# Une technique de relaxation pour la mise en correspondance d'images: Application à la reconnaissance d'objets et au suivi du visage.

Dro Désiré Sidibe

UNIVERSITÉ MONTPELLIER II

— SCIENCES ET TECHNIQUE DU LANGUEDOC —

# THÈSE

pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITÉ MONTPELLIER II**

DISCIPLINE : **INFORMATIQUE**
*Formation Doctorale* : **INFORMATIQUE**
*Ecole Doctorale* : **Information, Structure, Systèmes**

présentée et soutenue par

DRO DÉSIRÉ SIDIBE

le 07 décembre 2007

***Titre :***

# UNE TECHNIQUE DE RELAXATION POUR LA MISE EN CORRESPONDANCE D'IMAGES

## Application à la reconnaissance d'objets et au suivi du visage

**JURY**

Christine Fernandez-Maloigne, Professeur, Université Poitiers ..... Rapporteur
Frédéric Jurie, Professeur, Université Caen ..... Rapporteur
Valérie Gouet-Brunet, Enseignant-Chercheur, CNAM Paris ..... Examinateur
Réné Zapata, Professeur, Université Montpellier II ..... Examinateur
Philippe Montesinos, Enseignant-Chercheur, Ecole des Mines Alès ..... Encadrant de proximité
Jean-Claude Bajard, Professeur, Université Montpellier II ..... Directeur de thèse

UNIVERSITÉ MONTPELLIER II

— SCIENCES ET TECHNIQUE DU LANGUEDOC —

# THÈSE

pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITÉ MONTPELLIER II**

DISCIPLINE : **INFORMATIQUE**
*Formation Doctorale* : **INFORMATIQUE**
*Ecole Doctorale* : **Information, Structure, Systèmes**

présentée et soutenue par

DRO DÉSIRÉ SIDIBE

le 07 décembre 2007

***Titre :***

# UNE TECHNIQUE DE RELAXATION POUR LA MISE EN CORRESPONDANCE D'IMAGES

## Application à la reconnaissance d'objets et au suivi du visage

**JURY**

Christine Fernandez-Maloigne, Professeur, Université Poitiers ..... Rapporteur
Frédéric Jurie, Professeur, Université Caen ..... Rapporteur
Valérie Gouet-Brunet, Enseignant-Chercheur, CNAM Paris ..... Examinateur
Réné Zapata, Professeur, Université Montpellier II ..... Examinateur
Philippe Montesinos, Enseignant-Chercheur, Ecole des Mines Alès ..... Encadrant de proximité
Jean-Claude Bajard, Professeur, Université Montpellier II ..... Directeur de thèse

Université Montpellier II (Université des Sciences et Techniques du Languedoc)
LGI2P

---

**Références**

SIDIBE, Dro Désiré « Une technique de relaxation pour la mise en correspondance d'images *Application à la reconnaissance d'objets et au suivi du visage* », Thèse de doctorat, Université Montpellier II, 07 décembre 2007

---



---

*Aucunes choses ne méritent de détourner notre route ;*
*embrassons-les toutes en passant ;*
*mais notre but est plus loin qu'elles.*
***André Gide***

A mes parents, Jeanette et Bernard.

# Remerciements

Une thèse dure, généralement, trois ans. Trois années, cela est à la fois long et pas assez.

On est parfois seul, et souvent accompagné. Aussi, je voudrais remercier toutes les personnes qui d'une manière ou d'une autre m'ont aidé à passer ces trois années dans les meilleures conditions. Je remercie tout particulièrement mon encadrant Philippe Montesinos pour sa patience, ses explications et ses critiques toujours éclairées. J'ai particulièrement apprécié l'autonomie qu'il m'a accordée dans les choix et les orientations de mon travail. Je remercie aussi chaleureusement Stefan Janaqi, de fait mon second encadrant, pour ses conseils et son aide précieuse pour la résolution de nombreux problèmes mathématiques. Je remercie également Jean-Claude Bajard, mon directeur de thèse, pour ses conseils et son soutien pendant ces trois années.

Merci enfin aux membres du jury qui m'ont fait l'honneur d'accepter de juger mon travail. Merci à Mme Christine Fernandez-Maloigne et à Mr Frédéric Jurie d'avoir accepté d'être les rapporteurs de cette thèse. Je les remercie pour leurs remarques et suggestions pertinentes qui m'ont permis d'améliorer ce manuscrit. Merci à Mme Valérie Gouet-Brunet et à Mr Réné Zapata d'avoir accepté de faire partie de mon jury de thèse.

Cette thèse s'est déroulée au sein du laboratoire LGI2P sur le site ERIEE de l'Ecole des Mines d'Alès. Je remercie toutes les personnes qui y travaillent pour leur accueil et leur soutien. En particulier, Françoise pour sa bonne humeur contagieuse et Sylvie, qui a quitté le labo il y a quelques mois, pour son aide précieuse dans toutes les démarches administratives.

Il y a aussi tous les autres doctorants du labo avec qui une complicité et une amitié s'installent au cours du temps. Un grand merci à mon compagnon de route Saber, ainsi qu'aux jeunes Sylvain, Kamel, Sofiane, Wael, Gladys, et tous les autres.

Je remercie aussi ma famille pour son soutien sans faille malgré la distance qui me sépare d'elle. Un grand merci enfin à tous les copains pour leur aide ; je n'oublie pas les soirées ciné, poker et les parties de football qui permettent d'oublier la thèse un moment.

# Table des matières

# Liste des figures

# Liste des tableaux

# Chapitre 1

# Introduction

*Le regard ne s'empare pas des images,*
*ce sont elles qui s'emparent du regard.*
*Elles inondent la conscience.*
***Franz Kafka***

La vision est sans doute notre sens le plus développé et, du point de vue de l'évolution, le plus utile. Nous nous servons quotidiennement de notre système de vision pour nous déplacer, pour estimer les distances, pour identifier les personnes et les objets qui nous entourent, etc. Nous le faisons sans aucune difficulté, sans même y prêter attention, bien que les processus mis en jeu soient assez complexes. La vision par ordinateur est une discipline à la frontière de l'informatique, des mathématiques, de la physique, des neurosciences, et de diverses autres disciplines, qui a pour but de simuler la vision humaine, si ce n'est la comprendre, pour en doter les ordinateurs. Autrement dit, "faire voir les ordinateurs" selon la terminologie anglaise consacrée "make computers see".

La vision par ordinateur est un domaine de recherche qui n'a cessé de se développer depuis le début des années 40, et qui trouve aujourd'hui des applications dans de nombreux secteurs d'activité. Les systèmes d'imagerie, caméras et systèmes de vision, sont de plus en plus accessibles et performants, et induisent des progrès considérables dans les domaines de la santé (scanners, endoscopes, échographes, etc), de l'industrie (réalisation de tâches dans des environnements à risque), de la production (systèmes de production automatisés) ou de la communication (réalité virtuelle, télévision numérique, 3D TV, etc).

Des développements importants ont été réalisés, mais la vision par ordinateur reste un champ d'investigation très actif avec de nombreux problèmes difficiles et non entièrement résolus, et l'émergence de nouvelles perspectives dues à l'évolution des moyens de

communication.

D'une manière générale, la vision par ordinateur peut être considérée comme un processus de traitement de l'information, information issue d'images numérisées [84]. Les questions qui se posent alors concernent la nature de ces informations et leur représentation : quelle sorte d'information extraire de l'image ? comment décrire et/ou représenter cette information pour en faciliter l'interprétation ?

Il est clair que la nature et la représentation des informations dépendent de l'application envisagée. Cependant, dans tout processus d'analyse d'image, il faut pouvoir extraire certaines parties de l'image, mesurer des propriétés de ces parties ou des relations entre ces parties, et utiliser les valeurs de ces propriétés pour interpréter le contenu de l'image [115]. Ces trois étapes d'extraction, de caractérisation et d'interprétation, sont présentes dans presque toutes les applications.

Dans de nombreuses applications, une fois les informations utiles extraites et caractérisées, il faut résoudre le problème de l'appariement ou de la mise en correspondance d'images. En effet, en l'absence de toute autre information, une image seule nous apprend bien peu de choses et ne permet pas une interprétation complète et non ambigüe de la scène représentée. Mais lorsqu'on associe plusieurs images ou une image et d'autres types d'information, alors on est capable de réaliser des tâches difficiles telles que la reconnaissance d'objets, la localisation dans l'espace 3D ou l'estimation de distances.

Lorsque nous identifions un objet à partir d'une image, c'est parce que le système visuel est capable d'associer des éléments présents dans l'image à des informations déjà présentes dans notre mémoire. De même, c'est la mise en correspondance d'images qui nous permet d'estimer la distance d'un objet en utilisant nos deux yeux comme un système stéréoscopique et en estimant le relief à partir de la disparité entre les deux images.

Toutefois, en toute rigueure, la mise en correspondance d'images est un problème "*mal posé*" car nous ne possédons pas suffisamment d'information pour le résoudre. En effet, une image est la projection bidimensionnelle (2D) d'un monde tridimensionnel (3D) et cette projection entraîne nécessairement une perte d'information. Plusieurs éléments de la scène 3D peuvent avoir une même projection 2D et un même élément 3D peut avoir plusieurs projections 2D.

Pour résoudre ces difficultés, il faut d'une part, pouvoir extraire des images à apparier des éléments caractéristiques. Ceux-ci doivent être stables sous l'effet de diverses transformations pour être détectés dans chacune des images. D'autre part, il faut fournir une description assez robuste et discriminante de ces éléments qui permette d'identifier correc-

tement ceux qui se correspondent.

En fonction de l'application, des hypothèses peuvent être avancées pour simplifier le problème. Dans le cas de la stéréovision par exemple, on suppose que le changement de point de vue entre les images à apparier est assez faible. Les coordonnées des éléments caractéristiques et la distribution d'intensité lumineuse autour de chaque élément sont proches dans les images, et on peut trouver les correspondants en utilisant une méthode de corrélation. De même, si les paramètres des caméras sont connus, on dit alors qu'on travaille dans un environnement calibré, la géométrie épipolaire reliant deux images peut être utilisée pour réduire l'espace de recherche des correspondants des éléments d'une image dans l'autre image.

Dans le cas général cependant, on se trouve dans des environnements non calibrés et les changements de point de vue entre les images sont quelconques. Cela rend la mise en correspondance assez difficile et nécessite la mise en œuvre de méthodes capables de prendre en compte ces difficultés.

Ces dernières années, les invariants locaux se sont revélés être très adaptés et très efficaces pour l'appariement de différentes vues d'une même scène, notamment depuis les travaux de Schmid et Mohr [123]. Le terme d'invariants locaux désigne des régions de l'image invariantes aux transformations géométriques, principalement affines, ainsi qu'aux changements d'illumination de la scène. Le caractère local les rend robustes aux occultations ainsi qu'aux changements de fond et l'invariance assure la robustesse aux changements de point de vue et de l'échelle. Les points d'intérêt en sont un exemple largement utilisé, et de nombreux détecteurs de régions invariantes de l'image sont proposés dans la littérature [8, 91, 153, 80, 120, 86].

Une fois les points d'intérêt détectés, la région de l'image autour de chaque point est utilisée pour calculer un descripteur. L'invariance aux transformations affines est assurée d'une part, par le fait que chaque point est défini par une échelle caractéristique et d'autre part, par le fait que chaque région possède une orientation spécifique. Plusieurs descripteurs ont été proposés dans la littérature et le descripteur SIFT (Scale Invariant Feature Transform) est aujourd'hui considéré comme étant le plus performant [92]. Ce descripteur introduit par Lowe [79], décrit le voisinage local d'un point par un histogramme 3D de la distribution des orientations du gradient.

Il est alors possible d'établir des correspondances entre les points d'intérêt détectés dans les images en utilisant une mesure de similarité entre les descripteurs locaux.

## 1.1 Motivations

L'utilisation d'un détecteur et d'une description invariants aux transformations affines permet d'obtenir d'excellents résultats dans de nombreuses applications [122, 81, 54, 127]. Cependant, malgré ces bonnes performances, la robustesse de ces approches locales est limitée par la répétabilité du détecteur utilisé et par la difficulté de trouver des correspondants corrects en présence de fortes occultations ou de changements de points de vue important entre les images. Dans la plupart des cas, la mise en correspondance conduit à des faux appariements qu'il faut ensuite éliminer par des méthodes coûteuses telle que l'estimation de la transformation géométrique reliant les images avec l'algorithme RANSAC par exemple [38].

D'autre part, lorsque les images à apparier présentent de nombreuses structures répétitives, les invariants locaux ne permettent pas de trouver les correspondants corrects. Dans ces cas en effet, toutes les régions d'intérêt sont décrites presque de la même manière par un descripteur local et il est difficile, voire impossible, de trouver les correspondants corrects.

Dans une application de reconnaissance d'objets, où on souhaite identifier un objet, représenté par une image, dans une scène complexe qui peut contenir plusieurs autres objets, il faut pouvoir apparier l'image représentant l'objet avec une partie, relativement petite, de l'image de la scène contenant l'objet. Il peut donc y avoir des occultations, et il y a un nombre restreint de primitives de l'objet dans l'image de la scène parmi un nombre relativement important d'autres primitives.

Pour tirer partie de la robustesse des invariants locaux dans ces cas, il est nécessaire de mettre en œuvre une méthode de mise en correspondance robuste. C'est ce que nous faisons dans la première partie de cette thèse, consacrée à la mise en correspondance d'images. Nous montrons les limites de l'utilisation des invariants locaux et proposons une méthode robuste de mise en correspondance, particulièrement utile dans le cadre de la reconnaissance d'objets.

Dans une seconde partie, nous nous intéressons au problème de la détection et du suivi du visage dans une séquence d'images. Détecter le visage, le reconnaître si besoin et le suivre dans une séquence d'images est à la base de nombreuses applications faisant intervenir les intéractions homme-machine. La détection du visage est néanmoins une tâche difficile à cause de la variabilité de la taille, de l'apparence et de l'orientation que peut avoir un visage. De plus, les expressions faciales, les occultations et les conditions d'illumination affectent également l'apparence du visage. Nous proposons une méthode de détection du

visage basée sur la détection des yeux et nous montrons comment la méthode de mise en correspondance développée dans la première partie peut être utilisée pour le suivi du visage dans une séquence d'images.

## 1.2 Organisation de la thèse et principales contributions

Ce mémoire de thèse est divisé en deux parties, chaque partie correspondant à l'un des deux principaux problèmes abordés. Les deux parties, traitant de problèmes différents, peuvent être abordées de manière indépendante. Toutefois, le dernier chapitre de la seconde partie fait appel à une méthode développée dans la première.

La première partie est dédiée à la reconnaissance d'objets dans une image en utilisant des primitives locales et elle est organisée en quatre chapitres. Dans les chapitres 2 et 3, nous présentons les différentes méthodes de détection et de mise en correspondance des invariants locaux présentées dans la littérature, ainsi que leurs principales limitations. Dans le chapitre 4, nous proposons une méthode robuste de mise en correspondance, et nous présentons son application à la reconnaissance d'objets dans le chapitre 5.

La seconde partie de la thèse aborde le problème de la détection et du suivi du visage dans une séquence d'images. Pour suivre le visage, il faut dans un premier temps le détecter dans la première image de la séquence. La détection de la peau est abordée dans le chapitre 6. Dans le chapitre 7, nous proposons une méthode de détection du visage basée sur la détection des yeux. Ensuite, certains points particulier du visage, ici les yeux et le nez, sont utilisés pour le suivre au cours du temps dans le chapitre 8. Cette dernière partie fait appel à la méthode de mise en correspondance développée dans le chapitre 4.

Dans la première partie, nos principales contributions concernent la mise en évidence de la nécessité de prendre en compte des informations contextuelles, et la mise en œuvre d'un algorithme robuste de mise en correspondance des invariants locaux. Cet algorithme est basé sur la technique de relaxation et les résultats obtenus montrent la supériorité de notre approche par rapport à diverses autres méthodes. Les travaux effectués dans cette partie ont été publiés dans [132, 133, 134, 135].

Dans la seconde partie, nous proposons une méthode simple et efficace pour la détection du visage dans une image couleur et l'application de l'algorithme de mise en correspondance dans le cadre du suivi du visage dans une séquence d'images. Une partie de ces travaux a été publiée dans [131].

# Première partie

# Mise en correspondance d'invariants locaux : reconnaissance d'objets

*Vision is the process of discovering from images what is present and where it is.*
***David Marr***

Le principal problème abordé dans cette partie du mémoire est celui de la mise en correspondance d'images. Celui-ci, est l'un des problèmes les plus anciens et, par conséquent, l'un des plus étudiés dans le domaine de la vision par ordinateur. La littérature sur ce sujet étant assez dense, nous nous proposons dans le chapitre 2 de décrire brièvement le problème, de présenter la manière dont nous l'abordons ainsi que les approches les plus récentes. Le chapitre 3 présente les limites de la mise en correspondance par l'utilisation des invariants locaux et la nécessité de prendre en compte des informations contextuelles. Puis dans le chapitre 4, nous présentons une méthode robuste de mise en correspondance basée sur la technique de relaxation qui tient compte de l'information contextuelle. Enfin, dans le chapitre 5, nous présentons l'application de cette méthode à un problème particulier, celui de la reconnaissance d'objets.

# Chapitre 2

# Mise en correspondance d'images par invariants locaux : Un état de l'art

Ce chapitre présente un état de l'art de la mise en correspondance d'images basée sur l'utilisation des invariants locaux. La littérature sur ce sujet étant assez dense, le but de ce chapitre est de décrire brièvement le problème, de présenter la manière dont nous l'abordons ainsi que les approches les plus récentes. Cet état de l'art n'est en aucun cas exhaustif, et nous renvoyons le lecteur intéressé aux différentes références fournies dans le texte.

## 2.1 Introduction

Le problème de la mise en correspondance d'images consiste à identifier dans deux ou plusieurs images d'une même scène, les primitives qui "*se correspondent*". Le terme de primitives désigne des points ou des régions particulières de l'image riches en information. Les primitives utilisées seront présentées plus en détails dans la suite de ce chapitre. Par *se correspondre*, nous entendons les primitives 2D qui sont les projections d'un même point 3D de la scène, comme l'illustre la figure 2.1.

On peut aussi établir des correspondances entre des images ne représentant pas exactement la même scène. C'est le cas, par exemple, dans les applications de reconnaissance d'objets où l'on cherche à identifier une zone de l'image contenant l'objet en question. Ce problème particulier sera abordé plus en détail dans le chapitre 5. Dans tous les cas, il s'agit d'identifier les zones des deux images qui se correspondent.

La mise en correspondance est une étape essentielle dans de nombreuses applications

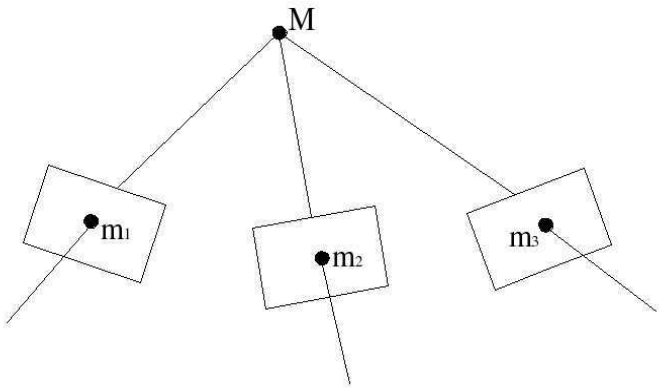



Fig. 2.1 – Illustration de la notion de correspondance : les trois points $m_1$, $m_2$ et $m_3$ se correspondent car ils sont issus de la projection du même point M.

de la vision par ordinateur. Nous énumérons ci-dessous quelques unes de ces applications :

- *La stéréovision.* Le problème de la stéréovision consiste à estimer la position d'un point $M$ de l'espace connaissant celles de ses projections $m_1$ et $m_2$ sur deux images. On parle de stéréovision binoculaire dans le cas de deux images, mais on peut également utiliser un nombre plus important d'images. Cela passe par l'estimation de la géométrie reliant les deux images, l'homographie ou la matrice fondamentale par exemple [164, 53], et il est donc nécessaire d'établir des correspondances entre les deux vues.

- *La reconnaissance d'objets.* Etant donné une ou plusieurs images d'un objet (définissant le modèle), déterminer si celui-ci est présent dans une nouvelle image (image test). Il faut pouvoir identifier des primitives de l'objet dans l'image test. Plus on trouve de correspondants, plus la présence de l'objet est probable, et plus il peut être localiser avec précision [41, 79, 36]. Nous abordons plus en détail ce problème dans le chapitre 5.

- *L'indexation d'images ou de vidéos.* L'indexation est un cas particulier de la reconnaissance d'objets lorsqu'il y a plusieurs images, de l'ordre de quelques centaines ou milliers, et que l'objet recherché est présent dans une petite portion de ces images. Le problème de la reconnaissance devient alors celui de la recherche d'un objet particulier dans une base de données [123, 152, 138]. Dans le cas de séquence d'images, on peut tirer avantage de la continuité temporelle de la vidéo pour identifier quelles

sont les images de la séquence qui contiennent l'objet.

- *La navigation de robots.* Un robot a besoin de se déplacer et de se localiser dans son environnement. Les obstacles et les positions dans l'environnement peuvent être obtenus à partir d'images grâce à l'établissement de correspondances. Dans cette application particulière, les contraintes sur le robot, ses mouvements et son environnement peuvent être prises en compte pour de meilleurs performances [154, 127].

Comme déjà évoqué, l'établissement de correspondances entre deux images passe par l'identification de primitives qui sont les projections d'un même point de la scène tridimensionnelle. On distingue généralement deux approches qui sont la mise en correspondance dense et la mise en correspondance éparse. Dans la première approche, l'objectif est la mise en correspondance de tous les pixels visibles dans les deux images (pixels non occultés). Rechercher le correspondant correct de chacun des pixels d'une image est un problème d'une complexité très élevée et il est indispensable d'utiliser des contraintes géométriques pour réduire cette complexité. On peut par exemple utiliser la contrainte épipolaire qui est toujours vérifiée dans le cas de scènes rigides. Dans la pratique, il faut estimer cette géométrie épipolaire à partir d'un certain nombre de primitives correctement appariés (sauf dans le cas où les caméras sont calibrés). On procède donc, de manière classique, à la mise en correspondance éparse de certaines primitives avant d'effectuer une mise en correspondance dense.

Dans la suite de ce chapitre et de cette partie, nous nous intéresserons uniquement à la mise en correspondance éparse qui consiste à apparier un certain nombre de primitives particulières détectées dans les images. Les points et les régions d'intérêt sont des primitives largement utilisées dans la littérature. En fait, la notion de région d'intérêt recouvre celle de point d'intérêt car chaque point d'intérêt détecté est décrit par une petite région de l'image qui l'entoure. Comme nous le verrons dans la section suivante, certains détecteurs donnent directement des régions, tandis que d'autres détectent dans un premier temps des points.

D'une manière générale, la mise en correspondance d'images par l'utilisation de primitives locales comporte trois principales étapes :

- La détection des primitives ;
- La caractérisation des primitives par un descripteur invariant aux principales transformations de l'image ;
- L'appariement des primitives.

Les deux étapes de détection et de caractérisation doivent êtres invariantes aux principales transformations de l'image pour assurer un nombre maximal de correspondants corrects dans l'étape d'appariement. Soient $I_1$ et $I_2$ deux vues d'une même scène prises dans des conditions différentes, il existe deux types de transformations possibles entre les deux images [45] :

1. Les transformations *photométriques* du type :

$$I_2(x, y) = f(I_1(x, y))$$

   qui traduisent un changement d'illumination. $f$ est généralement une translation ou une fonction affine des intensités lumineuses ;

2. Les transformations *géométriques* du type :

$$I_2(x, y) = I_1(g(x, y))$$

   qui traduisent un changement de point de vue. $g$ est généralement une homograpie du plan projectif.

Dans la pratique, la phase de détection assure l'invariance aux transformations géométriques, et l'invariance aux transformations photométriques est prise en compte dans la phase de caractérisation des régions.

Parce que les primitives sont caractérisées par de petites régions compactes de l'image, elles sont désignées par le terme d'*invariants locaux* (*local invariant features* en anglais). Les invariants locaux possèdent de nombreux avantages :

- *la robustesse* : d'une part, le caractère local assure une certaine robustesse aux occultations et aux variations du fond de l'image. D'autre part, l'invariance (de la détection et de la caractérisation) assure la robustesse à des changements de point de vue et d'échelle.
- *la répétabilité* : les détecteurs, présentés dans la section suivante, sont capables de trouver des primitives correspondantes simultanément dans deux images, en dépit de transformations géométriques et photométriques.
- *la compacité* : le nombre de primitives détectées est généralement très faible par rapport au nombre de pixels de l'image, conduisant à une représentation compacte de l'information contenue dans l'image.
- etc.

Du fait de ces avantages, les invariants locaux se sont avérés, ces dernières années, être particulièrement efficaces pour l'établissement de correspondances entre différentes vues d'une même scène. Il existe un nombre important de travaux concernant la détection, la description et l'appariement des invariants locaux. Dans les sections suivantes, nous présentons les principales approches.

## 2.2 Détection des invariants locaux

### 2.2.1 Les points d'intérêt

Les points d'intérêt sont largement utilisés dans la littérature pour la mise en correspondance d'images. La notion de point d'intérêt[1], introduite par Moravec [97], permet de caractériser les endroits où le signal est riche en information. Selon Moravec, un point d'intérêt est un point de l'image où l'intensité lumineuse varie beaucoup dans plusieurs directions (au moins deux directions simultanément). Le signal contient donc plus d'information en ces points qu'en des points correspondant à un changement unidimensionnel du signal (points de contour par exemple). Un grand nombre de travaux ont été réalisés concernant la détection des points d'intérêt [97, 68, 30, 139], et le détecteur le plus utilisé est celui de Harris [52]. Celui-ci est basé sur la fonction d'autocorrélation du signal et nous le décrivons sommairement ci-dessous.

Une mesure des variations locales de l'image $I$ au point $\mathbf{x} = (x, y)^T$ associée à un déplacement $\Delta\mathbf{x} = (\Delta x, \Delta y)$ est fournie par la *fonction d'autocorrélation* :

$$\chi(\mathbf{x}) = \sum_{\mathbf{x} \in W} (I(\mathbf{x}) - I(\mathbf{x} + \Delta\mathbf{x}))^2 \tag{2.1}$$

où $W$ est une fenêtre centrée au point $\mathbf{x}$.

En utilisant une approximation du premier ordre :

$$I(\mathbf{x} + \Delta\mathbf{x}) \simeq I(\mathbf{x}) + \begin{pmatrix} \frac{\partial I}{\partial x}(\mathbf{x}) & \frac{\partial I}{\partial y}(\mathbf{x}) \end{pmatrix} . \Delta\mathbf{x}$$

[1] on utlise aussi le terme de coin dans la littérature

On a donc :

$$\begin{aligned}\chi(\mathbf{x}) &= \sum_{\mathbf{x}\in W}\left[\left(\frac{\partial I}{\partial x}(\mathbf{x}) \quad \frac{\partial I}{\partial y}(\mathbf{x})\right).\Delta\mathbf{x}\right]^2 \\ &= \Delta\mathbf{x}^T\, M(\mathbf{x})\, \Delta\mathbf{x}\end{aligned}$$

où la matrice d'autocorrélation $M(\mathbf{x})$ représente la variation locale de l'image $I$ en $\mathbf{x}$ :

$$M(\mathbf{x}) = \begin{pmatrix} \sum_{(x_k,y_k)\in W}(\frac{\partial I}{\partial x}(x_k,y_k))^2 & \sum_{(x_k,y_k)\in W}\frac{\partial I}{\partial x}(x_k,y_k).\frac{\partial I}{\partial y}(x_k,y_k) \\ \sum_{(x_k,y_k)\in W}\frac{\partial I}{\partial x}(x_k,y_k).\frac{\partial I}{\partial y}(x_k,y_k) & \sum_{(x_k,y_k)\in W}(\frac{\partial I}{\partial y}(x_k,y_k))^2 \end{pmatrix}$$

Le point $\mathbf{x} = (x, y)$ est considéré comme un point d'intérêt, si pour tout déplacement $\Delta\mathbf{x}$, la quantité $\chi(\mathbf{x})$ est grande. En d'autres termes, les points d'intérêt sont les points $\mathbf{x}$ pour lesquels la matrice d'autocorrélation $M(\mathbf{x})$ a *deux valeurs propres grandes.*

Dans la pratique, la mesure d'autocorrélation est estimée à partir des dérivées premières calculées sur un support gaussien de taille $\sigma_D$.

$$M(\mathbf{x},\sigma_I,\sigma_D) = G(\sigma_I) \otimes \begin{bmatrix} I_x^2(\mathbf{x},\sigma_D) & I_x(\mathbf{x},\sigma_D)I_y(\mathbf{x},\sigma_D) \\ I_x(\mathbf{x},\sigma_D)I_y(\mathbf{x},\sigma_D) & I_y^2(\mathbf{x},\sigma_D) \end{bmatrix} \tag{2.2}$$

Dans cette expression, $\sigma_D$ représente l'écart type de la gaussienne utilisée pour calculer les dérivées de l'image, c'est la taille de la fenêtre de dérivation. La convolution par la gaussienne d'écart type $\sigma_I$, joue le rôle de la sommation sur la fenêtre $W$. $\sigma_I$ représente la taille de la fenêtre d'intégration. Ces deux paramètres, $\sigma_D$ et $\sigma_I$, peuvent avoir des valeurs différentes.

Pour éviter le calcul de ces valeurs propres, Harris et Stephens [52] proposent l'opérateur suivant :

$$k_H = Det(M) - \alpha.Trace^2(M) \tag{2.3}$$

Les points d'intérêt sont obtenus en prenant les maxima locaux de cet opérateur. $\alpha$ est une constante déterminée de manière empirique. Harris et Stephens proposent de prendre $\alpha = 0.04$.

L'opérateur de Harris et Stephens est en fait une version modifiée de celui proposé par

FIG. 2.2 – Le problème de l'invariance : les deux régions circulaires, de même taille, ne recouvrent pas la même zone dans les deux images.

Noble [101] :

$$1/k_N = \frac{det(M)}{trace(M)} \tag{2.4}$$

D'autres auteurs utilisent également la matrice $M$, par exemple Rorh dans [113] extrait les points d'intérêt en maximisant le déterminant de $M$.

Notons enfin que le détecteur de Harris a été étendu à la détection de points d'intérêt dans une image couleur par Montesinos et Gouet dans [95] et [45].

### 2.2.2 Le besoin de l'invariance affine

Pour établir les correspondances, il faut pouvoir comparer les points détectés dans les deux images. Pour cela, chaque point est caractérisé par un descripteur calculé dans une région avoisinant le point. La méthode de caractérisation la plus ancienne est la corrélation et dans ce cas, la fenêtre utilisée pour calculer le descripteur est de taille et de forme fixes.

Le problème de la mise en correspondance devient difficile quand la différence de point de vue entre les deux images est importante ou quand le changement d'échelle est significatif. Dans le premier cas, les fenêtres de corrélation dans les deux images ne recouvrent pas les mêmes parties de l'image. Dans le second cas, elles n'ont pas la même taille. Il est également acquis que la localisation des points de Harris varie en fonction de l'échelle de calcul (les paramètres $\sigma_D$ et $\sigma_I$) [30]. Par conséquent, la méthode classique "points de Harris + fenêtre fixe de corrélation" échoue quand les images à apparier présentent des différences de point de vue significatives. Un exemple de cette difficulté est présenté à la figure 2.2.

En général, pour une scène tridimensionnelle quelconque, il n'existe pas de transfor-

mation géométrique globale reliant un point d'une image à son correspondant dans l'autre image. Ce qui peut s'expliquer par le fait qu'une image est la projection bidimensionnelle (2D) d'un monde tridimensionnel (3D) et que cette projection entraîne nécessairement une perte d'information. Par conséquent, les distances, les angles et les formes ne sont pas toujours conservés.

La modélisation mathématique de la déformation projective évoquée ci-dessus est un problème difficile. En effet, la manière dont chaque région est déformée dépend de la profondeur inconnue de chacun de ses pixels. La transformation ne peut donc être modélisée par quelques paramètres. En revanche, pour des scènes contenant des surfaces planes, la transformation est une homographie qui peut elle même être approximée par une transformation affine.

Dans le cas qui nous intéresse, les petites régions locales autour de chaque point d'intérêt couvrent des surfaces *approximativement* planes, car elles sont de taille très petites par rapport à la distance au centre optique. Dans ces conditions, deux régions $R_1$ et $R_2$ sont reliées par une transformation affine :

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} . \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \tag{2.5}$$

où $(x_1, y_1)^T$ est un point de $R_1$ et $(x_2, y_2)^T$ sont correspondant dans $R_2$, et $\{a, b, c, d, e, f\}$ les six paramètres qui déterminent entièrement la transformation. Cette dernière est en fait une approximation de la réelle homographie qui relie les deux images, obtenue en négligeant les effets perspectifs.

Dans la pratique, les changements de point de vue étant limités, on utilise un modèle simplifié de la transformation affine qui s'écrit de la manière suivante :

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = s . \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix} . \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{2.6}$$

La transformation est composée d'une rotation d'angle $\theta$, d'une translation de vecteur $(t_x, t_y)^T$ et d'un changement d'échelle de facteur $s$.

### 2.2.3 Les détecteurs invariants aux transformations affines

Pour résoudre les difficultés évoquées dans la section précédente, le détecteur doit adapter la taille, la forme et l'orientation de chaque fenêtre pour que deux fenêtres correspon-

dantes recouvrent la même région de l'image. Le détecteur doit donc être invariant à un changement d'échelle, ainsi qu'à un changement de point de vue.

Concernant les premières tentatives pour résoudre le problème de l'invariance à l'échelle, on peut citer les travaux de Dufournaud *et al.* [31] et ceux de Hansen et Morse [51] dans lesquels les auteurs adoptent une approche multi-échelle pour estimer le facteur d'échelle entre deux images. Hansen et Morse proposent une méthode qui tient compte de la trace d'échelles construites par des filtres gaussiens calculés à différentes résolutions. Une trace d'échelles est un ensemble de valeurs calculées en un point sur des niveaux de résolution consécutifs. Dufournaud *et al.* calculent des points et des descripteurs à plusieurs niveaux d'échelle et un algorithme d'appariement robuste permet de sélectionner le facteur d'échelle correct entre deux images. Il est clair que ces deux méthodes ne sont pas des solutions satisfaisantes quant à la complexité et à la flexibilité de la mise en correspondance, car il faut apparier les points détectés à plusieurs échelles dans chaque image.

Une approche plus intéressante, consiste à sélectionner de manière automatique l'échelle de chaque point d'intérêt. Lindeberg [76] a proposé une méthode appelée *automatic scale detection* pour détecter l'échelle *caractéristique* de chaque point dans un espace échelle. Cette méthode est à la base de plusieurs détecteurs. De plus, les détecteurs sont rendus robustes à des transformations affines (voir la section 2.2.2). L'idée principale de ces détecteurs, consiste à calculer les points d'intérêt à plusieurs niveaux d'échelle et à sélectionner les points où une mesure locale (le Laplacien par exemple) est maximale dans la dimension d'échelle. Plus précisément, Lindeberg a montré que les extrema locaux des dérivées normalisées dans l'espace échelle, indiquent la présence de structures caractéristiques [76]. Nous présentons ci-dessous quelques unes des principales méthodes. Pour un état de l'art plus détaillé, nous renvoyons le lecteur intéressé aux références [93, 34].

Commençons tout d'abord par définir la notion d'espace échelle utilisée par quelques uns des détecteurs présentés. La notion d'espace échelle introduite sous sa forme continue par Witkin [158] et Koenderink [69] permet d'obtenir les dérivées en utilisant des arguments de géométrie différentielle. En particulier, il a été établi par Koenderink [69] et par Lindeberg [75] que le seul opérateur possible de l'espace échelle linéaire isotrope, sous des conditions raisonnables, est l'opérateur gaussien.

Une définition de l'espace échelle pour les signaux 1D est la suivante :

**Définition 1 (Espace échelle)** *Soit $f(x)$ une fonction et $G_\sigma(x)$ la gaussienne d'écart-type $\sigma$. On appelle espace échelle, le lieu des réalisations de la transformation $S$ définie*

*par :*

$$(Sf)(x,\sigma) = f * G_\sigma(x), (x,\sigma) \in \Re \times \Re^+ \tag{2.7}$$

*On appellera $S$ opérateur de changement d'échelle et l'on notera $E = (x,\sigma)$ l'espace échelle.*

Cette transformation conduit à la représentation d'une fonction sous la forme d'une surface décrite dans $E$. On peut donc étudier cette surface en utilisant des arguments de géométrie différentielle.

La représentation dans l'espace échelle d'une image $I$ est donc définie par une fonction $E(x,y,\sigma)$, obtenue par la convolution de $I$ avec une gaussienne $G(x,y,\sigma)$ d'écart-type variable :

$$E(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

où $*$ est l'opérateur de convolution en $x$ et $y$, et

$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x^2+y^2)/2\sigma^2}$$

**Le détecteur de Lowe : SIFT**

L'approche proposée par Lowe [79, 80] est aujourd'hui considérée comme l'une des plus performantes pour de nombreuses applications. Le détecteur calcule des points d'intérêt invariants à un changement d'échelle ainsi qu'un descripteur robuste. Nous reviendrons sur le calcul du descripteur à la section 2.3.1. Le détecteur est désigné par l'acronyme SIFT pour *Scale Invariant Feature Transform.*

L'approche consiste à détecter les points qui sont stables dans l'espace échelle. Pour ce faire, on utilise les extrema locaux de l'opérateur DoG (Difference of Gaussian) dans l'espace échelle. Etant donné deux échelles séparées par une constante multiplicative $s$, on a :

$$\begin{aligned} DoG(x,y,\sigma) &= (G(x,y,s\sigma) - G(x,y,\sigma)) * I(x,y) && (2.8)\\ &= E(x,y,s\sigma) - E(x,y,\sigma) && (2.9) \end{aligned}$$

L'opérateur DoG est rapide à calculer (simple soustraction d'images) et fournit une bonne approximation de l'opérateur Laplacien.

Dans la pratique, chaque octave de l'espace échelle (doublement de $\sigma$) est divisée en un nombre $s$ d'intervalles et on soustraie les images adjacentes pour obtenir les images

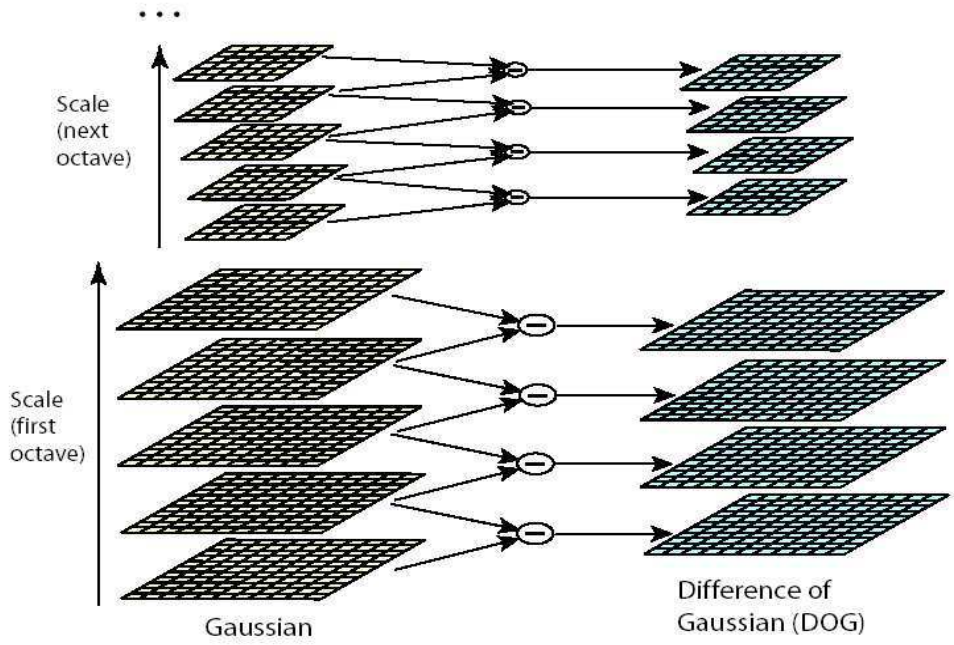



FIG. 2.3 – Construction de l'opérateur DoG dans l'espace échelle. Image reproduite d'après l'article de Lowe [80].

DoG (voir figure 2.3). Lowe montre que le nombre d'intervalles par octave qui donne les résultats les plus stables est $s = 3$. Ce nombre étant obtenu de manière empirique.

Pour détecter les points d'intérêt, chaque point d'une image $DoG(x, y, s\sigma)$ est comparé à ses 8 voisins de la même échelle $s\sigma$ ainsi qu'à ses 16 voisins des deux échelles immédiatement inférieure et supérieure. On détecte ainsi les minima et les maxima locaux.

Notons que le détecteur de Lowe n'est pas vraiment un détecteur invariant affine car il n'est invariant qu'au changement d'échelle. Néanmoins, le descripteur qui lui est associé est assez robuste et donne d'excellents résultats même dans le cas d'un changement de point de vue important.

### Les détecteurs basés sur le détecteur de Harris : Harris-Affine

Les premiers travaux visant à rendre le détecteur de Harris invariant aux transformations affines, sont ceux de Baumberg [8]. Le détecteur de Baumberg se base sur l'idée de sélection automatique d'échelle introduite par Lindeberg [76] pour détecter des points d'intérêt quand le changement de point de vue entre les deux images est important. Les mêmes idées ont été développées par Schaffalitzky et Zisserman [119], ainsi que par Mikolajczyk

et Schmid dans leurs nombreux travaux [89, 90, 91].

Dans la méthode proposée par Mikolajczyk et Schmid, on procède en trois étapes :

1. pour chaque point $\mathbf{x} = (x, y)^T$, on calcule un ensemble de réponses $F(\mathbf{x}, \sigma_n)$ à l'aide du détecteur de Harris (présenté à la section 2.2.1). On construit un espace échelle en prenant $\sigma_n = s_n\sigma$. On a donc :

$$F(\mathbf{x}, \sigma_n) = s_n^2 G(s_n\widetilde{\sigma}) \otimes \begin{bmatrix} I_x^2(\mathbf{x}, s_n\sigma) & I_x(\mathbf{x}, s_n\sigma)I_y(\mathbf{x}, s_n\sigma) \\ I_x(\mathbf{x}, s_n\sigma)I_y(\mathbf{x}, s_n\sigma) & I_y^2(\mathbf{x}, s_n\sigma) \end{bmatrix} \tag{2.10}$$

   où $\sigma$ est la fenêtre de dérivation et $\widetilde{\sigma}$ celle d'intégration.

2. on détermine l'*échelle caractéristique* de chaque point en utilisant l'opérateur Laplacien :

$$L(\mathbf{x}, \sigma_n) = |s_n(I_{xx}(\mathbf{x}, s_n\sigma) + I_{yy}(\mathbf{x}, s_n\sigma))| \tag{2.11}$$

   L'échelle caractéristique du point $\mathbf{x}$ est égale à $\sigma^* = max\{L(\mathbf{x}, \sigma_n)\}$.

3. on adapte la forme du voisinage de chaque point à l'aide d'un processus itératif basé sur la matrice des moment d'ordre 2 calculée à l'échelle caractéristique.

Ce détecteur sera désigné dans ce document par le nom de détecteur Harris-Affine. Voir [89, 90] pour plus de détails.

Dans les deux approches ci-dessus, on détecte dans un premier temps des points d'intérêt, puis on calcule une zone de l'image autour de chaque point. Il existe plusieurs autres approches permettant d'obtenir directement des régions invariantes aux transformations affines. Nous en présentons les trois principales ci-après.

**Les détecteurs de Tuytelaars : EBR et IBR**

Tuytelaars et Van Gool [152, 153] proposent deux détecteurs de régions invariantes à des transformations affines respectivement notés EBR (Edge Based Regions) et IBR (Intensity Based Regions). La première méthode est basée sur la détection des coins de Harris et sur la détection des contours de l'image. Les points d'intérêt qui sont à l'intersection d'au moins deux contours sont sélectionnés comme points d'ancrage. On construit ensuite, à partir de ces points d'ancrage, des parallélogrammes et on sélectionne ceux pour lesquelles une certaine fonction de la texture atteint un extremum. Voir [152] pour plus de détails. Cette méthode de détection est très peu stable car elle est basée sur les points de Harris et les contours dont la détection n'est pas stable.

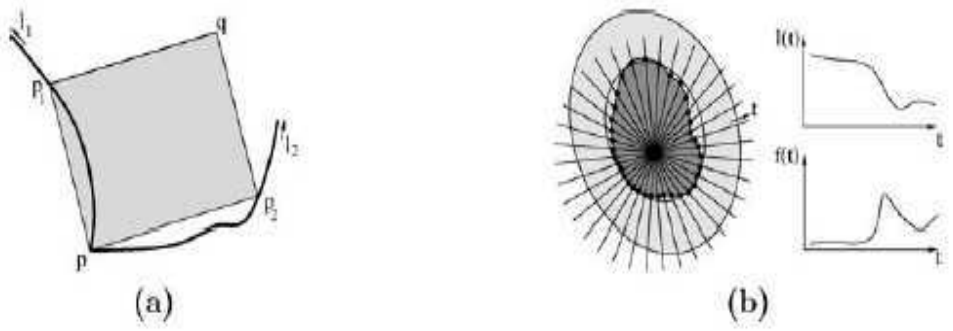



FIG. 2.4 – Principes des détecteurs EBR et IBR. a) EBR exploite les contours de l'image ; b) IBR exploîte l'information photométrique. Image reproduite d'après l'article de Tuytelaars et Van Gool [153].

