-Consistent Dense Multiview Stereo. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, **34**:5, pp. 889–901 (see pp. 80, 83, 84, 86, 87)

WANG, HANZI and DAVID SUTER (Nov. 2004). Robust Adaptive-Scale Parametric Model Estimation for Computer Vision. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**:11, pp. 1459–1474. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2004.109. URL: http://dx.doi.org/10.1109/TPAMI.2004.109 (see p. 23)

WANG, LU, ULRICH NEUMANN, and SUYA YOU (2009). Wide-baseline image matching using Line Signatures. In: *ICCV*. IEEE, pp. 1311–1318 (see pp. 4, 14)

WANG, WILEY, ILYA POLLAK, TAK-SHING WONG, CHARLES A. BOUMAN, and MARY P. HARPER (2006). Hierarchical stochastic image grammars for classification and segmentation. In: *IEEE Transactions on Image Processing*, **15**: pp. 3033–3052 (see p. 111)

WEICKERT, JOACHIM, SEIJI ISHIKAWA, and ATSUSHI IMIYA (May 1999). Linear Scale-Space has First been Proposed in Japan. In: *J. Math. Imaging Vis.*, **10**:3, pp. 237–252. ISSN: 0924-9907. DOI: 10.1023/A:1008344623873. URL: http://dx.doi.org/10.1023/A:1008344623873 (see p. 3)

YANG, CHAO, TIAN HAN, LONG QUAN, and CHIEW-LAN TAI (2012). Parsing façade with rank-one approximation. In: *Computer Vision and Pattern Recognition*. IEEE, pp. 1720–1727. ISBN: 978-1-4673-1226-4. URL: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#YangHQT12 (see p. 111)

ZASS, RON and AMNON SHASHUA (2008). Probabilistic graph and hypergraph matching. In: *Computer Vision and Pattern Recognition* (see pp. 26, 27, 29)

ZHENG, Y. and D. DOERMANN (2004). *Robust Point Matching for Non-Rigid Shapes: A Relaxation Labeling Based Approach*. Tech. rep. LAMP-TR-117. UMIACS (see p. 27)

ZHENG, YEFENG and DAVID DOERMANN (2006). Robust point matching for nonrigid shapes by preserving local neighborhood structures. In: *TPAMI*, **28**: p. 2006 (see pp. 12, 27)

ZHU, SONG-CHUN and DAVID MUMFORD (2006). A stochastic grammar of images. In: *Foundations and Trends in Computer Graphics and Visions*, **2**:4, pp. 259–362. ISSN: 1572-2740. DOI: http://dx.doi.org/10.1561/0600000018 (see pp. 5, 15, 110–112)

ZULIANI, MARCO, CHARLES S. KENNEY, and B. S. MANJUNATH (2005). The multiRANSAC algorithm and its application to detect planar homographies. In: *ICIP (3)*, pp. 153–156 (see p. 100)