La seconde méthode, IBR, exploîte l'information photométrique de l'image pour la détection des régions. Elle commence par détecter les extrema locaux de l'intensité lumineuse, puis explore la région autour de chaque extremum. Plus précisément, étant donné un extremum local **p** d'intensité $I_0$, on étudie une fonction de l'intensité le long des rayons issus de **p**. Pour chaque rayon, la fonction suivante est évaluée :

$$f_{I(t)} = \frac{abs(I(t) - I_0)}{max\left(\frac{\int_0^t abs(I(t)-I_0)}{t}, d\right)} \tag{2.12}$$

où $t$ est un paramètre quelconque le long du rayon, $I(t)$ est l'intensité à la position $t$, et $d$ est un nombre mis pour éviter une division par zéro.

La fonction $f(t)$ atteint un maximum lorsque l'intensité lumineuse le long du rayon change de signe. Voir figure 2.4(b). On sélectionne donc tous les points en lesquels cette fonction atteint un maximum [151]. Tous les points ainsi définis sont reliés et forment la région invariante qui est ensuite approximée par une ellipse.

EBR et IBR produisent des régions qui sont invariantes à des transformations affines, mais qui sont basées sur des points de Harris ou des extrema d'intensité, sensibles à un changement d'échelle. Pour assurer une certaine robustesse au changement d'échelle, les points de départ sont calculés à plusieurs échelles. La méthode de détection IBR est plus stable que EBR.

### Le détecteur de Matas : MSER

Matas *et al.* [86] proposent un détecteur qui exploîte également l'intensité lumineuse. Une MSER (Maximally Stable Extremal Region) est un ensemble connexe de pixels qui possèdent la propriété d'être tous plus clairs ou tous plus foncés que les pixels du bord de la région (d'où le terme *Extremal*). La méthode est donc basée sur le seuillage de l'image et le terme *maximally stable* dans MSER fait référence au fait que les régions détectées sont celles qui sont stables dans un large intervalle de seuils. Un algorithme rapide de recherche considère tous les seuils possibles de l'image (de 0 à 255 dans le cas d'une image en niveaux de gris) et retient les régions stables. Notons que le seuillage est global.

Les régions ainsi obtenues sont remplacées par des ellipses de mêmes moments d'ordre 1 et 2 pour faciliter les étapes ultérieures. Notons enfin que l'algorithme de détection est extrêmement rapide (voir page 25 pour une comparaison des temps d'exécution des différentes méthodes).

### Le détecteur de Kadir : SRD

Ce détecteur proposé par Kadir *et al.* [64] est basée sur l'entropie de la distribution des intensités dans une région elliptique de l'image et sur la notion de *saillance*. Plus précisément, soit $\mathbf{x}$ un pixel de l'image et $\mathcal{E}$ une région elliptique centré en $\mathbf{x}$. $\mathcal{E}$ est définie par sa taille $s$ (la longueur du grand axe), son orientation $\theta$ (celle du grand axe) et la rapport entre le grand et le petit axe $\lambda$. On calcule la densité de probabilité $p(I)$ des intensités dans $\mathcal{E}$ et l'entropie $\mathcal{H}$ est définie par :

$$\mathcal{H} = -\sum_{I} p(I) log p(I) \tag{2.13}$$

Une fois l'entropie calculée pour chaque région, on calcule les extrema locaux de $\mathcal{H}$ par rapport à la taille $s$ de l'ellipse. Pour chaque extremum, on calcule la dérivée de la densité de probabilité $p(I; s, \theta, \lambda)$ par rapport à $s$ de la manière suivante :

$$\mathcal{W} = \frac{s^2}{2s-1} \sum_{I} \left| \frac{\partial p(I; s, \theta, \lambda)}{\partial s} \right|$$

Enfin, la mesure de saillance $\mathcal{Y}$ de la région est défine par $\mathcal{Y} = \mathcal{H}\mathcal{W}$. Les régions sont triées selon leur saillance et on retient celles dont la saillance est supérieure à un seuil donné. Notons qu'en maximisant $\mathcal{Y}$, on recherche les régions qui ont une entropie maximale et qui

| détecteur | temps d'éxécution (min :sec) | nombre de régions |
|---|---|---|
| SIFT | 0 :01.87 | 3079 |
| Harris-Affine | 0 :02.83 | 2027 |
| MSER | 0 :00.31 | 533 |
| IBR | 0 :07.22 | 679 |
| EBR | 2 :33.78 | 1265 |
| SRD | 31 :29.82 | 513 |

TAB. 2.1 – Comparaison des différents détecteurs en utilisant l'image gauche de la figure 2.2.

possèdent des contours aux bords du grand axe de l'ellipse.

Ce détecteur sera noté SRD pour *Salient Region Detector*.

### Remarques sur les différents détecteurs

Il existe différents détecteurs de régions invariantes aux changements d'échelle et/ou aux transformations affines. Bien qu'ils soient basés sur des méthodes d'extractions différentes et trouvent des régions différentes, tous les détecteurs obéissent au même besoin d'invariance évoqué à la section 2.2.2. Ils partagent donc la même idée générale qui est l'adaptation de la taille, de la forme et de l'orientation de chaque région afin de pouvoir identifier des structures identiques dans des images différentes.

En fonction de l'image et de la scène qu'elle représente, un détecteur donnera plus ou moins de régions. Par exemple, MSER et EBR sont mieux adaptés à des scènes structurées alors que les détecteurs Harris-Affine et SIFT répondent mieux à des scènes texturées. D'autre part, les temps d'exécution des différents algorithmes sont assez variés comme le montre le tableau 2.1. Pour obtenir les données rassemblées dans ce tableau, nous avons utilisé les exécutables fournis par les auteurs des différentes méthodes[2]. Les temps d'exécution sont donnés pour un processeur 3 Ghz tournant sous Linux. Il est également important de souligner que certains détecteurs trouvent plus d'un point d'intérêt à une même position $(x, y)$ de l'image. C'est le cas de SIFT et Harris-Affine, d'où le grand nombre de régions fourni par ces deux détecteurs.

Une meilleure évaluation de la performance des détecteurs est donnée par le critère de répétabilité et la précision de la détection. La répétabilité, indique le nombre moyen de points (ou régions) correspondants simultanément détectés dans deux images [124, 93]. La

[2]L'ensemble des détecteurs est présenté sur le site internet à l'adresse suivante : `http://www.robots.ox.ac.uk/~vgg/research/affine/`

précision, renvoie à la localisation des points (ou centres des régions). Plus la répétabilité du détecteur entre deux images est grande, plus on peut, potentiellement, trouver de correspondants entre les deux images. Un moyen de calculer la répétabilité d'un détecteur est présenté par Mikolajczyk *et al* [93]. On commence par définir une *mesure de recouvrement* $\epsilon_s$ (*overlap error*) comme étant le rapport entre l'intersection et l'union de deux régions correspondantes. Soit $A$ et $B$ deux régions détectées respectivement dans $I_1$ et $I_2$. Alors, l'erreur de *recouvrement* est définie par :

$$\epsilon_s = 1 - \frac{A \cap (H^T B H)}{A \cup (H^T B H)} \tag{2.14}$$

où $H$ est l'homographie reliant $I_1$ et $I_2$.

Les deux régions $A$ et $B$ se correspondent si $\epsilon_s$ est suffisament faible : $\epsilon_s < \epsilon_0$. La répétabilité du détecteur pour la paire d'images $(I_1, I_2)$, est définie par le rapport entre le nombre de régions correspondantes et le plus petit nombre de régions détectées dans les images. Dans leur étude comparative, Mikolajczyk *et al* [93] montrent qu'aucun détecteur ne surpasse les autres dans toutes les situations (ce que nous avons souligné plus haut en notant que le nombre de régions détectées dépend du détecteur et du type de scène). Ils notent toutefois que MSER obtient la plus grande répétabilté dans de nombreux cas, suivi par Harris-Affine. Notons que le détecteur de Lowe, SIFT, n'est pas inclut dans cette étude.

## 2.3 Description des invariants locaux

Une fois les régions détectées, la seconde étape consiste à calculer un descripteur qui sera utilisé dans la phase d'appariement.

Le descripteur doit être robuste aux principales transformations évoquées à la section 2.1 (voir page 14), i.e. il doit tolérer de petites déformations de l'image, des changement d'illumination de la scène, ainsi que diverses autres sources de "bruit" telle que la compression. Il existe différentes méthodes pour décrire une région de l'image et chaque descripteur caractérise différentes propriétés de l'image telles que la couleur, la texture, les contours, etc. La méthode la plus simple consiste à stocker dans un vecteur les niveaux de gris des pixels de la région. Ce vecteur est alors le descripteur de la région. La méthode peut être appliquée aux dérivées de l'image (gradient ou laplacien), mais elle n'est pas invariante aux transformations euclidiennes (translation, rotation) et au changement d'échelle.

### 2.3.1 Les méthodes du type "*Shape Context*"

Ils existent différentes méthodes représentant les propriétés d'une région sous la forme d'histogramme, par exemple l'histogramme des intensités lumineuses. On peut toutefois, utiliser des informations plus riches et plus discriminantes que la simple intensité lumineuse. Nous les appelerons méthodes du type "*Shape Context*" car elles caractérisent chaque point par une distribution de l'apparance de son voisinage.

Johson et Hebert [61] caractérisent le voisinage de chaque point d'intérêt par un descripteur appelé *spin image*. Ce descripteur est adapté à la caractérisation des invariants locaux par Lazebnik [74]. Il est représenté par un histogramme bi-dimensionnel de la distribution des intensités lumineuses dans la région. Les deux dimensions étant la distance $d$ par rapport au centre de la région et l'intensité $i$ du pixel considéré. En divisant l'intervalle des valeurs de $d$ en 10 parties et celui des valeurs de $i$ en 10 parties, on obtient un vecteur d'invariants de taille 100. La contribution d'un pixel $\mathbf{x}$ à l'index $(d, i)$ de l'histogramme est donnée par :

$$exp(-\frac{(|\mathbf{x}-\mathbf{x}_0|-d)^2}{2\alpha^2}-\frac{|I(\mathbf{x})-i|^2}{2\beta^2})$$

où $\mathbf{x}_0$ est le centre de la région, et $\alpha$ et $\beta$ deux paramètres fixés [74].

Pour assurer l'invariance à un changement de luminosité (transformation affine des intensités lumineuses : $(I \rightarrow aI + b)$, il suffit d'effectuer une normalisation locale dans la région.

Belongie *et al.* [11] introduisent un descripteur noté *shape context* qui est représenté par un histogramme 2D des positions des points de contour dans la région. Les deux dimensions sont dans ce cas la distance $d$ par rapport au centre de la région et la position $\theta$ du pixel considéré. En divisant la région en 12 secteurs angulaires et l'intervalle des valeurs de $d$ en 5 parties, on obtient un vecteur de dimension égale à 60.

Lowe [80] propose, avec son détecteur présenté dans la section précédente (voir page 20), un descripteur basé sur la distribution des orientations et positions du gradient dans la région. Le descripteur, *SIFT* (Scale Invariant Feature Transform), est obtenu en divisant la région en 4x4 parties, et en divisant chaque partie en 8 secteurs angulaires. On obtient donc un vecteur d'invariants de taille 4x4x8 = 128. Notons que la contribution de chaque point de la région à l'histogramme est pondérée par la norme du gradient en ce point. La figure 2.5 montre le principe du calcul de ce descripteur.

Ces trois descripteurs sont basés sur la même idée et sont très similaires. L'invariance

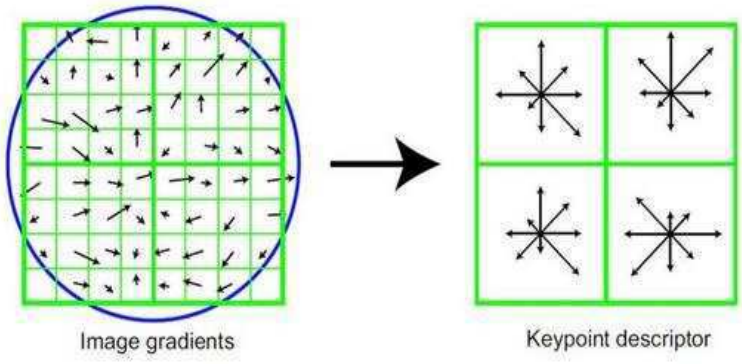



Fig. 2.5 – Principe du descripteur *SIFT*. Image reproduite d'après l'article de Lowe [80].

aux transformations affines est assurée d'une part, par le fait que chaque point est détecté à une échelle caractéristique qui définit la taille de la fenêtre de calcul du descripteur. D'autre part, par le fait que chaque région possède une orientation spécifique.

### 2.3.2 Les invariants différentiels

Une méthode, plus ancienne, de caractérisation consiste à décrire la géométrie d'une région par l'ensemble des dérivées de l'image. Cet ensemble de dérivées est appelé *jet local* [70]. Les dérivées sont calculées de manière stable en utilisant un filtre gaussien. Cette caractérisation est utilisée dans le cadre de l'indexation d'images par Schmid [123], qui calcule les invariants jusqu'à l'ordre 3 pour obtenir une caractérisation suffisamment riche (9 invariants) et invariante à la rotation.

Les invariants différentiels ont été étendus au cas des images couleurs par Gouet *et al* [46, 95, 45]. Les auteurs montrent, en particulier, que la prise en compte de la couleur fournit assez d'information pour limiter le calcul des dérivées à l'ordre 1 (8 invariants), rendant ainsi la caractérisation plus robuste au bruit.

Dans [39], Freeman et Adelson proposent d'utiliser des filtres directionnels, i.e. des filtres définis par des dérivées calculées dans n'importe quelle direction $\theta$. Plus récemment, Baumberg [8] et Schaffalitzky et Zisserman [120] proposent d'utiliser des filtres de variables complexes $K(x, y, \theta) = f(x, y)exp(i\theta)$, où $\theta$ désigne l'orientation du filtre. Pour la fonction $f(x, y)$, Baumberg utilise une gaussienne tandis que Schaffalitzky et Zisserman adoptent une fonction polynomiale.

### 2.3.3 Les invariants fréquentiels

La description d'un signal (une image) par son contenu fréquentiel est une vieille notion en traitement du signal (d'image). On peut notament citer les transformées de Fourier et de Mellin qui décomposent le signal sur une base de fonctions élémentaires. Ces deux représentations ne peuvent cependant pas être utilisées pour décrire des régions de l'image car elles ignorent les relations spatiales entre les points. La transformée de Gabor [40] et la transformée en ondelettes [83] sont plus adaptées à une description locale. Elles nécessitent néanmoins, pour être invariantes à la rotation, l'emploi d'un nombre élevé de filtres [160].

### 2.3.4 Les moments

Mindru *et al.* [94] introduisent la notion de moments généralisés pour caractériser la forme et la distribution d'intensité dans une région $\Omega$. Le moment d'ordre $p+q$ et de degré $n$ est défini par :

$$M_{pq}^{n} = \int_{\Omega} [I(x,y)]^{n} x^{p} y^{q} dxdy \tag{2.15}$$

Ces moments sont indépendants et faciles à calculer jusqu'à un ordre quelconque. Une caractérisation invariante est obtenue en combinant des moments de différents ordres et degrés. Tuytelaars et Van Gool [152, 151] utilisent ces moments pour la détection et la description des régions par les dédecteurs IBR et EBR.

### 2.3.5 De la performance de SIFT

Les descripteurs du type "*Shape Context*" sont aujourd'hui considérés comme étant les plus performants dans diverses applications. Dans une étude comparative de plusieurs détecteurs, Mikolajczyk et Schmid ont montré que SIFT donne de meilleurs résultats respectivement par rapport à Shape Context, aux moments généralisés et aux invariants différentiels [92].

La bonne performance de SIFT par rapport à Shape Context peut s'expliquer par le fait que le premier descripteur capture plus d'information que le second. En effet, SIFT prend en compte non seulement la position des points de la région, mais également l'orientation et la norme du gradient en ces points, alors que Shape Context ne considère que la position des points de contour. D'ailleurs, en incorporant dans le descripteur Shape Context une information de gradient comme dans SIFT, Mori *et al.* [98] obtiennent de meilleurs résultats. Leur approche est appelée *Generalized Shape Context*.

Il existe de nombreux autres descripteurs basés sur la même idée de distribution. Parmi les travaux représentatifs, on peut citer PCA-SIFT de Ke et Sukthankar [66]. Ce descripteur représente l'apparence locale par les composantes principales du champ de gradient normalisé. La seule idée similaire avec SIFT est l'emploi du gradient, bien que le nom PCA-SIFT puisse laisser penser à une plus grande proximité. On peut également noter des améliorations apportées à SIFT en vue de le rendre totalement invariant à la rotation par Lazebnik [74] et Mikolajczyk [92], ou en vue de le rendre plus rapide par Bay *et al.* qui introduisent un nouveau détecteur et descripteur noté SURF (Speeded Up Robust Features) [9].

## 2.4 Mise en correspondance des invariants locaux

Une fois les primitives caractérisées par des vecteurs d'invariants, le problème de l'appariement se ramène à la comparaison des ensembles d'invariants. En général, on établit les correspondances entre régions par une méthode du type *plus proche voisin* (PPV), i.e. une région de la première image est appariée avec la région de la seconde image qui est la plus proche pour une mesure de similarité donnée. Il faut donc trouver une bonne mesure de similarité entre les descripteurs.

### 2.4.1 Calcul des scores de similarité

Les méthodes de comparaisons les plus utilisées sont basées sur la corrélation et sur le calcul de distances entre vecteurs.

**La corrélation**

Il est possible de calculer un score de corrélation entre deux vecteurs à comparer et il existe plusieurs formules de corrélation. Les plus utilisées étant la NCC (Normalized Cross Correlation) et la ZNCC (Zero mean Normalized Cross Correlation).

Si $u_i$ désigne le vecteur d'invariant d'une région de l'image $I_i$ et $\overline{u_i}$ la valeur moyenne des composantes du vecteur $u_i$, alors les scores de corrélation entre deux vecteurs $u_1$ et $u_2$ calculés avec les formules NCC et ZNCC s'écrivent respectivement selon les équations 2.16 et 2.17 ci-dessous :

$$NCC(u_1, u_2) = \frac{u_1.u_2}{\|u_1\|.\|u_2\|} \tag{2.16}$$

$$ZNCC(u_1, u_2) = \frac{(u_1 - \overline{u_1}).(u_2 - \overline{u_2})}{\|u_1 - \overline{u_1}\|.\|u_2 - \overline{u_2}\|} \tag{2.17}$$

**La distance de Mahalanobis**

Le calcul de la distance entre deux vecteurs est un problème délicat et important en statistique lorsque chaque dimension du vecteur s'exprime dans une unité particulière. On utilise la formulation générale suivante [118] : la distance entre deux vecteurs $u_1$ et $u_2$ est définie par la forme quadratique :

$$d^2(u_1, u_2) = (u_1 - u_2)^T \mathbf{M}(u_1 - u_2)$$

où $\mathbf{M}$ est une matrice symétrique définie positive.

Le choix de la matrice $\mathbf{M}$ définit la distance :

- si $\mathbf{M} = \mathbf{I}$, alors $d$ est la distance euclidienne usuelle. Elle conduit à privilégier les variables les plus dispersées et à négliger les différences entre les variables ;
- la matrice la plus utilisée est la matrice diagonale des inverses des variances :

$$\mathbf{M} = \mathbf{D}_{1/s^2} = \begin{bmatrix} 1/\sigma_1^2 & 0 & \cdots & 0 \\ 0 & 1/\sigma_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/\sigma_p^2 \end{bmatrix}$$

  Ce qui revient à diviser chaque variable par son écart-type. On a alors des variables de même importance, quelque soit leur dispersion ;
- l'utilisation de la matrice de covariance $\Sigma$ définit la distance de Mahalanobis.

$$d^2(u_1, u_2) = (u_1 - u_2)^T \Sigma^{-1}(u_1 - u_2) \tag{2.18}$$

**Les distances statistiques**

Dans les cas où le descripteur est représenté par une distribution (sous forme d'histogramme par exemple), on peut utiliser une distance statistique pour évaluer la similarité de deux distributions. Une mesure de similarité naturelle est le test du $\chi^2$ :

$$d(u_1, u_2) = \frac{1}{2} \sum_{k=1}^{K} \frac{[u_1(k) - u_2(k)]^2}{u_1(k) + u_2(k)} \tag{2.19}$$

où $K$ est la dimension du descripteur.

Une autre mesure de similarité très utilisée est la distance de Bhattacharyya définie par :

$$d(u_1, u_2) = \sqrt{1 - \rho(u_1, u_2)} \tag{2.20}$$

où $\rho(u_1, u_2)$ est le coefficient de Bhattacharyya entre les deux distributions $u_1$ et $u_2$ :

$$\rho(u_1, u_2) = \sum_{i=1}^{K} \sqrt{u_1(i)u_2(i)}$$

Dans la pratique, les distances statistiques sont peu adaptées si les vecteurs d'invariants sont de dimension faible. Quant à la distance de Mahalanobis, la principale difficulté de sa mise en œuvre réside dans l'estimation de la matrice de covariance qui est inconnue. L'utilisation de la matrice diagonale des inverses des variances permet de calculer une distance euclidienne entre les vecteurs centrés-réduits. On peut aussi normaliser chaque composante du vecteur d'invariants dans un intervalle fixé, puis calculer des distances euclidiennes entre les vecteurs normalisés. Ce qui est équivalent à la réduction des variables. C'est cette approche que nous adoptons dans nos travaux.

### 2.4.2 Elimination des faux appariements

Les invariants ne sont pas totalement discriminants et plusieurs points peuvent avoir une caractérisation similaire. Le calcul des scores d'appariement fournit donc un ensemble de couples de régions non cohérent (une région peut avoir plusieurs correspondants dans l'autre image). Il faut donc une étape supplémentaire qui élimine les faux appariements pour respecter la *contrainte d'unicité* (une région doit avoir un correspondant unique). Plusieurs approches sont possibles :

#### Vérification croisée

La méthode de mise en correspondance par vérification croisée (ou appariement croisé) fournit un ensemble de couples de régions symétriques. On commence par mettre en correspondance les régions de $I_1$ avec celles de $I_2$ par la méthode du plus proche voisin. Puis on échange les rôles des images $I_1$ et $I_2$. Les couples de correspondants finalement retenus sont ceux composés de régions qui ont été mutuellement sélectionnées comme le montre la figure 2.6.

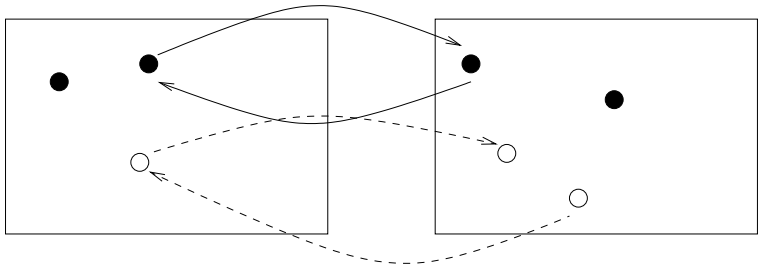


FIG. 2.6 – *Illustration de la mise en correspondance par vérification croisée : en trait plein, les correspondants corrects ; en pointillés, les correspondants incorrects.*

Cette méthode est plus satisfaisante que la méthode du plus proche voisin, mais en présence de bruit, il demeure des ambiguïtés : une région peut avoir un score de similarité élevé avec plusieurs régions dans l'autre image et il est alors difficile d'identifier le correspondant correct qui peut être rejeté à tort.

### Plus proche voisin avec rapport de distances

On peut réduire le nombre de faux appariements en prenant en compte une mesure d'ambiguïté, c'est-à-dire en éliminant les régions qui ont un score de similarité élevé avec plusieurs régions différentes. C'est l'approche utilisé par Zhang *et al* [164], ainsi que par Lowe [79]. Soit $r_i$ une région de $I_1$ et soient $r_{j1}$ et $r_{j2}$ deux régions de $I_2$. Supposons que $r_{j1}$ soit le plus proche voisin de $r_i$ et $r_{j2}$ son second plus proche voisin dans l'image $I_2$. On note $d_{ij1}$ le score de similarité entre $r_i$ et $r_{j1}$, et $d_{ij2}$ le score de similarité entre $r_i$ et $r_{j2}$. Alors, on définit l'ambiguïté de l'appariement de $r_i$ avec $r_{j1}$ par :

$$\mathcal{A} = \frac{d_{ij1}}{d_{ij2}}$$

Les régions $r_i$ et $r_{j1}$ correspondent si l'ambiguïté est inférieure à un certain seuil. Lowe prend comme seuil 0.6.

Cette méthode, que nous appelons plus proche voisin avec rapport de distances (PPVRD), associe une région à son plus proche voisin, si ce dernier est beaucoup plus proche que le second plus proche voisin. Dans la pratique, elle permet de réduire le nombre de faux appariement en éliminant les apprariements les plus ambigus, mais elle réduit également le nombre d'appariements corrects entre deux images.

### Estimation de la transformation géométrique

Une approche couramment utilisée pour l'élimination de faux appariements est l'estimation de la transformation géométrique entre les deux images lorsque cela est possible. En particulier, dans le cas d'images représentant des scènes planes, on peut estimer l'homographie entre les deux images par des méthodes robustes telles que RANSAC (RANdom SAmple Consensus) [38].

Une fois la transformation connue, il est facile de vérifier l'exactitude d'un couple de correspondants $(r_i, r_j)$. En effet, si $\mathcal{H}$ désigne l'homographie entre $I_1$ et $I_2$, alors $r_i$ est correctement apparié avec $r_j$ si :

$$\|r_j - \mathcal{H}r_i\| < \varepsilon$$

où $\varepsilon$ est l'erreur de localisation (généralement entre 1 et 5 pixels).

On peut aussi utiliser l'*erreur de recouvrement* introduite par Mikolajczyk *et al* [93], voir équation 2.14, pour sélectionner les correspondants corrects connaissant l'homographie entre les images.

$\mathcal{H}$ étant estimée à partir de l'ensemble initial de correspondants, il est important que celui-ci comporte peu de correspondants incorrects. L'expérience montre que RANSAC échoue lorsque la proportion de faux appariements est supérieure à 50%, voir [80, 23, 34].

### Cas particulier de la stéréo

Dans le cas d'un couple d'images stéréoscopiques, la contrainte épipolaire permet de ramener le problème bidimensionnel de la mise en correspondance à un problème unidimensionnel.

En effet, la géométrie épipolaire est caractérisée par une matrice, dite matrice fondamentale, $F_{3\times3}$ qui vérifie :

$$m_2^T F m_1 = 0 \tag{2.21}$$

pour tout couple de points appariés $(m_1, m_2)$.

Cette équation traduit le fait que le point $m_2$ dans la seconde image est situé sur la droite épipolaire $Fm_1$ et réciproquement, que le point $m_1$ de la première image est situé sur la droite épipolaire $F^T m_2$.

Ainsi, la connaissance de $F$ réduit la complexité de la méthode d'appariement, puisque la zone de recherche du correspondant d'un point devient une droite. Dans le cas général cependant, la matrice fondamentale n'est pas connue et elle doit être estimée. Des méthodes robustes d'estimation de la matrice fondamentale existent, par exemple celle développée

par Zhang *et al* dans [164]. Cependant, pour estimer $F$, il faut avoir un ensemble initial de correspondants corrects.

## 2.5 Conclusion

Dans ce chapitre, nous avons présenté le problème de la mise en correspondance d'images par l'utilisation d'invariants locaux en présentant les principaux avantages dus à ce type de primitives locales. Nous avons passé en revue les principales méthodes de détection et de caractérisation des invariants locaux proposées dans la littérature. Il ressort de cet état de l'art, que les méthodes de caractérisation basées sur une distribution des propriétés (couleur, texture, contours, etc) des régions détectées donnent les meilleurs résultats.

Néanmoins, les différentes méthodes de mise en correspondance utilisées conduisent à de nombreux faux appariements qu'il faut ensuite éliminer. Une solution pour éviter cette étape, ou au moins faciliter cette étape, peut être l'enrichissement de la description des régions, ou la mise en œuvre d'une méthode de mise en correspondance robuste.

Dans le chapitre suivant, nous évoquons les différentes approches visant à utiliser des informations complémentaires pour améliorer les résultats de la mise en correspondance.

# Chapitre 3

# La prise en compte de l'information contextuelle

Ce chapitre présente les principales difficultés liées à l'utilisation des invariants locaux pour la mise en correspondance d'images et la nécessité de prendre en compte une information contextuelle pour résoudre les ambiguïtés. Nous présentons également les principales méthodes proposées dans la littérature pour améliorer les résultats, nous les comparons entre elles et tirons des conclusions quant à la manière d'utiliser cette information contextuelle.

## 3.1 Limites des invariants locaux

Comme nous l'avons vu au chapitre précédent, les invariants locaux sont largement utilisés pour la mise en correspondance d'images. Le principal intérêt de leur utilisation, réside dans le caractère local qui les rend robustes aux occultations et aux changements de fond, et invariants aux transformations géométriques (principalement affines) et photométriques. Cependant, en dépit des excellents résultats obtenus dans de nombreuses applications, l'utilisation des invariants locaux pour la mise en correspondance présente de nombreuses difficultés. La principale difficulté concerne le pouvoir de discrimination de la caractérisation locale. En effet, comme souligné dans la section 2.4.2, la simple comparaison des invariants locaux conduit souvent à de nombreux faux appariements qu'il faut ensuite éliminer par des méthodes coûteuses, par exemple l'estimation de la transformation géométrique entre les images lorsque cela est possible. Mais même dans ce cas, il faut disposer au départ d'un nombre suffisant de correspondants corrects car, comme nous l'avons

Fig. 3.1 – Un cas difficile de mise en correspondance. La présence de structures répétitives rend impossible la mise en correspondance par plus proche voisin.

souligné à la page 34, une méthode telle que RANSAC échoue lorsque la proportion de faux appariements est supérieure à 50% [80, 23, 34].

### 3.1.1 Cas de structures répétitives

La difficulté devient plus importante lorsque les deux images à apparier présentent des structures répétitives. Dans ce cas, toutes les régions d'intérêt sont décrites presque de la même manière par un descripteur local et il est difficile, voire impossible, de trouver les correspondants corrects à cause de l'ambiguïté élevée. Considérons par exemple le couple d'image de la figure 3.1. Il y a respectivement 318 et 304 points d'intérêt détectés dans chacune des deux images en utilisant le détecteur Harris-Affine (voir page 21). La mise en correspondance par la méthode du plus proche voisin (PPV), en utilisant SIFT comme descripteur, donne 117 appariements dont seuls 39 sont corrects. Autrement dit, le taux de faux appariements atteint dans ce cas plus de 66% ! Même une méthode robuste d'estimation de la transformation géométrique entre les deux images telle que RANSAC, échouera dans ce cas.

On peut chercher à réduire le nombre de faux appariements en utilisant la méthode du plus proche voisin avec rapport de distances (PPVRD), voir page 33. Avec cette méthode, on a 6 appariements dont 3 corrects. Donc, PPVDR réduit effectivement le nombre de faux appariements en éliminant les appariements les plus ambigus, mais elle réduit également, et de manière considérable dans ce cas, le nombre d'appariements corrects entre les deux images. En effet, PPVRD élimine tous les points qui ont un score de similarité élevé avec


a) b)

FIG. 3.2 – Un cas difficile de mise en correspondance. La faible répétabilité du détecteur de points d'intérêt rend difficile la mise en correspondance.

plusieurs points différents. Ce qui a pour conséquence d'éliminer de nombreux appariements corrects. On a alors trop peu d'appariements pour mettre en œuvre une méthode d'estimation de la transformation géométrique entre les deux images.

Cet exemple, met en évidence un fait important : *la localité du descripteur limite son pouvoir discriminant.*

Ce qui fait apparaître une sorte de contradiction. D'une part, la localité du descripteur le rend robuste et invariant, ses principaux avantages. D'autre part, cette même localité limite son pouvoir discriminant et il est impossible de distinguer des structures localement similaires.

### 3.1.2 Cas de la reconnaissance d'objets

Considérons le cas de la reconnaissance d'objets où l'on cherche à détecter et à localiser un objet dans une scène complexe. La figure 3.2(a) montre un objet que l'on souhaite détecter dans la scène de la figure 3.2(b). En utilisant le détecteur Harris-Affine, il y a respectivement 313 points d'intérêt détectés sur l'objet, et 750 points d'intérêt détectés sur l'image de la scène.

Cependant, à cause du changement d'échelle entre les deux vues, du changement important de point de vue et des occultations, très peu de points d'intérêt détectés dans la scène appartiennent à l'objet. En effet, il y a moins de 100 points d'intérêt détectés sur l'objet dans l'image de la figure 3.2(b).

La mise en correspondance par la méthode PPV ne permet de trouver aucun corres-

pondant correct, de même que la méthode PPVRD.

Cet autre exemple met en évidence le fait que la répétabilité du détecteur décroît lorsque le changement de point de vue entre les deux images devient important, i.e. qu'il y a peu de points simultanément détectés dans les mêmes zones des images à des échelles différentes. Une méthode de mise en correspondance efficace doit donc être capable d'apparier l'image de l'objet avec une partie, relativement petite, de l'image représentant la scène.

Il est évident que la prise en compte d'informations ou de contraintes supplémentaires peut permettre d'éliminer certaines ambiguïtés. La question est de savoir quelle information utiliser et comment l'utiliser ?

Les informations supplémentaires peuvent être prises en compte de deux manières possibles. D'une part, nous avons les méthodes qui tentent d'augmenter le pouvoir discriminant des descripteurs locaux en utilisant des informations de couleur, de texture ou de courbure [99, 155]. D'autre part, il y a les méthodes qui tentent, dans la phase d'appariement, d'éliminer les appariements incorrects en utilisant une information plus globale telle que les relations entre les primitives voisines [29, 132].

Rappellons que la mise en correspondance avec des primitives locales comporte trois phases : la détection, la description et l'appariement des primitives. Les méthodes de la première catégorie utilisent une information supplémentaire, que nous appelerons information contextuelle, dans la phase de description, tandis que celles de la seconde catégorie utilisent l'information contextuelle dans la phase de mise en correspondance. Dans la suite de ce chapitre, nous présentons les différentes méthodes proposées dans la littérature, leurs limitations, et nous verrons quelle est la meilleure façon d'utiliser l'information contextuelle disponible.

## 3.2 Prise en compte du contexte dans la phase de description

Il existe de nombreux travaux visant à enrichir la description locale, notamment SIFT, par la prise en compte de divers types d'information. Nous décrivons brièvement ici deux approches récentes. La première utilise la couleur, la seconde utilise une information de courbure.

### 3.2.1 SIFT+Color

Une idée, presque naturelle, lorsque l'on traite des images en couleur, est d'ajouter à la caractérisation géométrique donnée par SIFT, une caractérisation basée sur la couleur. C'est ce qui est fait par Van de Weijer et Schmid [155] et par Abdel-Hakim et Farag [2].

Van de Weijer et Schmid concatennent les descripteurs géométrique et photométrique :

$$K = (\widehat{S}, \lambda\widehat{C}) \tag{3.1}$$

où $S$ désigne le descripteur SIFT, $C$ le descripteur couleur, $\lambda$ un terme de pondération, et $\widehat{A}$ indique que le vecteur $A$ est normalisé.

Dans leur article [155], les auteurs essaient différents descripteurs couleur robustes aux différents changements géométriques et photométriques. Ils montrent que les résultats obtenus, en terme de gain par rapport à SIFT seul, dépendent de l'application. Pour un problème de classification ou d'indexation d'images, la combinaison de SIFT avec la couleur donne des performances dépassant largement celles de SIFT. En revanche, pour un problème de mise en correspondance, le gain obtenu est très faible. D'une manière générale, les auteurs recommandent l'emploi de la teinte :

$$t = arctan\left(\frac{O1}{O2}\right)$$

où

$$O1 = \frac{1}{\sqrt{2}}(R - G) \text{ et } O2 = \frac{1}{\sqrt{6}}(R + G - 2B)$$

Le descripteur couleur $C$ est obtenu en calculant un histogramme de la teinte. L'histogramme est rendu robuste en pondérant chaque valeur de la teinte par sa saturation $sat = O1^2 + O2^2$.

### 3.2.2 SIFT+Global Context

Mortensen *et al.* [99] proposent un descripteur qui combine à la fois des caractéristiques locales et globales. Les auteurs utilisent SIFT comme descripteur local $L$, et une approche similaire à shape context (voir page 27) pour calculer un descripteur global $G$. Le descripteur final $F$ s'écrit :

$$F = \begin{bmatrix} \omega L \\ (1-\omega)G \end{bmatrix} \tag{3.2}$$

où $\omega$ est un paramètre de pondération.

Le descripteur local $L$ est calculé dans la région affine détectée dans la phase de détection. Le descripteur global $G$ est quant à lui calculé sur toute l'image. $G$ est obtenu en calculant la courbure maximale en chaque point de l'image, et en accumulant ces valeurs sous la forme d'un histogramme. Les valeurs de l'histogramme sont pondérées par une gaussienne de manière à donner plus d'importance aux points qui se situent en dehors de la région décrite par SIFT.

Le descripteur final $F$ est censé être plus performant que SIFT dans la mesure où l'information globale, introduite par $G$, permet de distinguer entre deux régions décrites de manière similaire par le descripteur local $L$ [99].

Nous reviendrons sur ce point à la section 3.5 et verrons si l'ajout d'une information globale de courbure permet de résoudre les cas ambigus comme celui de la figure 3.1.

## 3.3 Prise en compte du contexte dans la phase d'appariement

La prise en compte du contexte dans la phase d'appariement pour éliminer les appariements incorrects est une idée adoptée de longue date dans la communauté de la vision par ordinateur [114]. Le contexte d'une région ou d'un point d'intérêt, est en général défini par son voisinage immédiat et par les relations, géométriques et photométriques, entre les différents éléments de ce voisinage.

Il est évident que la prise en compte d'informations globales telles que les relations spatiales entre différentes régions, peut permettre d'éliminer des ambiguïtés et des faux appariements. Il est néanmoins important, de définir avec soin les relations utilisées.

Soient $u = \{u_1, \ldots, u_n\}$ et $v = \{v_1, \ldots, v_m\}$, deux ensembles de points détectés dans deux images. Chaque point est décrit par un descripteur, ici SIFT. Nous décrivons ci-dessous deux méthodes qui utilisent l'information fournie par le voisinage de chaque point pour réduire le nombre de faux appariements.

### 3.3.1 Relaxation

La technique de relaxation, introduite par Rosenfeld *et al.* [114], est un schéma itératif qui vise à accroître la cohérence et à réduire l'ambiguïté de la mise en correspondance en utilisant l'information fournie par le voisinage de chaque point.

On définit pour chaque point $u_i$ un ensemble de probabilités initiales $p_i^0(k), k = 1, ..., m$ ; $p_i^0(k)$ étant la probabilité que $u_i$ soit apparié avec $v_k$. Les probabilités sont alors mises à jour par un processus itératif jusqu'à ce qu'un état stationnaire soit atteint. Celui-ci correspond à un ensemble non ambigu d'appariements. La mise à jour est basée sur une fonction de compatibilité $q_i$ définie dans le voisinage $V_i$ du point $u_i$. Cette fonction de compatibilité indique la probabilité que $u_i$ soit apparié avec $v_k$ connaissant les appariements de ses voisins.

Plusieurs schémas de relaxation ont été proposés et ils diffèrent principalement par la définition de la fonction de compatibilité et la règle de mise à jour des probabilités. Un exemple standard de règle de mise à jour est défini comme suit par Hummel et Zucker [58] :

$$p_i^{t+1}(k) = \frac{p_i^t(k) q_i^t(k)}{\sum_k p_i^t(k) q_i^t(k)} \tag{3.3}$$

où

$$q_i^t(k) = \sum_j w_{ij} \left[ \sum_l p_{ij}(k,l) p_j^t(l) \right] \tag{3.4}$$

et $p_{ij}(k,l)$ est la probabilité que le point $u_i$ soit apparié avec le point $v_k$ sachant que le point $u_j$ est apparié avec $v_l$. $p_{ij}(k,l)$ est l'information contextuelle qui permet d'augmenter la cohérence. Les nombres $w_{ij}$ sont des poids qui indiquent l'influence de $u_j$ sur $u_i$. Ils sont normalisés et vérifient la relation $\sum_j w_{ij} = 1$.

La convergence de l'algorithme, établie par Hummel et Zucker [58], vers une solution correcte, i.e. un ensemble de correspondants corrects, dépend tès fortement des probabilités initiales et des probabilités conditionnelles $p_{ij}(k,l)$. Nous reviendrons plus en détail sur la relaxation au chapitre suivant.

### 3.3.2 Reinforcement Matching

L'idée de la méthode notée *reinforcement matching* introduite par Deng *et al.* [29] est très similaire à celle de la relaxation. Elle consiste dans la pratique à augmenter le score d'appariement d'un couple de points, si les deux points en question ont des voisinages similaires.

On commence par calculer une matrice de coût qui contient les distances Euclidiennes entre chaque paire de primitives :

$$C = \{c_{ij}\}_{1 \leq i \leq n \, , \, 1 \leq j \leq m} \tag{3.5}$$

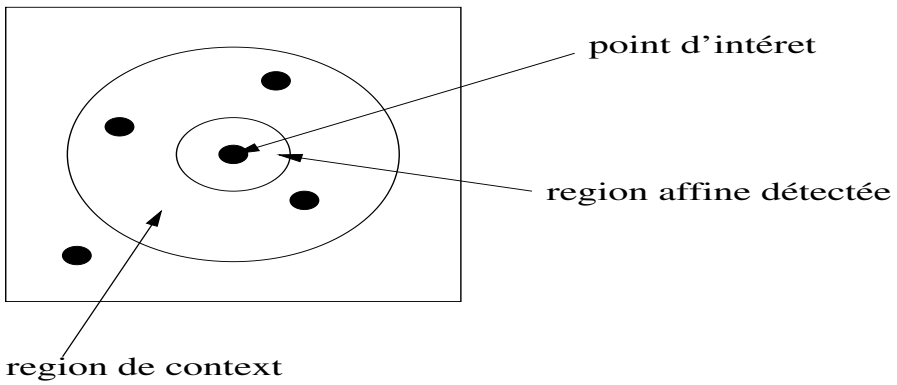



Fig. 3.3 – Région de contexte utilisée par la méthode *reinforcement matching*.

Ensuite, on choisit une fraction, par exemple 20%, des meilleurs appariements basés sur la matrice $C$ pour former les points d'ancrage (*anchor features*). Finalement, chaque région est élargie, par un facteur compris entre 3 et 8, pour définir la région de contexte (*region context*), voir figure 3.3. Les distances de la matrice de coût sont alors mises à jour en utilisant un score de contexte obtenu en comptant le nombre de points d'ancrage appariés entre deux régions de contexte :

$$c'_{ij} = \frac{c_{ij}}{log_{10}(10 + num_{support})} \tag{3.6}$$

où, $num_{support}$ est le nombre de points d'ancrage appariés entre les deux régions de contexte des deux points $u_i$ et $v_j$. Cette mise à jour est comparable à celle effectuée dans la relaxation par la fonction de compatibilité $q_i$.

## 3.4 Autres alternatives

Il existe d'autres méthodes qui ne considèrent pas l'information contextuelle dans le sens où elle a été définie ci-dessus, i.e. l'information fournie par le voisinage d'un point, mais qui visent à améliorer la performance des invariants locaux. Nous en présentons deux dans les sections suivantes. La première est basée sur la décomposition spectrale d'une matrice de proximité, et la seconde sur la notion d'entropie.

### 3.4.1 SVD Matching

L'utilisation des méthodes de décomposition spectrale pour la mise en correspondance d'images remonte au travaux de Scott et Longuet-Higgins en 1991 [126]. Plus récemment, Delponte *et al.* [28] utilisent cette approche pour la mise en correspondance de régions caractérisées par SIFT.

L'approche consiste à calculer dans un premier temps une matrice de proximité $G$ ( la terminologie date des travaux de Scott et Longuet-Higgins qui calculent une vraie matrice de proximité dans le sens où ils utilisent la distance Euclienne entre les points) :

$$G_{ij} = exp(\frac{-d_{ij}^2}{2\sigma^2}) \tag{3.7}$$

où $d_{ij}$ est la distance Euclidienne entre les descripteurs des points $u_i$ et $v_j$ et $\sigma$ un terme de lissage.

On réalise ensuite une décomposition en valeurs singulières (SVD) de la matrice $G : G = VDU^T$, puis on définit une nouvelle matrice $P$ en remplaçant toute les valeurs singulières par 1 : $P = VEU^T$ où $E$ est la matrice diagonale telle que $E_{ii} = 1$.

Finalement, $u_i$ est apparié avec $v_j$, si l'élément $P_{ij}$ de $P$ est le plus grand élément de la ligne $i$ et de la colonne $j$. Pour plus de robustesse, Delponte utilise une approche du type PPVRD en considérant le plus grand et le second plus grand élément de chaque ligne et de chaque colonne. Voir [28] pour plus de détails.

Remarquons que si dans la méthode initiale de Scott et Longuet-Higgins [126], la décomposition en valeurs singulières a un sens car la matrice $G$ est effectivement une matrice de proximité (les points dans les deux images ont des positions voisines), dans le cas de transformations plus importantes il n'est pas sûr que la décomposition SVD conduise à des résultats corrects. Les méthodes de détection de points d'intérêt sont sensibles au bruit dans les images, et les points ne sont pas précisément localisés. D'autre part, il n'est pas certain que le remplacement de toutes les valeurs singulières par 1, réduise les ambiguïtés.

### 3.4.2 Prise en compte de l'entropie

Zivkovic et Kröse [166] introduisent une mesure de similarité pour la mise en correspondance d'invariants locaux basée sur la théorie de l'information. En particulier, les auteurs utilisent la notion d'information mutuelle pour comparer deux régions. L'information mutuelle ne dépend pas seulement de la similarité entre deux signaux, mais également de leur

complexité. En adoptant une approximation de l'information mutuelle, Zivkovic et Kröse proposent la mesure de similarité suivante :

$$S_{MI}(u_i, v_j) = S_E + \frac{1}{2}(H_{u_i} + H_{v_j}) \tag{3.8}$$

où $S_E$ est la distance Euclidienne entre $u_i$ et $v_j$ et $H_z$ est la mesure de l'entropie de la région $z$ :

$$H_z = -\sum_{1}^{N} z_i log z_i$$

avec $\mathbf{z} = [z_1, \ldots, z_N]^T$ le descripteur de la région $z$.

Cette mesure de similarité ajoute à SIFT, une mesure de la complexité de la distribution de gradient dans la région. On peut donc classer cette méthode dans la catégorie de celles qui cherchent à enrichir la description locale.

## 3.5 Comment faut-il utiliser l'information contextuelle ?

Nous adoptons comme cas d'étude le couple d'image de la figure 3.1. Sur cet exemple difficile, nous verrons comment les différentes méthodes décrites plus haut permettent d'améliorer ou non les résultats.

Il y a respectivement 318 et 304 points d'intérêt détectés dans chacune des deux images en utilisant le détecteur Harris-Affine. Nous évaluons la performance de chaque méthode en utilisant les critères de précision et de rappel. La précision est définie par le rapport entre le nombre d'appariements corrects et le nombre total d'appariements :

$$precision = \frac{\#\ appariements\ corrects}{\#\ total\ d'appariements} \tag{3.9}$$

Le rappel est défini par le rapport entre le nombre d'appariements corrects trouvés et le nombre d'appariements corrects possibles entre deux vues :

$$rappel = \frac{\#\ appariements\ corrects}{\#\ appariements\ possibles} \tag{3.10}$$

Le nombre d'appariements possibles est obtenu grâce à l'erreur de recouvrement (voir équation 2.14, page 26). Un couple de points $(u_i, v_j)$ est un appariement possible si l'erreur de recouvrement est inférieure à 0.5. Le nombre d'appariements possibles entre les deux

| Méthode | # appariements | # appariements corrects | précision | rappel | temps en s |
|---|---|---|---|---|---|
| PPVRD | 6 | 3 | 0.5 | 0.064 | 0.165 |
| SIFT+Color | 6 | 3 | 0.50 | 0.064 | 0.997 |
| SIFT+Global Context | 6 | 3 | 0.5 | 0.064 | 177.62 |
| Entropie | 4 | 3 | 0.75 | 0.064 | 0.174 |
| Reinforcement | 16 | 8 | 0.5 | 0.17 | 0.205 |
| SVD matching | 60 | 21 | 0.35 | 0.45 | 1.63 |

TAB. 3.1 – Comparaison des différents algorithmes avec le couple d'images de la figure 3.1.

images de la figure 3.1 est égal à 47.

Les deux critères de précision et de rappel définissent ensemble la performance d'une méthode. Une précision élevée traduit la confiance que l'on peut accorder à la méthode car la plupart des appariements trouvés sont corrects, et un rappel élevé traduit le fait que la plupart des appariements corrects entre les deux images ont été trouvés.

Nous évaluons les performances des 5 méthodes : SIFT+Color, SIFT+Global Context, Reinforcement Matching, SVD Matching et Entropie, et nous les comparons à celle d'une mise en correspondance en utilisant uniquement le descripteur SIFT et une méthode du plus proche voisin avec rapport de distance PPVRD. Notons que pour les 5 méthodes évaluées, les appariements sont calculés en utilisant la technique du PPVRD soit après enrichissement du descripteur (dans les cas de SIFT+Color, SIFT+Global Context et Entropie), soit après la mise à jour des scores d'appariement (dans les cas de Reinforcement Matching et SVD Matching).

Les résultats sont rassemblés dans le tableau 3.1. Soulignons que pour obtenir ces résultats, nous avons utilisé nos propres implémentations des différents algorithmes. On note que toutes les méthodes sont très rapides à l'exception de SIFT+Global Context dont le temps de calcul très élevé est dû au fait que le descripteur global est long à calculer et qu'il est calculé sur l'image entière. Les images utilisées sont de résolution 480x640. On note aussi que le temps d'exécution de SVD Matching est relativement élevé par rapport à celui des autres méthodes, excepté SIFT+Global Context bien entendu. Une grande partie de ce temps est employé par l'algorithme de décomposition en valeurs singulières (nous utilisons l'algorithme standard de *Numerical Recipes* [110]).

Une première remarque ressort de ce tableau. La prise en compte du contexte dans la phase de description (SIFT+Color, SIFT+Global Context et Entropie) n'améliore pas les résultats obtenus par SIFT seul (PPVRD). On obtient le même nombre d'appariements

avec la même précision et le même rappel pour SIFT+Color et pour SIFT+Global Context. Avec la prise en compte de l'entropie, la précision est un peu meilleure mais le nombre d'appariements corrects reste très faible, i.e. égal à 3. Le rappel est donc lui aussi très faible (rappellons qu'il y a 47 appariements possibles entre les deux images).

Comme nous l'avons déjà souligné, avec 3 appariements corrects, on peut difficilement mettre en œuvre une méthode telle que RANSAC pour estimer de la transformation géométrique entre les deux images.

De meilleurs résultats sont obtenus par les méthodes qui tiennent compte du contexte dans la phase d'appariement. Avec Reinforcement, le nombre d'appariements corrects est multiplié par 3 tout en maintenant une précision égale à 0.5. Le rappel de la méthode est donc amélioré. SVD Matching fournit le plus grand nombre d'appariements corrects, sept fois plus que PPVRD, correspondant à un rappel égal à 0.45. Mais, la précision obtenue par SVD est très faible, inférieure à 0.5, et une méthode telle que RANSAC échouera dans ce cas.

### 3.5.1 Remarques

Bien que sommaires, car obtenus pour une seule paire d'images, ces résultats donnent une indication importante sur la manière d'utiliser l'information contextuelle. En effet, son utilisation s'avère nécessaire dans les cas ambigus où le descripteur local SIFT seul n'arrive pas à distinguer entre des régions similaires, ce qui est le cas dans nos expériences dans le cas de structures répétitives. On peut supposer qu'une méthode qui permettrait d'améliorer, de façon notable, les résultats dans ce cas difficile, donnerait de meilleur résultats dans des situations moins ambigües. Nous évaluerons la justesse de cette hypothèse dans les chapitres suivants à partir d'expériences plus nombreuses.

On peut donc déduire des résultats présentés ci-dessus que l'information contextuelle est mieux prise en compte dans la phase d'appariement. La faible performance des méthodes qui utilisent l'information contextuelle dans la phase de description peut être expliquée par le fait que la caractérisation reste locale. Par conséquent, l'ajout d'information supplémentaire ne suffit pas pour distinguer entre des structures localement similaires. Il est donc nécessaire de prendre en compte une information plus "*globale*" pour distinguer des régions localement similaires. Toutefois, la méthode SIFT+Global Context de Mortensen *et al.* [99] donne des résultats peu satisfaisants parce que l'information globale est calculée sur l'image entière. On perd donc le caractère local du descripteur car pour tous les points, l'information globale est la même. Au lieu de réduire l'ambiguïté de l'appariement, on

l'augmente. Un bon compromis est obtenu par la méthode Reinforcement dans laquelle la région de contexte est plus grande que la région de calcul du descripteur, mais reste petite par rapport à la taille de l'image.

## 3.6 Conclusion

Ce chapitre a mis en évidence les limites des invariants locaux pour la mise en correspondance d'images. En particulier, lorsque les images présentent des structures répétitives, les invariants locaux seuls ne suffisent plus pour obtenir un nombre suffisant d'appariements corrects. Il devient nécessaire de prendre en compte une information contextuelle et nous avons vu que la prise en compte de cette information dans la phase d'appariement donne de meilleurs résultats. Néanmoins, aucune des méthodes rencontrées ne donne des résultats totalement satisfaisants.

Nous proposons dans le chapitre suivant un algorithme, basé sur une technique de relaxation, qui permet d'améliorer les résultats de manière significative. En particulier, la méthode est rapide, et on obtient un nombre de faux appariements réduit tout en ayant un rappel élevé même dans les cas les plus difficiles.

# Chapitre 4

# Une méthode robuste de mise en correspondance

Ce chapitre présente un algorithme robuste de mise en correspondance basé sur la technique de relaxation. Après avoir présenté la méthode sur laquelle nous nous basons et présenté les principales limitations, nous montrons comment celle-ci peut être rendue rapide pour apparier des ensembles de points de grande taille. Nous décrivons ensuite différentes manières de prendre en compte l'information contextuelle. Enfin, nous comparons cet algorithme aux principales approches présentées dans le chapitre précédent dans le cadre de la mise en correspondance d'images présentant des changements géométriques importants.

## 4.1 Introduction

Dans le chapitre 3, nous avons vu que la simple comparaison des invariants locaux est insuffisante pour résoudre les ambiguïtés qui peuvent se présenter dans le cadre de la mise en correspondance d'images. En particulier, lorsque les images présentent des structures répétitives ou lorsqu'il faut détecter un objet dans une scène complexe, les invariants locaux seuls ne suffisent plus pour obtenir un nombre suffisant d'appariements corrects. Il est alors nécessaire de prendre en compte des informations supplémentaires et nous avons vu que la prise en compte de l'information contextuelle dans la phase d'appariement donne de meilleurs résultats.

La technique de relaxation, décrite à la section 3.3.1, permet de tenir compte de l'information contextuelle dans la phase d'appariement. En effet, le processus itératif de mise à

jour des probabilités d'appariement, tient compte de l'information fournie par le voisinage de chaque point pour accroître ou diminuer sa probabilité d'appariement avec un point de l'autre image. La relaxation semble donc être une idée intéressante et nous présentons dans la section suivante les différents algorithmes de relaxation proposés dans la littérature en mettant l'accent sur celui qui a servi de base à notre méthode de mise en correspondance.

## 4.2 Mise en correspondance par relaxation

### 4.2.1 Définitions et notations

Comme souligné à la section 3.3.1, l'idée principale de la relaxation consiste à utiliser l'information fournie par le voisinage de chaque point pour accroître la cohérence et réduire l'ambiguïté de la mise en correspondance. Commencons par définir plus précisément ces deux notions de cohérence et d'ambiguïté.

Soient deux ensembles de primitives détectées dans deux images $I_1$ et $I_2$ : $u = \{u_1, \ldots, u_n\}$ et $v = \{v_1, \ldots, v_m\}$. Chaque primitive est décrite par un descripteur. On note $p_i(k)$ la probabilité que $u_i$ soit appariée avec $v_k$.

Pour chaque primitive $u_i$, on définit un voisinage $V_i$ et pour $u_j \in V_i$ on définit un ensemble de probabiltés conditionnelles $p_{ij}(k,l)$. $p_{ij}(k,l)$ indique la probabilité que la primitive $u_i$ soit appariée avec la primitive $v_k$, sachant que $u_j$ est appariée avec $v_l$. Les probabilités $p_{ij}(k,l)$ représentent l'information contextuelle et on les suppose connues a priori. Elles sont utilisées pour calculer la fonction de compatibilité $q_i$ :

$$q_i(k) = \sum_{j \in V_i} w_{ij} \left[ \sum_l p_{ij}(k,l) p_j(l) \right] \tag{4.1}$$

Dans la pratique, les probabilités $p_i(k)$ sont obtenues à partir de mesures bruitées et souffrent de deux inconvénients [32] :

1. **L'incohérence** : elles ne vérifient pas la règle de Bayes

$$p_i(k) = \sum_{l=1}^{m} p_{ij}(k,l) p_j(l) \text{ pour } u_j \in V_i$$

   Ce qui indique que les probabilités $p_i(k)$ ne sont pas compatibles avec l'information contextuelle représentée par les $p_{ij}(k,l)$.

2. **L'ambiguïté** : elles ne fournissent pas un appariement non-ambiguë. Ce qui signifie qu'on ne peut pas décider, de manière certaine, du correspondant de la primitive $u_i$ ou, de manière équivalente, que le vecteur de probabilités $p_i$ est différent d'un vecteur unité $[0, \ldots, 0, 1, 0, \ldots, 0]^T$

Les différentes méthodes de relaxation ont pour but d'augmenter la cohérence et de réduire l'ambiguïté de la mise en correspondance.

On peut utiliser différents types de primitives (des points, des segments, des régions, etc). Dans la suite de ce chapitre, on considère que les primitives à apparier sont des points d'intérêt.

### 4.2.2 Différentes approches

La technique de relaxation, introduite par Rosenfeld *et al.* [114] pour accroître la cohérence et réduire l'ambiguïté de la mise en correspondance est très utilisée en vision par ordinateur. Le principe de la relaxation est l'adaptation (ou la mise à jour) des probabilités d'appariement en utilisant l'information fournie par le voisinage de chaque point.

Plusieurs schémas de relaxation ont été proposés et ils diffèrent principalement par la définition de la fonction de compatibilité ainsi que par la règle de mise à jour des probabilités. Une des premières approches de la relaxation en vision par ordinateur a été proposée par Rosenfeld *et al.* [114] et elle est similaire à la méthode proposée par Hummel et Zucker [58] qui utilisent la règle de mise à jour suivante :

$$p_i^{t+1}(k) = \frac{p_i^t(k) q_i^t(k)}{\sum_k p_i^t(k) q_i^t(k)} \tag{4.2}$$

L'algorithme converge vers un point stationnaire après un nombre réduit d'itérations.

Plus récemment, la relaxation a été utilisée par Zhang *et al.* [164] ainsi que Gouet *et al.* [46] pour la mise en correspondance de points d'intérêt. Dans tous ces travaux, seul un critère de cohérence est pris en compte à travers la fonction de compatibilité $q_i$. Soulignons que dans la pratique, cette fonction de compatibilité est généralement définie par des contraintes géométriques. Schmid [121] utilise la conservation des angles, l'angle défini par deux voisins d'un point doit être constant pour toutes les vues de ce point. Zhang [164] utilise la distance entre le point étudié et ses voisins. Gouet [46, 45] utilise aussi la conservation des angles entre les points voisins, mais elle met en place une contrainte angulaire basée sur le gradient multi-spectral des points considérés.

L'ensemble de correspondants obtenus par la règle de Hummel et Zucker n'est pas

totalement non ambigu car aucun critère d'ambiguïté n'est pris en compte. C'est la raison pour laquelle, Zhang [164] ainsi que Gouet [46, 45] utilisent une méthode du type PPVRD à la fin du processus de relaxation pour éliminer les appariements ambigus.

### 4.2.3 Algorithme de Faugeras et Berthod

Faugeras et Berthod [32] proposent une approche par optimisation de la relaxation. Ils définissent un critère global à optimiser qui tient compte à la fois de la cohérence et de l'ambiguïté :

$$C = \alpha C_1 + (1 - \alpha) C_2 \tag{4.3}$$

$C_1$ mesure la cohérence de la mise en correspondance, i.e. la compatibilité de l'appariement d'un point avec ceux de ses voisins. Cette mesure de la cohérence est définie par :

$$C_1 = \frac{1}{2n} \sum_{i=1}^{n} \|p_i - q_i\|^2 \tag{4.4}$$

$C_2$ mesure l'ambiguïté de la mise en correspondance comme la somme des entropies de chaque appariement, et est définie par :

$$C_2 = \frac{m}{m-1} \left[ 1 - \frac{1}{n} \sum_{i=1}^{n} \|p_i\|^2 \right] \tag{4.5}$$

Notons que les termes $\frac{1}{2n}$ et $\frac{m}{m-1}$ ne servent qu'à normaliser $C_1$ et $C_2$.

Le terme $C_1$ est minimal lorsque la mise en correspondance est cohérente, i.e. lorsque les appariements de tous les points sont compatibles avec ceux de leurs voisins. Le terme $C_2$ est minimal lorsque les appariements sont non ambigus, i.e. les vecteurs de probabilités sont égaux à des vecteurs unités. Le but est donc de minimiser le critère $C$.

Si on note $p$ le vecteur obtenu par concaténation des vecteurs $p_i$, i.e. $p = [p_1, \ldots, p_n]^T$, alors le problème de la mise en correspondance se ramène à la minimisation de $C(p)$ sous les contraintes linéaires définies par :

$$\begin{cases} \displaystyle\sum_{k=1}^{m} p_i(k) = 1 \quad i = 1, \ldots, n \\ p_i(k) \geq 0 \quad i = 1, \ldots, n \quad k = 1, \ldots, m \end{cases} \tag{4.6}$$

Ces contraintes définissent un sous espace convexe $\mathbf{K}$ de $\mathbf{R}^{mn}$.

Le problème d'optimisation est résolu par la méthode du gradient projété et pour chaque point $u_i$, le point $v_k$ qui a la plus grande probabilité finale est choisi comme correspondant.

L'approche par optimisiation peut être vue comme une généralisation de la méthode de relaxation présentée plus haut. En effet, si on prend $\alpha = 1$ dans la définition du critère (équation (4.3)), i.e. si on prend $C = C_1$, on montre que les deux approches sont équivalentes, voir [32] pour plus de détails.

En résumé, la règle de Humel et Zucker ne tient compte que de la cohérence, tandis que l'approche de Faugeras et Berthod permet de prendre en compte les deux critères de cohérence et d'ambiguïté en même temps. Cette dernière approche semble donc plus intéressante. Elle est toutefois limitée dans la pratique par sa grande complexité comme nous allons le voir dans la section suivante.

### 4.2.4 Limitations

La principale limitation de l'approche par optimisation, comme l'approche standard, est sa grande complexité. En effet, si $n$ et $m$ sont les nombres de points dans chacune des images et $V$ le cardinal de $V_i$ pour $i = 1, \ldots, n$, alors l'algorithme est d'une complexité de l'ordre de $O(nm^2V)$.

En théorie, il faut considérer dans le calcul de la fonction de compatibilté $q_i$ (équation (4.1)) les $m$ points de l'ensemble $v$. La méthode est donc appropriée pour des applications telles que la classification ou la segmentation, dans lesquelles il faut assigner un nombre réduit d'étiquettes, i.e. $m \approx O(10^2)$, à chaque point d'une image. Pour une application telle que la mise en correspondance où il faut apparier un nombre élevé de points, i.e. $m \approx O(10^4)$, la méthode devient impraticable car l'occupation de la mémoire est importante, et le temps de calcul est élevé. Cela, d'autant plus que la fonction de compatibilté $q_i$ doit être estimée à chaque itération.

Prenons par exemple le couple d'images de la figure 4.1. Il y a un changement d'échelle important (un facteur 4 environ), de même qu'une rotation entre les deux images. En utilisant le détetcteur Harris-Affine, il y a respectivement 1889 et 685 points d'intérêt détectés dans chaque image. On ne peut raisonnablement pas, pour chaque point $u_i$ de la première image et pour chacun de ses voisins $u_j \in V_i$, considérer tous les points de la seconde image dans le calcul de la fonction de compatibilté $q_i$. Cela nécessiterait un espace mémoire qui dépasse largement celle dont nous disposons sur les ordinateurs de bureau (de l'ordre de plusieurs Go).

Fig. 4.1 – Exemple d'images à apparier. Il y a respectivement 1889 et 685 points d'intérêt détectés dans chaque image.

| Méthode | # appariements | # appariements corrects | précision | temps en s |
|---|---|---|---|---|
| Hummel et Zucker [58] | 235 | 36 | 0.15 | 6.87 |
| Faugeras et Berthod [32] | 414 | 100 | 0.24 | 7.28 |

Tab. 4.1 – Résultats de la mise en correspondance des images de la figure 4.1.

Il faut donc sélectionner pour chaque point $u_i$ de la première image, un ensemble réduit de points de la seconde image comme ensemble des correspondants potentiels. On peut pour ce faire utiliser une mesure de similarité entre les descripteurs des points dans les deux images, et prendre pour chaque points ses $K$ plus proches voisins. En prenant $K = V = 10$ on obtient les résultats présentés dans le tableau 4.1. Comme on peut le voir sur ce tableau, les deux méthodes ont une complexité équivalente, mais l'approche par optimisation donne de meilleurs résultats que l'approche de Hummel et Zucker. Cependant, les précisions obtenues sont faibles et ne permettent pas d'estimer la transformation entre les deux images.

Dans le cas de structures répétitives, comme l'exemple de la figure 3.1, les résultats obtenus sont peu satisfaisants comme le montre le tableau 4.2. Les précisions obtenues sont très faibles.

Une autre limitation de la méthode vient du fait que l'algorithme converge vers un

| Méthode | # appariements | # appariements corrects | précision | temps en s |
|---|---|---|---|---|
| Hummel et Zucker [58] | 56 | 13 | 0.23 | 1.68 |
| Faugeras et Berthod [32] | 80 | 22 | 0.27 | 1.78 |

Tab. 4.2 – Résultats de la mise en correspondance dans le cas de structures répétitives (couple d'images de la figure 3.1).

minimum local. Autrement dit, les probabilités finales dépendent fortement des probabilités initiales, $p_i^0(k), k = 1, ..., m$, et des probabilités conditionnelles, $p_{ij}(k, l)$ [58, 111]. Si ces quantités ne sont pas correctement estimées, alors les probabilités finales conduisent à de nombreux faux appariements.

En résumé, la prise en compte d'un terme d'ambiguïté dans l'approche de Faugeras et Berthod permet d'obtenir des résultats légèrement meilleurs que ceux obtenus par la méthode classique de Hummel et Zucker. Cependant, les deux méthodes sont d'une complexité élevée, et les résultats obtenus sont peu satisfaisants lorsque le changement de point de vue entre les images à apparier est important, ou lorsque les images présentent des structures répétitives.

Nous pensons que ces limitations sont dues, pour une grande part, à l'estimation de l'information contextuelle représentée par les probabilités conditionnelles. Nous montrons dans la section suivante comment réduire la complexité de l'algorithme pour pouvoir apparier rapidement un nombre de points élevé, et comment estimer les probabilités initiales et conditionnelles nécessaires pour obtenir des résultats corrects.

## 4.3 Une mise en œuvre rapide et robuste

### 4.3.1 Réduction de la complexité

Comme mentionné ci-dessus, la fonction de compatibilité doit être réévaluée à chaque itération dans l'algorithme de Faugeras et Berthod. Ce qui a pour conséquence d'accroître la complexité de la méthode. Pour réduire cette complexité, on peut essayer d'écrire le critère à minimiser sous une forme plus "compacte". Il s'agit de trouver une représentation du critère de telle sorte que, toute l'information nécessaire pour calculer la fonction de compatibilté $q_i$ soit obtenue une seule fois.

Le critère $C$, équation (4.3), étant quadratique en $p$, il peut se mettre sous la forme suivante :

$$C(p) = \frac{1}{2} p^T H p + cte \tag{4.7}$$

L'obtention de cette équation, décrite dans l'annexe B, n'est pas aisée. En effet, pour calculer $q_i$ il faut avoir les probablilités conditionnelles $p_{ij}(k, l)$ (qu'on suppose connue à priori), mais aussi les probabilités $p_i$ dont l'estimation dépend de $q_i$.

Nous montrons, voir annexe B, que le critère $C$ peut s'écrire :

$$C([p_1, \ldots, p_n]^T) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} p_i^T H_{ij} p_j + cte \tag{4.8}$$

où chaque matrice $H_{ij}$ contient les probabilités conditionnelles $p_{ij}(k, l)$, i.e. l'information contextuelle nécessaire pour évaluer la fonction de compatibilité $q_i$.

La matrice $H$ est donc formée de plusieurs matrices $H_{ij}$ :

$$H = \begin{pmatrix} H_{11} & \cdots & H_{1n} \\ \vdots & H_{ij} & \vdots \\ H_{n1} & \cdots & H_{nn} \end{pmatrix}$$

Cette écriture présente deux principaux avantages. D'une part, en ne considérant dans la définition de la fonction de compatibilité que les points $u_j$ qui sont dans le voisinage $V_i$ de $u_i$, certaines matrices $H_{ij}$ sont nulles. En particulier, il est facile de montrer que pour $i = 1, \ldots, n$ et pour $j = 1, \ldots, n$ :

$$H_{ij} \neq 0 \quad \text{si} \quad \begin{cases} i = j \quad \text{ou} \\ u_j \in V_i \quad \text{ou} \\ \exists k / (u_i, u_j) \in V_k \times V_k \end{cases} \tag{4.9}$$

Voir l'annexe C pour l'obtention de ces conditions.

Donc en utilisant une structure de matrice creuse pour représenter $H$, on réduit la complexité de l'algorithme. Pour réduire la complexité plus encore, on ne considère pour chaque point $u_i$ qu'un certain nombre de points $v_k$ comme correspondants potentiels. Nous choisissons les $K$ plus proches voisins pour une mesure de similarité donnée. Ainsi, chaque matrice $H_{ij}$ est de taille $K \times K$ au lieu de $m \times m$, avec $K \ll m$.

D'autre part, la matrice $H$ est calculée une seule fois, ce qui rend l'algorithme rapide. A chaque itération, le gradient du critère est obtenu par l'équation suivante :

$$\frac{\partial C}{\partial p} = \frac{1}{2}(H + H^T)p \tag{4.10}$$

En général, $H$ n'est pas une matrice symétrique. Mais si elle l'est, alors le gradient est

obtenu par l'équation classique :

$$\frac{\partial C}{\partial p} = Hp \tag{4.11}$$

Toute l'information nécessaire pour calculer la fonction de compatibilité $q_i$ est contenue dans la matrice $H$. On n'a donc plus besoin de re-estimer $q_i$ à chaque itération comme cela est le cas dans l'algorithme de Faugeras et Berthod.

### 4.3.2 Estimation des probabilités

Nous avons déjà souligné à la section 4.2.4 que les résultats, i.e. les probabilités finales, dépendent fortement des probabilités initiales et conditionnelles. Par conséquent, l'estimation de ces quantités est d'une importance capitale.

#### Probabilités initiales

Les probabilités initiales sont obtenues par comparaison des descripteurs locaux. Nous utilisons SIFT qui est considéré comme étant le descripteur le plus performant [92], et pour chaque point $u_i$, nous choisissons les $K$ plus proches voisins comme correspondants potentiels. Les probabilités initiales sont alors données par l'équation suivante :

$$p_i^0(k) = \frac{1/d_{ik}}{\sum_{k=1}^{K} 1/d_{ik}} \quad i = 1, \ldots, n \quad k = 1, \ldots, K \tag{4.12}$$

où, $d_{ik}$ est la distance Euclidienne entre les descripteurs des points $u_i$ et $v_k$.

#### Probabilités conditionnelles

Pour chaque point d'intérêt $u_i$, la fonction de compatibilité $q_i$ permet de mesurer la cohérence de l'appariement de $u_i$ avec ceux de ses voisins. On peut donc interpréter $q_i$ comme une estimation de $p_i$ connaissant l'information à priori représentée par les $p_{ij}(k,l)$ pour les points $u_j$ appartenants à $V_i$. L'estimation des $p_{ij}(k,l)$ peut être basée soit sur la géométrie de la scène, soit sur l'information photométrique. Des contraintes géométriques sont utilisées par de nombreux auteurs. Par exemple, Schmid [123] utilise la conservation des angles, en considérant que l'angle défini par deux voisins d'un point doit être constant pour toutes les vues de ce point. Zhang [164] utilise la distance entre le point étudié et ces voisins. Gouet et Montesinos [46, 96] utilisent une contrainte angulaire basée sur le gradient multi-spectral. Nous basons l'estimation de nos probabilités conditionnelles sur

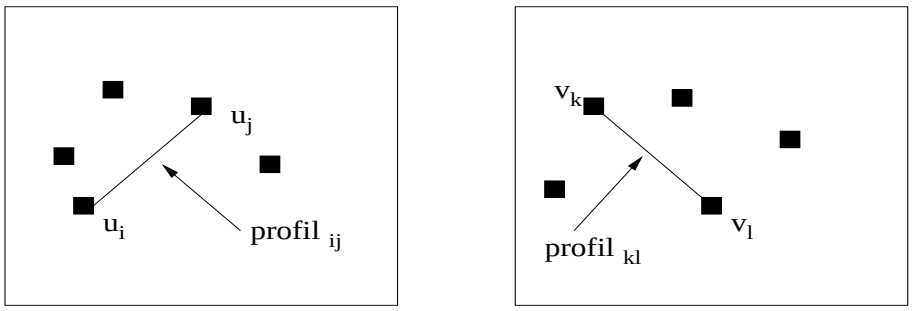



Fig. 4.2 – Calcul des probabilités conditionnelles avec des profils d'intensité.

l'information photométrique pour deux raisons. Premièrement, dans le cas d'un changement important de point de vue, la géométrie de la scène est mal conservée (les angles et les distances ne sont pas conservés). Deuxièmement, le descripteur SIFT fournissant une description géométrique du voisinage d'un point, il paraît intéressant d'utiliser une information photométrique complémentaire pour résoudre les ambiguïtés.

Nous essayons deux manières de calculer les probabilités conditionnelles : les profils d'intensité entre les points voisins et les régions de contexte définies par des points voisins.

*Profils d'intensité*

Soient deux couples de points $(u_i, u_j)$ et $(v_k, v_l)$, $u_j$ étant un voisin de $u_i$ et $v_l$ un voisin de $v_k$. Soit $profil_{ij}$ le profil d'intensité le long du segment reliant les points $u_i$ et $u_j$. De même, soit $profil_{kl}$ le profil d'intensité le long du segment reliant les points $v_k$ et $v_l$. Voir figure 4.2 pour une illustration. La probabilité que $u_i$ soit apparié avec $v_k$ sachant que $u_j$ est apparié avec $v_l$, est obtenue en comparant les deux profils d'intensité $profil_{ij}$ et $profil_{kl}$.

La similarité entre deux profils est évaluée de manière à assurer l'invariance aux changements affines géométriques et photométriques entre les deux images. L'invariance au changement de luminosité est assurée en normalisant chaque profil d'intensité de la manière suivante :

$$T(i) = \frac{T(i) - min_T}{max_T - min_T} \quad i = 1, \ldots, N \tag{4.13}$$

où $N$ est la longueur du profil $T$, et $min_T$ et $max_T$ les valeurs minimale et maximale d'intensité le long de $T$.

Enfin, l'information de chaque profil est représentée par les coefficients de Fourier d'ordre 1 du signal. Plus précisément, nous considérons les $q$ premiers coefficients défi-

nis par les formules ci-dessous :

$$\begin{cases} c_T^k &= \frac{1}{N}\sum_{i=0}^{N} T(i) sin(\frac{ki\pi}{N}) & k = 1, \ldots, q/2 \\ c_T^k &= \frac{1}{N}\sum_{i=0}^{N} T(i) cos(\frac{(k+1-q/2)i\pi}{N}) & k = q/2, \ldots, q \end{cases}$$

La similarité entre deux profils est évaluée par la distance entre les coefficients de Fourier. Nous prenons dans nos expériences, $q = 6$, et dans le cas d'une image couleur, nous faisons une moyenne des distances obtenues dans chacun des trois plans R, G et B.

Les profils d'intensité sont également utilisés par Tell et Carlsson [145]. Cependant, les auteurs de cet article utilisent les profils d'intensité pour calculer un descripteur de chaque point d'intérêt et ils mettent ensuite en correspondance les points par une méthode du plus proche voisin suivi de l'estimation de la transformation géométrique par RANSAC. Nous utilisons les profils d'intensité en plus de SIFT, comme information complémentaire dans la relaxation pour éliminer les ambiguïtés.

*Régions de contexte*

Une autre manière d'obtenir l'information contextuelle consiste à définir pour chaque point $u_i$ et pour chacun de ses voisins $u_j$, une région de contexte. On définit une région circulaire $C_{ij}$ dont le diamètre est égal à la distance entre les points $u_i$ et $u_j$. Voir figure 4.3. Dans chaque région ainsi définie, on calcule un histogramme de l'intensité lumineuse $histo_{ij}$. La probabilité que $u_i$ soit apparié avec $v_k$ sachant que $u_j$ est apparié avec $v_l$, est obtenue en calculant la distance entre les histogrammes $histo_{ij}$ et $histo_{kl}$. Une bonne mesure de la similarité entre deux histogrammes est donnée par le test du $\chi^2$ :

$$d(histo_1, histo_2) = \frac{1}{2}\sum_{k=1}^{N} \frac{[histo_1(k) - histo_2(k)]^2}{histo_1(k) + histo_2(k)}$$

Dans nos expérimentations, nous prenons des histogrammes de dimension égale $N = 16$.

### 4.3.3 Prise en compte des occultations

Dans de nombreuses applications, on souhaite avoir un correspondant unique pour chaque point. Aussi, pour prendre en compte les occultations, les changements de fond et les changements de point de vue, on ajoutte à l'ensemble des correspondants potentiels un point abstrait que l'on note $v_{nul}$. Les points de l'ensemble $u$ qui n'ont aucun correspondant dans $v$ seront appariés avec le point $v_{nul}$. Pour chaque point $u_i \in u$, l'ensemble des

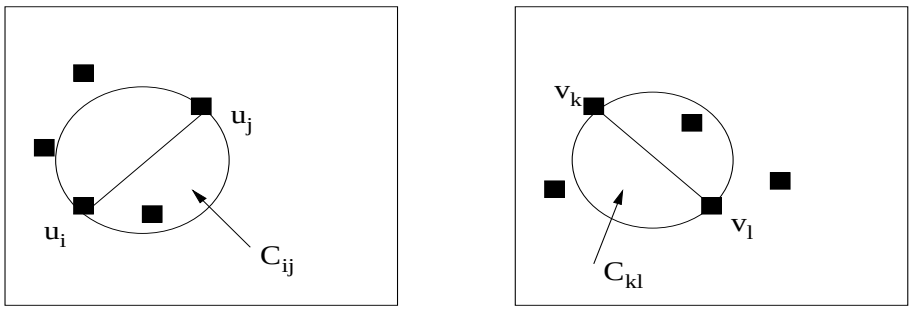



FIG. 4.3 – Calcul des probabilités conditionnelles avec des régions de contexte.

correspondants potentiels est donc :

$$PM_i = \{v_1^i, \ldots, v_K^i, v_{nul}\} \tag{4.14}$$

où $v_1^i, \ldots, v_K^i$ sont les $K$ plus proches voisins de $u_i$.

Les matrices $H_{ij}$ sont de taille $(K+1) \times (K+1)$ et peuvent se mettre sous la forme suivante :

$$H_{ij} = \left( \begin{array}{ccc|c} p_{ij}(1,1) & \cdots & p_{ij}(1,K) & p^{**} \\ \vdots & & \vdots & p^{**} \\ p_{ij}(K,1) & \cdots & p_{ij}(K,K) & p^{**} \\ \hline p^{**} & \cdots & p^{**} & p^{**} \end{array} \right) \tag{4.15}$$

où $p^{**}$ est une constante qui désigne la probabilité conditionnelle pour le point $v_{nul}$. Les probabilités initiales pour ce point sont également fixées à une valeur constante :

$$p_i^0(nil) = p^* \quad i = 1, \ldots, n \tag{4.16}$$

Une fois les matrices $H_{ij}$ obtenues, la matrice $H$ est calculée (voir section 4.3.1) et le problème d'optimisation est résolu par la méthode du gradient projété.

L'algorithme converge vers un minimum local en un nombre réduit d'itérations et pour chaque point $u_i$, le correspondant potentiel avec la plus grande probabilité finale est retenu comme son correspondant. Les points d'une image qui n'ont pas de correspondant dans l'autre image, doivent être appariés avec le point $v_{nul}$.

La méthode de mise en correspondance par relaxation est résumée par l'algorithme de la figure 4.4.

Etant donnés deux ensembles de points d'intérêt $u = \{u_1, \ldots, u_n\}$ et $v = \{v_1, \ldots, v_m\}$
- calculer les voisinages $V_i$ et $V_j$ pour $i = 1, \ldots, n$ et $j = 1, \ldots, m$ ;
- pour $i = 1, \ldots, n$ :
  - calculer les probabiltés initiales $p_i(k)$ (équation (4.12)) ;
  - calculer l'ensemble $MP_i$ des correspondants potentiels
    en prenant les $K$ plus proches voisins de $u_i$ ;
- pour $i = 1, \ldots, n$ et pour $u_j \in V_i$ :
  - pour $v_k \in MP_i$ et pour $v_l \in MP_j$ :
    calculer la probabilité conditionnelle $p_{ij}(k, l)$
  - calculer la matrice $H_{ij}$ (équation (4.15))
- calculer la matrice $H$ ;
- minimiser le critère $C$ (équation (4.7)) par la méthode du gradient projeté ;
- pour $i = 1, \ldots, n$ :
  - prendre comme correspondant du point $u_i$, le point $v_k$ tel que :
    $p_i(k) = max\{p_i(l);\ l = 1, \ldots, K + 1\}$

FIG. 4.4 – Résumé de la méthode de mise en correspondance par relaxation.

Etant données une condition initiale $p_0 \in \mathbf{K}$ et une précision $\varepsilon > 0$,
- calculer $p_1 = p_0 - \rho_0 P_{\mathbf{K}}(C'(p_0))$
- tant que $\|p_k - p_{k-1}\| > \varepsilon$ :
  calculer $p_{k+1} = p_k - \rho_k P_{\mathbf{K}}(C'(p_k))$

où $P_{\mathbf{K}}$ un est opérateur de projection sur le sous espace $\mathbf{K}$ défini par l'équation 4.6.

FIG. 4.5 – Algorithme du gradient projeté.

### 4.3.4 Détails d'implémentation : définition de l'opérateur de projection

Dans la méthode de mise en correspondance par relaxation, nous avons besoin de l'algorithme du gradient projeté pour minimiser le critère $C$. Dans ce dernier algorithme, présenté sur la figure 4.5, il est important de définir correctement l'opérateur de projection $P_{\mathbf{K}}$. Pour illustrer notre propos, nous nous plaçons dans le cas où les vecteurs de probabilité sont de taille égale à 3, cas plus simple à représenter de manière graphique. La figure 4.6 a) montre le domaine admissible $\mathbf{K}$ sur lequel il faut projeter les vecteurs de probabilité.

En partant d'un point initial $p_0 \in \mathbf{K}$, l'opérateur le plus simple est défini par :

$$P_{\mathbf{K}}(p_i) = p_i - \frac{1}{3}\sum_{k=1}^{3} p_i(k) \tag{4.17}$$

Dans la pratique, en utilisant cet opérateur on se retrouve très vite sur l'un des bords du domaine et on oscille d'un bord à l'autre du domaine à chaque itération. Ce phénomène d'oscillation est illustré par la figure 4.6 b). L'algorithme prend plus de temps pour converger (nombre élevé d'itérations) et, de plus, si on fixe un nombre maximal d'itérations à priori, le point d'arrêt obtenu risque de ne pas être un point stationnaire. Ce problème est connu dans le domaine de l'optimisation avec contraintes sous le nom de contre-exemple de Wolfe, voir par exemple [15].

Pour utiliser cet opérateur simple tout en évitant le problème d'oscillation, il faut faire en sorte de demeurer à l'intérieur du domaine $\mathbf{K}$. Il faut donc choisir le pas $\rho_k$ en conséquence. A chaque itération, on calcule le pas maximal $\rho_{max}$ qui permet de rester dans le domaine, puis on détermine le pas optimal $\rho_k$ dans l'intervalle $[0.05; 0.95\rho_{max}]$. Ce qui évite d'atteindre les bords du domaine et l'algorithme converge plus rapidement, figure 4.6 c).

Soulignons que Faugeras et Berthod [32] définissent un opérateur de projection qui tient compte des composantes nulles de $p$. L'opérateur est défini de telle sorte que les composantes qui sont nulles à une itération, restent nulles au cours des itérations suivantes. Ce qui permet d'éviter les oscillations. Cette méthode se rapproche de la méthode d'activation des contraintes utilisée dans le cadre de l'optimisation avec contraintes [15]. Le but étant d'obtenir un vecteur unité à la fin du processus.

La solution que nous proposons ne permet pas d'obtenir un vecteur de probabilité finale égale au vecteur unité. Néanmoins, l'une des composantes de ce vecteur sera plus grande que les autres, et c'est celle qui sera retenue. De plus, l'un de nos buts étant de proposer un algorithme rapide, nous adoptons cette solution car l'opérateur de projection est extrement simple à mettre en œuvre par rapport à celui proposé dans [32].

## 4.4 Evaluations expérimentales

### 4.4.1 Images tests et critères d'évaluation

#### Images tests

Nous évaluons la performance de notre algorithme dans le cadre de la mise en correspondance d'images avec un changement de point de vue important. Pour ce faire, nous utilisons des images largement utilisées dans la littérature et mises à disposition par Mikolajczyk et Schmid [93]. Les images sont disponible à l'adresse `http://www.robots.ox`.

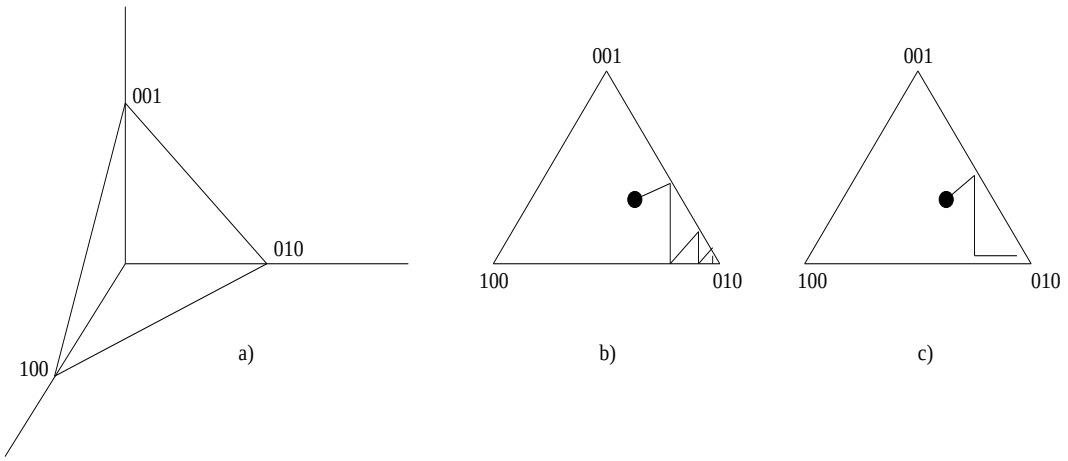



FIG. 4.6 – Phénomène d'oscillations : exemple d'un vecteur de probabilité de dimension 3. a) Sous espace convexe **K** définissant le domaine admissible ; b) oscillations sur les bords du domaine ; c) cas sans oscillations en restant à l'intérieur du domaine.

`ac.uk/~vgg/research/affine/`.

Nous utilisons quatre séquences contenant chacune six images. Les cinq dernières images de chacune des séquences présentent des transformations géométriques croissantes par rapport à la première image de la séquence. Nous choisissons les quatres séquences : *Graffiti*, *Boat*, *Wall* et *Bark*, car elles représentent trois types de transformations géométriques (changement de point de vue, changement d'échelle et rotation) et deux types de scènes différentes (scène structurée et scène texturée). Les premières, troisièmes et cinquièmes images de chaque séquence sont présentées sur la figure 4.7.

Pour évaluer la performance de l'algorithme en présence de structures répétitives, nous utlisons les quatres paires d'images de la figure 4.11.

### Critères d'évaluation

Les résultats sont évalués en utilisant les critères de précision et de rappel introduits au chapitre 3 (voir page 46). Rappelons que ces deux termes sont définis par :

$$precision = \frac{\#\ appariements\ corrects}{\#\ total\ d'appariements} \tag{4.18}$$

et

$$rappel = \frac{\#\ appariements\ corrects}{\#\ appariements\ possibles} \tag{4.19}$$

Dans toutes les expériences, nous prenons $V = K = 5$, i.e. nous considérons pour

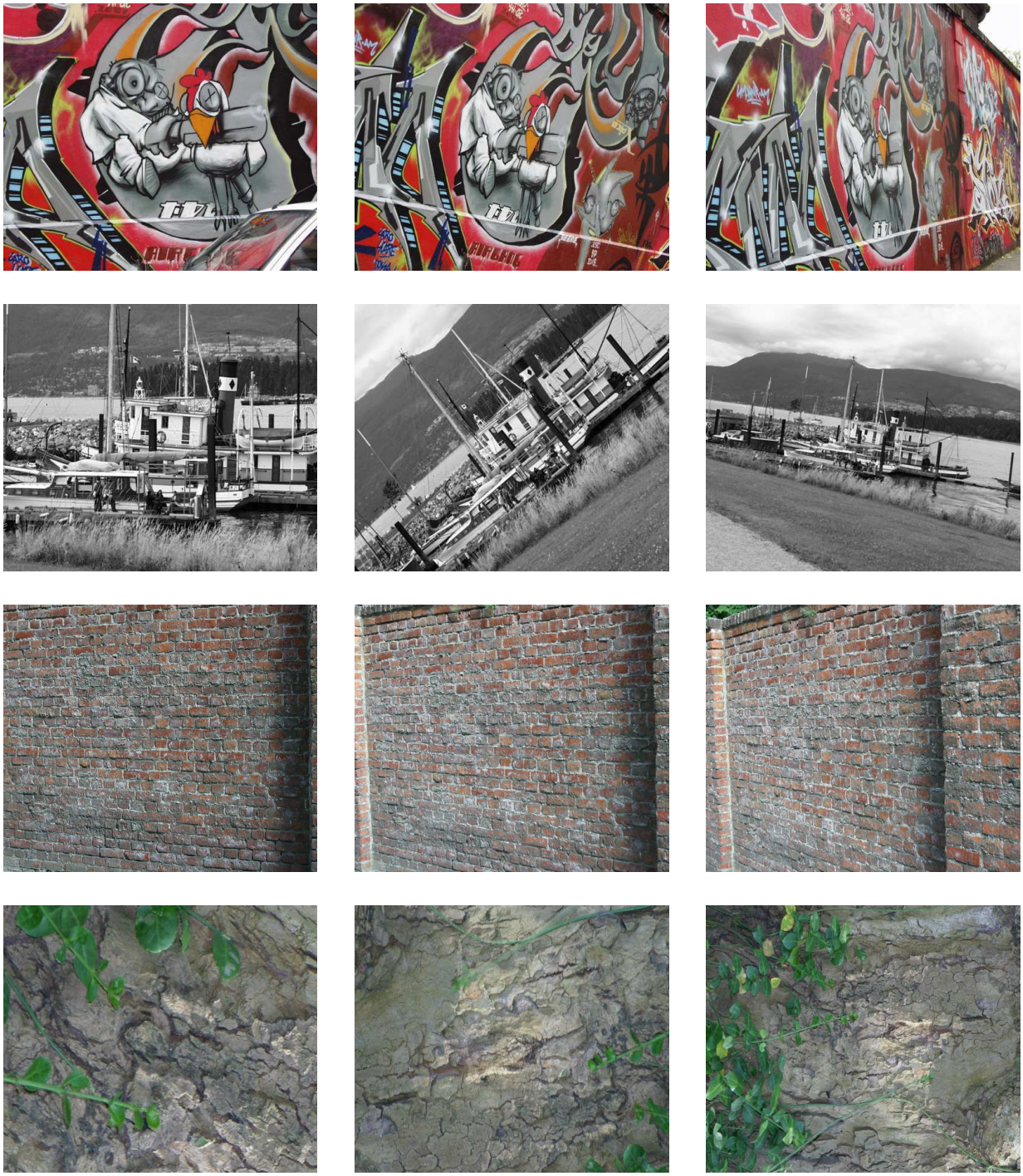


Fig. 4.7 – Première, troisième et cinquième image de chaque séquence. De haut en bas : **Graffiti** (changement de point de vue, scène structurée), **Boat** (changement d'échelle + rotation, scène structurée), **Wall** (changement de point de vue, scène texturée), **Bark** (changement d'échelle + rotation, scène texturée).

chaque points 5 voisins ainsi que 5 correspondants potentiels. Les probabilités initiales et conditionnelles pour le point $v_{nul}$ sont prises égales à 0.1. Enfin, la constante $\alpha$ du critère $C$, voir équation (4.3), est fixée à 0.5, i.e. les termes de cohérence et d'ambiguïté ont la même importance.

### 4.4.2 Comparaison des deux méthodes d'estimation des probabilités conditionnelles

Dans cette section, nous comparons les deux méthodes de calcul des probabilités conditionnelles, les profils d'intensité et les régions de contexte, présentées à la section 4.3.2. La comparaison tient compte de trois paramètres :

- le nombre d'appariements obtenus ;
- la précision ;
- le rappel.

Nous utilisons les quatre séquences d'images et pour chaque séquence, nous mettons en correspondance la première image avec les cinq images suivantes de la séquence.

Les résultats, figure 4.8 et 4.9, montrent que les deux méthodes donnent des performances très similaires. Le nombre d'appariements, la précision et le rappel obtenus pour chaque paire d'images avec les deux méthodes sont presque toujours les mêmes. Quand il y a une différence, celle-ci est très faible.

Les résultats sont les mêmes principalement parce que les deux méthodes utilisent la même information de couleur. De plus, en prenant $V = 5$, i.e. 5 voisins pour chaque point $u_i$, les régions de contexte sont assez petites. Par conséquent, l'information représentée par les histogrammes dans les régions de contexte, est équivalente à celle représentée par les profils d'intensité.

La différence la plus importante concerne le temps d'exécution. En moyenne, le calcul des probabilités conditionnelles avec les profils d'intensité est deux à trois fois plus rapide que le calcul avec les régions de contexte. C'est la raison pour laquelle, nous utiliserons dans la suite les profils d'intensité.

### 4.4.3 Comparaison avec la méthode originale

Dans cette section, nous comparons notre algorithme de relaxation avec l'algorithme de Faugeras et Berthod qui a servi de base à notre travail. Nous utilisons le couple d'images de la figure 4.1 comme exemple, et prenons les valeurs suivantes pour les paramètres de

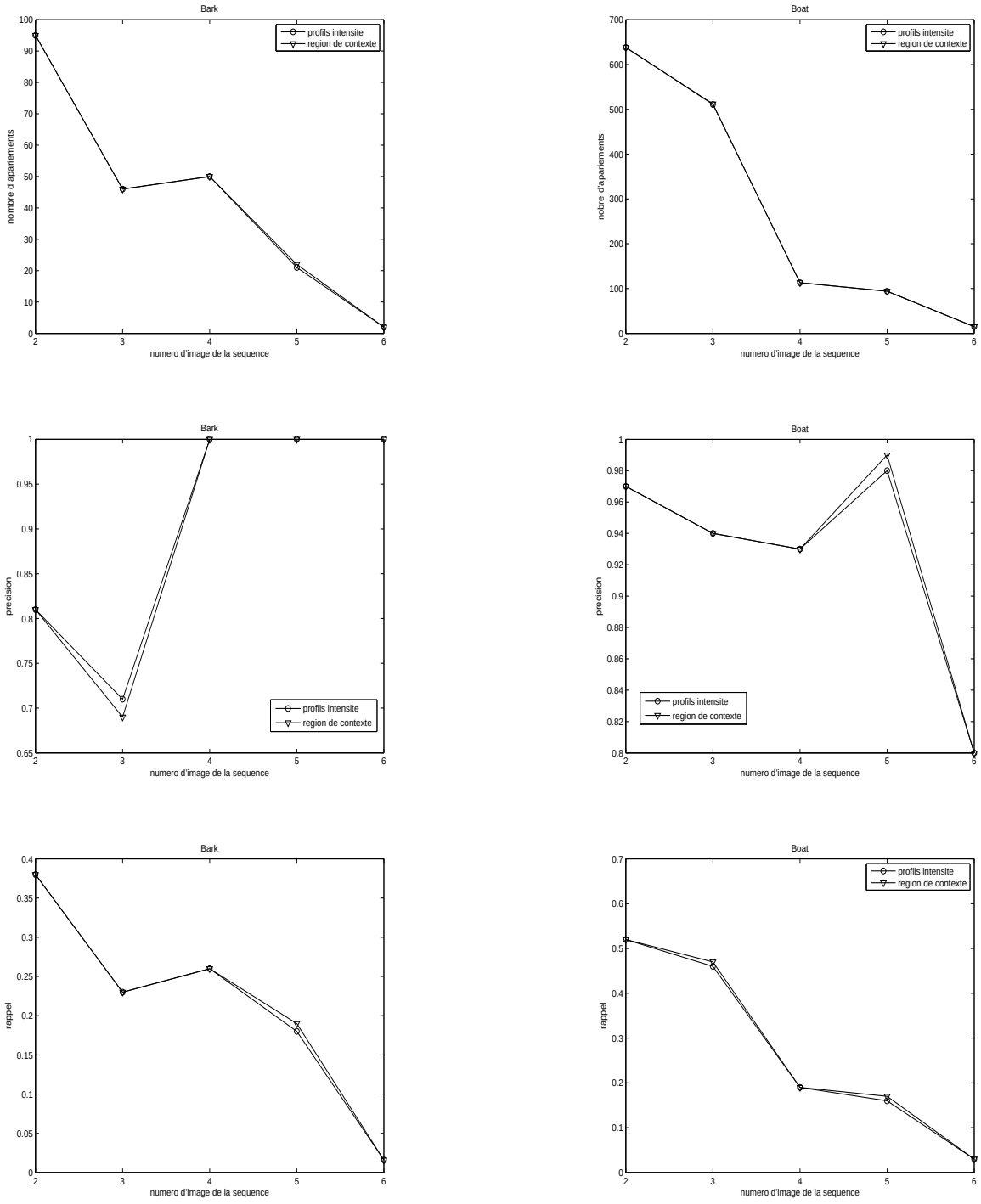



Fig. 4.8 – Comparaison des deux méthodes d'estimation des probabilités conditionnelles. De haut en bas : nombre d'appariements, précision et rappel. A gauche, résultats pour la séquence **Bark**. A droite, résulats pour la séquenc **Boat**.

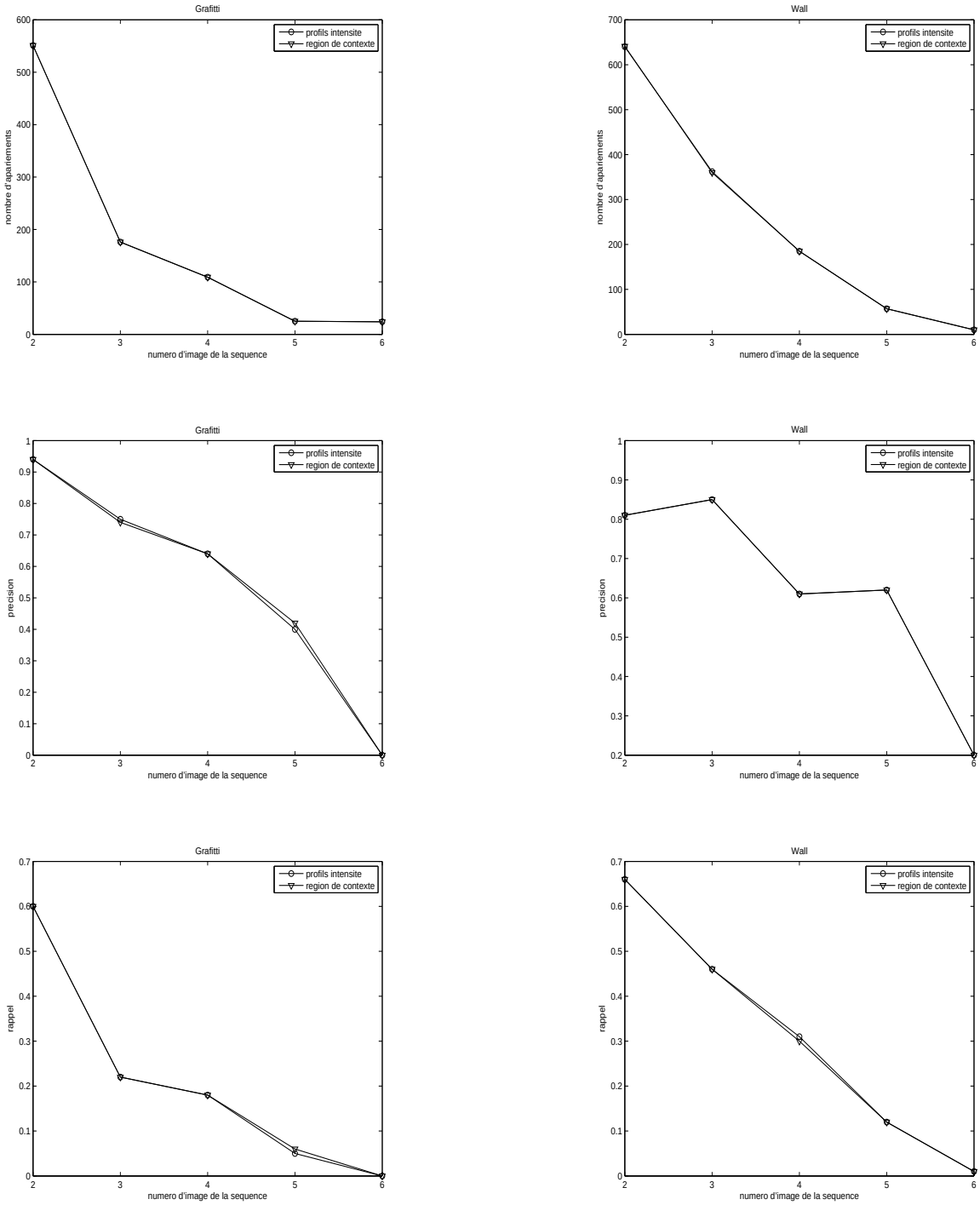



FIG. 4.9 – Comparaison des deux méthodes d'estimation des probabilités conditionnelles. De haut en bas : nombre d'appariements, précision et rappel. A gauche, résultats pour la séquence **Graffiti**. A droite, résulats pour la séquence **Wall**.

| Méthode | # appariements | # appariements corrects | précision | paramètres | | | temps en s |
|---|---|---|---|---|---|---|---|
| | | | | $\alpha$ | $V$ | $K$ | |
| Notre mise en œuvre | 94 | 92 | 0.98 | 0.5 | 5 | 5 | 4.80 |
| Faugeras et Berthod [32] | 576 | 98 | 0.17 | 0.5 | 5 | 5 | 5.9 |
| | 414 | 100 | 0.24 | 0.5 | 10 | 10 | 7.28 |
| | 101 | 96 | 0.95 | 0.5 | 10 | 50 | 28.24 |

Tab. 4.3 – Comparaison de notre algorithme de relaxation avec l'algorithme de Faugeras et Berthod en utilisant les images de la figure 4.1.

l'algorithme : $V = K = 5$, $\alpha = 0.5$ et $p^* = p^{**} = 0.1$. $p^*$ et $p^{**}$ étant respectivement les probabilités initiale et conditionnelle pour le point $v_{nul}$.

Les résultats obtenus sont rassemblés dans le tableau 4.3. Comme on peut le noter, avec les valeurs des paramètres définis ci-dessus, notre mise en œuvre de la relaxation donne d'excellents résultats. On obtient une précision égale à 0.98 en un temps de calcul égal à 4.8 s.

L'algorithme initial de Faugeras et Berthod, avec les mêmes valeurs de paramètres, donne des résultats peu satisfaisants (très faible précision). Pour obtenir des résultats comparables à ceux obtenus avec notre mise en œuvre, il faut accroître les valeurs des paramètres de l'algorithme, notamment, le nombre de correspondants potentiels de chaque point. Ce qui se traduit par un temps de calcul plus important.

La rapidité de notre mise en œuvre est due à la réécriture du critère sous forme matricielle. L'information contextuelle est calculée une seule fois et est représentée par la matrice $H$ (voir section 4.3.1). Il n'est donc pas nécessaire de réévaluer la fonction de compatibilité à chaque itération du processus d'optimisation, et le gradient est obtenu par l'équation (4.10).

### 4.4.4 Comparaison de différentes méthodes d'appariement

#### Les algorithmes

Dans cette section, nous comparons notre algorithme de relaxation, noté RELAX, avec les méthodes suivantes :

- PPVRD : plus proche voisin avec rapport de distances [79] ;
- SVD : méthode de décomposition en valeurs singulières [28] ;
- SIFT+COLOR : ajout de la couleur à la description locale SIFT [155] ;
- ENTROPIE : mesure de similarité basée sur l'entropie [166] ;

– REINF : renforcement des scores d'appariement par régions de contexte [29].

La méthode PPVRD (sans prise en compte d'information contextuelle) est considérée comme la méthode de référence pour la comparaison. Dans un premier temps, nous évaluons la robustesse de chaque méthode face à des transformations géométriques croissantes (changement de point de vue ou d'échelle et rotation) en utilisant les images de la figure 4.7. Puis, dans un second temps, nous évaluons la performance de chacune des méthodes en présence de structures répétitives avec les images de la figure 4.11.

### Robustesse aux transformations géométriques

La robustesse face à des transformations géométriques est évaluée en utilisant les images de la figure 4.7. Les tableaux 4.4, 4.5, 4.6 et 4.7 présentent les résultats obtenus avec les quatres séquences d'images. On note que les performances de chacune des méthodes de mise en correspondance dépend de la nature de la scène représentée par les images d'une part, et de l'importance de la transformation géométrique entre les images d'autre part.

D'une manière générale cependant, on note que la méthode de décomposition en valeurs singulières, SVD, donne de moins bons résultats que PPVRD. Cela peut, en partie, s'expliquer par le fait que l'algorithme de décomposition en valeurs singulières a des problèmes de stabilité lorsque les matrices sont de taille importante. Dans nos expériences, le nombre de points détectés dans chaque image est de l'ordre de $n \approx 10^3$ et nous utilisons l'algorithme de décomposition décrit dans Numerical Recipes [110]. Une raison plus importante de ces mauvais résultats est due à la manière dont les correspondants sont trouvés avec la méthode SVD. En effet, il n'est pas du tout évident que le fait de remplacer toutes les valeurs singulières de la matrice de proximité par 1 (voir la section 3.4.1, page 45) conduise à la sélection des points qui se correspondent. SVD donne presque toujours beaucoup plus de correspondants que PPVRD, mais avec une précision nettement inférieure.

On note également que la prise en compte de l'entropie fournit des résultats comparables à ceux obtenu avec SIFT seul (PPVRD). L'ajout de la couleur à la caractérisation locale donne des résultats peu satisfaisants. Si la précision obtenue par SIFT+COLOR est dans la plupart des cas comparable à celle obtenue par PPVRD, SIFT+COLOR donne deux à trois fois moins de correspondants corrects. Ce qui peut s'expliquer par le fait que pour les images utilisées, notamment les séquences *Wall* et *Bark*, la couleur n'est pas discriminante. Presque tous les points dans chacune des paires d'images, ont la même couleur. L'addition de l'information couleur à la caractérisation locale augmente donc l'ambiguïté de la mise

| Numéro image | PPVRD | | SIFT+COLOR | | SVD | | ENTROPIE | | REINF | | RELAX | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N | p | N | p | N | p | N | p | N | p | N | p |
| 2 | 55 | 0.70 | 12 | 0.91 | 101 | 0.49 | 46 | 0.76 | 60 | 0.73 | 95 | 0.81 |
| 3 | 32 | 0.62 | 12 | 0 | 74 | 0.39 | 30 | 0.6 | 33 | 0.63 | 46 | 0.71 |
| 4 | 45 | 1 | 42 | 0.64 | 100 | 0.53 | 39 | 1 | 46 | 1 | 50 | 1 |
| 5 | 19 | 1 | 16 | 1 | 77 | 0.28 | 16 | 1 | 20 | 1 | 21 | 1 |
| 6 | 1 | 1 | 2 | 1 | 73 | 0.13 | 1 | 1 | 1 | 1 | 2 | 1 |

TAB. 4.4 – Comparaison des différents algorithmes avec la séquence *Bark* (changement d'échelle + rotation, scène texturée). $N = \#correspondants$ et $p = precision$.

| Numéro image | PPVRD | | SIFT+COLOR | | SVD | | ENTROPIE | | REINF | | RELAX | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N | p | N | p | N | p | N | p | N | p | N | p |
| 2 | 429 | 0.91 | - | - | 562 | 0.76 | 403 | 0.91 | 456 | 0.92 | 638 | 0.97 |
| 3 | 299 | 0.91 | - | - | 476 | 0.79 | 286 | 0.91 | 306 | 0.91 | 511 | 0.94 |
| 4 | 98 | 87 | - | - | 193 | 0.67 | 89 | 0.88 | 95 | 0.87 | 113 | 0.93 |
| 5 | 76 | 0.97 | - | - | 119 | 0.61 | 74 | 0.97 | 82 | 0.97 | 94 | 0.98 |
| 6 | 18 | 0.66 | - | - | 90 | 0.28 | 16 | 0.68 | 17 | 0.70 | 15 | 0.8 |

TAB. 4.5 – Comparaison des différents algorithmes avec la séquence *Boat* (changement d'échelle + rotation, scène structurée). $N = \#correspondants$ et $p = precision$. La méthode SIFT+COLOR n'est pas évaluée car les images sont en niveaux de gris.

en correspondance, et réduit le nombre de correspondants trouvés. Ce qui fait baisser le rappel de la méthode.

D'autre part, SIFT donne de très bons résultats pour les images utilisées car les changements géométriques observés sont de nature affine. SIFT étant conçu pour être invariant aux transformations affines, on obtient de bons résultats avec PPVRD.

Malgré la bonne performance de SIFT seul, nous notons que des gains substantiels de performance sont obtenus par les méthodes de relaxation (RELAX) et de renforcement des scores (REINF) pour toutes les paires d'images. Ces deux méthodes permettent d'obtenir plus de correspondants corrects que PPVRD tout en ayant une précision élevée.

En moyenne, on obtient entre 20% et 50% de correspondants corrects en plus avec RELAX et entre 5% et 30% de correspondants corrects en plus avec REINF. Toutefois, la performance de chaque méthode dépend de la nature de la scène et des transformations géométriques entre les images.

- *Types de scènes*

Concernant le type la scène, on constate que l'écart de performance entre la relaxation et

| Numéro image | PPVRD | | SIFT+COLOR | | SVD | | ENTROPIE | | REINF | | RELAX | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N | p | N | p | N | p | N | p | N | p | N | p |
| 2 | 385 | 0.89 | 140 | 0.92 | 459 | 0.76 | 350 | 0.88 | 387 | 0.88 | 551 | 0.94 |
| 3 | 178 | 0.56 | 66 | 0.54 | 316 | 0.5 | 157 | 0.58 | 195 | 0.56 | 176 | 0.74 |
| 4 | 72 | 0.55 | 30 | 0.56 | 229 | 0.32 | 68 | 0.55 | 78 | 0.6 | 109 | 0.64 |
| 5 | 16 | 0.18 | 14 | 0.28 | 132 | 0.10 | 12 | 0.08 | 16 | 0.18 | 25 | 0.4 |
| 6 | 29 | 0.03 | 10 | 0.1 | 97 | 0.02 | 24 | 0.04 | 31 | 0.03 | 24 | 0 |

TAB. 4.6 – Comparaison des différents algorithmes avec la séquence *Grafitti* (changement de point de vue, scène structurée). $N = \#correspondants$ et $p = precision$.

| Numéro image | PPVRD | | SIFT+COLOR | | SVD | | ENTROPIE | | REINF | | RELAX | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N | p | N | p | N | p | N | p | N | p | N | p |
| 2 | 387 | 0.72 | 138 | 0.81 | 489 | 0.63 | 355 | 0.73 | 408 | 0.73 | 640 | 0.81 |
| 3 | 219 | 0.85 | 59 | 0.84 | 384 | 0.61 | 192 | 0.85 | 234 | 0.84 | 360 | 0.85 |
| 4 | 120 | 0.52 | 34 | 0.5 | 273 | 0.35 | 112 | 0.51 | 127 | 0.51 | 185 | 0.61 |
| 5 | 36 | 0.61 | 8 | 0.75 | 200 | 0.28 | 30 | 0.63 | 39 | 0.64 | 57 | 0.62 |
| 6 | 2 | 0 | 0 | 0 | 111 | 0.08 | 1 | 0 | 2 | 0 | 10 | 0.2 |

TAB. 4.7 – Comparaison des différents algorithmes avec la séquence *Wall* (changement de point de vue, scène texturée). $N = \#correspondants$ et $p = precision$.

le renforcement des scores par rapport à PPVRD est faible pour des scène texturées. En revanche, la relaxation améliore les résultats de manière notable pour des scènes structurées.

Notons aussi que pour les scènes texturées, la mise en correspondance avec SIFT seul donne de très bons résultats, et le gain en performance obtenu par REINF et RELAX est faible. Dans le cas de scènes structurées au contraire, le gain en performance obtenu par la prise en compte de l'information contextuelle est significatif (en terme de rappel notamment). Cela est dû au fait que l'information de gradient, capturée localement par le descripteur SIFT, est plus importante dans des scène texturées que dans des scènes structurées.

- *Types de transformation*

En ce qui concerne le type de transformation, on constate que toutes les méthodes (RELAX, REINF, ENTROPIE, PPVRD, SVD et SIFT+COLOR) obtiennent un nombre de correspondants et une précision plus importantes dans le cas d'un changement d'échelle et d'une rotation (paires d'images *Boat* et *Bark*), que dans le cas d'un changement de point de vue (paires d'images *Grafitti* et *Wall*). Ceci s'explique par le fait que le descripteur utilisé, SIFT, est plus adapté aux rotations et aux changements d'échelle qu'à des changements

de point de vue. La même observation est faite par Mikolajczyk et Schmid [92].

Pour des scènes avec un changement de point de vue important, la performance de SIFT est très limité, i.e. la répétabilité du détecteur Harris-Affine diminue. Pour cette raison, il est difficile d'établir des correspondances correctes entre deux images de la même scène. Notons par exemple qu'il y a un changement de point de vue de près de 50° entre la première image et la cinquième image de la séquence *Grafitti*. La figure 4.10 montre l'évolution de la précision et du rappel en fonction de l'importance de la transformation pour les séquences *Boat* et *Grafitti*.

### Robustesse en présence de structures répétitives

Pour évaluer la performance des différentes méthodes en présence de structures répétitives, nous utilisons les images de la figure 4.11. Dans ces cas, la mise en correspondance est difficile parce que tous les points sont décrits presque de la même manière par SIFT.

Nous avons vu au chapitre 3, voir page 46, que l'on obtient de meilleurs résultats avec les méthodes qui tiennent compte du contexte dans la phase d'appariement, contrairement à celles qui utilisent le contexte dans la phase de description. Cette observation est ici confirmée par les résultats rassemblés dans les tableaux 4.8.

Pour des scènes structurées présentant des structures répétitives (paires d'images *Eerie* et *Clavier* de la figure 4.11), la performance de la relaxation dépasse largement celles des autres méthodes avec un rappel et une précision nettement plus élevés. Les deux meilleures performances sont obtenues par RELAX et REINF. Cependant, RELAX fournit environ 3 fois plus de correspondants corrects que REINF avec la séquence *Eerie*, et environ 4 fois plus de correspondants corrects avec la séquence *Clavier*.

Notons que pour ces deux exemples, le nombre de correspondants corrects obtenu par SIFT+COLOR, PPVRD et ENTROPIE est beaucoup trop faible. SVD fournit un nombre élevé de correspondants corrects, comparable à celui obtenu par RELAX, mais avec une précision très faible, inférieure à 50%.

Dans le cas de scènes texturées (paires d'images *Arènes* et *Batiment* de la figure 4.11), il y a plus de points d'intérêt détectés dans les deux images et le descripteur local SIFT est plus riche en information. Toutes les méthodes permettent donc d'obtenir une précision élevée. La méthode de relaxation permet d'obtenir une légère amélioration de la précision, mais elle fournit plus de correspondants que les autres améthodes. Le rappel est donc sensiblement amélioré (de l'ordre de 50% avec les paires d'images *Arènes* et *Batiment*).

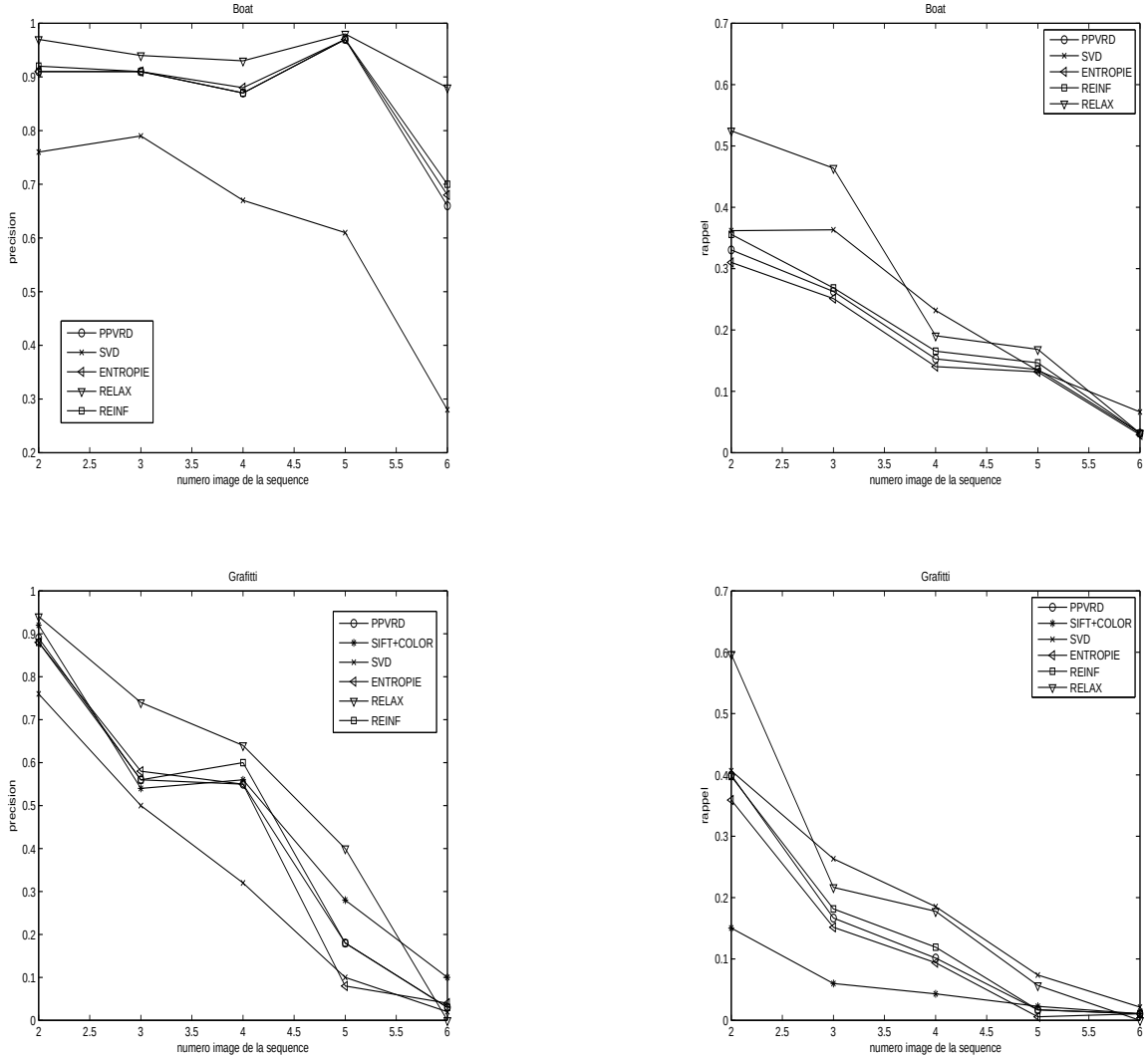



FIG. 4.10 – Evolution de la précision et du rappel avec la transformation géométrique. En haut : dans le cas d'un changement d'échelle et d'une rotation (la séquence **Boat**) ; en bas : dans le cas d'un changement de point de vue (séquence **Grafitti**).

Fig. 4.11 – Images de structures répétitives. De haut en bas : séquence **Eerie**, séquence **Clavier**, séquence **Arènes** et séquence **Batiment**.

| Images | Méthode | # appariements | # appariements corrects | précision | rappel | temps en s |
|---|---|---|---|---|---|---|
| Eerie | PPVRD | 6 | 3 | 0.5 | 0.064 | 0.165 |
| | SIFT+COLOR | 6 | 3 | 0.50 | 0.064 | 0.997 |
| | ENTROPIE | 4 | 3 | 0.75 | 0.064 | 0.174 |
| | REINF | 16 | 8 | 0.5 | 0.17 | 0.205 |
| | SVD | 60 | 21 | 0.35 | 0.45 | 1.63 |
| | RELAX | 38 | 25 | 0.66 | 0.53 | 1.364 |
| Clavier | PPVRD | 18 | 8 | 0.44 | 0.1 | 1.46 |
| | SIFT+COLOR | 7 | 3 | 0.42 | 0.03 | 2.89 |
| | ENTROPIE | 16 | 8 | 0.5 | 0.1 | 1.53 |
| | REINF | 18 | 8 | 0.44 | 0.1 | 2.17 |
| | SVD | 129 | 43 | 0.34 | 0.53 | 68.78 |
| | RELAX | 40 | 30 | 0.75 | 0.37 | 3.56 |
| Arènes | PPVRD | 353 | 339 | 0.94 | 0.48 | 2.18 |
| | SIFT+COLOR | 170 | 164 | 0.96 | 0.23 | 3.66 |
| | ENTROPIE | 328 | 314 | 0.95 | 0.44 | 2.30 |
| | REINF | 347 | 333 | 0.96 | 0.47 | 3.34 |
| | SVD | 471 | 406 | 0.86 | 0.57 | 135.27 |
| | RELAX | 568 | 560 | 0.98 | 0.79 | 5.22 |
| Batiment | PPVRD | 276 | 252 | 0.91 | 0.44 | 1.72 |
| | SIFT+COLOR | 93 | 77 | 0.82 | 0.16 | 3.00 |
| | ENTROPIE | 243 | 224 | 0.92 | 0.39 | 1.80 |
| | REINF | 300 | 277 | 0.92 | 0.48 | 2.61 |
| | SVD | 360 | 290 | 0.8 | 0.51 | 93.66 |
| | RELAX | 420 | 414 | 0.98 | 0.72 | 5.82 |

TAB. 4.8 – Comparaison des différents algorithmes dans le cas de structures répétitives, en utilisant les images de la figure 4.11. Pour chaque paire d'images, la précision maximale et le rappel maximal sont soulignés.

### Courbes de précision-rappel

Nous présentons ici les courbes de précision-rappel obtenues en mettant en correspondance pour chacune des séquences la première image avec la quatrième image, et en faisant varier le seuil de détection.

Les résultats comparatifs obtenus sont présentés par les courbes des figures 4.12 et 4.13. Soulignons qu'il n'y a pas de courbe représentant les résultats de la méthode SIFT+COLOR pour la paire d'images *Boat* (voir figure 4.12), parce que celle-ci est une paire d'images en niveau de gris. Rappelons qu'une précision élevée traduit la confiance que l'on peut accorder à la méthode car la plupart des appariements trouvés sont corrects, et qu'un rappel élevé traduit le fait que la plupart des appariements corrects entre les deux images sont trouvés. Par conséquent, une méthode parfaite devrait fournir un rappel égal à 1 pour toutes les précisions.

Les résultats obtenus sont conformes à ceux décrit ci-dessus pour des transformations géométriques croissantes. La performance de chaque méthode dépend de la nature de la scène et du type de transformation, mais les meilleurs résultats sont obtenus par les méthodes RELAX et REINF. En moyenne, RELAX fournit un rappel supérieur de 40% à celui de PPVRD pour une précison égale à 0.7. Avec REINF, on obtient un rappel supérieur de 20% à celui obtenu par PPVRD pour la même précision. La performance de ENTROPIE est comparable à celle PPVRD et SVD et SIFT+COLOR donne des résultats inférieurs à ceux obtenus par PPVRD.

Les meilleurs résultats, rappel et précision élevés, sont obtenus avec les séquences *Bark* et *Boat* pour toutes les méthodes. Avec les séquences *Wall* et *Grafitti*, on obtient de très faibles rappels pour des précisions supérieures à 0.5. Cela confirme le fait que SIFT est plus adapté aux rotations et aux changements d'échelle qu'à des changements de point de vue.

### Remarques

Les résultats ci-dessus apportent la confirmation de l'observation effectuée au chapitre précédent. A savoir que la prise en compte de l'information contextuelle dans la phase d'appariement donne des résultats supérieurs par rapport à sa prise en compte dans la phase de description. C'est la raison pour laquelle les méthodes de relaxation et de renforcement sont celles qui donnent les meilleurs résultats. Toutefois, d'une manière générale, RELAX obtient des résultats supérieurs par rapport à REINF, notamment en terme de rappel.

Cette dernière méthode tente dans un premier temps, d'accroître les scores d'apparie-

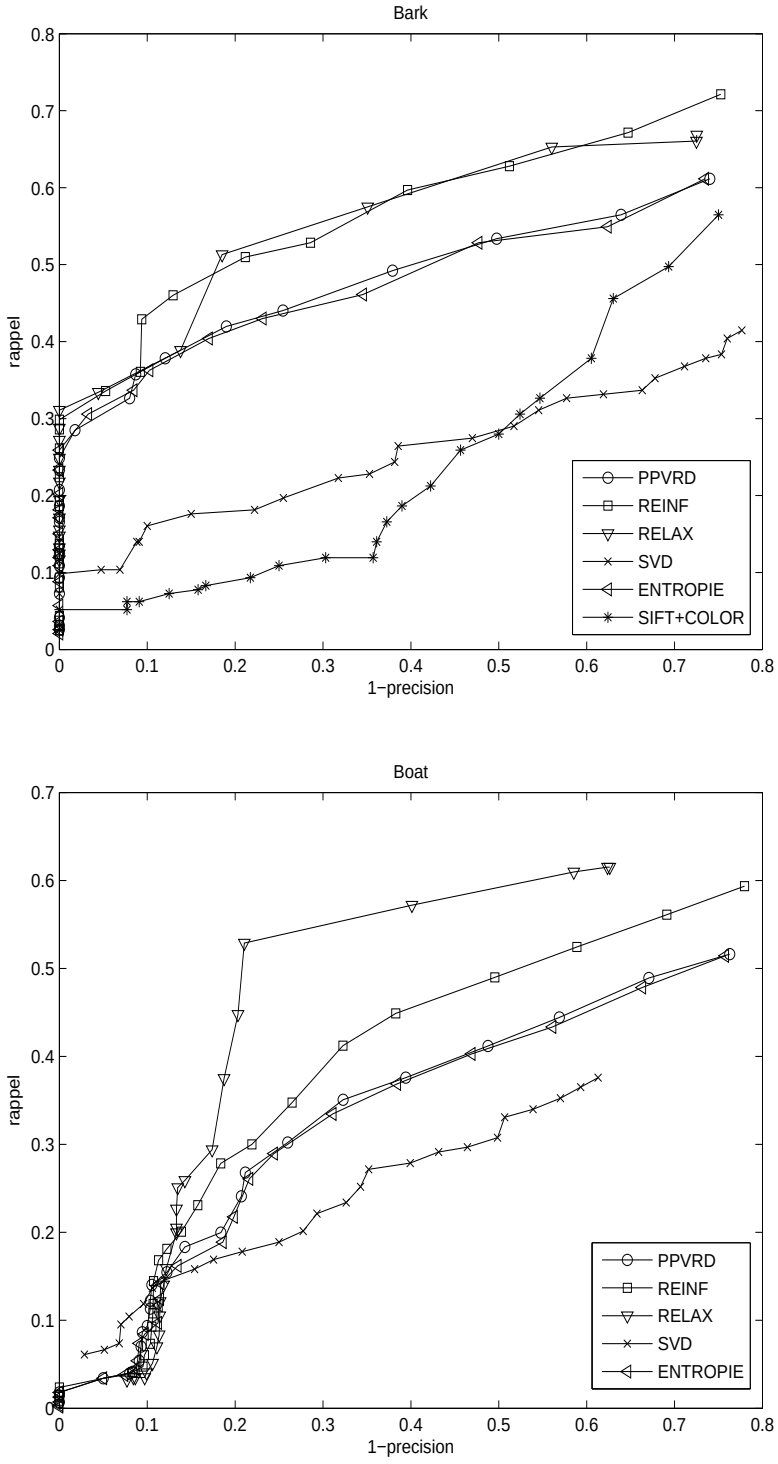



FIG. 4.12 – Courbes de précision-rappel avec les séquences **Bark** (en haut) et **Boat** (en bas).

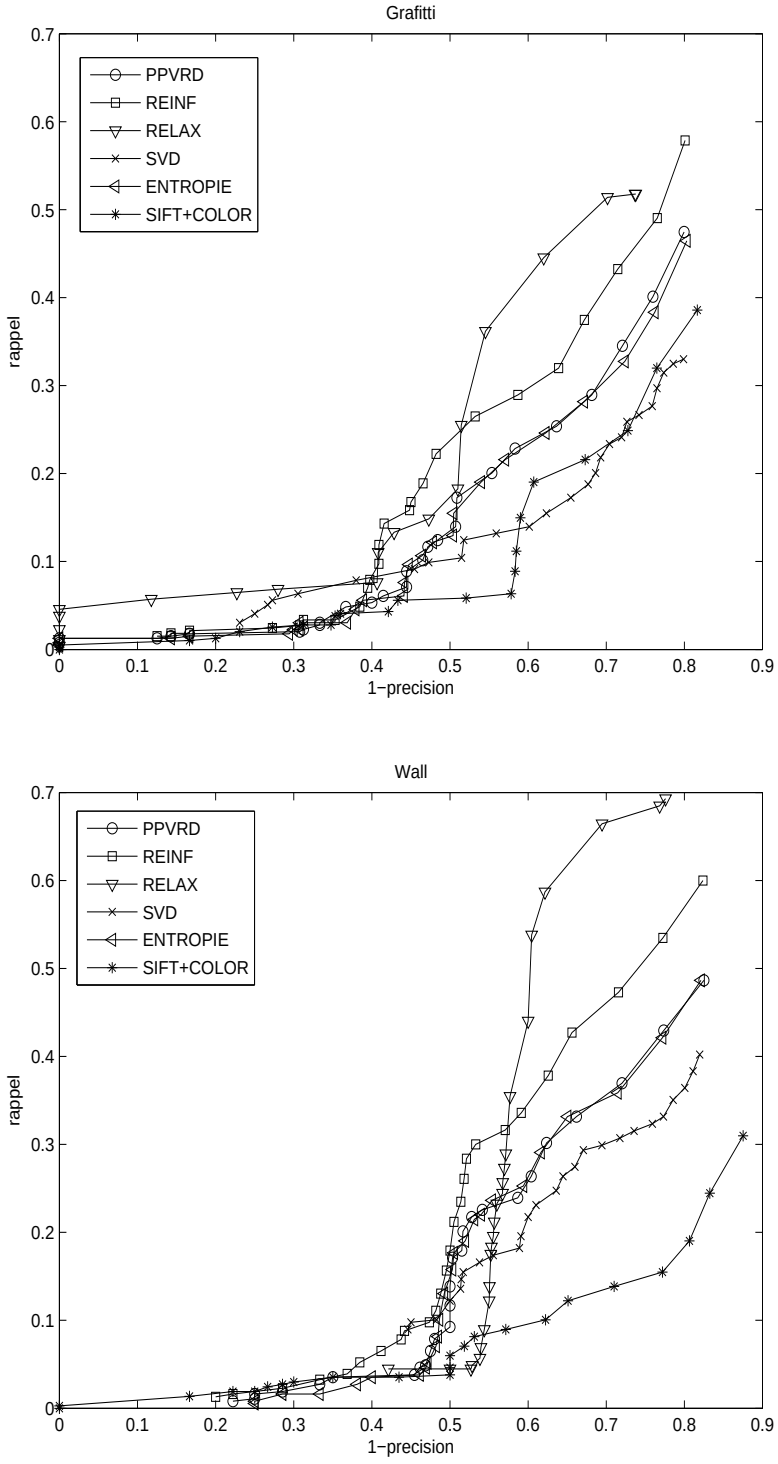



Fig. 4.13 – Courbes de précision-rappel avec les séquences **Grafitti** (en haut) et **Wall** (en bas).

| | SIFT+COLOR | ENTROPIE | REINF | SVD | RELAX |
|---|---|---|---|---|---|
| précision | $\approx$ | $\approx$ | $\approx$ | $-$ | $+$ |
| rappel | $-$ | $\approx$ | $+$ | $+$ | $+$ |

TAB. 4.9 – Comparaison des différentes méthodes de mise en correspondance. Le signe $+$ indique une amélioration par rapport à l'approche PPVRD, $-$ indique une moins bonne performance et $\approx$ indique des performances comparables.

ment en se basant sur la répartition spatiale des points d'ancrage (voir section 3.3.2). Les correspondants sont ensuite trouvés par une méthode du plus proche voisin avec rapport de distances. Si les points d'ancrage ne sont pas corrects, la mise à jour des scores d'appariement conduira à de faux appariements. La sélection des points d'ancrage est basée sur les distances euclidiennes entre les descripteurs SIFT et nous avons vu que SIFT n'est pas robuste à des déformations importantes.

La relaxation, quant à elle, augmente la probabilité d'appariement d'un point en considérant la configuration de son voisinage. Dans la méthode proposée, si l'appariement d'un point n'est pas compatible avec ceux de ses voisins, alors la probabilité de cet appariement diminue et il finit par être écarté.

La bonne performance de REINF dans le cas de scènes texturées, s'explique par le fait que dans ce cas, il y a plus de points d'intérêt détectés dans les images. Par conséquent, les points d'ancrage sont mieux distribués, ce qui n'est pas le cas pour des scènes structurées.

Il nous semble que la supériorité de RELAX par rapport à REINF, particulièrement dans le cas de structures répétitives, est certainement due à l'utilisation de l'information couleur dans la relaxation. En effet, SIFT est basé uniquement sur la géométrie de la scène. Il est donc intéressant de prendre en compte une information photométrique complémentaire. Nous avons cependant vu que le simple ajout de l'information photométrique à la caractérisation locale est insufisante car elle réduit le nombre d'appariements trouvés dans la plupart des cas. La couleur doit être utilisée en plus de SIFT dans la phase d'appariement à travers les relations entre points voisins.

Enfin, pour résumer l'ensemble des résultats obtenus dans ces expériences, nous présentons dans le tableau 4.9 la manière dont chacune des méthodes se situe globalement par rapport à celle qui nous sert de référence, PPVRD. Un signe $+$ indique que la méthode apporte une amélioration par rapport à PPVRD, un signe $-$ qu'elle donne une moins bonne performance que PPVRD, et le signe $\approx$ indique que les deux méthodes ont des performances comparables. Comme on peut le noter, notre méthode de relaxation améliore à la fois, le rappel et la précision de la mise en correspondance.

### 4.4.5 Stabilité de l'algorithme

Les résultats de la mise en correspondance dépendent des différents paramètres de l'algorithme de relaxation, en particulier de $\alpha$ qui définit l'influence de chacun des termes de cohérence et d'ambiguïté, et de la taille $V$ du voisinage de chaque point. Nous prenons comme cas d'étude, pour mesurer l'influence de ces paramètres, la paire d'images *Eerie* de la figure 4.11 présentant des stuctures répétitives.

#### Influence du paramètre $\alpha$

Les résultats du tableau 4.10 montrent l'influence du paramètre $\alpha$ sur les résulats. On note que si on accorde plus d'importance au terme de cohérence, $\alpha > 0.5$, alors la précision augmente tandis que le nombre d'appariements diminue, ce qui a pour conséquence de diminuer le rappel. A l'inverse, si on accorde plus d'importance au terme d'ambiguïté, $\alpha < 0.5$, alors le nombre d'appariements augmente, et donc le rappel, mais la précision diminue.

Ces résultats sont conformes à l'expérience, i.e. le rappel augmente quand la précision diminue. La baisse de la précision lorsque $\alpha < 0.5$ traduit le fait que l'information contextuelle est prise en compte, principalement, dans le terme de cohérence du critère.

Rappelons que le terme de cohérence s'écrit :

$$C_1 = \frac{1}{2n} \sum_{i=1}^{n} \|p_i - q_i\|^2$$

où, $p_i$ désigne le vecteur de probabilité et $q_i$ le vecteur de compatibilté qui est lui-même obtenu grâce à l'information contextuelle représentée par les probabilités conditionnelles $p_{ij}(k,l)$.

#### Influence de la taille du voisinage

Le tableau 4.11 montre que les résultats sont relativement stables quand la taille du voisinage $V$ varie. On pourrait s'attendre à ce qu'une grande valeur de $V$ conduise à des résultats plus précis. En fait, si le rappel augmente avec $V$, la précision ne varie que très peu en fonction de $V$. Cette dernière quantité est donc plus liée au paramètre $\alpha$ qu'à la taille du voisinage.

Une augentation de la valeur de $V$ se traduit toutefois par une complexité plus élevée. Le temps d'exécution passe de 1.33 s pour $V = 5$, à 5.15 s pour $V = 15$, tandis que le

| $\alpha$ | # appariements | # appariements corrects | précision | rappel |
|---|---|---|---|---|
| 0.3 | 88 | 42 | 0.48 | 0.89 |
| 0.5 | 38 | 25 | 0.66 | 0.53 |
| 0.7 | 23 | 17 | 0.74 | 0.36 |
| 0.9 | 15 | 13 | 0.87 | 0.28 |

TAB. 4.10 – Influence du paramètre $\alpha$ : exemple de la paire d'images *Eerie* de la figure 4.11.

| # V | # appariements | # appariements corrects | précision | rappel | temps en s |
|---|---|---|---|---|---|
| 3 | 35 | 23 | 0.66 | 0.49 | 0.98 |
| 5 | 38 | 25 | 0.66 | 0.53 | 1.33 |
| 7 | 36 | 25 | 0.69 | 0.53 | 1.82 |
| 10 | 41 | 27 | 0.66 | 0.57 | 2.79 |
| 15 | 48 | 30 | 0.62 | 0.64 | 5.15 |

TAB. 4.11 – Influence de la taille du voisinage : exemple de la paire d'images *Eerie* de la figure 4.11.

nombre d'appariements corrects passe de 25 à 30. Le gain de performance est donc faible par rapport à la complexité plus élevée.

### Remarques

Les résultats obtenus en faisant varier les paramètres de l'algorithmes indiquent qu'il faut trouver un compromis entre le rappel et la précision de la méthode. Pour le couple d'images utilisé, les paramètres "optimaux", i.e. ceux qui donnent le meilleur compromis entre le rappel et la précision, sont $\alpha^* = 0.45$ et $V^* = 14$. Ce sont les points d'intersections des courbes de rappel et de précision de la figure 4.14.

Dans la pratique, il est impossible d'obtenir ces paramètres "optimaux" pour chaque paire d'images, car il faut apparier plusieurs fois les images pour différentes valeurs de $\alpha$ et de $V$.

Nous avons utilisé pour nos expériences, les valeurs $\alpha = 5$ et $V = 5$ qui donnent des résultats satisfaisants. Ces valeurs ont été choisies pour un besoin de rapidité (le temps de calcul étant lié à $V$).

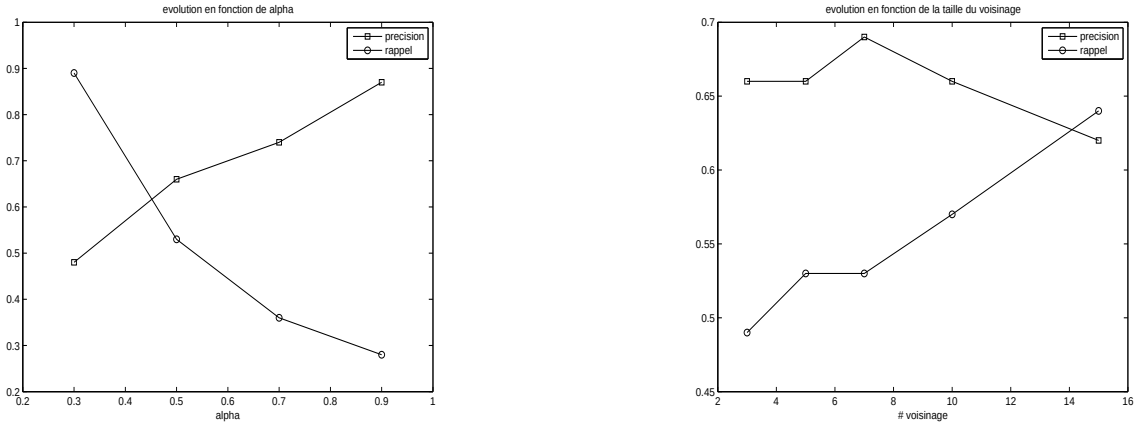



Fig. 4.14 – A gauche, influence du paramètre $\alpha$ ; A droite, influence de la taille du voisinage.

## 4.5 Conclusion

Dans ce chapitre, nous avons proposé une méthode de mise en correspondance rapide et robuste. La méthode est basée sur une technique de relaxation et l'optimisation d'un critère qui tient compte à la fois de la cohérence et de l'ambiguïté des appariements. Nous montrons qu'en écrivant le critère sous une forme convenable, la complexité de l'algorithme peut être réduite de manière significative. De plus, nous proposons deux manières différentes de calculer l'information contextuelle nécessaire pour réduire l'ambiguïté ainsi que le nombre de faux appariements.

Les résultats expérimentaux obtenus indiquent que notre approche donne des résultats supérieurs ou comparables, en terme de précision et de rappel, à ceux obtenus avec les diverses méthodes présentées au chapitre précédent. En particulier, l'utilisation de l'information colorimétrique pour le calcul des probabiltés conditionnelles, permet d'obtenir des résultats satisfaisants dans le cas difficile de structures répétitives. Cas dans lequel la plupart des autres méthodes échouent. Notre méthode permet d'obtenir un rappel et une précision compatibles avec les méthodes d'estimation de la transformation géométrique entre deux images, par exemple RANSAC. Rappelons que RANSAC échoue si la proportion de faux appariements est supérieure à 50% [80, 23], ce qui est le cas avec les autres méthodes évaluées dans ce chapitre.

Dans le chapitre suivant, nous appliquons notre méthode de mise en correspondance au problème de la reconnaissance d'objets dans des scènes complexes.

# Chapitre 5

# Application à la reconnaissance d'objets

Dans ce chapitre, nous abordons le problème de la reconnaissance de formes, ou d'objets, basée sur la mise en correspondance de primitives. Après une brève description du problème de la reconnaissance d'objets et des principales approches, nous évaluons les performances de différentes méthodes de mise en correspondance des invariants locaux dans le cadre de deux exemples d'application : la recherche d'un objet dans une base d'images, et la détection d'un objet dans une scène complexe.

## 5.1 Introduction

Le problème de la reconnaissance d'objets (RO) en vision par ordinateur peut être simplement formulé de la manière suivante :

*Etant donné une ou plusieurs images d'un objet (définissant le modèle), déterminer si celui-ci est présent dans une nouvelle image (image de la scène).*

Si la réponse est positive, on doit pouvoir déterminer la position de l'objet dans la scène.

La formulation très simple du problème, masque la difficulté de la tâche dans la plupart des cas pratiques. Les principales difficultés sont dues aux occultations, aux changements de point de vue ainsi qu'aux variations des conditions de prise de vue des images.

D'une manière générale, on divise les méthodes de la RO en deux familles d'approches : l'approche basée sur les modèles (model-based approach) et celle basée sur l'apparence (appearance-based approach).

La première, l'approche basée sur les modèles, nécessite une modélisation 3D explicite de la forme de l'objet, des différentes parties de l'objet et des relations entre celles-ci. La


a)

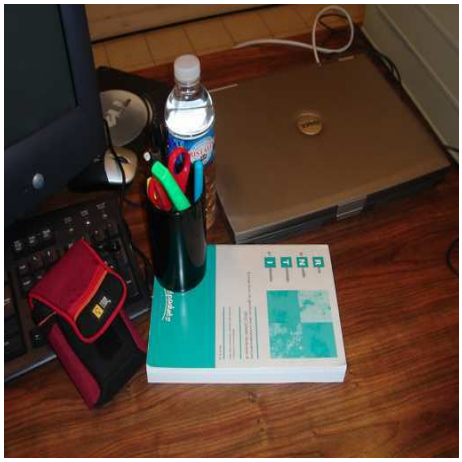

b)

Fig. 5.1 – Formulation du problème de la reconnaissance d'objets : le livre (a) est-il présent dans la scène (b) ?

reconnaissance revient alors à identifier une projection du modèle dans une image de la scène. On peut soit extraire de l'image des informations tri-dimensionnelles (par exemple la forme par des techniques de type *shape from X*, où $X$ désigne l'ombre, la texture ou les contours) et les comparer à la description du modèle, soit extraire des primitives 2D (courbes, segments, jonctions, etc) de l'image et les comparer à une projection 2D du modèle [77, 129, 78, 10]. La plupart du temps, les relations spatiales entre les parties de l'objet sont représentées sous la forme d'un graphe et le problème de reconnaissance se ramène à celui d'isomorphisme de graphes (graph matching) [55].

Un exemple de méthode basée sur les modèles est le système proposé par Biederman [13, 12]. L'auteur représente un objet ou une scène sous la forme d'un arrangement de primitives volumétriques appelées *geons*. Ces derniers sont obtenus à partir de la déformation d'un cylindre. Les contours détectés dans une image sont d'abord regroupés sous forme de *geons*, qui sont ensuite utilisés pour reconnaître l'objet.

La principale limitation des méthodes de cette approche est liée au fait qu'il est difficile, voire souvent impossible, de définir un modèle explicite pour les objets de forme complexes ou ceux qui sont déformables. Cette remarque, limite le champ d'application à certaines classes d'objets de forme simple telles que les objets polyhédriques. D'autre part, il est extremement difficile d'interpréter les primitives géométriques d'une image de la scène comme étant des projections d'un modèle 3D, en particulier lorsque l'on souhaite reconnaître plusieurs objets dans une même scène. A ces difficultés, s'ajoute une autre plus importante liée aux occultations. Une partie, plus ou moins importante, de l'objet peut

ne pas être visible dans la scène. Dans ce cas, l'identification du modèle d'objet devient impossible.

La philosophie des méthodes basées sur l'apparence est radicalement différente. En effet, l'approche basée sur l'apparence ne nécessite pas de modèle explicite de l'objet, mais utilise des images qui représentent l'objet selon différents angles ou points de vue. Chaque image décrit l'apparence de l'objet selon un point de vue particulier. Le modèle de l'objet est donc directement extrait de ses différentes images. Parce qu'elle n'utilise pas de connaissance à priori sur l'objet, cette approche peut s'appliquer à des formes très variées. Le terme apparence se refère ici aux caractéristiques de couleur, de texture et de forme de l'objet et à la manière dont celles-ci apparaissent dans l'image. Les méthodes de cette famille peuvent être classées dans deux catégories : les méthodes globales et les méthodes locales.

Les méthodes globales représentent l'objet en utilisant la totalité de l'information présente dans l'image. Les méthodes les plus utilisées sont les histogrammes couleur [142] et les représentations en composantes principales (*eigenimages*) [100]. La première méthode capture l'apparence d'un objet sous la forme de la distribution spatiale des couleurs présente dans l'image et la reconnaissance consiste à comparer les histogrammes du modèle et de l'image de la scène. La deuxième méthode capture l'apparence d'un objet sous la forme de composantes principales issues d'une ACP et l'image de la scène est projetée sur l'espace défini par ces composantes principales. Malheureusement, parce qu'elles utilisent l'information présente dans l'image entière, les méthodes globales sont très sensibles aux occultations et aux changements de fond dans l'image. Elles nécessitent souvent une présegmentation de l'image pour éliminer le fond. De plus, pour être efficace, il faut disposer d'un nombre important d'images de référence, pour capturer l'apparence de l'objet sous différents points de vue, particulièrement dans le cas de l'ACP. Les méthodes globales sont donc la plupart du temps utilisées dans des environements controlés, i.e. des environements avec peux d'objets et de légères variations de point de vue, de forme, d'échelle et d'illumination.

Les méthodes locales représentent l'objet sous la forme d'une collection de petites primitives locales. Ces primitives peuvent être des points ou régions d'intérêt, des segments, etc. Parce qu'elles sont basées sur des primitives locales, ces méthodes sont plus robustes aux occultations car les parties non visibles de l'objet, n'affectent pas celles qui le sont. De plus, un changement du fond de l'image, par l'apparition de nouveaux objets par exemple, induit simplement des primitives supplémentaires sans affecter celles détectées sur l'objet. La reconnaissance se fait en comparant les primitives de l'objet à celles de l'image de la

scène et en identifiant les primitives qui se *correspondent*. Cette étape de mise en correspondance est cruciale, car c'est elle qui détermine la présence ou non de l'objet dans une scène. Concrêtement, plus il y a de correspondants corrects, plus la décision prise quant à la présence de l'objet est renforcée.

Du fait de la robustesse aux occultations et de l'invariance aux variations des conditions de prise de vue, les méthodes locales se sont imposées ces dernières années comme une alternative efficace pour le problème de la reconnaissance d'objets et plus généralement pour celui de la mise en correspondance d'images depuis les travaux de Schmid [123, 121] utilisant les points d'intérêt dans le cadre de l'indexation d'images.

## 5.2 L'utilisation des invariants locaux

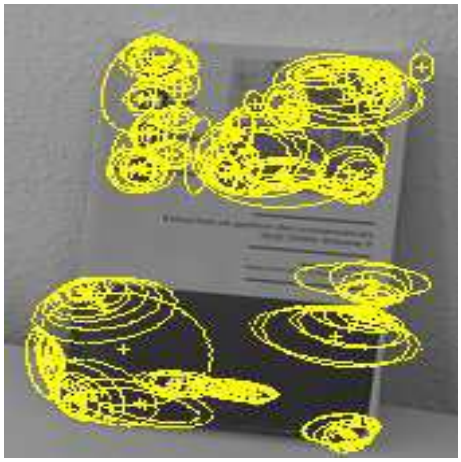
Les travaux précurseurs de Schmid [123, 121] ont montré que les invariants locaux peuvent être une solution efficace pour le problème de la reconnaissance d'objets. Toutefois, il faut pouvoir établir des correspondances entre les primitives de l'image représentant le modèle et celles de l'image représentant la scène. La reconnaissance passe donc par la mise en correspondance d'images. Plus on trouve de correspondants, plus la présence de l'objet est évidente, de même que sa localisation. A condition, que les correspondants trouvés soient corrects. Le nombre de correspondants corrects est donc un bon critère de détection.

Ici, nous distinguerons deux problèmes souvent confondus dans la littérature sous la même appelation de *reconnaissance d'objets* (ou *object recognition* en anglais) :

- la détection d'objets : qui consiste à identifier et à localiser un ou plusieurs objets dans une scène complexe.
- l'indexation d'images : qui consiste à rechercher un objet particulier dans une base d'images.

Dans le second problème, l'indexation, chaque objet est représenté par une image dans une base de données. Etant donné une image représentant un objet quelconque, l'objectif est de trouver l'image de la base qui représente le même objet. En général, chaque image ne contient qu'un seul objet avec un fond fixe.

Dans le premier problème au contraire, on souhaite identifier un objet, représenté par une image, dans une scène complexe qui peut contenir plusieurs autres objets. Ce dernier problème est plus difficile dans la mesure où on doit pouvoir apparier l'image représentant l'objet avec une partie, relativement petite, de l'image de la scène contenant l'objet. Il peut donc y avoir des occultations, et il y a un nombre restreint de primitives de l'objet dans


a)

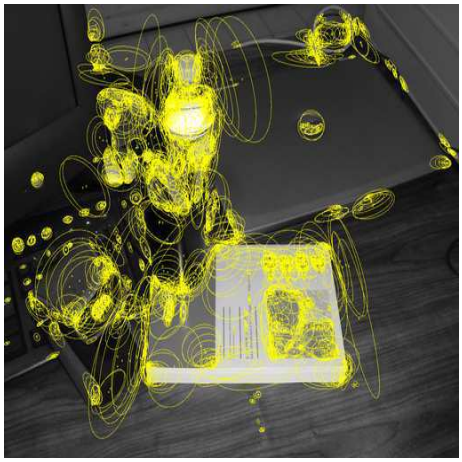

b)

FIG. 5.2 – Détection de points d'intérêt à l'aide du détecteur Harris-Affine.

l'image de la scène parmis un nombre relativement important d'autres primitives.

Notons toutefois que cette distinction ne vaut que dans le cas de requêtes globales. Dans le cas de requêtes partielles, on a les mêmes contraintes que dans le cas de la détection d'objets.

Dans la section 5.3, nous présentons des résultats obtenus dans chacune de ces deux applications avec différentes méthodes de mise en correspondance.

### 5.2.1 Reconnaissance

Prenons comme exemple les deux images de la figure 5.1. Avec le détecteur Harris-Affine, nous avons respectivement 98 points d'intérêt détectés sur le livre, figure 5.2(a), et 1286 points détectés dans la scène entière, figure 5.2(b).

La méthode de mise en correspondance, doit être capable d'identifier un petit nombre de correspondants corrects parmi un nombre important de primitives. En d'autres termes, la précision et le rappel de la méthode doivent être élevés.

Dans le chapitre précédent, nous avons proposé une méthode de mise en correspondance, RELAX, capable de trouver un nombre suffisant d'appariements corrects dans des cas difficiles. Dans le cas du couple d'images de la figure 5.1, RELAX permet d'obtenir 20 appariements tous corrects. La précision de la méthode, ici égale à 1, permet de s'assurer de la présence de l'objet et d'estimer sa position avec précision. La figure 5.3 montre les appariements obtenus avec RELAX et le tableau 5.1 récapitule les résultats obtenus avec les méthodes SIFT+COLOR, PPVRD, ENTROPIE, SVD et REINF. Comme on peut le voir sur ce tableau, toutes les méthodes à l'exception de SVD et SIFT+COLOR

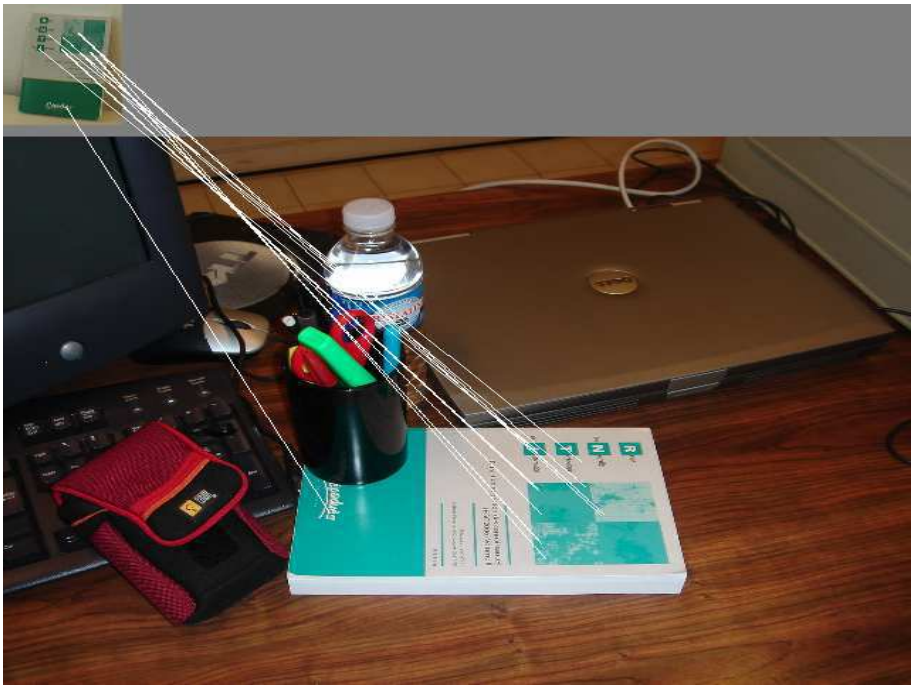


Fig. 5.3 – Exemple de reconnaissance d'objets avec RELAX.

permettent d'obtenir 100% d'appariements corrects. Cependant, RELAX fournit environ trois fois plus d'appariements que PPVRD, ENTROPIE et REINF. SVD permet d'obtenir 18 appariements, mais il y a 33,3% d'appariements incorrects. Notons enfin que la prise en compte de la couleur dans la caractérisation locale, SIFT+COLOR, ne fournit aucun appariement à cause de l'ambiguïté élevée.

| Méthode | # appariements | # appariements corrects | temps en s |
|---|---|---|---|
| PPVRD | 7 | 7 | 1.719 |
| SIFT+COLOR | 0 | 0 | 4.519 |
| ENTROPIE | 7 | 7 | 1.817 |
| REINF | 7 | 7 | 2.179 |
| SVD | 18 | 12 | 657.967 |
| RELAX | 20 | 20 | 15.129 |

Tab. 5.1 – Comparaison de différents algorithmes avec le couple d'image de la figure 5.1.

### 5.2.2 Localisation

Pour déterminer la pose, i.e. la position, l'orientation et la taille, de l'objet dans la scène, on adopte une approche de type prédiction-vérification. Dans un premier temps, on cherche une pose probable de l'objet dans la scène en utilisant la transformée de Hough [7, 56]. On définit un système de vote à quatre paramètres : deux paramètres de position, un paramètre d'orientation et un paramètre d'échelle. Chaque point d'intérêt vote dans cet espace de dimension 4 et on considère les points d'accumulation de cet espace comme représentant des poses possibles de l'objet.

Chaque point d'accumulation qui contient au moins 3 votes constitue une pose possible qui est vérifiée par l'estimation de la transformation affine reliant l'image de l'objet à celle de la scène. Cette transformation, comme on l'a vu au chapitre 2 (voir page 17), est en fait une approximation de la réelle homographie qui relie les deux images.

Un point $(x, y)^T$ du modèle est transformé en un point $(u, v)^T$ de la scène par l'équation :

$$\begin{pmatrix} u \\ v \end{pmatrix} = s \,.\, \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix} \,.\, \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{5.1}$$

où, $\theta$ désigne la rotation, $s$ le facteur d'échelle et $(t_x, t_y)^T$ le vecteur de translation.

Si on note $a = s \,.\, cos(\theta)$ et $b = s \,.\, sin(\theta)$, alors l'équation peut aussi s'écrire :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \,.\, \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

La détermination des paramètres de la transformation peut se faire par la résolution du système linéaire suivant :

$$\begin{pmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \\ & \cdots & & \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} u \\ v \\ \vdots \end{pmatrix}$$

Si l'on note $\mathbf{x}$ le vecteur des paramètres de la transformation, alors le système s'écrit :

$$\mathbf{Ax} = \mathbf{b}$$

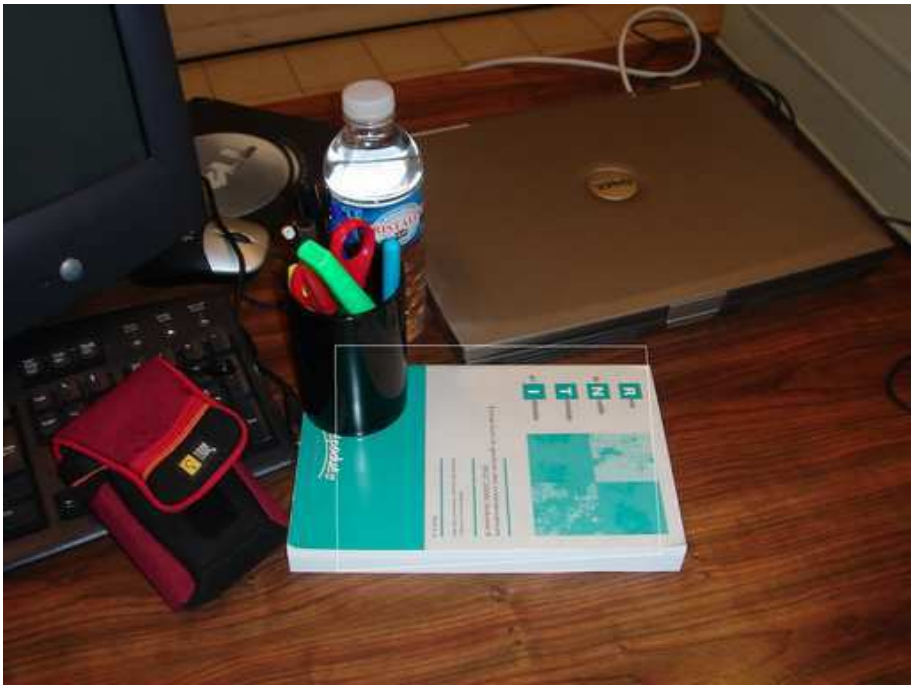


Fig. 5.4 – Détermination de la position de l'objet.

Et une solution est donnée par la méthode des moindres carrés :

$$\mathbf{x} = [\mathbf{A}^T\mathbf{A}]^{-1}\mathbf{A}^T\mathbf{b} \tag{5.2}$$

Il est important de souligner que plus il y a de correspondants corrects obtenus, plus l'estimation de la transformation affine est correcte. Encore une fois, la précision de la méthode de mise en correspondance est d'une importance cruciale.

La figure 5.4 montre la pose du livre estimée dans la scène en utilisant les appariements obtenus avec la méthode de relaxation.

## 5.3 Evaluation expérimentale

Les résultats préliminaires, comme ceux présentés dans la section précédente, montrent que notre méthode de relaxation donne de bons résultats dans le cadre de la reconnaissance d'objets. Il nous faut toutefois évaluer la performance de l'algorithme avec une base d'images plus importante et la comparer à d'autre méthodes. C'est ce que nous faisons dans cette section. Nous nous intéressons dans un premier temps à la recherche d'objets dans une base d'images, puis dans un second temps à la reconnaissance d'objets dans des

scènes complexes.

### 5.3.1 Recherche d'objets dans une base d'images

#### Base d'images et critère d'évaluation

Pour cette expérience, nous utilisons les images de la base SOIL-47A (Surrey Object Image Library) [1]. La base est publiquement disponible et est utilisée dans de nombreux travaux [72, 4]. Elle comprend 47 objets (des objets d'utilisation quotidienne tels que des boîtes de céréales, des livres ou des jouets) et chaque objet est photographié sous 21 angles différents compris entre -90 et +90 degrés. Soit un total de 987 images.

La vue de face de chaque objet représente le modèle inclus dans une base d'images. Les autres vues de l'objet sont utilisées pour évaluer la performance des méthodes de reconnaissance. L'image de la figure 5.5(a) présente quelques modèles (vues de face) d'objets de la base, et l'image de la figure 5.5(b) présente les 20 vues d'un même objet utilisées pour les tests. Les images représentant les modèles d'objets ont une résolution de 720x576, tandis que les images tests ont une résolution de 360x288. Il y a donc un changement d'échelle de facteur égal à 2 entre les images à apparier. L'ensemble des objets de la base est présenté dans l'annexe D (figure D.1).

L'intérêt de cette base d'images est de permettre l'évaluation de la robustesse des méthodes de mise en correspondance par rapport à des changements de point de vue importants. Pour chaque angle de vue, la performance est évaluée en mettant en correspondance toutes les images acquises sous cet angle avec les images de la base de modèles. Pour chaque image test, les images de la base sont classées dans l'ordre décroissant du nombre de correspondants obtenus, i.e. la première image est celle qui donne le plus grand nombre de correspondants. La reconnaissance est jugée correcte si l'objet recherché se trouve parmis les $k$ premières images extraites de la base.

Nous nous limiterons dans nos expériences au cas où $k = 1$, i.e. l'objet recherché correspond à la première image extraite de la base, et au cas où $k = 3$, i.e. l'objet recherché se trouve parmis les 3 premières images extraites de la base. Le taux de reconnaissance pour un angle de vue est donc égal au rapport entre le nombre d'objets correctement extraits de la base, et le nombre total d'objets dans la base.

Nous évaluons les performances de six méthodes de mise en correspondance dans le cadre de cette application :

– RELAX : la méthode basée sur la relaxation décrite dans la chapitre 4 ;

- PPVRD : la méthode du plus proche voisin avec rapport de distance [79] ;
- SVD : la méthode de décomposition en valeurs singulières [28] ;
- SIFT+COLOR : l'ajout de la couleur à la description locale SIFT [155] ;
- ENTROPIE : calcul de la similarité basée sur l'entropie [166] ;
- REINF : le renforcement des scores d'appariement par régions de contexte [29].

La méthode PPVRD sert de méthode de référence.

### Résultats

Les résultats obtenus dans les deux séries d'expériences, $k = 1$ et $k = 3$, sont rassemblés dans les tableaux 5.2, 5.3, 5.4 et 5.5.

- *Premier cas : $k = 1$*

Dans le cas où on ne considère que la première image extraite de la base, $k = 1$, les performances obtenues sont assez faibles. Le meilleur taux moyen de reconnaissance est obtenu avec la méthode de relaxation et est de 42.86%. Cela est principalement dû au fait que pour des angles de vue élevés, i.e. supérieurs à 60°, très peu de points d'intérêt détectés sur la vue de face (le modèle) le sont sur l'image test. Il est donc difficile d'obtenir des correspondants corrects dans ces cas extrêmes. Pour des changements de point de vue plus faibles, les performances augmentent de manière considérable. On obtient ainsi une performance maximale de 89.36% avec RELAX pour un angle de vue de -9°. Globalement, et ce n'est pas une surprise, les meilleures performances sont obtenues pour chacune des méthodes, lorsque le changement de point de vue est faible. C'est ce que montrent les résultats rassemblés dans le tableau 5.3. Comme on peut le noter, le taux moyen de reconnaissance est nettement plus élevé quand on ne considère que les angles de vue inférieurs à 60° et à 20°. Les performances maximales dans ce cas sont respectivement de 67.77% et de 84.04% avec la méthode RELAX.

- *Deuxième cas : $k = 3$*

Si l'objet est recherché parmis les trois premières images extraites de la base, les performances obtenues sont, comme on pouvait s'y attendre, meilleures. Le taux moyen de reconnaissance maximal pour l'ensemble des 20 angles de vue est obtenu avec la méthode de relaxation et est de 56.36%. Comme dans le cas où $k = 1$, plus le changement de point de vue est imporatnt, moins les performances sont élevées. On notera que dans ce cas, les deux méthodes REALX et REINF obtiennent un taux de reconnaissance de 100% lorsque l'angle de vue est inférieur à 10°. Pour des angles de vue inférieurs à 60° et à 20°, les taux


a)

b)

FIG. 5.5 – (a) : Exemple d'objets de la base SOIL-47A. (b) : Les 20 vues d'un objet de la base SOIL-47A.

| Angle en degrés | Taux de reconnaissance en % | | | | | |
|---|---|---|---|---|---|---|
| | PPVRD | SIFT+COLOR | RELAX | REINF | SVD | ENTROPIE |
| -90 | 0 | 0 | 0 | 0 | 0 | 0 |
| -81 | 0 | 2.12 | 2.12 | 2.12 | 4.25 | 0 |
| -72 | 2.12 | 8.51 | 4.25 | 2.12 | 6.38 | 2.12 |
| -63 | 8.51 | 8.51 | 10.63 | 8.51 | 17.02 | 6.38 |
| -54 | 21.27 | 10.63 | 31.91 | 25.53 | 25.53 | 17.02 |
| -45 | 42.55 | 38.29 | 55.31 | 48.93 | 36.17 | 42.55 |
| -36 | 53.19 | 29.78 | 61.70 | 57.44 | 57.44 | 51.06 |
| -27 | 74.46 | 68.08 | 76.59 | 74.46 | 76.59 | 63.82 |
| -18 | 61.70 | 68.08 | 80.85 | 72.34 | 80.85 | 57.44 |
| -9 | 85.85 | 51.06 | 89.36 | 87.23 | 85.10 | 85.85 |
| +9 | 80.85 | 59.57 | 85.10 | 82.97 | 80.85 | 80.85 |
| +18 | 76.59 | 48.93 | 80.10 | 76.59 | 78.72 | 68.08 |
| +27 | 63.82 | 42.55 | 70.21 | 70.21 | 61.70 | 59.57 |
| +36 | 51.06 | 40.42 | 61.70 | 53.19 | 61.70 | 48.93 |
| +45 | 44.68 | 40.42 | 57.44 | 48.93 | 55.31 | 38.29 |
| +54 | 48.93 | 48.93 | 57.44 | 57.44 | 38.29 | 46.80 |
| +63 | 12.76 | 21.27 | 12.76 | 12.76 | 17.02 | 12.76 |
| +72 | 10.63 | 12.76 | 10.63 | 12.76 | 4.25 | 6.38 |
| +81 | 6.38 | 6.38 | 4.25 | 6.38 | 4.25 | 6.38 |
| +90 | 2.12 | 0 | 4.25 | 2.12 | 0 | 2.12 |
| Moyenne | 37.48 | 31.64 | 42.86 | 40.10 | 39.57 | 34.67 |

Tab. 5.2 – Résultats de la recherche d'objets avec la base SOIL-47A pour $k = 1$. Pour chaque angle, la performance maximale est soulignée.

moyens de reconnaissance obtenus avec la méthode RELAX sont respectivement de 80.13% et de 97.33%, voir le tableau 5.5.

**Remarques**

On note que, globalement, la méthode de relaxation RELAX donne les meilleurs résultats dans le cadre de cette application. Elle est suivie par la méthode de renforcement REINF, puis par les méthodes SVD et PPVRD. La prise en compte de l'entropie donne des résulats légèrement inférieurs à ceux obtenus par PPVRD et la méthode SIFT+COLOR donne de moins bons résultats comparés à PPVRD.

Cette mauvaise performance de la prise en compte de la couleur dans la caractérisation locale, méthode SIFT+COLOR, s'explique par le fait que la base SOIL-47A contient de nombreux objets de couleur similaire (voir figure 5.5). La prise en compte de la couleur

| Angles en degrés | Taux moyen de reconnaissance en % | | | | | |
|---|---|---|---|---|---|---|
| | PPVRD | SIFT+COLOR | RELAX | REINF | SVD | ENTROPIE |
| ±60 | 58.68 | 45.56 | 67.37 | 62.93 | 61.52 | 54.95 |
| ±20 | 76.06 | 56.91 | 84.04 | 79.78 | 81.38 | 72.86 |

TAB. 5.3 – Performances moyennes pour des angles de vue inférieurs à 20° et à 60° pour $k = 1$. Pour chaque angle, la performance maximale est soulignée.

| Angle en degrés | Taux de reconnaissance en % | | | | | |
|---|---|---|---|---|---|---|
| | PPVRD | SIFT+COLOR | RELAX | REINF | SVD | ENTROPIE |
| -90 | 10.63 | 12.76 | 6.38 | 10.63 | 4.25 | 12.76 |
| -81 | 8.51 | 23.40 | 6.38 | 8.51 | 21.27 | 8.51 |
| -72 | 21.27 | 19.14 | 23.40 | 21.27 | 17.02 | 21.27 |
| -63 | 25.53 | 25.63 | 42.25 | 23.40 | 27.65 | 19.14 |
| -54 | 31.91 | 25.53 | 44.68 | 42.55 | 42.55 | 29.78 |
| -45 | 57.44 | 51.06 | 63.82 | 63.82 | 53.19 | 55.31 |
| -36 | 76.59 | 53.19 | 78.72 | 78.72 | 70.21 | 63.82 |
| -27 | 80.85 | 72.34 | 85.10 | 78.72 | 82.97 | 74.46 |
| -18 | 87.23 | 74.46 | 95.74 | 89.36 | 93.61 | 76.59 |
| -9 | 89.36 | 72.34 | 100 | 100 | 85.10 | 91.48 |
| +9 | 89.36 | 78.72 | 100 | 100 | 85.10 | 85.10 |
| +18 | 91.48 | 70.21 | 93.61 | 89.36 | 87.23 | 85.10 |
| +27 | 70.21 | 68.08 | 85.10 | 74.46 | 72.34 | 68.08 |
| +36 | 72.34 | 59.57 | 82.97 | 78.73 | 72.34 | 72.34 |
| +45 | 65.95 | 53.19 | 70.21 | 61.70 | 68.08 | 59.57 |
| +54 | 53.19 | 61.70 | 61.70 | 57.44 | 51.06 | 55.31 |
| +63 | 25.53 | 19.14 | 29.78 | 25.53 | 34.04 | 23.40 |
| +72 | 21.27 | 29.78 | 25.53 | 21.27 | 12.76 | 14.89 |
| +81 | 10.63 | 19.14 | 21.27 | 8.51 | 10.63 | 8.51 |
| +90 | 8.51 | 4.25 | 10.63 | 8.51 | 6.38 | 4.25 |
| Moyenne | 49.89 | 44.68 | 56.36 | 52.12 | 49.89 | 46.48 |

TAB. 5.4 – Résultats de la recherche d'objets avec la base SOIL-47A pour $k = 3$. Pour chaque angle, la performance maximale est soulignée.

| Angles en degrés | Taux moyen de reconnaissance en % | | | | | |
|---|---|---|---|---|---|---|
| | PPVRD | SIFT+COLOR | RELAX | REINF | SVD | ENTROPIE |
| ±60 | 72.16 | 61.69 | 80.13 | 76.23 | 71.98 | 68.07 |
| ±20 | 89.36 | 73.93 | 97.33 | 94.68 | 87.76 | 84.56 |

TAB. 5.5 – Performances moyennes pour des angles de vue inférieurs à 20° et à 60° pour $k = 3$. Pour chaque angle, la performance maximale est soulignée.

ajoute donc de la confusion dans la mise en correspondance des images et c'est la raison pour laquelle elle est dépassée en performance par les méthodes qui ne tiennent compte que de l'information de gradient à travers la caractérisation par SIFT.

Il existe toutefois des cas dans lesquelles la prise en compte de la couleur donne de meilleurs résultats. En particulier, lorsque l'angle de vue devient très important, supérieur à 70°, la partie visible de l'objet dans l'image test n'est pas la même que celle visible dans la vue de face. Dans ce cas, l'information colorimétrique est plus utile que la seule caractérisation par SIFT (voir les tableaux 5.2 et 5.4).

Les performances de chacune des méthodes décroit quand l'angle de vue augmente comme le montre la figure 5.6 dans le cas où $k = 1$. Pour un angle de vue supérieur à 60°, les performances obtenues sont très faibles, car très peu de points d'intérêt détectés sur le modèle de l'objet (la vue de face) le sont également sur l'image test. Les différentes méthodes de mise en correspondance sont adaptées à des variations de points de vue raisonables, i.e. jusqu'à 50°.

Soulignons enfin que les résultats obtenus avec notre méthode de relaxation sont comparables ou supérieurs à ceux présentés par différents auteurs dans la littérature avec la même base d'images. Par exemple, Koubaroulis *et al.* évaluent deux méthodes de reconnaissance d'objets en utilisant la base SOIL-47A [72]. Les deux méthodes étant une méthode basée uniquement sur la couleur MNS (Multimodal Neighbourhood Signature) introduit par Matas *et al.* [87] et ARG (Attributed Relational Graph) une méthode représentant les relations entre les paires de primitives sous la forme d'un graphe [3]. Dans leurs expériences, les auteurs utilisent un sous-ensemble de la base SOIL-47A, composé uniquement des 24 objets de forme plane (les boîtes de céréales par exemple). Nous noterons cette base, SOIL-24A.

Les résultats rapportés par Koubaroulis *et al.* [72] et ceux obtenus par notre méthode de relaxation sont présentés dans le tableau 5.6. Soulignons qu'on se place dans le cas où $k = 1$, i.e. l'objet recherché correspond à la première image extraite de la base. Comme on peut le voir, la relaxation donne des résultats supérieurs aux deux autres approches.

Dans un autre article, Obdrzalek et Matas [102] rapportent un taux de reconnaissance de 100% en utilisant la base SOIL-24A et en se limitant à des angles de vue inférieurs à 45°. Nous obtenons un taux de reconnaissance de 98.83% avec la base SOIL-24A pour des angles de vue inférieurs à 45°. La méthode LAF (Local Affine Frames) de Obdrzalek et Matas est donc légèrement plus performante que la relaxation pour cette application.

Ces excellentes performances obtenues avec la base SOIL-24A s'expliquent principale-

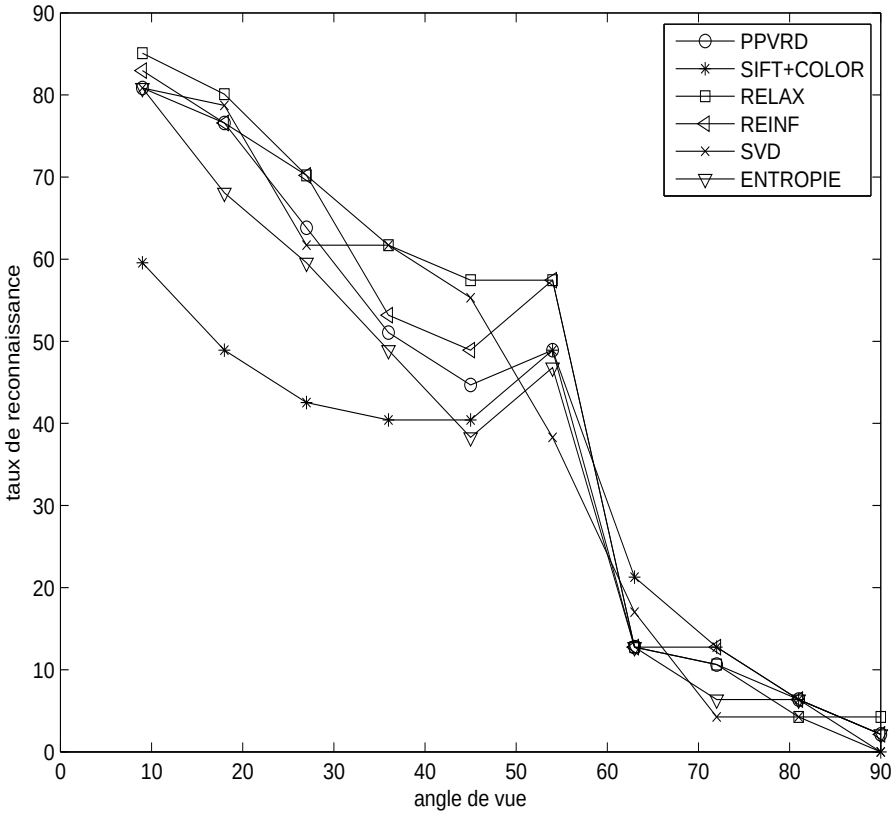



FIG. 5.6 – Evolution des résultats de la recherche d'objets avec la base SOIL-47A en fonction de l'angle de vue, pour $k = 1$.

| Angles en degrés | Taux de reconnaissance en % | | |
|---|---|---|---|
| | RELAX | MNS | ARG |
| ±60 | <u>82.29</u> | 74.6 | 67.8 |
| ±20 | <u>100</u> | 71 | 87.5 |

TAB. 5.6 – Comparaison de différntes approches avec la base SOIL-24A ($k = 1$). Pour chaque angle, la performance maximale est soulignée.

ment par le fait que les différents détecteurs présentés au chapitre 2, ont une très bonne répétabilité lorsque les scènes contiennent des structures planes. Par conséquent, les différentes méthodes de mise en correspondance permettent de trouver de nombreux appariements corrects.

### 5.3.2 Reconnaissance d'objets dans des scènes complexes

#### Base d'images et critère d'évaluation

Pour cette application, nous utilisons une base d'images proposée par Ferrari [36, 34] et accessible à l'adresse suivante : `http://www.vision.ee.ethz.ch/~ferrari`. La base est constituée de 9 modèles d'objets et de 23 images tests. Certaines images tests contiennent plusieurs objets et au total, les objets apparaissent 43 fois dans les images tests.

Nous avons 3 objets planaires représentés chacun par une seule image, 2 objets de forme fortement courbée représentés chacun par 6 images, 3 objets 3D de formes complexes représentés chacun par 8 images et un objet 3D représenté par une vue frontale. La figure 5.7 montre une image de chaque objet et la figure 5.8 montre quelques unes des images tests. L'ensemble des images représentant les modèles d'objets est donné en annexe (annexe D).

Nous utilisons cette base d'images à cause de son degré de difficulté élevé. En effet, les images tests présentent des changements de point de vue et d'échelle très importants par rapport aux images représentant les modèles des objets. De plus, il y a des déformations non rigides et non planaires, et les occultations de la surface de l'objet dans la scène peuvent atteindre 80%. Comme on peut le voir sur la figure 5.8, les objets sont très occultés dans les scènes, et ils apparaissent plus petits que dans les modèles. Cela rend la mise en correspondance très difficile.

Nous évaluons la performance à l'aide des courbes ROC (Receiver Operating Characteristics). Chaque modèle d'objet est mis en correspondance avec chaque image test et on compte le nombre d'appariements. L'objet est détecté dans la scène si ce nombre d'appariement dépasse un seuil pré-défini. Les courbes ROC sont obtenues en faisant varier le seuil de 0 à 200 appariements. Notons que pour les objets modélisés par plusieurs images, nous effectuons la somme des appariements obtenus avec chacune des images représentant l'objet.

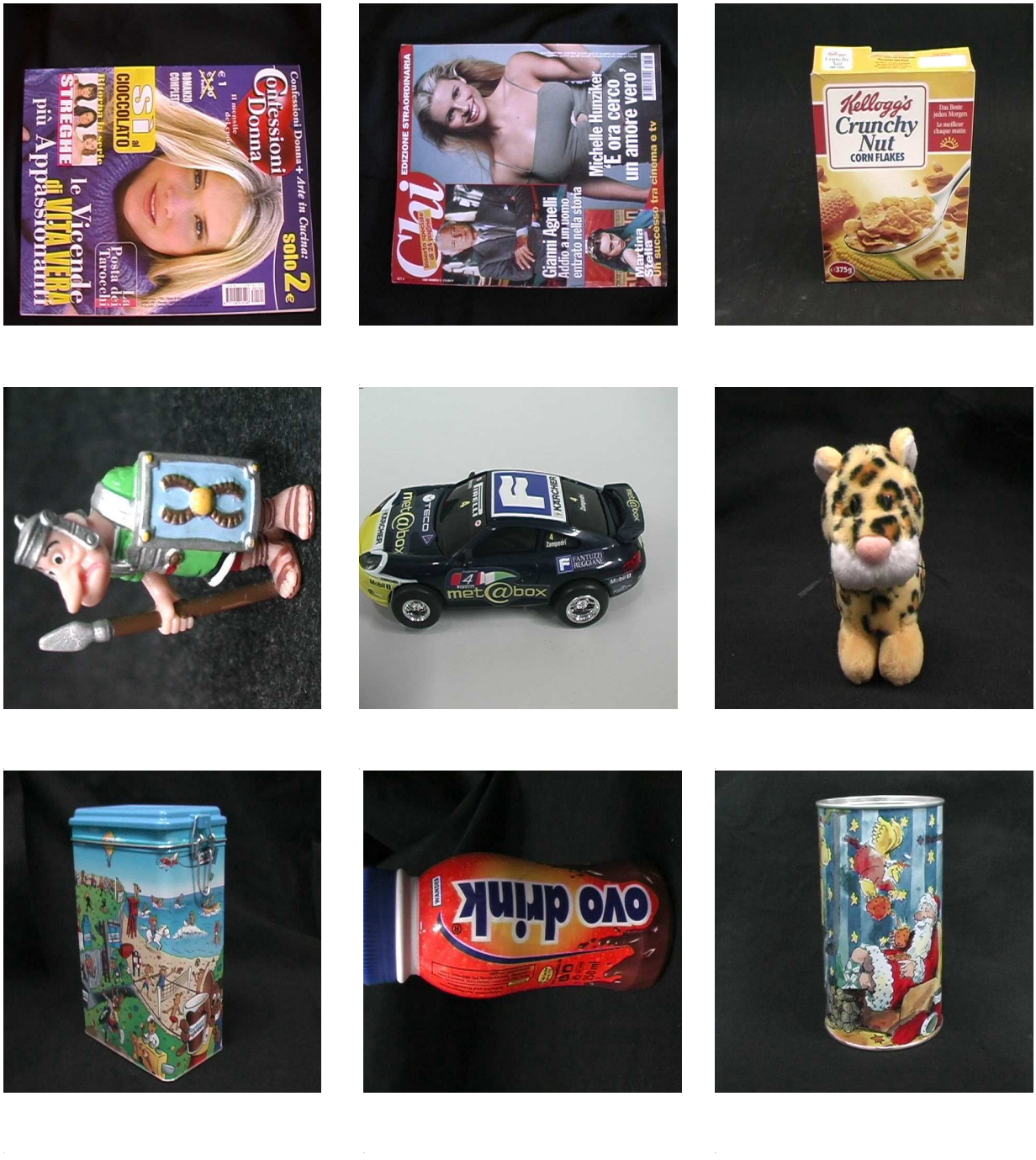



FIG. 5.7 – Modèles des objets utilisés dans le cadre de la reconnaissance d'objets. Certains objets sont modélisés par une seule vue, d'autre le sont par plusieurs vues. L'ensemble des vues représentants les objets est donné dans l'annexe D.


Fig. 5.8 – Exemples de scènes complexes. On notera que les objets sont déformés, occultés et à des échelles réduites dans les scènes. La mise en correspondance est dans ces cas, un véritable chalenge.

| | RELAX | REINF | PPVRD | SIFT+COLOR |
|---|---|---|---|---|
| Taux de détection | 65% | 55% | 40% | 20% |

TAB. 5.7 – Taux de détection pour un taux d'erreur égal à 10%.

**Résultats**

Les figures 5.9 et 5.10 montrent des exemples de détection et de localisation d'un objet dans une scène complexe. Sur l'exemple de la figure 5.9, en dépit des occultations, de la présence de nombreux autres objets et des déformations non planaires de l'objet, notre méthode de relaxation trouve 24 appariements, tous corrects, qui permettent de déterminer avec précision la pose de l'objet.

Dans l'exemple de la figure 5.10, la relaxation permet d'obtenir 18 appariements tous corrects. la méthode de renforcement des scores, REINF, donne quand à elle quelques appariements incorrects qui faussent l'estimation de la position de l'objet dans la scène. Notons aussi que dans ce cas difficile, les méthodes qui tiennent compte du contexte dans la phase de description des points d'intérêt, PPVRD, SIFT+COLOR, ENTROPIE et SVD, ne fournissent aucun appariement correct.

Nous évaluons les performances de quatre méthodes avec l'ensemble de la base : les deux méthodes qui donnent les meilleurs résultats dans les expériences du chapitre précédent, RELAX et REINF, la méthode du plus proche voisin avec rapport de distance, PPVRD, qui sert de méthode de référence, et l'ajout de la couleur à la description locale, SIFT+COLOR.

Les résultats comparatifs sont présentés par la figure 5.11. Comme on peut le noter sur cette figure, la prise en compte de l'information contextuelle améliore les résultats de façon considérable. RELAX donne de meilleurs résultats par rapport, respectivement, à REINF, PPVRD et SIFT+COLOR.

Le tableau 5.7 indique les taux de détections correctes obtenus pour un taux d'erreur égal à 10%. RELAX obtient un taux de détections correctes de 65%, REINF obtient un taux de 55%, PPVRD de 40% et SIFT+COLOR de 20%.

**Remarques**

Aucune des méthodes évaluées ne donne des résultats totalement satisfaisants du fait de la difficulté posée par la base d'images que nous utilisons. Les performances sont assez faibles pour des taux d'erreur inférieurs à 10%. La principale explication de ces faibles performances est la faible répétabilté du détecteur de points d'intérêt utilisé dans ces cas


a)

b)

Fig. 5.9 – Exemple de résultat de reconnaissance d'objet avec notre méthode de relaxation. (a) détection de l'objet dans la scène. (b) localisation de l'objet.


a)


b)


c)

FIG. 5.10 – Exemple de résultat de reconnaissance d'objet. (a) détection de l'objet dans la scène avec la méthode de renforcement des scores (REINF). (b) détection de l'objet avec la méthode de relaxation (RELAX). (c) localisation de l'objet à partir des résultats obtenus par RELAX.

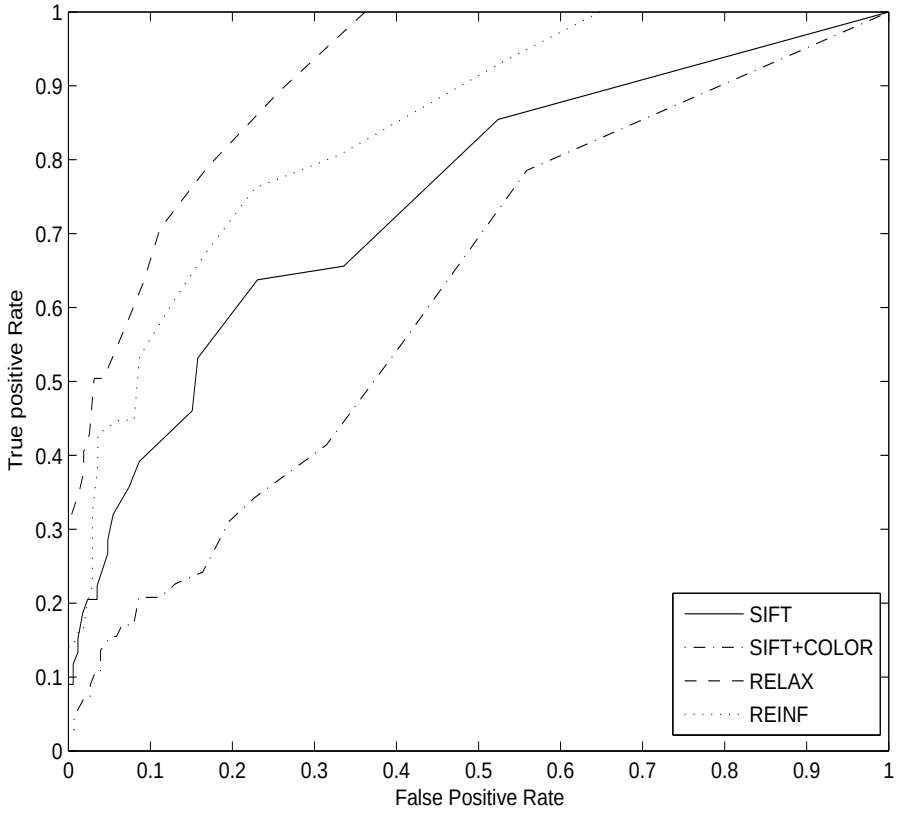



Fig. 5.11 – Résultats de reconnaissance d'objets dans des scènes complexes.

difficiles. Dans l'exemple de la figure 5.10, il y a respectivement 313 et 750 points d'intérêt détectés sur l'image de la voiture et sur l'image de la scène avec le détecteur Harris-Affine. Cependant, il y a très peu de points d'intérêt détectés sur la voiture dans l'image de la scène à cause des fortes occultations et du changement important de point de vue. Par conséquent, très peu d'appariements sont trouvés quand l'objet est présent dans la scène, et avec peu d'appariements, la performance dimunue quand le seuil de détection augmente. Toutefois, la méthode de relaxation donne de meilleurs résultats car elle est capable de trouver plus d'appariements corrects.

Notons enfin que Ferrari *et al.* [36] proposent une méthode de reconnaissance qui marche particulièrement bien avec cette base d'images. La méthode commence par établir un ensemble initial d'appariements, avec PPVRD par exemple, puis explore de façon progressive toute l'image test pour trouver de plus en plus d'appariements. Les auteurs obtiennent un taux de détection de 98% pour un taux d'erreur égal à 6%. Cependant, la méthode est assez coûteuse en temps de calcul. Il faut en moyenne 4 à 5 minutes pour traiter un couple d'images. Ce qui est très lent comparé aux quelques secondes qu'il faut pour apparier deux images avec la relaxation.

## 5.4 Conclusion

Les résultats obtenus dans ce chapitre montrent que la mise en correspondance d'invariants locaux permet la reconnaissance et la localisation des objets à partir d'un nombre réduit d'images réprésentant l'objet. On peut même reconnaître certains objets de forme simple (surfaces planes) à partir d'une seule image. C'est le cas en particulier, dans l'application de recherche d'objets dans une base d'images. On obtient des résultats très satisfaisants lorsque le changement de point de vue est relativement faible, inférieur à 20° par exemple.

Cependant, plus la scène présente des occultations, de nombreux autres objets et des déformation géométriques importantes, plus la mise en correspondance s'avère difficile comme on a pu le voir dans les expériences de reconnaissance d'objets dans des scènes complexes.

La méthode de mise en correspondance proposée au chapitre précédent permet d'obtenir de bons résultats par rapport à diverses autres méthodes basées sur les invariants locaux car elle permet d'obtenir plus d'appariements corrects. Toutefois, dans des cas les plus difficiles, comme ceux étudiés dans les expériences de la section 5.3.2, la mise en correspondance seule est insuffisante pour donner des résultats totalement satisfaisants. Elle peut, par contre,

servir de point de départ pour des approches plus élaborées comme celle proposée par Ferrari *et al.* [36].

# Deuxième partie

# Détection et suivi du visage dans une séquence d'images

*My philosophy, like color television, is all there in black and white*
***Monty Python***

Dans cette partie, nous abordons le problème de la détection du visage dans une image ou une séquence d'images en utilisant principalement les informations colorimétriques. Nous nous intéressons, plus particulièrement, au suivi d'un visage dans une séquence d'images. Il est important de souligner que notre but principal n'est pas la détection du visage, mais le suivi d'un certain nombre de points détectés sur le visage. L'étape de détection ne servant qu'à initialiser le suivi. Nous nous plaçons donc dans le cas d'une application telle que la réalisation d'une interface homme-machine, nécessitant le suivi et l'estimation de l'orientation du visage d'un utilisateur. Le chapitre 6 aborde le problème de la détection de la peau dans une image en mettant l'accent sur le choix de l'espace de représentation de la couleur. Ensuite, dans le chapitre 7 nous présentons une méthode de détection du visage basée sur la détection de certains éléments du visage tels que les yeux, la bouche et le nez. A partir de ces éléments du visage, nous proposons dans le chapitre 8 un algorithme de suivi qui utilise la méthode de mise en correspondance présentée au chapitre 4.

# Chapitre 6

# La détection de la peau dans une image couleur

Ce chapitre traite de la détection de la peau dans une image, une étape essentielle dans de très nombreuses applications de la vision par ordinateur. Nous commençons par présenter la perception de la couleur à travers la théorie trichromatique. Nous mettons ensuite l'accent sur le choix de l'espace de représentation de la couleur pour la détection de la peau dans une image couleur. Enfin, nous décrivons la méthode de détection utilisée, qui est basée sur une analyse statistique de la distribution des couleurs dans l'espace de représentation choisi.

## 6.1 Introduction

De nombreux systèmes de vision ont pour but la détection, la localisation et la reconnaissance de personnes. Par exemple, les systèmes de vidéo-surveillance font tous appel à des méthodes de détection de personnes.

L'un des moyens les plus efficaces pour détecter une personne dans une image est la détection de la peau lorsque cela est possible. Car, d'une manière générale, au moins quelques parties du corps telles que la tête et le visage, les bras et les mains sont visibles. Cependant, la détection de la peau n'est pas une tâche facile dans la mesure où la couleur ou la teinte de la peau, comme la couleur de tout autre objet, varie en fonction du matériel utilisé pour acquérir l'image ainsi que des conditions d'acquisition, i.e l'environnement dans lequel l'image est acquise. De plus, dans des scènes complexes telles que les scènes d'extérieur, il peut être difficile, voire impossible, de distinguer la peau d'une personne de

la couleur de divers autres objets.

D'un point de vue de la classification, la détection de la peau peut être vue comme un problème de décision binaire : *peau vs non-peau*. Les principales étapes de la détection sont : (i) le choix d'un espace de réprésentation de la couleur ; (ii) la modélisation des classes peau et non-peau par des distributions appropriées ; (iii) la classification basée sur ces distributions.

La méthode généralement employée pour détecter la peau repose sur la modélisation de la distribution des vecteurs de chrominance dans un espace de représentation choisi. Les résultats obtenus dépendent surtout de l'espace et de la modélisation adoptés [146, 162].

Dans ce chapitre, nous commencons par passer en revue les principaux espaces de représentation de la couleur, puis nous adopterons une approche paramétrique pour l'estimation de la distribution de la couleur de la peau dans l'espace choisi.

## 6.2 La perception de la couleur et la théorie trichromatique

Peut-on imaginer un monde sans couleur ? La couleur est un élément important de la vie. Elle n'apporte pas seulement de la beauté aux objets, elle fournit aussi une information utile sur ces objets afin de faciliter leur localisation et leur identification. Par exemple, la couleur est utile pour distinguer un fruit apte à la consommation d'un fruit qui ne l'est pas ou pour identifier son équipe favorite au cours d'un match de football !

La couleur est un phénomène psycho-physiologique provoqué par l'excitation de photorécepteurs situés sur la rétine de l'œil par une onde électromagnétique. Elle est donc le résultat conjugué de [149] :

- la source lumineuse qui éclaire la scène ;
- la géométrie d'observation (angles d'éclairement et d'observation) ;
- la scène et ses caractéristiques physiques ;
- l'œil de l'observateur ou le capteur de la caméra ;
- le cerveau de l'observateur.

### 6.2.1 La théorie trichromatique

La compréhension de la perception de la couleur a commencé avec Newton et son expérience sur la dispersion de la lumière à travers un prisme en 1672. En 1801, Thomas

Young a suggéré que trois couleurs primaires étaient suffisantes pour produire toutes les couleurs de façon additive. Ces travaux ont été poursuivis par Helmholtz et la théorie trichromatique a été prouvée en 1960 par la découverte de trois types de récepteurs dans la rétine qui correspondent aux trois type de cônes L, M et S. Leur réponse maximale se situe respectivement dans les teintes bleues à 440 nm pour les cônes de type S (*Short*), dans les teintes vertes à 545 nm pour les cônes de types M (*Medium*) et dans les teintes rouges à 580 nm pour les cônes de type L (*Long*).

Du fait de cette trichomie, il est possible de représenter les couleurs dans un espace tridemensionel dont les vecteurs de base correspondent aux couleurs primaires. Ainsi, un vecteur couleur $[S]$ est défini par une combinaison linéaire des vecteurs de la base $([R], [G], [B])$ :

$$[S] = r[R] + g[G] + b[B] \tag{6.1}$$

où les nombres $r$, $g$ et $b$ sont les composantes trichromatiques et représentent les quantités respectives des primaires utilisées.

D'important travaux ont été effectués afin d'obtenir les fonctions colorimétriques qui permettent de calculer facilement les composantes trichromatiques d'une lumière colorée. Ces travaux ont donné naissance au standard défini par la CIE (Commission Internationale de l'Eclairage) en 1931 [109].

## 6.3 Les espaces de représentation de la couleur

En référence au système visuel humain et à la théorie trichromatique, on considère, de manière générale, que la couleur se définit selon trois composantes qui conduisent à différentes familles de systèmes de représentation : les systèmes primaires, les systèmes luminance-chrominance, les systèmes perceptuels, les systèmes d'axes indépendants, etc. Le lecteur intéressé consultera avec intérêt l'ouvrage collectif [149] pour plus de précision.

Bien qu'il existe une forte dépendance au système de primaires RGB en raison notamment de la dépendance aux matériels (cartes d'acquisition, cartes vidéo, caméras, écran, ...) qui effectuent leurs échanges d'information uniquement en utilisant les triplets (R,G,B), les systèmes de représentation les plus utilisés sont les systèmes de type luminance-chrominance et les systèmes perceptuels.

### 6.3.1 Les systèmes intensité-chromaticité

L'intérêt des espaces de type intensité-chromaticité est qu'ils dissocient la composante d'intensité des composantes de chrominance. De nombreux espaces de représentation se rattachent à cette famille et ils se différencient essentiellement par la façon dont sont calculées les coordonnées d'intensité et de chrominance.

#### Les espaces de type $YC_1C_2$

Ce système a été à l'origine développé pour assurer une compatibilité entre les téléviseurs couleurs et les téléviseurs noir et blanc, d'où la séparation des composantes de luminance et de chrominance. Une simple transformation linéaire permet de passer d'un système RGB au sytème de type $YC_1C_2$, mais cette transformation diffère suivant les standards de télévision (NTSC, PAL ou SECAM).

La forme générale des composantes chromatiques est donnée par :

$$\begin{cases} C_b = a_1(R-Y) + b_1(B-Y) \\ C_r = a_2(R-Y) + b_2(B-Y) \end{cases} \tag{6.2}$$

où les coefficient $a_1$, $a_2$, $b_1$ et $b_2$ sont spécifiques au standard considéré et $Y$ est la luminance.

Comme déjà souligné, il existe plusieurs systèmes de type $YC_1C_2$. Ainsi le système YIQ est celui qui correspond à la norme NTSC, le système YUV est celui qui correspond à la norme PAL et le système YCrCb dédié au codage digital des images de la télévision numérique, correspond à la norme ITU.BT-601 et fait partie du nouveau standard de compression JPEG2000 [144].

Les principales transformations sont données par les équations suivantes :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6.3}$$

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6.4}$$

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6.5}$$

### Les systèmes perceptuellement uniformes

Les systèmes preceptuellement uniformes correspondent aux systèmes uniformes au sens de la perception visuelle. Plus précisément, la spécificité de ces systèmes est de pouvoir décrire fidèlement, par rapport à la vision humaine, les écarts de couleur entre couleurs proches. Ils sont donc dits, perceptuellement uniformes. Pour une présentation détaillée de ces espaces, voir [149].

### L'espace L*a*b*

Le système L*a*b* peut être considéré comme le système de référence de la CIE (il a été introduit en 1976) pour calculer des écarts de couleurs. La transformation qui permet de passer de l'espace XYZ (l'espace XYZ est un espace de primaires introduit par la CIE en 1931 pour pallier à certains inconvenients du système colorimétriques RGB) à l'espace L*a*b* est une transformation non linéaire qui prend en compte les coordonnées trichromatiques du blanc de référence $W = (X_0, Y_0, Z_0)$.

Les composantes L*, a* et b* sont obtenues par les équations suivantes :

$$L^* = \begin{cases} 116 * (\frac{Y}{Y_0})^{\frac{1}{3}} - 16 & \text{si} \quad \frac{Y}{Y_0} \geq 0.008856 \\ 903.3 * \frac{Y}{Y_0} & \text{si} \quad \frac{Y}{Y_0} \leq 0.008856 \end{cases} \tag{6.6}$$

$$a^* = 500 * (f(\frac{X}{X_0}) - f(\frac{Y}{Y_0})) \tag{6.7}$$

$$b^* = 200 * (f(\frac{Y}{Y_0}) - f(\frac{Z}{Z_0})) \tag{6.8}$$

avec

$$f(x) = \begin{cases} x^{\frac{1}{3}} & \text{si} \quad \frac{Y}{Y_0} \geq 0.008856 \\ 7.787x + \frac{16}{116} & \text{si} \quad \frac{Y}{Y_0} \leq 0.008856 \end{cases} \tag{6.9}$$

### 6.3.2 Les espaces perceptuels

Les espaces perceptuels correspondent à une interprétation des couleurs par le système visuel humain. Les espaces représentatifs de cette famille sont l'espace HSI et HSV.

#### L'espace HSI

La modélisation de ce système de représentation communément utilisé en traitement d'images couleur, est issue de la rotation du cube des couleurs RGB. En effet, il suffit de faire pivoter le cube sur le coin représentant le noir ; ainsi, l'axe achromatique constitue l'axe des intensités I et la couleur est définie par une position sur un pallier circulaire où la saturation S représente le rayon et la teinte H représente l'angle.

Les formules exprimant la transformation de l'espace RGB à l'espace HSI sont données par :

$$\begin{cases} I = \frac{R+G+B}{3} \\ S = 1 - \frac{3*min(R,G,B)}{R+G+B} \\ H = arcos(\frac{0.5*(R-G)+(R-B)}{\sqrt{(R-G)^2+(R-B)(G-B)}}) \end{cases} \tag{6.10}$$

#### L'espace HSV

L'espace HSV est un système de cône hexagonal qui représente la couleur sous la forme d'un triplet : teinte H (*Hue*), saturation S (*Saturation*) et luminosité V (*Value*). Les transformations sont effectuées comme suit :

$$V = max(R, G, B) \tag{6.11}$$

$$S = \frac{V - min(R, G, B)}{V} \tag{6.12}$$

$$H = \begin{cases} \frac{G-B}{V-min(R,G,B)} & \text{si} \quad V = R \\ 2 + \frac{B-R}{V-min(R,G,B)} & \text{si} \quad V = G \\ 4 + \frac{R-G}{V-min(R,G,B)} & \text{si} \quad V = B \end{cases} \tag{6.13}$$

Notons que les espaces intensité-chromaticité exprimés en coordonnées polaires peuvent être interprétés en terme de teinte et de saturation et deviennent donc des espaces perceptuels.

### 6.3.3 Les systèmes d'axes indépendants

Suivant la distribution des couleurs considérée, autrement dit suivant l'image traitée et le système de représentation utilisé, on peut observer une corrélation plus ou moins forte entre les différentes composantes couleur. Si cette corrélation est relativement forte, traiter indépendamment chacune des composantes conduit à une perte d'information. Pour pallier ce problème, on utilise les système d'axes indépendants dont le système de représentation $I_1I_2I_3$.

#### L'espace $I_1I_2I_3$

Cet espace introduit par Otha *et al.* [103] est inspiré de la transformation de *Karhunen-Loeve* afin de déterminer les trois axes de plus grande variance de l'ensemble des couleurs. A partir d'un échantillon de quelques images (8 en fait), les auteurs parviennent à obtenir un système d'axes qui est une bonne approximation de la transformation de *Karhunen-Loeve*. La transformation qui permet de passer de passer de l'espace RGB à l'espace $I_1I_2I_3$ est une transformation linéaire définit par l'équation suivante :

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6.14}$$

L'espace $I_1I_2I_3$ appartient également à la famille des systèmes de type luminance-chrominance, puisque $I_1$ correspond à la luminance, et $I_2$ et $I_3$ aux composantes de chrominance.

## 6.4 La détection de la peau dans une image couleur

La détection de la peau est une étape très importante dans de nombreux systèmes de vision qui ont pour but la détection, la localisation et la reconnaissance de personnes. La détection de la peau est en effet, un des moyen efficace pour détecter une personne dans une image car, d'une manière générale, au moins quelques parties du corps telles que la tête et le visage, les bras et les mains sont visibles.

La construction d'un système de détection de la peau doît cependant répondre à deux principaux problèmes :

- quel espace de représentation des couleurs choisir ?
- comment modéliser la distribution de la couleur de la peau dans cet espace ?

### 6.4.1 Choix de l'espace couleur

Il existe de nombreux espaces de représentation de la couleur (voir la section précédente) et le choix d'un espace dépend en priorité de l'application envisagée.

Dans le cas de la détection de la peau, ce choix peut être guidé par deux observations :

- bien que la couleur de la peau varie d'une personne à une autre, différentes études ont montré que la différence se situe plus au niveau de la composante de luminance qu'au niveau des composantes de chrominance [47, 62] ;
- dans l'espace défini par les composantes de chrominance, les pixels correspondant à la peau sont "assez bien" regroupés, voir par exemple les travaux de Yang *et al.* [161].

Ces deux observations conduisent à privilégier les espaces de représentation qui séparent les composantes de luminance et de chrominance et à s'intéresser aux composantes de chrominance.

De nombreux auteurs utilisent les composantes de chrominance pour la détection de la peau dans différentes applications. Mais l'espace de représentation le plus utilisé est l'espace normalisé $rgb$ [162]. Cet espace est obtenu à partir de l'espace RGB par simple normalisation de la manière suivante :

$$r = \frac{R}{R+G+B} \quad g = \frac{G}{R+G+B} \quad b = \frac{B}{R+G+B} \tag{6.15}$$

Comme les trois composantes $r$, $g$ et $b$ vérifient la relation $r+g+b = 1$, on peut se contenter de la connaissance des deux premières composantes $r$ et $g$. Ce qui a pour conséquence la réduction de la dimension de l'espace de représentation.

D'autres espaces de type luminance-chrominance sont aussi utilisés pour la détection de la peau. Par exemple, l'espace $YCrCb$ est utilisé par Hsu *et al.* dans [57], tandis que Saber et Tekalp [117] utilisent l'espace $YES$.

Certains auteurs s'intéressent à la sélection des meilleurs composantes couleur pour la détection de peau. Ainsi, Gomez *et al.* [42] travaillent dans plusieurs espaces de représentation et choisissent pour chacun des ces espaces la meilleure composante, en insistant sur la complémentarité des différentes composantes. La conclusion de ce travail est que la combinaison des composantes E (de l'espace YES), r/g (de l'espace normalisé rgb) et H (de l'espace HSV), forme un espace tridimensionnel dans lequel les pixels représentant la peau sont bien séparés des autres pixels. Si de très bons résultats de détection sont présentés, les auteurs précisent cependant que leur résultats ne "doivent pas être extrapolés à des images issues de scanners ou de l'Internet", car, avancent-ils, ces images de qualité médiocre ne

respectent pas les standards de la CIE [42].

Le choix d'un espace de représentation particulier n'est donc pas aisé, mais nous basons notre choix sur les conclusions de deux travaux. D'une part, l'étude comparative de 9 espaces de représentation par Terrillon *et al.* dans [146] a montré que les meilleurs espaces de représentation des couleurs pour la détection de la peau sont les espaces normalisés $rgb$ et $TSL$. L'espace $TSL$ est un espace de représentation perceptuel qui sépare les composantes de teinte, de saturation et de luminance. Il est donc semblable à l'espace $HSI$.

D'autre part, les travaux de Shin *et al.* dans [130] portant sur l'influence des transformations d'espaces couleur sur les résultats de la détection de la peau, ont montré que la meilleure séparation entre les deux classes peau et non-peau est obtenue dans l'espace $RGB$, i.e. l'espace sans transformation.

En nous basant sur ces deux études, nous nous intéresserons à deux espaces de type luminance-chrominance, à savoir les espaces $rgb$ et $YCrCb$, et à un espace perceptuel $HSI$ (semblable à $TSL$).

### 6.4.2 Modélisation de la peau

#### Distribution dans l'espace de chrominance

Afin de déterminer la distribution des composantes chromatiques caractérisant la peau dans les espaces $rgb$, $YCrCb$ et $HSI$, nous utilisons un ensemble de 110 images dont sont extraits 1 448 273 pixels correspondant à la peau. Les images sont choisies (sur Internet) de manière à représenter des couleurs de peau différentes et des conditions d'illumination variées. Les conditions d'acquisition des images sont inconnues.

Comme souligné dans la section précédente, on note sur les histogrammes des figures 6.1, 6.2 et 6.3 que les pixels correspondants à la peau sont assez bien regroupés dans le plan des chrominances.

On peut donc segmenter une image en deux régions en classant chaque pixel dans une des deux catégories peau et non-peau. Pour ce faire, il faut modéliser la zone de l'espace du plan de chrominances qui caractérise les pixels de peau et plusieurs approches sont possibles :

- les approches non-paramétriques qui visent à estimer la distribution de la couleur de la peau sans une modélisation explicite à partir d'un ensemble d'apprentissage. Elles se traduisent généralement par un histogramme et une carte de probabilité

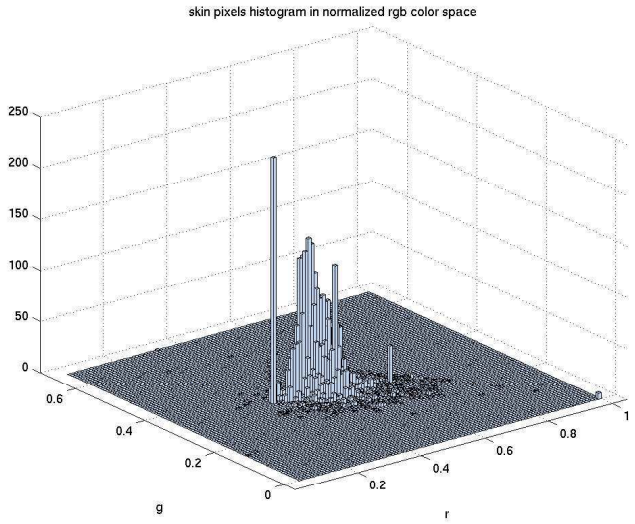



Fig. 6.1 – Histogramme des pixels de peau dans l'espace $rgb$

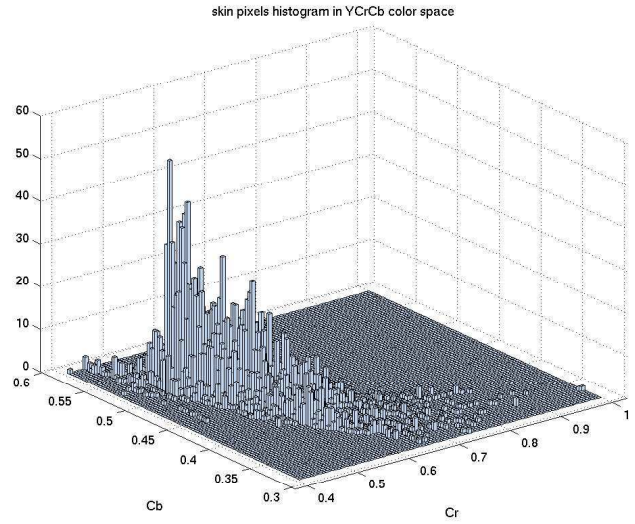



Fig. 6.2 – Histogramme des pixels de peau dans l'espace $YCrCb$

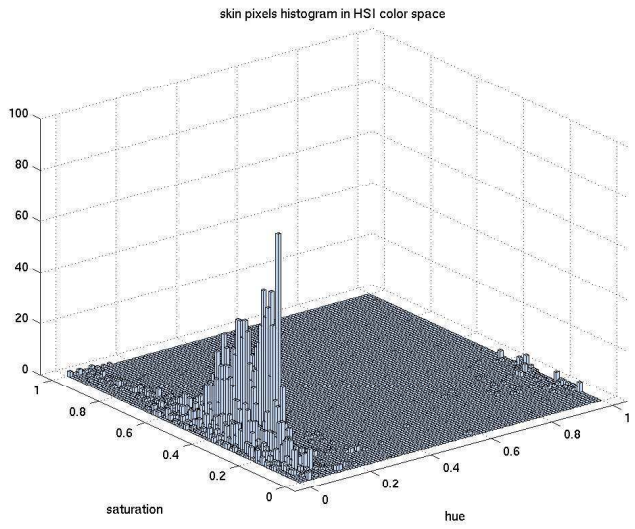



FIG. 6.3 – Histogramme des pixels de peau dans l'espace $HSI$

SPM (Skin Probability Map) qui affecte une valeur de probabilité à chaque pixel de l'espace discrétisé [42, 19].

– les approches paramétriques qui estiment la distribution de la couleur de la peau sous forme de modèles explicites. La distribution est généralement caractérisée par une densité ou une somme de densités de probabilité dont les paramètres sont obtenus à partir d'un ensemble d'apprentissage [162, 48].

Les approches non-paramétriques possèdent l'avantage de la rapidité (à la fois dans les phases d'apprentissage et de classification) et sont, en théorie, indépendantes de la forme de la distribution. Elles nécessittent cependant un espace de stockage important (pour représenter la carte de probabilité) ainsi qu'un ensemble d'apprentissage de grande taille.

La phase d'apprentissage peut être plus longue avec les approches paramétriques, mais ces dernières possèdent l'avantage de fournir une bonne estimation de la densité avec un ensemble d'apprentissage plus réduit.

Les deux approches sont largement employées dans la littérature et donnent des résultats comparables [156]. Nous utiliserons dans la suite, une méthode paramétrique pour caractériser la distribution de la couleur de la peau dans les trois espaces $rgb$, $YCrCb$ et $HSI$, car celle-ci nécessite un ensemble d'apprentissage de faible taille.

### Modélisation paramétrique

Comme nous l'avons déjà souligné, les pixels correspondants à la peau sont assez bien regroupés dans le plan des chrominances. On peut donc représenter la distribution de la couleur de la peau dans ce plan sous la forme d'une densité de probabilité. Les modèles les plus couramment utilisés sont le modèle gaussien qui représente cette distribution sous la forme d'une gaussienne [146, 117], et le modèle de mélange de gaussiennes (Mixture of Gaussians) qui représente la distribution sous la forme d'une somme de gaussiennes. Ce dernier modèle est semble-t-il plus adapté pour prendre en compte la variabilité des conditions d'acquisition des images et la présence de populations hétérogènes [48, 21].

La distribution d'une variable aléatoire $X \in \Re^d$ est un mélange de $k$ gaussiennes si sa fonction de densité est de la forme :

$$f(x|\theta) = \sum_{j=1}^{k} \alpha_j \frac{1}{\sqrt{(2\pi)^d |\Sigma_j|}} * exp\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\} \tag{6.16}$$

$f$ est donc la somme pondérée de $k$ gaussiennes $f_j$, de moyennes et et de matrices de covariance respectives $\mu_j$ et $\Sigma_j$. Les termes $\alpha_j$ sont des coefficients de pondération et vérifient la relation $\sum_{j=1}^{k} \alpha_j = 1$.

L'estimation de l'ensemble des paramètres $\theta = \{\alpha_j, \mu_j, \Sigma_j\}_{j=1}^{k}$ du modèle, à partir d'un ensemble d'apprentissage, peut se faire grâce à l'algorithme EM (Expectation Maximization) bien connu en statistiques [88].

L'algorithme EM est une méthode itérative qui permet d'obtenir l'estimé au sens du maximum de vraisemblance de $\theta$ :

$$\theta_{ML} = \arg\max_{\theta} f(x_1, ..., x_n|\theta)$$

Les principales difficultés de mise en œuvre de cet algorithme concernent l'étape d'initialisation, car l'algorithme EM converge vers un extremum local qui dépend de l'itéré initial, et le choix du nombre de classes $k$. La plupart du temps, ce nombre est fixé à priori. Pour plus de détails sur l'algorithme EM, voir [88].

Pour chaque pixel de vecteur de chrominance $x = [r, g]^T$
(1) calculer $f_j(x)$, pour $j = 1 \cdots k$ (équation (6.16)) ;
(2) prendre $f(x) = \arg\max_j(f_j(x))$ ;
(3) classer $x$ en $\begin{cases} \text{peau} & \text{si} \quad f(x) > s \\ \text{non-peau} & \text{sinon} \end{cases}$

FIG. 6.4 – Algorithme de détection de la peau dans une image.

**Mise en œuvre**

Pour mettre en œuvre l'algorithme EM et estimer les densités de probabilité, nous utilisons le logociel MIXMOD[1] disponible à l'adresse `htpp://www-math.univ-fcomte.fr/MIXMOD/index.php`. Cet outil permet d'estimer des densités de probabilité et il offre la possibilté d'estimer le bon nombre de classes.

A partir de l'ensemble de 1 448 273 pixels de peau, nous avons constitué, de manière aléatoire, un ensemble d'apprentissage de taille 250 000 pixels. Les autres pixels de peau, 1 198 273, auxquels on ajoutte 4 688 525 pixels non-peau sont utilisés pour la phase de test. L'ensemble de 250 000 pixels est lui-même divisé en 25 sous-ensembles de 10 000 pixels chacun. On calcule les densités de probabilité pour chacun des 25 sous-ensembles, et on fait la moyenne des paramètres obtenus.

En utilisant MIXMOD, nous calculons les paramètres des densités de probabilité caractérisant la peau dans chacun des trois espaces $rgb$, $YCrCb$ et $HSI$ et pour différentes valeurs de $k$. Nous nous intéressons à un modèle gaussien simple (SGM pour Simple Gaussian Model), i.e $k = 1$, et à trois modèles de mélange de gaussiennes (MGM pour Multiple Gaussian Model) pour $k \in \{2, 3, 4\}$.

Les paramètres obtenus avec MIXMOD pour l'espace $rgb$ sont présentés dans le tableau 6.1.

Une fois les paramètres du modèle connus, on peut segmenter une image en deux régions en classant chaque pixel dans une des deux catégories peau et non-peau. La méthode de détection de la peau est résumée par l'algorithme de la figure 6.4 :

Un pixel est classé comme étant de la peau, si sa probabilité d'appartenance à l'une des $k$ classes est supérieure à un seuil prédéfinit $s$, dans le cas contraire, il est considéré comme ne correspondant pas à de la peau.

[1] MIXture MODelling Software

| | $\mu$ | $\Sigma$ | $\alpha$ |
|---|---|---|---|
| $k=1$ | $\begin{pmatrix} 0.460264 \\ 0.310444 \end{pmatrix}$ | $\begin{pmatrix} 0.003517 & -0.00085 \\ -0.00085 & 0.000613 \end{pmatrix}$ | 1 |
| $k=2$ | $\begin{pmatrix} 0.446199 \\ 0.314148 \end{pmatrix}$ | $\begin{pmatrix} 0.001454 & -0.000321 \\ -0.000321 & 0.00019 \end{pmatrix}$ | 0.798159 |
| | $\begin{pmatrix} 0.515885 \\ 0.295798 \end{pmatrix}$ | $\begin{pmatrix} 0.013291 & -0.002932 \\ -0.002932 & 0.001741 \end{pmatrix}$ | 0.201841 |
| $k=3$ | $\begin{pmatrix} 0.415723 \\ 0.321385 \end{pmatrix}$ | $\begin{pmatrix} 0.000255 & -0.000047 \\ -0.000047 & 0.000037 \end{pmatrix}$ | 0.258389 |
| | $\begin{pmatrix} 0.53291 \\ 0.284281 \end{pmatrix}$ | $\begin{pmatrix} 0.016125 & -0.002978 \\ -0.002978 & 0.002354 \end{pmatrix}$ | 0.106202 |
| | $\begin{pmatrix} 0.466235 \\ 0.310369 \end{pmatrix}$ | $\begin{pmatrix} 0.001913 & -0.000353 \\ -0.000353 & 0.000279 \end{pmatrix}$ | 0.63541 |
| $k=4$ | $\begin{pmatrix} 0.472818 \\ 0.309014 \end{pmatrix}$ | $\begin{pmatrix} 0.002533 & -0.000446 \\ -0.000446 & 0.000384 \end{pmatrix}$ | 0.50656 |
| | $\begin{pmatrix} 0.413434 \\ 0.321843 \end{pmatrix}$ | $\begin{pmatrix} 0.000251 & -0.000044 \\ -0.000044 & 0.000038 \end{pmatrix}$ | 0.273769 |
| | $\begin{pmatrix} 0.54285 \\ 0.276157 \end{pmatrix}$ | $\begin{pmatrix} 0.018032 & -0.003172 \\ -0.003172 & 0.002732 \end{pmatrix}$ | 0.076815 |
| | $\begin{pmatrix} 0.461085 \\ 0.31211 \end{pmatrix}$ | $\begin{pmatrix} 0.00024 & -0.000042 \\ -0.000042 & 0.000036 \end{pmatrix}$ | 0.142855 |

Tab. 6.1 – Paramètres des densités de probabilité dans l'espace $rgb$ pour un modèle gaussien simple et 3 modèles de mélange de gaussiennes.

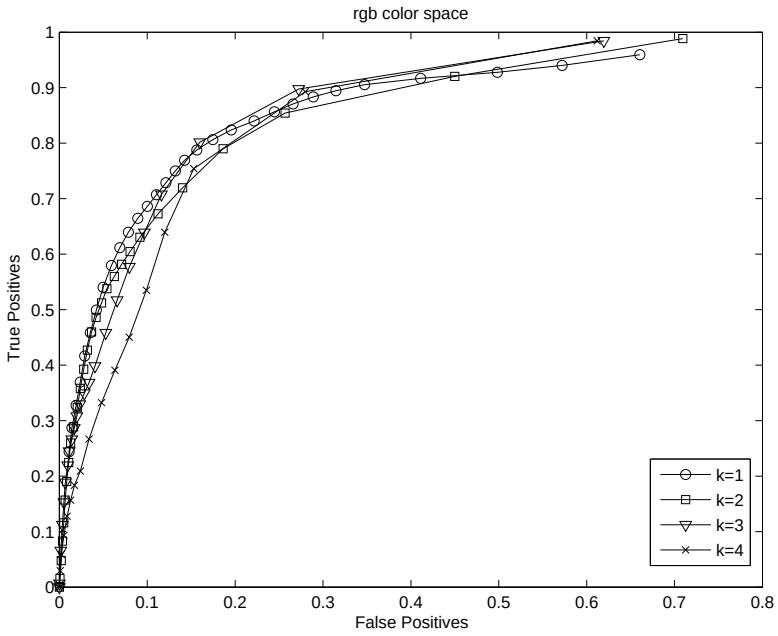



FIG. 6.5 – Courbes ROC dans l'espace *rgb*

Le choix du seuil est crucial, car la qualité de la détection y est directement liée. Afin de déterminer ce seuil, nous traçons les courbes ROC (Receiver Operating Characteristics) pour chacun des modèles et dans chacun des trois espaces. Une courbe ROC est obtenue en calculant, pour un seuil donné, le taux de fausses détections (i.e le taux de pixels non-peau incorrectement classé comme de la peau) et le taux de bonnes détection (i.e le taux de pixels peau correctement classé comme peau). Les tests sont réalisés avec un ensemble de 5 886 798 pixels dont 1 198 273 pixels peau et 4 688 525 pixels non-peau. Les résultats pour les trois espaces sont présentés sur les figures 6.5, 6.6 et 6.7.

Nous pouvons tirer quelques conclusions de l'analyse des courbes ROC des figures 6.5, 6.6 et 6.7. D'une part, le modèle gaussien simple ($k = 1$) donne des résultats comparables à ceux des modèles de mélange. Les taux de bonne détection obtenus avec le modèle SGM sont comparables, voire meilleurs, à ceux obtenus avec les modèles MGM. Ce qui peut sembler surprenant, car il est souvent admis dans la littérature que les modèles de mélange de gaussiennes donnent de meilleurs résultats [48]. En fait, les modèles MGM ne donnent de meilleurs résultats que pour des taux de bonnes détections très élévés ($> 90\%$). On peut donc se contenter d'utiliser un modèle gaussien simple. La même observation a été faite par Caetano *et al.* dans [21].

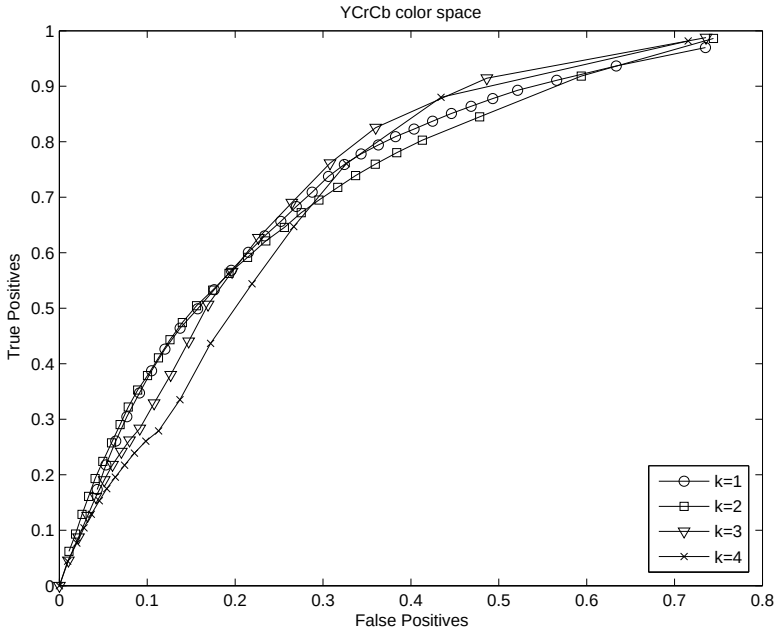



Fig. 6.6 – Courbes ROC dans l'espace $YCrCb$

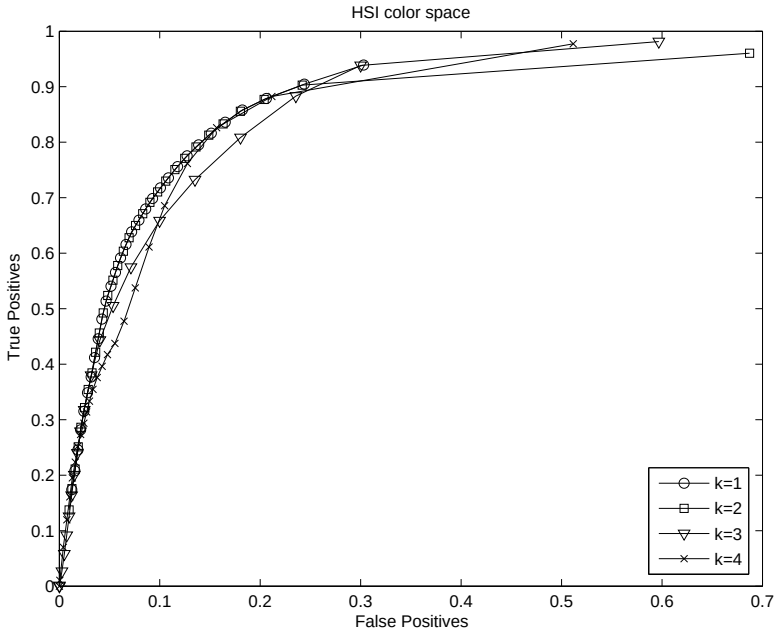



Fig. 6.7 – Courbes ROC dans l'espace $HSI$

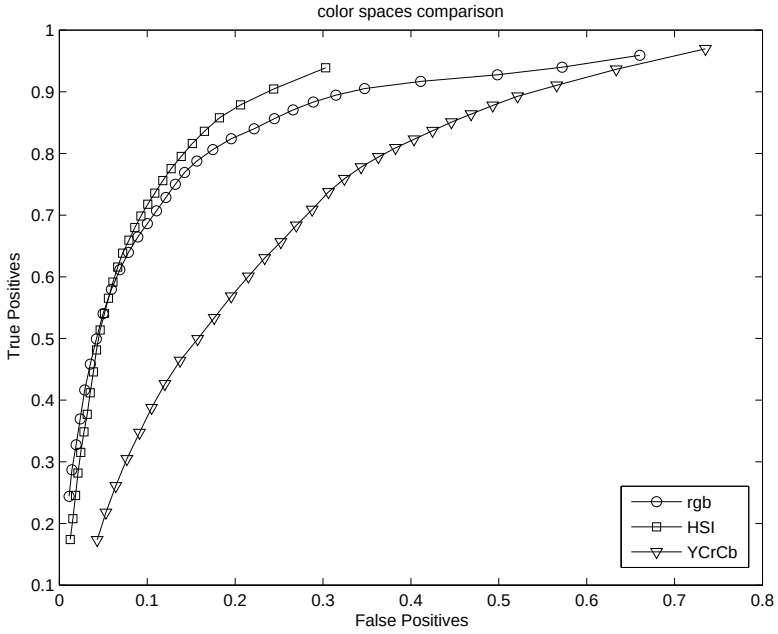



FIG. 6.8 – Comparaison des trois espaces avec un modèle gaussien simple.

D'autre part, on note que les résultats obtenus dans les espaces $rgb$ et $HSI$ sont meilleurs que ceux obtenus dans l'espace $YCrCb$ comme le montre la figure 6.8 qui présente les courbes ROC dans le cas d'un modèle gaussien simple. Cela confirme les observations de Terrillon *et al.* [146], à savoir que les meilleurs espaces de représentation des couleurs pour la détection de la peau sont les espaces normalisés $rgb$ et $TSL$. Rappelons que l'espace $HSI$ utilisé ici est comparable à l'espace $TSL$.

Dans la suite de nos travaux, nous utiliserons l'espace normalisé $rgb$ comme espace de représentation du fait de la simplicité de la transformation : $RGB \rightarrow rgb$.

### 6.4.3 Détection

La détection des régions de peau dans une image est réalisée par l'algorithme de la figure 6.4. Pour chaque modèle, le seuil de détection a été choisi à partir de la courbe ROC de façon à avoir un taux de bonnes détections supérieur à 80%, et un taux de fausses détections inférieur à 20%.

Sur la figure 6.9, nous présentons les résulats de la détection sur une image avec les différents modèles. Comme on peut le noter, le modèle gaussien simple ($k = 1$) est suffisant pour obtenir un très bon résultat. Sur la figure 6.9(b), toutes les régions de peau sont

correctement détectées et il y a très peu de fausses détections (une partie des cheveux de la fille en haut à gauche). Avec un modèle de mélange de deux gaussiennes ($k = 2$), on obtient un résultat similaire comme le montre la figure 6.9(c). L'utilisation d'un nombre plus élevé de gaussiennes ($k = 4$) permet de capturer plus détails, mais conduit également à des fausses détections. Voir la figure 6.9(d) où le vêtement de la fille en haut à droite est incorrectement détecté.

Nous utiliserons donc le modèle gaussien gaussien simple dans la suite de nos travaux. D'autres exemples de détection sont présentés sur la figure 6.10.

### 6.4.4 Remarques

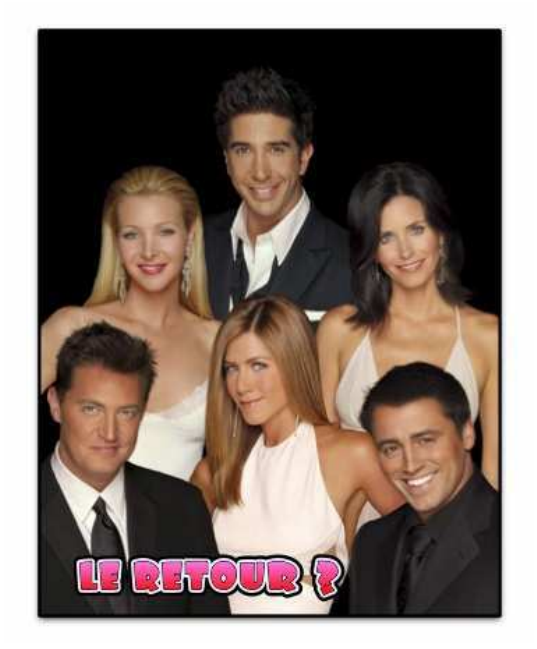
Le modèle gaussien simple (SGM) suffit pour obtenir de bons résultats de détection dans nos expériences. Cependant, dans de très nombreux travaux il est admis et/ou affirmé que les modèles de mélanges de gaussiennes (MGM) donnent de meilleurs résultats [162, 48]. Nous avons pu constaté dans nos expériences que cela n'est pas toujours le cas.

En fait, il est important de souligner que les résultats obtenus dépendent très fortement de l'ensemble des pixels utilisés comme base d'apprenstissage. Les paramètres des densités de probabilité sont obtenus à partir d'un ensemble de 250 000 pixels dans nos expériences et nous adoptons une modélisation paramétrique qui est adapté pour des ensembles d'apprentissage de faible taille. Mais même en utilisant une base d'apprentissage de plusieurs millions de pixels, Caetano *et al.* arrivent à la même conclusion que nous [21], i.e. les modèles MGM ne donnent de meilleurs résultats que pour des taux de bonnes détections très élévés ($> 90\%$).

Nous pensons que les modèles de mélange sont très adaptés lorsqu'on souhaite batir un modèle pour détecter différents types de peau : africain, asiatique, caucasien, etc. Dans ce cas, chaque famille ou type de peau peut être raisonablement représentée par une gaussienne (SGM), et le modèle général par la somme de ces gaussiennes (MGM).

## 6.5 Conclusion

Dans ce chapitre, nous avons vu comment l'information colorimétrique présente dans une image couleur peut être utilisée pour la détection de la peau. Cela passe par le choix d'un espace de représentation convenable et par une modélisation adéquate de la distribution des couleurs dans cet espace.




a)

b)

c)

d)

FIG. 6.9 – Exemple de détection de la peau. (a) image originale ; (b) résultat de la détection avec un modèle gaussien simple (k=1) ; (c) résultat de la détection avec un modèle de mélange de gaussiennes (k=2) ; (d) résultat de la détection avec un modèle de mélange de gaussiennes (k=4).

Fig. 6.10 – Exemple de détection de la peau. De gauche à dorite : image originale et résultat de la détection avec un modèle gaussien simple.

Nos expériences nous ont permis de confirmer deux observations importantes déjà faites par de nombreux auteurs. D'une part, les meilleurs espaces de représentation pour la détection de la peau sont les espaces de type intensité-chromaticité. En particulier, les meilleurs résultats sont obtenus avec l'espace perceptuel $HSI$ et l'espace normalisé $rgb$. D'autre part, dans le cas d'une représentation paramétrique de la distribution des couleurs dans le plan de chrominances, l'utilisation d'un modèle gaussien simple suffit pour obtenir de bons résultats.

Dans les chapitres suivants, nous utilisons la détection de la peau comme étape initiale pour la détection et le suivi d'un visage dans une séquence d'images.

# Chapitre 7

# Détection des yeux dans une image

Dans ce chapitre, nous abordons le problème de la détection des yeux dans une image couleur. Notre méthode de détection est basée sur l'utilisation de l'information colorimétrique et fait donc appel à la méthode de détection de la peau présentée au chapitre précédent. Une fois les yeux détectés, il est possible d'obtenir une localisation du visage dans l'image.

## 7.1 Introduction

La détection et la reconnaissance du visage est un domaine de recherche qui a reçu une attention particulière au cours de ces dernières années dans la communauté de la vision par ordinateur. Cela est dû principalement à l'émergence de nombreuses applications telles que la vidéo surveillance, l'identification et l'authentification de personnes et les interfaces homme-machine intelligentes. Dans toutes ces applications, la détection et la localisation du visage est une étape cruciale. En effet, pour reconnaître une personne, il faut dans un premier temps localiser le visage dans l'image, en extraire des caractéristiques importantes qui seront ensuite utilisées pour interroger une base de données afin d'identifier la personne [165, 71].

La détection du visage dans une image est cependant une tâche difficile à cause de la variabilité de la taille, de l'apparence et de l'orientation que peut avoir un visage. Les expressions faciales, les occultations et les conditions d'illumination affectent également l'apparence du visage.

Il existe de nombreux travaux concernant la détection du visage dans une image et un excellent état de l'art est proposé par Yang *et al.* dans [162]. Dans cette étude, les auteurs

classent les différentes méthodes de détection du visage dans quatre catégories :

- *Les méthodes basées sur des connaissances* : elles utilisent des règles définies à partir de connaissances à priori sur le visage humain. Typiquement, il s'agit de relations entre les différents éléments caractéristiques du visage.
- *Les méthodes basées sur les invariants* : elles utilisent des éléments du visage qui sont invariants aux changements d'orientation, de point de vue ou d'illuminantion. Les éléments couramment utilisés sont les yeux, la bouche, le nez, les sourcils, etc.
- *Les méthodes basées sur la corrélation* : elles utilisent un ou plusieurs modèle(s) calculé(s) pour décrire le visage entier ou les différents éléments du visage séparement. Le modèle est défini manuellement, il n'est pas appris. Le score de corrélation entre une image donnée et le(s) modèle(s) indique la présence ou non du visage.
- *Les méthodes basées sur l'apparence* : à la différence des méthodes basées sur la corrélation, le modèle (ou template) est obtenu par apprentissage. L'ensemble d'apprentissage est choisi pour capturer la variabilité de l'apparence du visage.

Les méthodes de la première catégorie sont simples à mettre en œuvre mais il est difficile de les étendre à différentes poses et orientations du visage. Les méthodes basées sur les invariants sont plus robustes aux changements de pose et d'orientation, mais il peut être difficile de détecter les éléments du visage en cas d'occultation, de bruit ou de présence de nombreux autres objets dans l'image. Les méthodes de la troisième catégorie sont simples et souffrent des mêmes inconvénients que celles de la première catégorie. Enfin, les méthodes basées sur l'apparence sont très utilisées dans la littérature. La détection du visage peut en effet être considérée comme un problème de décision binaire dans lequel chaque région d'une image est affectée à l'une des deux classes : "visage", "pas visage". Par conséquent, de nombreuses méthodes de classification telles que l'ACP [150], les SVM [106], les champs de Markov [112] et les réseaux de neurones [116] ont été utilisées pour la détection du visage dans une image. Si ces méthodes donnent parfois de très bons résultats, elles sont dans la pratique limitées à de petites variations de l'orientation et de la pose du visage. D'autre part, les résultats dépendent très fortement de l'ensemble d'apprentissage utilisé [57].

Soulignons que cette classification est très sommaire et que de nombreuses méthodes peuvent être classées dans plusieurs de ces catégories. En particulier, les méthodes les plus efficaces sont celles qui utilisent différents types de connaissance et d'information pour assurer la robustesse.

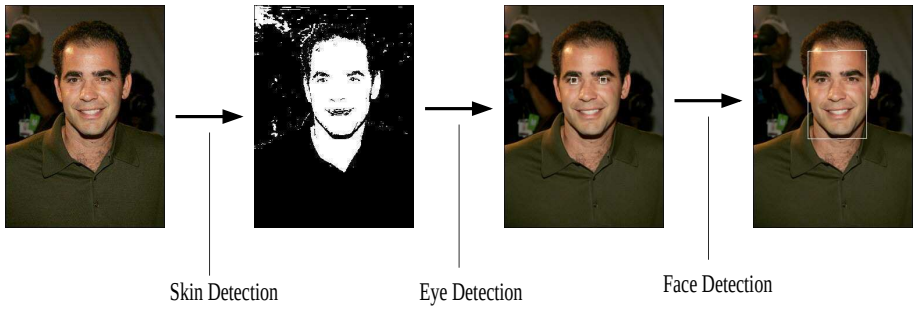



FIG. 7.1 – Principe de la méthode de détection des yeux et du visage.

Il est important de souligner que notre but dans ce chapitre n'est pas la détection du visage, mais la détection des yeux qui sont des éléments caractéristiques du visage et qui seront utilisés pour le suivi dans le chapitre suivant. Dans la plupart des travaux sur la détection des yeux, on commence par la détection du visage afin de réduire l'espace de recherche des yeux [159]. Notre méthode de détection, décrite dans la section suivante, utilise l'apparence pour sélectionner les régions de l'image susceptibles de représenter un visage, ce sont les candidats. Elle extrait ensuite de ces régions, les éléments caractéristiques du visage que sont les yeux.

## 7.2 Une méthode simple et robuste de détection des yeux

La méthode de détection des yeux est basée sur la détection de la peau car une région contenant les yeux, est une région de l'image qui contient beaucoup de pixels correspondant à la peau.

La détection de la peau est utilisée comme étape initiale afin de réduire l'espace de recherche des yeux.

L'image de la figure 7.1 montre les différentes étapes de notre méthode de détection.

### 7.2.1 Détection de la peau

La première étape consiste à détecter les régions de l'image pouvant contenir des yeux. Pour ce faire, nous utilisons le détecteur de peau dans une image couleur présenté dans le chapitre précédent (chapitre 6).

Nous utilisons comme espace de représentation l'espace normalisé $rgb$, et nous adoptons

le modèle gaussien simple pour modéliser la distribution des couleurs dans cet espace.

La figure 7.2 montre quelques exemples de détection de la peau. Comme on peut le constater sur ces exemples, les yeux ne sont pas détectés par le détecteur de peau. C'est cette observation qui va nous guider dans la section suivante pour élaborer une méthode simple et robuste de détetction des yeux.

### 7.2.2 Détection des yeux

Parmis les éléments caractéristiques du visage (les yeux, la bouche, le nez, etc), les yeux peuvent être considérés comme étant les plus stables. De plus, des études en psychologie et en neurosciences ont montré que les yeux sont les éléments du visage qui attirent en premier l'attention des humains et des animaux lorsqu'un visage leur est présenté [43].

La détection des yeux est donc utilisée dans de nombreux algorithmes de détection du visage [159, 57]. Les méthodes classiques inclus la corrélation, la transformée de Hough et l'analyse en composantes principales. La corrélation simple n'est pas robuste à la variation de la taille, et à la rotation du visage. Brunelli et Poggio proposent une approche multi-échelle pour résoudre le problème de l'invariance à l'échelle [20]. Yuile *et al.* proposent un modèle déformable pour suivre les éléments du visage, dont les yeux [163]. Ils décrivent un œil par un modèle paramétrique et utilisent une fonction d'énergie pour lier les contours, les pics et les vallées de l'image aux paramètres du modèle. Pentland *et al.* utilisent une ACP pour la détection des yeux en capturant les variations d'apparance, d'orientation et d'illumination à partir d'un ensemble d'apprentissage [107].

Plusieurs autres méthodes de détection ont été proposées plus récemment. Han *et al.* [49] utilisent des opérations de morphologie mathématique (fermerture et ouverture conditionnelle) pour localiser des ensembles de pixels "ressemblant aux yeux" (eye-analogue pixels) dans une image. Puis, un processus d'étiquetage est utilisé pour rechercher les visages potentiels. Finalement, un réseau de neurones permet d'identifier les visages et leurs positions. Des idées similaires sont utilisées par Wu et Zhou dans leurs travaux [159]. Ils utilisent les informations de taille et d'intensité pour trouver les "eye-analogue segments", et exploitent les relations géométriques entre les différentes paires pour trouver les yeux. Le visage obtenu par la position des yeux est ensuite vérifié par corrélation avec un ensemble de 8 composantes principales obtenues par apprentissage. Kawaguchi et Rizon utilisent quant à eux, les contours de l'image pour localiser l'iris de l'œil à l'aide de la transformée de Hough [65].

Dans les travaux présentés dans [49] et [159], les yeux sont détectés en se basant sur l'hy-

FIG. 7.2 – Exemples de détection de la peau. De gauche à droite : image originale et résultat de la détection.

pothèse que la région d'un œil est plus sombre que les autres parties du visage. Wu et Zhou cherchent donc les "eye-analogue segments" en recherchant les parties de l'image qui sont plus sombres que les régions avoisinantes. Cette approche présente quelques inconvenients. Outre le fait que l'hypothèse n'est pas toujours vérifiée (les sourcils sont généralement plus sombres que les yeux), il faut rechercher les pairs d'yeux possibles dans toute l'image. Ce qui augmente la complexité de la méthode quand la taille de l'image augmente. D'autre part, il est possible de trouver des régions qui satisfont à cette hypothèse en dehors du visage. Ce qui justifie la nécessité d'une étape de vérification par réseaux de neurones dans [49] ou par ACP dans [159]. On a donc besoin d'une phase d'apprentissage.

Nous proposons une méthode de détection simple qui ne nécessite aucun apprentissage de l'apparence de l'œil et qui est assez robuste, notamment à la rotation.

### Notre méthode de détection des yeux

La méthode de détection des yeux est basée sur l'observation suivante : *dans une région représentatnt un visage, les yeux ne sont pas détectés par le détecteur de peau.* On peut vérifier cette observation sur les images de la figure 7.2 par exemple.

Il est clair que les yeux se situent dans une région de peau, elle même susceptible de représenter un visage. Dans une région de l'image obtenue par le détecteur de peau, on recherche donc les yeux parmis les *trous* de la région. Un trou est une zone de "non peau" à l'intérieur d'une région de peau. Nous extrayons les trous dans une région de peau en utillisant un processus d'étiquetage en composantes connexes. La taille maximale des régions recherchées par l'analyse en composantes connexes est fixée en fonction de la taille de la région de peau dans laquelle l'on cherche les trous.

La recherche de trous permet d'identifier les zones de l'images susceptibles de correspondre à des yeux. On trouve en général, les deux yeux, la bouche, les sourcils et parfois les narines ou d'autres zones due à l'imperfection du détecteur de peau. Nous désignons ces zones de l'image par le terme de "œil potentiel". Un exemple d'extraction des yeux potentiels est donné par la figure 7.3.

Chaque œil potentiel est ensuite représenté par une ellipse. Soit $R_k$ une région définissant un œil potentiel et $(x_k, y_k)$ le centre de $R_k$. On extrait de $R_k$ les paramètres $\{a_k, b_k, \theta_k\}$ qui sont respectivement, la longueur du grand axe, la longueur du petit axe et l'orientation du grand axe de l'ellipse.

Une fois les yeux potentiels détectés, les deux régions représentant les deux yeux sont sélectionnées en utilisant des connaissances anthropologiques qui caractérisent les yeux sur

FIG. 7.3 – Recherche des yeux potentiels. De gauche à droite : résultat de la détection de la peau et les zones représentatnt les yeux potentiels.

un visage humain. Les règles utilisées sont basées sur les propriétés géométriques des yeux, et sur la distance inter-oculaire qui est une bonne mesure de caractérisation des éléments du visage [44].

Soient $R_i$ et $R_j$ deux régions définissant deux yeux potentiels. Alors, le couple $(R_i, R_j)$ correspond à une paire d'yeux si les contraintes définies par les équations suivantes sont satisfaites :

$$\begin{cases} 1 < \frac{a_i}{b_i} < 3 \\ 1 < \frac{a_j}{b_j} < 3 \end{cases} \tag{7.1}$$

$$|\theta_i - \theta_j| < 15^o \tag{7.2}$$

$$\frac{a_i + a_j}{2} < d_{ij} < 3\,\frac{a_i + a_j}{2} \tag{7.3}$$

Les contraintes des équations (7.1) et (7.3) traduisent le fait que, pour un être humain, si on note, respectivement, $w_e$ et $h_e$ la largeur et la hauteur d'un œil, alors on a [5] :

$$\begin{cases} w_e \simeq 2h_e \\ d_{ij} \simeq 2w_e \end{cases} \tag{7.4}$$

$d_{ij}$ étant la distance entre les centres des deux régions.

La contrainte de l'équation (7.2) traduit le fait que les grands axes des deux ellipses


FIG. 7.4 – Règles utilisées pour la détection des yeux. (a) : la distance inter-oculaire est proportionnelle à la taille des yeux ; (b) : les axes des deux ellipses sont alignés.

possèdent la même orientation. Une contrainte supplémentaire est l'alignement des grands axes, i.e. que les grands axes des deux ellipses appartiennent à une même droite. Par exemple, sur l'exemple de la figure 7.4(b), les deux régions $R_i$ et $R_j$ respectent la contrainte d'alignement alors que les régions $R_i$ et $R_l$ ne la satisfont pas, bien qu'elles aient la même orientation.

En utilisant ces régles simples, la méthode détecte les yeux mais aussi, souvent, les sourcils qui respectent ces mêmes règles. La figure 7.5 montre des exemples de détection en utilsant les règles ci-dessus. Comme on peut le voir sur les exemples des figures 7.5(c) et 7.5(d), les sourcils sont égalements détectés.

Pour distinguer les yeux des sourcils, nous utilisons l'information de l'intensité lumineuse de chacune des régions détectées en nous appuyant sur l'observation suivante : *un œil contient une région centrale plus sombre que les deux régions extérieurs.*

Une simple analyse de l'histogramme des niveaux de gris de la région permet donc d'écarter les sourcils, car l'histogramme d'une région représentant les sourcils a un seul pic, tandis que l'histogramme d'une région représentant un œil en a deux. Si les niveaux de gris sont compris entre 0 et 1, l'histogramme d'une région représentant les sourcils ne montre aucun pic au-delà de la valeur égale à 0.4, voir figure 7.6(a). Pour une région représentant un œil, l'histogramme présente deux pics, un pic de part et d'autre de cette valeur comme le montre la figure 7.6(b). Nous prendront donc comme seuil pour l'analyse des histogrammes, la valeur $s = 0.4$. Des exemples de résultats obtenus après l'analyse d'histogramme sont donnés par la figure 7.7.

### 7.2.3 Détection du visage

Une fois les yeux détectés, la position du visage peut être très simplement déduite de celles des yeux en utilisant la distance inter-oculaire. Un algorithme de détection des yeux peut donc être déduit de la méthode de détection des yeux proposée. Voir figure 7.8.

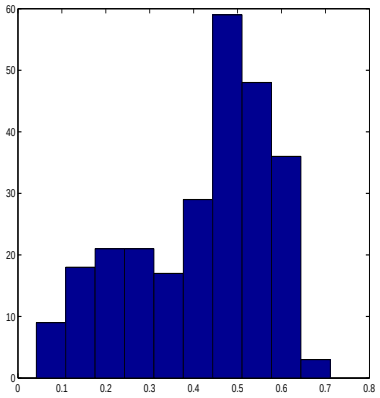
a)

b)

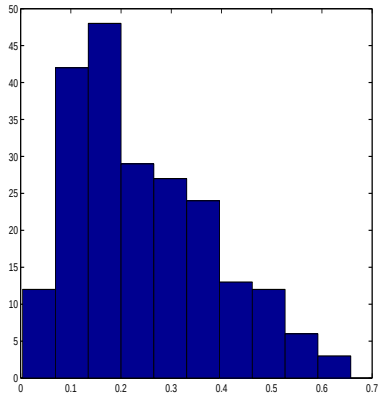
c)

d)

FIG. 7.5 – Exemple de détection des yeux. (a) et (b) détection correcte des yeux ; (c) et (d) détection incorrecte des sourcils.




a)




b)

Fig. 7.6 – Analyse d'histogrammes. (a) histogramme d'une région représentant un œil ; (b) histogramme d'une région représentant les sourcils.

Fig. 7.7 – Exemple de détection des yeux après analyse d'histogramme. De gauche à droite : résultats avant et après l'analyse d'histogramme.

Etant donné une image $I$
(1) détecter les régions de peau dans $I$ ;
(2) détecter les yeux :
    2.1) trouver les trous dans les régions de peau ;
    2.2) utiliser les règles des équations 7.1, 7.2 et 7.3 pour trouver les yeux potentiels ;
    2.3) sélectionner les yeux par une analyse d'histogramme.
(3) localiser le(s) visage(s) dans $I$.

FIG. 7.8 – Algorithme de détection du visage dans une image couleur.

## 7.3 Evaluation Expérimentale

Dans cette section, nous évaluons plus en détails la performance de notre algorithme en utilisant une base d'image publiquement disponible, ainsi que quelques autres images issues de l'Internet. Nous comparons notre méthode à deux autres méthodes qui utilisent la même base d'images de test.

Il est important de souligner que nous évaluons ici, la performance du détecteur des yeux dans une image couleur, et non celle du détecteur de visage.

### 7.3.1 Critère d'évaluation

Même si l'on peut évaluer de manière qualitative les résultats présentés sur les figures 7.5 et 7.7, il est nécessaire de définir un critère d'évaluation quantitatif.

Un premier critère est *l'erreur relative* introduite par Jesorsky *et al.* dans [60]. Celle-ci est définie par l'équation suivante :

$$err = \frac{max(d_l, d_r)}{d_{lr}} \tag{7.5}$$

où $d_l$ et $d_r$ désignent respectivement la disparité de l'œil gauche et celle de l'œil droit, i.e. la distance entre entre la vraie position de l'œil (définie manuellement) et la position détectée. $d_{lr}$ désigne la distance Euclidienne entre les vraies positions des deux yeux.

La détection est généralement considérée comme correcte si $err < 0.25$ [159].

Song *et al.* [140] définissent un autre critère d'évaluation. Ils considèrent que la détection d'une paire d'yeux est correcte si :

$$max(d_l, d_r) < \alpha.r \tag{7.6}$$

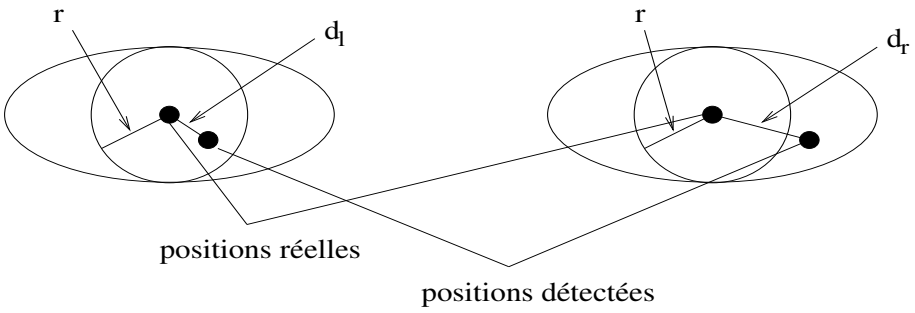



FIG. 7.9 – Evaluation de la détection des yeux. La détection est correcte si la position détectée se situe à l'intérieur de l'iris de l'œil.

où $r$ désigne le rayon de l'iris de l'œil et $\alpha$ est un facteur constant.

Les deux critères sont en fait équivalents. En effet, si on considère que le rayon de l'iris est égal au quart de la largeur d'un œil (ce qui est une hypothèse valable), il est facile de montrer que le critère de l'équation 7.5 est équivalent à celui de l'équation 7.6 pour $\alpha = 2$.

Nous utilisons le critère de l'équation 7.6 avec $\alpha = 1$. En d'autre termes, la détection d'une paire d'yeux est considérée comme correcte, si la position détectée de chaque œil se situe à l'intérieur de l'iris de l'œil. La figure 7.9 montre un exemple dans lequel l'œil gauche est correctement détecté tandis que l'œil droit sera considéré comme incorrecte.

### 7.3.2 Résultats avec la base AR

Afin de comparer notre méthode de détection des yeux avec celles proposées par Kawaguchi et Rizon [65], et par Song *et al.* [140], nous utilisons la base d'images AR[1] [85]. Il s'agit d'une base d'images couleur représentant des visages vus de face avec différentes expressions faciales, des conditions variées d'illumination et des occultations.

Pour une comparaison directe de nos résultats avec ceux fournis par les auteurs dans [65] et [140], nous utilisons le même sous-ensemble d'images de la base AR. Cet ensemble, que nous noterons AR-63, contient 63 images de 21 personnes (12 hommes et 9 femmes). Les images de AR-63 sont prises dans des conditions d'illumination normale et elles présentent trois types d'expressions faciales : neutre, sourire et colère.

En utilisant le critère de détection précédemment défini, nous obtenons un taux de détections correctes égale à 98.4% (il y a un seul échec) pour l'ensemble de la base AR-63. Quelques exemples de détection sont donnés sur la figure 7.10. La figure 7.11 présente

[1]LA base AR est disponible à l'adresse suivante : `http://cobweb.ecn.purdue.edu/~aleix/aleix_face_DB.html`.

| Méthode | Taux de détection |
|---|---|
| Notre méthode | 98.4 |
| Song *et al.* [140] | 98.4 |
| Kawaguchi et Rizon [65] | 96.8 |

TAB. 7.1 – Comparaison des différents méthodes de détection des yeux avec la base AR-63.

l'exemple dans lequel la méthode de détection échoue. Cet échec s'explique par le fait qu'il est impossible de distinguer les yeux des sourcils dans ce cas, car les yeux sont presque fermés. L'histogramme de la région des yeux ne présente donc qu'un seul pic dans ce cas.

Ces résultats sont comparables à ceux obtenus par Kawaguchi et Rizon et par Song *et al.* avec les mêmes images. L'ensemble des résultats est rassemblé dans le tableau 7.1. On note que nous obtenons le même résultat que la méthode décrite dans [140], et un résultat légèrement supérieur à celui de la méthode développée dans [65].

D'autre part, les méthodes développées dans [140] et [65] supposent que le visage est déjà détecté et ne s'intéressent donc qu'à la détection des yeux sur le visage présenté dans une position verticale. Elles ne peuvent donc pas être employées dans les cas où la position du visage n'est pas connue a priori. Notre méthode de détection au contraire, peut faire partie du processus de détection du visage. Elle est donc capable de détecter les yeux même lorsque le visage n'est pas dans une position verticale. Voir la section suivante.

### 7.3.3 Résultats avec des images de scènes complexes

Les images de la base AR sont prises dans un environnement contrôlé et présentent un fond fixe et homgène. Dans cette section, nous évaluons notre algorithme en utilisant des images de scènes complexes. Les images sont issues de l'Internet et présentent donc des conditions d'illumination variées et des fonds complexes. Il peut également y avoir une ou plusieurs personnes sur une même image.

Les résultats obtenus, voir les figure 7.12 et 7.13 pour des exemples, montrent que la méthode est capable de détecter les yeux même en cas de rotation plane, i.e que le visage est vu de face mais n'est pas vertical, ou lorsque le visage est vu légèrement de profil (figure 7.12(b)). Elle est également capable de détecter plusieurs paires d'yeux lorsque plusieurs visages sont présents dans la scène (figure 7.13).

Soulignons toutefois qu'il y a des cas où la méthode échoue. En particulier, lorsque l'un des deux yeux est fermé (ou les deux yeux sont fermés) comme sur l'exemple de la figure 7.13(b). On notera que la fille en bas à gauche de la figure a les yeux fermés. L'analyse

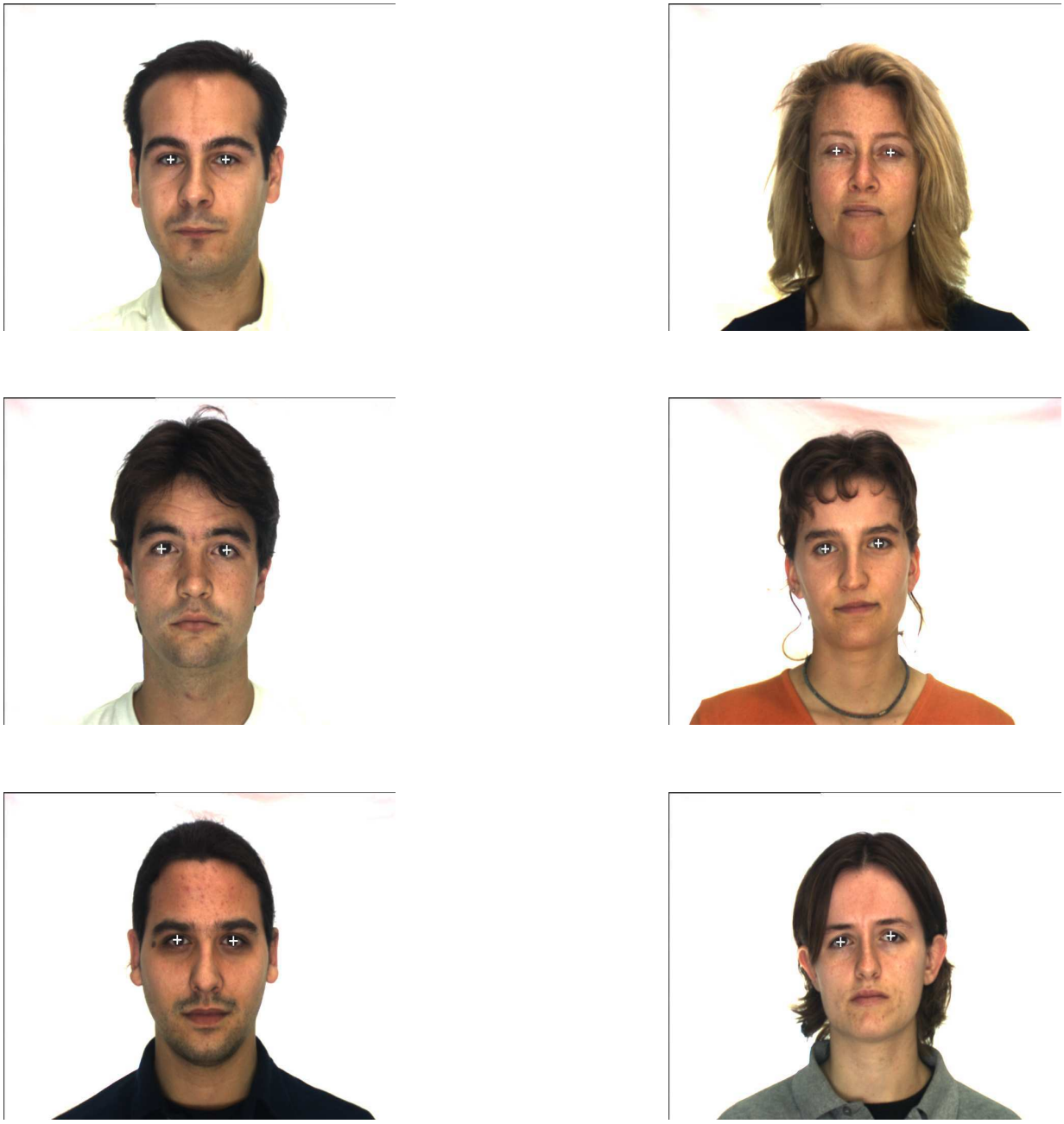
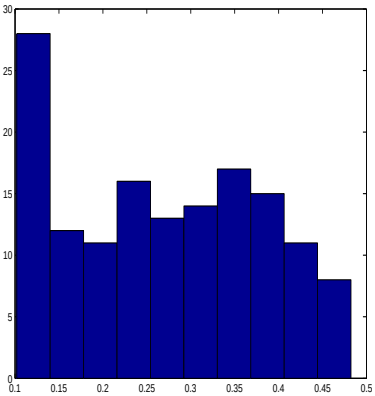


Fig. 7.10 – Exemple de détection des yeux avec la base AR-63.

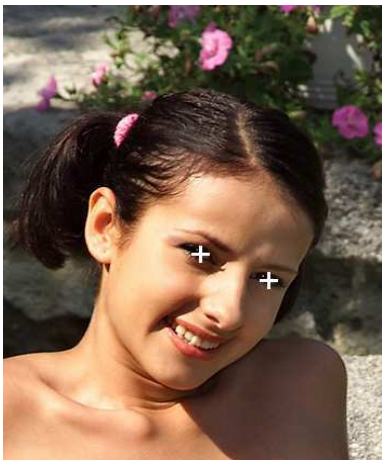
a)




b)

FIG. 7.11 – Cas dans lequel la détection des yeux échoue. (a) les yeus et les sourcils sont détectés. (b) l'histogramme de la région de l'œil ne permet pas de distinguer les yeux des sourcils.

a)



b)

FIG. 7.12 – Exemple de détection des yeux dans des scènes complexes.


a) b)

Fig. 7.13 – Exemple de détection multiple.

d'histogramme ne peut pas, dans ce cas, distinguer les yeux des sourcils. Il existe une autre situation dans laquelle la méthode de détection des yeux échoue à cause du détecteur de la peau. C'est le cas de la fille au centre de l'image de la figure 7.13(a). Les yeux sont ouverts, mais la fille étant légèrement de profil, son œil n'apparaît pas comme un trou dans la région de peau détectée.

### 7.3.4 Remarques

La méthode de détection des yeux proposée, permet d'obtenir des résultats très satisfaisants dans les cas simples (visage de face et condition d'illumination normale) et donne des résultats acceptables dans des cas plus complexes. Elle possède plusieurs avantages dont les principaux sont :

- la rapidité, due à la vérification par des règles simples ;
- l'invariance à la rotation ;
- la capacité de détecter plusieurs visages ;

De plus, elle ne nécessite aucune phase d'apprentissage, contrairement aux méthodes présentées dans [65] et [140]. Cependant, les résultats obtenus par l'algorithme de détection des yeux, dépendent très fortement de ceux obtenus par l'algorithme de détection de la peau. Si les régions de peau obtenues sont incorrectes, alors l'algorithme de détection des yeux échoue.

La méthode de détection du visage qui se rapproche le plus de la notre est celle développée par Hsu *et al.* dans [57]. Les auteurs basent leur algorithme de détection du visage sur un détecteur robuste de la peau également. Le détecteur de la peau est défini dans l'espace

de chrominance $YCrCb$ (voir chapitre 6). La différence avec notre méthode se situe au niveau de l'extraction des yeux et de la bouche. Hsu *et al.* définissent un modèle de l'œil et de la bouche en utilisant les composantes de chrominance et de luminance. Les yeux et la bouche sont ensuite extraits dans les régions de peau détectées dans l'image en utilisant les modèles précédemment définis. Chaque triangle formé par les deux yeux et la bouche est enfin vérifié en utilisant la transformée de Hough. Il faut donc une phase préalable de définition des modèles des yeux et de la bouche, ce qui n'est pas le cas dans notre approche.

De plus, l'utilsation de contraintes simples rend notre méthodes beaucoup de rapide que celle proposée dans [57]. En effet, le temps d'excécution moyen rapporté par les auteurs pour traiter une image de résolution 640x480 est de 24.71 s avec un processeur de 1.7 GHz. Le temps d'exécution pour traiter une image de la même taille est de 2.3 s avec notre méthode sur Pentium IV (2.8 GHz).

Soulignons enfin que notre méthode de détection du visage est assez simple et qu'elle échoue dans les cas où la taille du visage dans l'image est très petite. Dans ces cas, le détecteur de la peau identifie les régions à explorer mais l'extraction des yeux échoue du fait de la taille très réduite. Nos expériences nous permettent de noter que la méthode de détection du visage échoue si la distance inter-oculaire est inférieure à 20 pixels.

Dans des cas aussi difficiles, un étape préalable de détection du visage s'avère nécessaire. Les méthodes basées sur l'apprentissage sont plus robustes, et capables de détecter des visages dont la taille est de l'ordre de 20x20 pixels. En particulier, la méthode développée par Viola et Jones [157] permet d'obtenir d'excellents résultats et elle est aujourd'hui considéré comme l'une des meilleures approches [44]. Elle est basée sur l'utilisation de plusieurs classiffieurs disposés en cascade. Chaque étage de la cascade ne considère que les éléments de l'image correctement classés par l'étage précédente. Ce qui rend la méthode assez rapide.

## 7.4 Conclusion

Dans ce chapitre, nous avons développé une méthode simple et efficace de détection des yeux dans une image couleur. La méthode est basée sur le détecteur de la peau présenté dans le chapitre 6, suivi par la détection des yeux. La détection des yeux utilise des règles simples fondées sur la configuration géométrique du visage et le rapport entre la distance inter-oculaire et la taille des yeux.

Cette méthode nous permet d'obtenir d'excellents résultats dans les cas simples (par

exemple avec les images de la base AR) mais échoue quand le visage est vu de profil ou lorsque la taille du visage est tès petite (distance inter-oculaire inférieure à 20 pixels). Des méthodes de détection plus robustes existent. Elles sont basées sur la détection robuste du visage par exemple, par la méthode de Viola et Jones [157]. Mais notre méthode possède l'avantage de la simplicité et de la rapidité. De plus, elle ne nécessite pas de phase d'apprentissage.

Elle est donc largement suffisante pour l'application envisagée, à savoir le suivi du visage dans une séquence d'images. C'est ce nous verrons dans le chapitre suivant.

# Chapitre 8

# Suivi du visage dans une séquence d'images

Ce chapitre aborde le problème du suivi d'objets mobiles dans une séquence d'images. Nous nous intéressons plus particulièrement au suivi d'un ou de plusieurs visage(s) dans une séquence d'images en utilisant la méthode de détection présentée dans le chapitre précédent comme étape d'initialisation. Nous commencons par présenter la problématique du suivi de points ainsi que deux des principales méthodes utilisées dans la littérature. Nous décrivons ensuite une méthode de suivi basée sur la méthode de relaxation présentée dans le chapitre 4. Nous terminons le chapitre par une évaluation expérimentale des différentes méthodes et par la mise en évidence de l'intérêt de l'utilisation d'informations contextuelles dans le suivi.

## 8.1 Introduction

Le suivi d'objets dans une séquence d'images est une tâche importante dans de nombreuses applications de la vision par ordinateur telles que la vidéo-surveillance [67, 73, 24], les interfaces homme-machine intelligentes [17, 148, 104], la réalité augmentée [35], l'assistance à la conduite [50], pour n'en citer que quelques unes.

L'objectif principal du suivi est, grossièrement, de prédire et d'estimer la position de l'objet cible dans chacune des images de la séquence en dépit des changements de l'apparence, de l'illumination et de la pose de l'objet [25].

Il existe différentes méthodes de suivi et le choix d'une méthode dépend de l'application envisagée.

Nous nous intéressons ici au suivi d'un ou de plusieurs visages dans une séquence d'images. Détecter le visage, le reconnaître si besoin et le suivre dans une séquence d'images est à la base de nombreuses applications faisant intervenir les intéractions homme-machine. Dans l'exemple présenté par Toyama [148], l'utilisateur déplace un curseur sur un écran sans utiliser les mains. Un système de vision par ordinateur suit le visage de l'utilisateur et déplace le curseur en temps réel en fonction de l'orientation du nez de ce dernier. Un autre exemple d'application est la reconnaissance et la classification d'expressions faciales. Par exemple, le système LAFTER proposé par Oliver *et al.* [104] permet de détecter un visage, la bouche sur ce visage, puis de reconnaître les expressions du type : mécontent, sourire, bouche ouverte et bouche fermée.

Les différentes méthodes de suivi du visage peuvent se classer dans deux principales catégories. D'une part, nous avons les méthodes qui considèrent comme zone d'intérêt de l'image le visage entier. Après détection dans la première image de la séquence, la position de cette zone d'intérêt est estimée dans les images suivantes de la séquence [125, 14]. L'algorithme Mean-Shift [22] ou ses variantes, par exemple Camshift (Continuously Adaptive Mean Shift) [18], sont très utilisés dans ce cas. La zone d'intérêt est détectée dans l'image en utilisant par exemple, un détecteur de peau. D'autre part, nous avons les méthodes qui s'intéressent à des points d'intérêt définis sur le visage et ce sont ces points qui sont suivis dans les autres images de la séquence [25, 141, 16]. L'algorithme génaralement utilisé pour le suivi de points d'intérêt est l'algorithme KLT [82]. Nous le détaillons dans la section 8.3.1.

Dans le chapitre précédent, nous avons proposé une méthode de détection du visage basée sur la détection des yeux. Puisque le détecteur du visage nous donne directement les positions de certains points d'intérêt sur le visage, nous nous intéressons dans la suite de ce chapitre aux méthodes de la seconde catégorie. Ces méthodes se rapprochent des méthodes de mise en correspondance d'images, car il faut trouver dans deux images des points qui sont "similaires". Toutefois, dans le cas du suivi, on suppose que les changements de position et d'apparence d'un objet dans deux images consécutives de la séquence sont faibles [26]. Cette propriété est utilisée pour mettre en œuvre des méthodes de suivi efficaces.

Dans la section suivante, nous indiquons comment sont sélectionnés les points à suivre. Puis, nous présentons deux méthodes de suivi couramment utilisées dans la littérature. Enfin, nous proposons une méthode de suivi qui utilise la méthode de relaxation développée dans le chapitre 4.

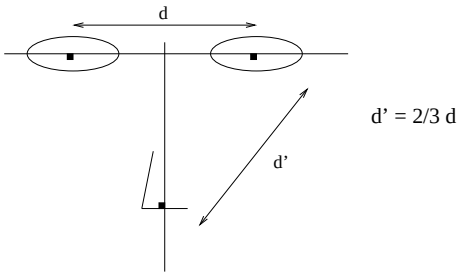



FIG. 8.1 – Configuration des points d'intérêt sur le visage.

## 8.2 La détection des points d'intérêt

Afin de suivre le visage dans toute la séquence, il faut dans un premier temps le détecter dans la première image de la séquence. L'objectif étant ici de suivre quelque points particuliers du visage, par exemple par l'algorithme KLT, nous utilisons la méthode que nous avons développé dans le chapitre 7 pour détecter les yeux dans une image couleur. Cette méthode nous permet d'obtenir la position des yeux et donc celle du visage dans la première image de la séquence. Nous considérerons également pour le suivi, la position du nez sur le visage. Celle-ci peut être facilement obtenue à partir des positions des deux yeux en utilisant la distance inter-occulaire et la configuration géométrique des éléments du visage. Nous utilisons le fait que le nez se situe (de manière approximative) à égale distance des deux yeux, et que la distance entre un œil et le nez est environ égale à la distance inter-occulaire multipliée par 2/3. Voir figure 8.1 pour une représentation graphique.

Nous nous limitons ici volontairement au suivi de trois points d'intérêt, mais on peut suivre un nombre plus important de points d'intérêt définis sur le visage. Par exemple, Bourel *et al.* [16] suivent un ensemble de 12 points d'intérêt sélectionnés manuellement sur la première image de la séquence.

Un exemple de détection des régions d'intérêt à suivre est présenté sur la figure 8.2.

## 8.3 Deux méthodes de suivi des points d'intérêt

Une fois les points d'intérêt détectés dans la première image de la séquence, leur position est estimée dans les images suivantes en utilisant une méthode de suivi. Dans cette section, nous présentons deux méthodes utilisées dans la littérature pour réaliser le suivi de points d'intérêt dans une séquence d'image.

Fig. 8.2 – Exemple de détection des points d'intérêt. De gauche à droite : première image de la séquence ; les zones d'intérêt, yeux et nez, détectées de manière automatique.

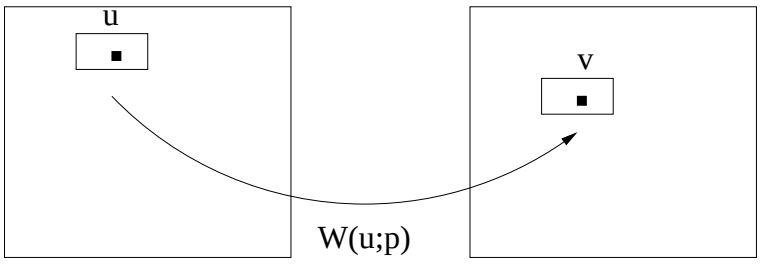



FIG. 8.3 – Principe de la méthode KLT. On recherche la transformation **W** qui minimise la somme des erreurs quadratiques.

### 8.3.1 L'algorithme KLT

L'algorithme KLT est une méthode largement utilisée en vision par ordinateur aussi bien pour l'alignement ou le recalage d'images, que pour le suivi de régions d'intérêt dans une séquence d'images. Il tire son nom des initiales de ses auteurs. La première version a été proposé en 1981 par Lucas et Kanade [82] dans le cadre du recalage d'images. Elle a ensuite été développé par Tomasi et Kanade [147, 128], d'où le nom KLT pour Kanade-Lucas-Tomasi.

Une description très complète de l'algorithme KLT est présenté par Baker et Matthews dans [6]. Ici, nous nous contentons de décrire brièvement le principe de la méthode.

Soient $I$ et $J$ deux images, et $\mathbf{u} = (u_x, u_y)^T$ un point de l'image $I$. On souhaite trouver le point $\mathbf{v}$ de $J$ tel que les quantités $I(\mathbf{u})$ et $J(\mathbf{v})$ soient "similaires". On notera que cela s'apparente au problème de la mise en correspondance d'image, et l'algorithme KLT est d'ailleurs souvent utilisé pour cette application.

L'objectif de la méthode décrite par Lucas et Kanade [82] consiste à trouver une transformation $\mathbf{W}$ telle que :

$$\mathbf{v} = \mathbf{W}(\mathbf{u}; \mathbf{p})$$

où $\mathbf{p}$ désigne les paramètres de la transformation reliant les deux images $I$ et $J$.

Par exemple, si $\mathbf{W}$ est une translation, alors $\mathbf{W}(\mathbf{u}; \mathbf{p}) = \mathbf{u}+\mathbf{p}$. Autrement dit, $\mathbf{v} = \mathbf{u}+\mathbf{p}$.

Pour trouver la meilleure transformation, on minimise la quantité suivante, qui évalue la similarité entre les régions des deux images autour des points $\mathbf{u}$ et $\mathbf{v}$ :

$$\epsilon(\mathbf{p}) = \sum_{\mathbf{x}\in\Omega} [J(\mathbf{W}(\mathbf{x}; \mathbf{p})) - I(\mathbf{x})]^2 \tag{8.1}$$

avec $\Omega$ un voisinage du point $\mathbf{u}$, appelé aussi fenêtre d'intégration.

Le résolution de l'équation (8.1) est un problème difficile d'optimisation non-linéaire. En effet, comme souligné par Baker et Matthews [6], même si $\mathbf{W}(\mathbf{u};\mathbf{p})$ est linéaire en $\mathbf{p}$, les valeurs des pixels $I(\mathbf{x})$ ne sont en général pas linéaires en $\mathbf{x}$. En fait, les valeurs des pixels $I(\mathbf{x})$ sont indépendantes des coordonnées $\mathbf{x}$ du pixel.

Pour optimiser l'expression de l'équation (8.1), l'algorithme KLT procède donc de manière itérative. On suppose connu une estimation courante de $\mathbf{p}$ et on cherche un incrément $\Delta\mathbf{p}$ tel que la quantité suivante soit minimale :

$$\epsilon(\mathbf{p}) = \sum_{\mathbf{x}\in\Omega}[J(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - I(\mathbf{x})]^2 \tag{8.2}$$

On minimise $\epsilon(\mathbf{p})$ par rapport à $\Delta\mathbf{p}$ et les paramètres sont mis à jour :

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \tag{8.3}$$

Les deux étapes, optimisation et mise à jour, sont repétées jusqu'à ce que la méthode converge. Un test de convergence simple est défini par la norme de l'incrément : $\|\Delta\mathbf{p}\| \leq \varepsilon$.

Si les mouvements de l'objet à suivre sont faibles d'une image à la suivante, on peut raisonnablement supposer que la transformation est une simple translation, i.e. $\mathbf{W}(\mathbf{x};\mathbf{p}) = \mathbf{x}+\mathbf{p}$. Dans ce cas, il est facile de montrer que l'incrément $\Delta\mathbf{p}$ est obtenu par l'équation suivante :

$$\Delta\mathbf{p} = H^{-1}\sum_{\mathbf{x}\in\Omega}\nabla J[I(\mathbf{x}) - J(\mathbf{x}+\mathbf{p})] \tag{8.4}$$

avec

$$H = \sum_{\mathbf{x}\in\Omega}(\nabla J)^T(\nabla J)$$

et $\nabla J = (\frac{\partial J}{\partial x}, \frac{\partial J}{\partial y})$ le gradient image au point $\mathbf{x}$.

$H$ est donc une matrice de taille 2x2 qui s'écrit :

$$H = \begin{pmatrix} (\frac{\partial J}{\partial x})^2 & \frac{\partial J}{\partial x}\frac{\partial J}{\partial y} \\ \\ \frac{\partial J}{\partial x}\frac{\partial J}{\partial y} & (\frac{\partial J}{\partial y})^2 \end{pmatrix}$$

Le vecteur de déplacement $\Delta\mathbf{p}$ est obtenu par la résolution d'un système linéaire de deux équations à deux inconnues. L'algorithme qui en découle est présenté sur la figure 8.4.

Le lecteur intéressé peut se repporter à l'annexe E pour une description de l'algorithme

Itérer
(1) Calculer $J(\mathbf{x}+\mathbf{p})$
(2) Calculer la différence $I(\mathbf{x}) - J(\mathbf{x}+\mathbf{p})$
(3) Evaluer le gradient image $\nabla J$ en $(\mathbf{x}+\mathbf{p})$
(4) Calculer la matrice Hessienne $H = \sum(\nabla J)^T(\nabla J)$
(5) Calculer $\Delta\mathbf{p}$ en utilisant l'équation (8.4)
(6) Mettre les paramètres à jours : $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$
Tant que $\|\Delta\mathbf{p}\| > \varepsilon$

FIG. 8.4 – Algorithme KLT dans le cas d'une translation

KLT dans le cas d'une transformation quelconque. Rappellons qu'une présentation très complète de l'algorithme est donnée dans [6].

### 8.3.2 L'algorithme "block-matching"

L'algorithme "block-matching" décrit dans [143] et utilisé par Spors et Rabenstein [141] pour le suivi des yeux, consiste à rechercher pour chaque point d'intérêt de l'image $I_t$, le point de l'image $I_{t+1}$ qui maximise un score de corrélation. La recherche est effectuée dans une fenêtre d'intégration $\Omega$.

On commence par détecter les positions des points d'intérêt dans la première image de la séquence, et pour chaque point on extrait un masque de corrélation qui servira de référence. Dans les images suivantes de la séquence, les nouvelles positions des points sont estimées par la recherche des points qui minimisent la distance de block définie par :

$$\eta_{x,y} = \frac{3}{4}\|\mathbf{p}_{ref} - \mathbf{p}_{x,y}\|^2 + \frac{1}{4}\|\mathbf{p}_{tmp} - \mathbf{p}_{x,y}\|^2 \tag{8.5}$$

où $\mathbf{p}_{ref}$ désigne le masque de référence, $\mathbf{p}_{tmp}$ le masque extrait de l'image précédente et $\mathbf{p}_{xy}$ celui extrait de l'image traitée.

Pour chaque point d'intérêt $\mathbf{x}$, le point $\mathbf{x}^* \in \Omega$ qui donne la distance de block minimale est retenu comme la nouvelle position de $\mathbf{x}$.

On recherche donc les correspondants dans $I_{t+1}$, des points de $I_t$, en utilisant comme mesure de similarité la distance définie par l'équation (8.5).

La prise en compte d'un masque de référence dans le calcul de la distance, permet de limiter les effets dus à un changement d'illumination pendant l'acquisition de la séquence

d'images. Les masques de référence sont mis à jour par l'équation suivante :

$$\mathbf{p}_{ref}[t+1] = \mathbf{p}_{ref}[t] + (\overline{\mathbf{p}_{ref}[t]} - \overline{\mathbf{p}_{\mathbf{x}^*}}) \tag{8.6}$$

où $\overline{A}$ désigne la valeur moyenne du masque $A$.

### 8.3.3 Mise en œuvre et remarques

Pour la mise en œuvre des deux méthodes de suivi décrits ci-dessus, il faut choisir la taille des masques utilisés pour évaluer les mesures de similarité, ainsi que la taille de la fenêtre d'intégration $\Omega$ (la fenêtre de recherche des points dans $I_{t+1}$). Pour que les algorithmes soient indépendants des séquences traitées, nous prenons un masque rectangulaire dont la largeur est égale à la distance inter-occulaire, et la hauteur égale à la moitié de cette distance. La fenêtre d'intégration est une fenêtre carrée dont de côté est égal à la distance inter-occulaire. Ainsi, ces paramètre ne dépend que de la taille du visage détecté dans la première image de la séquence.

Nous utilisons deux séquences d'images. La première séquence, *Antonio*, est disponible à l'addresse suivante `http://research.microsoft.com/vision/cambridge/i2i/`. Elle présente une personne assise face à la caméra et bougeant la tête de gauche à droite et de haut en bas, tout en parlant.

La deuxième séquence, *Sylvain*, est une séquence acquise par nous même. Elle présente une personne faisant face à la caméra et se déplaçant latéralement par rapport à celle-ci.

Les images sont de résolution égale à 320x240. La distance inter-occulaire vaut environ à 32 pixels dans le cas de la séquence *Antonio*, et elle vaut environ 26 pixels dans le cas de la séquence *Sylvain*.

#### Résultats avec la séquence *Antonio*

Les résultats obtenus avec la séquence *Antonio* sont présentés sur la figure 8.5. Comme on peut le constater, l'algorithme KLT permet un suivi efficace des régions d'intérêt dans une séquence d'images. Les yeux et le nez sont correctement suivis dans les 50 premières images de la séquence. On note toutefois, que les erreurs de localisation des points à suivre s'accumulent d'une image à la suivante. Ce qui conduit à des localisations moins précises, par exemple la 50ème image de la séquence, et à la perte de certains points comme sur la 60ème image de la séquence. On considérera dans ce cas, que l'œil gauche du visage est

perdu au cours du suivi car le résidu, la somme des erreurs entre les valeurs des pixels de deux régions $R_t$ et $R_{t+1}$, dépasse un seuil prédéfini.

Les résultats obtenus par l'algorithme "block-matching", partie basse de la figure 8.5, sont moins bons que ceux obtenus avec KLT. La position du nez est mal estimée dès la 15ème image de la séquence et le reste du suivi est incorrect. Cela est dû principalement au fait que, les nouvelles positions des points sont estimées en utilisant un score de corrélation (voir équation (8.5)) et que ce score conduit souvent à des erreurs de localisation. Notons que la séquence présente des changements importants de luminosité. D'autre part, comme dans le cas de l'algorithme KLT, les erreurs s'accumulent d'une image à la suivante. De plus, si on se trompe à un instant $t$, on se trompe à tous les instants suivants car le masque de référence sera incorrectement mis à jour (voir équation (8.6)).

#### Résultats avec la séquence *Sylvain*

La figure 8.6 montre les résultats du suivi avec la séquence *Sylvain*. Les deux méthodes, KLT et "block-matching", donnent des résultats peu satisfaisants dans ce cas. En effet, le nez est vite perdu (à partir de la 10ème image), et seul un œil est correctement suivi dans le reste de la séquence. Ces mauvais résultats s'expliquent par le fait que les mouvements du sujet, les déplacements des points d'intérêt, sont plus importants dans la séquence *Sylvain* que dans la séquence *Antonio*.

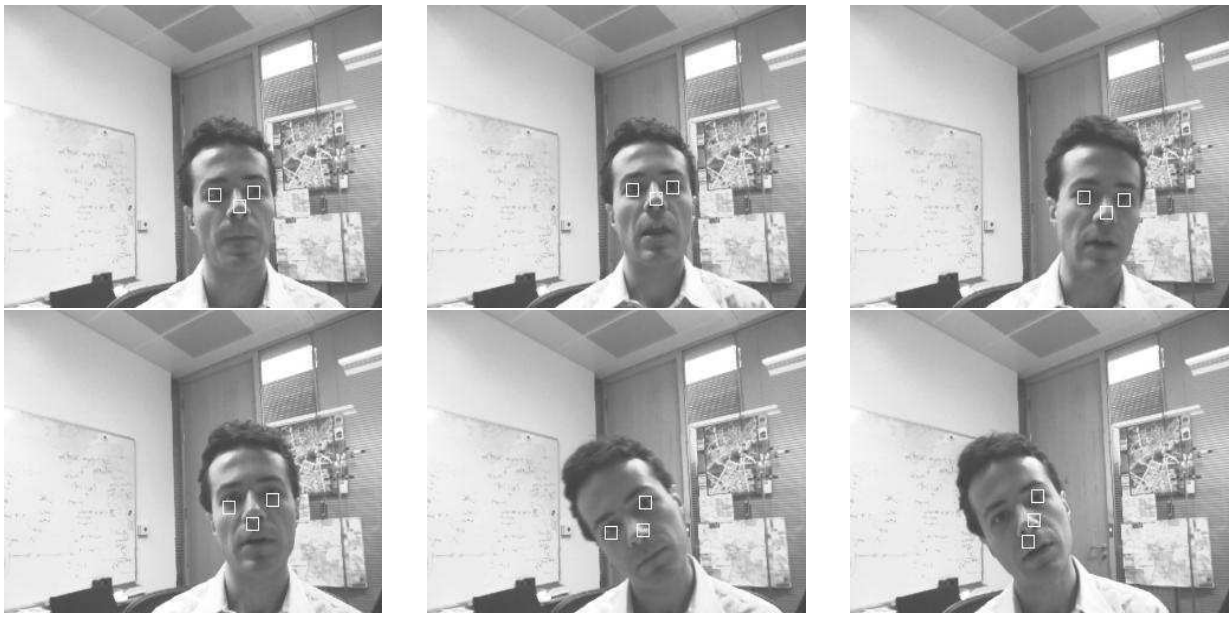
#### Remarques

La principale hypothèse utilisée dans l'algorithme KLT est celle de la "constance de la couleur" (brightness consistency en anglais), i.e. si on suppose que l'intervalle de temps entre deux images $I_t$ et $I_{t+\Delta t}$ est assez faible, alors, on peut raisonnablement supposer que les positions des points d'intérêt varient mais pas leurs intensités lumineuses. Autrement dit :

$$I(\mathbf{x}, t) = I(\mathbf{W}(\mathbf{x}), t + \Delta t)$$

Cette hypothèse est également importante dans le cas de l'algorithme "block-matching" car la mesure de similarité est basée uniquement sur l'intensité lumineuse.

Or, si les déplacements des points d'intérêt sont relativement importants, cette hypothèse n'est plus vérifiée. C'est ce qui explique les mauvais résultats obtenus avec la séquence *Sylvain* par les deux méthodes. Dans cette séquence, le déplacement d'un point entre deux images successives peut être de l'ordre de 2 fois la distance inter-occulaire.



Avec l'algorithme KLT.

Avec l'algorithme "block-matching".

Fig. 8.5 – Résultats avec la séquence *Antonio*. De haut en bas et de gauche à droite : 1ère, 20ème, 30ème, 40ème, 50ème et 60ème image de la séquence.

Avec l'algorithme KLT.

Avec l'algorithme "block-matching".

FIG. 8.6 – Résultats avec la séquence *Sylvain*. De haut en bas et de gauche à droite : 1ère, 10ème, 20ème et 30ème image de la séquence.

D'autre part, dans les séquences utilisées, le sujet a les yeux fermés dans certaines images. Lorsque d'une image $I_t$ à l'image $I_{t+1}$, le sujet ferme un œil ou les deux yeux, la quantité $I_t(\mathbf{x}) - I_{t+1}(\mathbf{x}+\mathbf{p})$ est mal estimée et par conséquent, le vecteur de déplacement $\mathbf{\Delta p}$ est lui aussi mal estimé dans le cas de la méthode KLT. Dans le cas de l'algorithme "block-matching", c'est le minimum de la distance de block qui est mal estimé.

On note aussi, que les déplacements des points sont indépendants les uns des autres. Or si l'on suit un visage, ou un objet rigide, il est normal de considérer que tous déplacements des points détectés sur le visage sont liés. On peut donc pendre en compte dans l'algorithme de suivi, des contraintes géométriques pour garantir la robustesse de la méthode. Nous avons vu dans le chapitre 4 comment la relaxation permet de tenir compte de la configuration du voisinage de chaque point dans le processus de mise en correspondance. Dans la section suivante, nous utilisons cette méthode de relaxation dans le cadre du suivi du visage.

## 8.4 La prise en compte de contraintes géométriques par la relaxation

Dans le cas où l'on suit un ensemble de points d'intérêt dans une séquence d'images, le problème du suivi est équivalent à celui de la mise en correspondance d'images. En effet, trouver les positions des points $\mathbf{x}_i$ dans l'image $I_{t+1}$ connaissant leurs positions dans l'image $I_t$, revient à mettre en correspondance les images $I_t$ et $I_{t+1}$, les points d'intérêt étant dans ce cas les $\mathbf{x}_i,\ i = 1, \ldots, n$.

Il y a toutefois, une différence notable entre les deux problèmes dans le cadre de notre application. En effet, pour la mise en correspondance d'images, on sait détecter les points d'intérêt dans chacune des deux images. On a donc deux ensembles de points $\{\mathbf{u}_i,\ i = 1, \ldots, n\}$ et $\{\mathbf{v}_j,\ j = 1, \ldots, m\}$ et on cherche les couples de points $(\mathbf{u}_i, \mathbf{v}_j)$ qui se correspondent (voir les chapitres 2 et 4). Dans le cas qui nous intéresse, si les poisitions des yeux et du nez sont connues dans l'image $I_t$, elles sont inconnues dans l'image $I_{t+1}$. Ce sont les inconnues de notre problème. On peut certes utiliser un détecteur tel que le détecteur des yeux décrit dans le chapitre 7, mais dans ce cas, on a plus besoin de mettre en œuvre une méthode de suivi car les positions des points sont déjà connues.

L'intérêt du suivi, consiste à n'utiliser le détecteur que pour l'initialisation avec la première image de la séquence. Il nous faut donc présenter autrement, le problème de la mise en correspondance dans le cadre du suivi.

### 8.4.1 Formulation du suivi comme un problème de mise en correspondance

La méthode de mise en correspondance par relaxation (voir le chapitre 4) nécessite la définition de différents termes :

*- le voisinage*

Le nombre de points étant faible (de l'ordre de la dizaine en général), tous les points sont considérés comme étant voisins les uns des autres. Autrement dit :

$$\text{pour } i = 1, \ldots, n;\ V_i = \{\mathbf{u}_j, j \neq i\}$$

*- les correspondants potentiels*

Les mouvements étant supposés faibles, on recherche l'ensemble des correspondants potentiels de chaque point $\mathbf{u}_i$ de l'image $I_t$, dans une fenêtre $\Omega$ centrée en ce point dans $I_{t+1}$. Nous utilisons la même taille de fenêtre que dans les méthodes KLT et "block-matching" décrites ci-dessus.

Pour $\mathbf{x} \in \Omega$, on calucle le score de corrélation entre la fenêtre centrée en $\mathbf{x}$ et celle centrée en $\mathbf{u}_i$. Rappelons que la première fenêtre est extraitre de l'image $I_{t+1}$, tandis que la seconde est une fenêtre de l'image $I_t$.

On retient comme correspondants potentiels, les $K$ points de $\Omega$ qui donnent les scores de corrélation les plus élevés. Dans la pratique, nous prenons $K = 5$, comme dans le cas de la mise en correspondance d'images.

*- les probabilités conditionnelles*

Les probabilités conditionnelles constituent l'information contextuelle qui permet de sélectionner le correspondant correct parmis les correspondants potentiels.

Elles doivent prendre en considération le fait que les points à suivre sont situés sur le visage, et que par conséquent, les déplacements de ces points sont liés. Plus précisément, dans le cas où nous suivons trois points par exemple, si les deux yeux bougent dans une direction, alors le nez doit se déplacer dans la même direction. En un mot, le triangle formé par les trois points doit être conservé au cours du suivi.

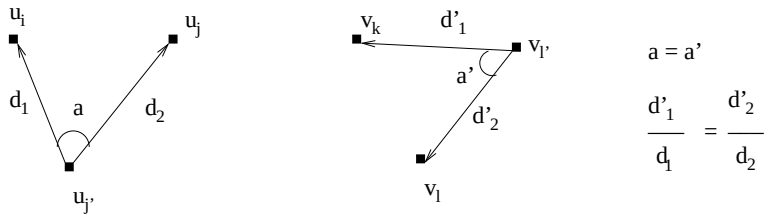



FIG. 8.7 – Calcul des probabilités conditionnelles.

Cette contrainte est exprimée par l'équation suivante :

$$p_{ij}(k,l) = \prod_{\substack{j' \neq i;\ j' \neq j \\ l' \neq k;\ l' \neq l}} f(|(\widehat{\overrightarrow{u_{j'}u_i}, \overrightarrow{u_{j'}u_j}}) - (\widehat{\overrightarrow{v_{l'}v_k}, \overrightarrow{v_{l'}v_l}})|)\ e^{-(|\frac{d_{kl'}}{d_{ij'}} - \frac{d_{ll'}}{d_{jj'}}|)} \tag{8.7}$$

où $f$ est la fonction défine par :

$$f(x) = \begin{cases} \frac{1-x}{\eta} & \text{si } x < \eta \\ 0 & \text{sinon} \end{cases}$$

Dans l'équation (8.7) ci-dessus :

- $p_{ij}(k,l)$ désigne la probabilité que le point $u_i$ soit apparié avec le point $v_k$ sachant que le point $u_j$ est apparié avec $v_l$ ;
- $d_{ij}$ désigne la distance Euclidienne entre les points $u_i$ et $u_j$ (entre les points $v_i$ et $v_j$) ;
- $(\widehat{\overrightarrow{u_{j'}u_i}, \overrightarrow{u_{j'}u_j}})$ désigne l'angle formé par les vecteurs $\overrightarrow{u_{j'}u_i}$ et $\overrightarrow{u_{j'}u_j}$.

Cette équation traduit donc le fait que, pour tout point $u_i$ et pour l'un de ses voisins $u_j$, les triangles formés par $u_i$,$u_j$ et chacun des autres points $u'_j$, conservent leurs propriétés géométriques dans l'autre image. Voir la figure 8.7 pour une représentation graphique.

### 8.4.2 Résultats

En utilisant les contraintes ci-dessus définies et la méthode de relaxation décrite dans le chapitre 4 (voir page 63), on obtient les résultats présentés sur la figure 8.8.

Séquence *Antonio*

Séquence *Sylvain*

FIG. 8.8 – Résultats avec la méthode de relaxation. Séquence *Antonio*, de haut en bas et de gauche à droite : 1ère, 20ème, 30ème, 40ème, 50ème et 60ème image de la séquence. Séquence *Sylvain*, de haut en bas et de gauche à droite : 1ère, 10ème, 20ème et 30ème image de la séquence.

#### Résultats avec la séquence *Antonio*

Les résultats obtenus avec la séquence *Antonio*, sont un peu meilleurs que ceux obtenus par l'algorithme KLT. Cette dernière méthode donne de bons résultats avec cette séquence car les déplacements des points du visage entre deux images consécutives sont faibles. Toutefois, les points d'intérêt sont mal localisés dans les dernières images à cause de l'accumulation progressive des erreurs de localisation dans l'algorithme KLT. Avec la relaxation, les erreurs de localisation sont moins importantes.

#### Résultats avec la séquence *Sylvain*

Dans le cas de la séquence *Sylvain*, tous les points d'intérêt sont correctement suivis dans l'ensemble de la séquence. En dépit des déplacements importants entre les images consécutives, la relaxation permet de suivre correctements les points du visage. On a donc une amélioration notable par rapport aux algorithmes KLT et "block-matching".

### 8.4.3 Remarques

La relaxation donne de bien meilleurs résultats parce que les contraintes prise en compte dans le calcul des probabilités conditionnelles, permettent de suivre les déplacements de tous les points en même temps. Ce qui n'est pas le cas dans la méthode KLT, ni dans la méthode "block-matching".

Il existe cependant des travaux visant à rendre ces deux méthode plus robustes. Par exemple, Spors et Rabenstein [141] utilisent un filtre de Kalman pour améliorer le résultat du suivi avec l'algorithme "block-matching". Ils obtiennent de meilleurs résultats, mais une mise en œuvre efficace du filtre de Kalman nécessite une pré-segmentation de l'image. Les auteurs utilisent pour ce faire une technique de soustraction du fond de l'image, ce qui revient à écarter de l'espace de recherche, les zones qui sont immobiles dans des images consécutives de la séquence.

En ce qui concerne l'algorithme KLT, des améliorations sont rapportés par Singh *et al.* dans [136]. Les auteurs utilisent des fonctions de pondération dans le calcul de la somme des erreurs à minimiser (voir équation (8.2)). Les fonctions utilisées sont la Gaussienne et le Laplacien. On peut aussi considérer que la transformation entre deux images successives est plus complexe qu'une simple translation. On utilise généralement, une transformation affine, i.e. la composée d'une rotation, d'une translation et d'un changement d'échelle. Nous avons utilisé un modèle de transformation affine dans nos expériences, sans noter une

| | Block-Matching | KLT | Relaxation |
|---|---|---|---|
| temps par image | 44 ms | 66 ms | 46 ms |
| # images par seconde | 23 | 15 | 22 |

TAB. 8.1 – Temps d'exécution moyen avec des images de résolution 320x240.

amélioration notable des résultats avec les séquences utilisées.

Enfin, soulignons que si notre méthode détecte de manière automatique les points d'intérêt (ici les yeux et la bouche) dans la première image de la séquence, il est possible de définir manuellement ces points. L'avantage de la détection autmatique est de permettre une ré-initialisation du système lorsqu'il échoue. Nous donnons un exemple d'une telle application dans la section 8.5.

#### Temps d'exécution

Dans la plupart des applications réelles, la phase de suivi doit être rapide car seule une petite partie des ressources du système peut y être allouée. Le reste étant utilisé pour des phases de pré-traitement ou pour des tâches de plus haut niveau telles que la reconnaissance et l'interprétation de la trajectoire [26].

Les trois méthodes présentées ci-dessus sont assez rapides. Avec des images de résolution égale à 320x240, le temps moyen de traitement d'une image est de 44 ms avec la méthode "block-matching", de 46 ms avec la relaxation et de 66 ms avec KLT. Ces données sont rassembmées dans le tableau 8.1. Soulignons que l'algorithme KLT est plus lent parce le temps d'exécution donné inclus celui nécessaire pour le calcul des dérivées de l'image. Si les dérivées sont calculées par ailleurs, alors le temps moyen de traitement d'une image avec KLT est de 32 ms.

## 8.5 Application au suivi dans des scènes complexes

Dans cette section nous présentons un exemple dans lequel nous devons suivre deux personnes se déplaçant dans une pièce. Les deux personnes avancent vers la caméra. La taille des visages dans les images n'est donc pas fixe. Plus une personne se rapproche de la caméra, plus son visage apparaît grand dans l'image. Dans ce cas difficile, la méthode KLT échoue après un suivi correct dans quelques images (une dizaine), de même que notre méthode de relaxation (au bout d'une vingtaine d'images).

La principale raison de cet échec est la non adaptation des paramètres de l'algorithme,

i.e. la taille des masques de corrélation et la taille de la fenêtre d'intégration, à la taille du visage. Pour suivre correctement le visage dans toute la séquence, il faut adapter ces paramètres à la taille du visage dans chaque image. Or nous avons vu que ces paramètres sont déterminés à partir de la première image de la séquence.

Pour résoudre cette difficulté, on peut pour adopter une approche multi-échelle, i.e. définir un ensemble de paramètres qui permet de suivre le visage à plusieurs échelles. Concrètement, on recherche les yeux en faisant varier la taille de la fenêtre d'intégration dans un intervalle et on retient l'échelle (la taille) pour laquelle les points sont correctement localisés.

Cette approche si elle est intéressante, est coûteuse en temps de calcul car il faut appliquer l'algorithme de suivi plusieurs fois pour chacune des images. Nous adoptons une approche différente.

Notre méthode de suivi étant basée sur la détection des yeux, nous adaptons les paramètres de l'algorithme en utilisant le détecteur des yeux (chapitre 7). Plus précisément, on détecte les yeux dans la première image de la séquence, et on les suit (de même que le nez) avec la méthode de relaxation ci-dessus décrite. Pour chaque image, on définit un critère qui mesure la qualité du suivi :

$$q_{t+1} = min\{sc(u_i^{t+1}, v_i^t),\ i = 1, \ldots, n\} \tag{8.8}$$

où $sc(u_i^{t+1}, u_i^t)$ est le score de corrélation entre le masque centré au point $u_i$ de l'image $I_{t+1}$, et le masque centré au point $v_i$ de l'image $I_t$. Le point $v_i$ étant la position estimée de $u_i$.

Si ce critère dépasse un certain seuil, alors on suppose que les points ne sont pas correctement suivis et on ré-initialise l'ensemble de la procédure en utilisant le détecteur des yeux. La nouvelle distance inter-occulaire permet d'ajuster les paramètres de l'algorithme.

La figure 8.9 montre un exemple de résultat. Dans toute la séquence de quatre secondes (soit 100 images), la procédure est ré-initialisée deux fois. Le détecteur des yeux est donc utilisé toutes les 30 images environ.

## 8.6 Conclusion

Dans ce chapitre, nous nous sommes intéressé au problème du suivi du visage dans une séquence d'images, basée sur la détection et le suivi de quelques points d'intérêt. Nous avons vu qu'il est parfois important de prendre en considération la nature de l'objet suivi, pour

FIG. 8.9 – Exemple de suivi avec des visages de taille variable. De gauche à droite et de haut en bas : 1ère, 15ème, 29ème, 55ème, 61ème et 80ème image de la séquence ; La détection des yeux est réalisée à partir des 29ème et 61ème images.

garantir la robustesse, en particulier lorsque les déplacements sont importants. Nous avons montré comment la méthode de relaxation développée dans le chapitre 4 peut s'appliquer dans le cadre de cette application. Les résultats obtenus montrent des gains de performance significatifs. Les résultats sont nettement meilleurs que ceux obtenus avec une méthode de suivi standard telle que l'algorithme KLT, tout en assurant un temps d'exécution assez faible. Nous obtenons une performance de l'ordre de 22 images par seconde avec des images de résolution 320x240), qui peut être améliorée en optimisant notre implémentation en C++.

Dans le cas où la taille du visage varie tout au long de la séquence, il est nécessaire d'adapter les paramètres de l'algorithme. Nous proposons de ré-initialiser le suivi grâce au détecteur des yeux (chapitre 7) à chaque fois qu'on estime les points incorrectement localisés. Cela permet le suivi de visages de taille variable, mais la méthode est plus lente à cause de la phase de détection des yeux.

# Chapitre 9

# Conclusions et Perspectives

*A conclusion is simply the place where someone got tired of thinking.*
***Arthur Block***

## 9.1 Conclusions

Dans la première partie de cette thèse, nous avons abordé le problème de la mise en correspondance d'images. Celui-ci est un problème difficile, particulièrement, lorsque les images ne sont pas acquises dans un environnement contrôlé et quand les transformations entre deux vues peuvent être quelconques. Ces dernières années, l'utilisation des invariants locaux a permis d'obtenir d'excellents résultats dans diverses applications. Après une présentation des différentes méthodes d'extraction, de caractérisation et d'appariement des invariants locaux, nous avons montré les principales limites de l'utilisation des invariants locaux, en particulier en présence de structures répétitives, et la nécessité de mettre en œuvre des méthodes d'appariements plus robustes.

Nous avons proposé une méthode de mise en correspondance qui permet d'obtenir de bons résultats dans les cas les plus difficiles. La méthode, basée sur une technique de relaxation, utilise l'information contextuelle fournie par le voisinage de chaque point d'intérêt pour sélectionner les correspondants corrects. Elle est rapide (quelques secondes en fonction des images traitées) et permet d'obtenir un nombre réduit de faux appariements. La méthode permet de reconnaître des objets de formes quelconques dans des scènes complexes, i.e. des scènes dans lesquelles les objets recherchés sont occultés et occupent une petite région de l'image parmi de nombreux autres objets.

Dans la seconde partie de cette thèse, nous nous sommes intéressés à la détection et

au suivi du visage dans une séquence d'images. Détecter le visage, le reconnaître si besoin et le suivre dans une séquence d'images est à la base de nombreuses applications faisant intervenir les intéractions homme-machine. La détection du visage est néanmoins une tâche difficile à cause de la variabilité de la taille, de l'apparence et de l'orientation que peut avoir un visage. De plus, les expressions faciales, les occultations et les conditions d'illumination affectent également l'apparence d'un même visage.

En nous plaçant dans un cadre simplifié, vue de face des visages et condition d'illumination normale, nous avons proposé une méthode simple et efficace pour la détection des yeux (du visage) dans une image couleur. Et nous avons utilisé l'algorithme de mise en correspondance développé dans la première partie dans le cadre du suivi du visage dans une séquence d'images.

## 9.2 Limites et Perspectives

Les travaux décrits dans cette thèse présentent des limites, et peuvent être améliorés et/ou prolongés, notamment sur les aspects applicatifs.

### 9.2.1 Limites

La principale limite de la méthode de mise en correspondance, est liée à celle du détecteur de points d'intérêt et du descripteur utilisés. En effet, comme nous l'avons souligné dans les chapitres 4 et 5, les performances obtenues sont très faibles lorsque les transformations (géométriques et photométriques) entre les deux images à apparier sont importantes. Dans ce cas, la faible répétabilité du détecteur ne lui permet pas de détecter les mêmes régions d'intérêt dans les images. De même, le descripteur SIFT est plus adapté à des changements d'échelle qu'à des changement de point de vue entre les images.

Le problème devient plus difficle lorsque la scène présente des déformations non planaires. C'est ce que nous avons vu dans le cas de la recherche d'objets dans une base d'images (voir section 5.3.1, chapitre 5) où les résultats sont bien meilleurs quand on se limite à des objets de forme plane (les boîtes). Pour des objets de forme plus complexe, comme dans les expériences de la section 5.3.2, les performances sont assez faibles. Il faut utiliser les résultats de la mise en correspondance comme base pour des approches plus élaborées comme la méthode d'exploration de Ferrari [36].

Dans le cas de la détection des yeux et du suivi du visage, les performances obtenues

si elles sont satisfaisantes, restent limitées à un cadre simplifié : vue de face des visages et condition d'illumination normale. La méthode de détection des yeux échoue quand le visage est vu de profil, quand un œil est fermé ou lorsque la taille du visage est tès petite (distance inter-oculaire inférieure à 20 pixels) dans l'image. Il va de soi, qu'on ne peut suivre correctement le visage que lorsque les éléments caractéristiques (les yeux et le nez) sont correctement détectés.

### 9.2.2 Perpectives

Un premier développement concerne une implémentation efficace de la méthode de mise en correspondance décrite dans le chapitre 4. En effet, les temps de calcul peuvent être encore réduit par l'utilisation de structures de données appropriées. Par exemple, on pourrait représenter les relations entre les points et leurs voisins sous la forme d'un graphe. Dans notre travail, nous avons utilisé uniquement les points d'intérêt et les profils d'intensité pour la mise en correspondance. On pourrait cependant utiliser divers types de primitives, par exemple des segments, comme information contextuelle dans l'algorithme de mise en correspondance. Les travaux récents de Opelt *et al.* [105] utilisant des fragments de contour, ou ceux de Ferrari *et al.* [37] basés sur les groupes de segments adjacents, tendent à montrer que les segments apportent une information utile pour la reconnaissance d'objets.

Une perspective intéressante pour la reconnaissance d'objets concerne le groupement de primitives. Il s'agit de mettre en correspondance des groupes ou densités de points détectés dans chacune des images. D'une part, la complexité de l'algorithme de mise en correspondance se trouve réduite en ne considérant qu'un nombre restreint d'ensemble de points, d'autre part, les densités de points correspondent souvent à des parties de l'objet recherché ou à des zones fortement texturées. On peut donc représenter un objet par ses parties, par exemple les roues d'un véhicule ou les fenêtres d'un batiment, et rechercher les parties dans la scène. Une fois les ensembles de points appariés, on peut affiner la mise en correspondance en considérant les points de chaque ensemble. Cette approche de l'appariement par groupement si elle est intéressante, présente plusieurs difficultés dont la principale est l'obtention des ensembles de points. Comment répartir les points d'intérêt détectés dans une image en différents ensembles représentatifs de l'objet ?

Une autre piste de recherche intéressante concerne la catégorisation d'images ou la reconnaissance de classes d'objets. Dans ce cas, on ne cherche pas à identifier un objet particulier dans une image, mais plutôt à associer cet objet à une catégorie (visage, voiture, moto, avion, etc). Les méthodes récemment développées reposent sur la création de

vocabulaires visuels (*bag of features* en anglais) par quantification des descripteurs locaux [137, 33]. Les primitives sont dans ce cas regroupées dans l'espace du descripteur (un espace de dimension 128 dans le cas du descripteur SIFT par exemple) et non dans l'espace de représentation 2D de l'image. Toutefois, il semble intéressant d'essayer d'adapter les méthodes de création de vocabulaires visuels [27, 63] à l'identification des parties des objets évoquée ci-dessus.

Concernant le suivi du visage, il peut être intéressant d'utiliser la relaxation non seulement pour suivre le visage, mais également pour mettre en œuvre une méthode de détection plus robuste. Une telle approche est suggérée par les travaux de Iwata et *al.* [59]. Les résultats présentés par les auteurs montrent que la méthode peut détecter les yeux même lorsque le visage est vu de profil.

Toutefois, il nous semble que pour des applications plus complexes, que la simple réalisation d'interface homme-machine par exemple, les méthodes de suivi de points d'intérêt présentées dans le chapitre 8 ne sont pas les mieux adaptées. Les méthodes probabilistes ou prédictives telles que le filtre de Kalman, les Champs de Markov ou les filtres à particules sont largement employées dans la littérature et donnent d'excellents résultats. Voir une présentation générale par Comaniciu *et al.* dans [26]. D'autre part, les approches probabilistes permettent la fusion de différents attributs, ce qui augmente la robustesse du suivi. Voir par exemple les travaux de Perez *et al.* [108].

# Troisième partie

# Annexes

# Annexe A

# Liste des publications

## Revues internationales avec comité de lecture

- D. Sidibe, P. Montesinos, S. Janaqi, "Matching local invariant features with contextual information : An experimental evaluation", *soumis à ELCVIA (Electronic Letters on Computer Vision and Image Analysis).* Soumis en septembre 2007.

## Conférences internationales avec comité de lecture

- D. Sidibe, P. Montesinos, S. Janaqi, "Matching Local Invariant Features : How Can Contextual Information Help ?", *6th EURASIP Conference Focused on Speech and Image Processing, Multimedia Communications and Services*, Maribor, Solvenia, 2007.

- D. Sidibe, P. Montesinos, S. Janaqi, "Fast and Robust Image Matching using Contextual Information and Relaxation", *2nd International Conference on Computer Vision Theory and Applications*, Barcelona, Spain, 2007.

- D. Sidibe, P. Montesinos, S. Janaqi, “A simple and efficient eye detection method in color images", *21th International Conference Image and Vision Computing New Zealand*, Great Barrier Island, New Zealand, 2006.

## Conférences nationales avec comité de lecture

- D. Sidibe, P. Montesinos, S. Janaqi, "Mise en correspondance robuste d'invariants locaux par relaxation", *ORASIS'07 : 11ième congrès francophone des jeunes chercheurs en vision par ordinateur*, Obernai, France, 2007.

## Autres publications

- D. Sidibe, "Mise en correspondance d'images par l'utilisation d'invariants locaux", *Doctiss'07 : 11ème journée des doctorants de l'école doctorale I2S de l'Université de Montpellier II*, Montpellier, France, 19 avril 2007.
- D. Sidibe, P. Montesinos, "Application de la colorimétrie à la détection de personnes dans une image couleur", *Rapport de recherche du LGI2P*, RR 07/002, 2007.

# Annexe B

# Ecriture du critère sous forme matricielle

Le critère à minimiser peut s'écrire :

$$\begin{aligned} C(p) &= \alpha C_1(p) + (1-\alpha) C_2(p) \\ &= \frac{\alpha}{2n} \sum_{i=1}^{n} \|p_i - q_i\|^2 + \frac{(1-\alpha)m}{m-1} \left[ 1 - \frac{1}{n} \sum_{i=1}^{n} \|p_i\|^2 \right] \\ &= c_1 \sum_{i=1}^{n} \|p_i - q_i\|^2 - c_2 \sum_{i=1}^{n} \|p_i\|^2 + c_3 \end{aligned}$$

avec $c_1 = \frac{\alpha}{2n}$, $c_2 = \frac{(1-\alpha)m}{(m-1)n}$ et $c_3 = nc_2$.

On veut mettre $C$ sous la forme :

$$C([p_1, \ldots, p_n]^T) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} p_i^T H_{ij} p_j + cte$$

On peut montrer que $C$ est la somme pondérée de trois termes $A$, $B$ et $C$ :

$$\begin{aligned} C(p) &= \sum_{i=1}^{n}(c_1\|p_i - q_i\|^2 - c_2\|p_i\|^2) + c_3 \\ &= \sum_{i=1}^{n}(c_1(p_i - q_i)^T(p_i - q_i) - c_2 p_i^T p_i) + c_3 \\ &= (c_1 - c_2)\underbrace{\sum_{i=1}^{n} p_i^T p_i}_{A} - 2c_1\underbrace{\sum_{i=1}^{n} p_i^T q_i}_{B} + c_1\underbrace{\sum_{i=1}^{n} q_i^T q_i}_{C} + c_3 \end{aligned}$$

Définissons les deux symboles suivants :

$$\delta_{ij} = \begin{cases} 1 & \text{si} \quad i = j \\ 0 & \text{sinon} \end{cases}$$

$$\Lambda_{ij} = \begin{cases} 1 & \text{si} \quad u_j \in V_i \\ 0 & \text{sinon} \end{cases}$$

On montre alors que :

$$\begin{aligned} A &= \sum_{i=1}^{n} p_i^T p_i \\ &= \sum_{i=1}^{n} p_i^T (\sum_{j=1}^{n} \delta_{ij} p_j) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} j_i^T (\delta_{ij}\mathcal{I}_m) p_j \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} j_i^T A_{ij} p_j \end{aligned}$$

où $\forall \quad i, j \in (1 \ldots n)^2$, $A_{ij} = \delta_{ij}\mathcal{I}_m$

$\forall \quad k = 1 \ldots m, \quad$ on a $\quad q_i(k) = \sum_{j \in V_i} \alpha_{ij} \left[\sum_l p_{ij}(k,l) p_j(l)\right]$

soit $q_i = \alpha_{ij} P_{ij} . p_j$

où $P_{ij}$ est la matrice $m \times m$ des probabilités conditionnelles $p_{ij}(\lambda_k, \lambda_l)$.

On a donc :

$$\begin{aligned} B &= \sum_{i=1}^{n} p_i^T q_i \\ &= \sum_{i=1}^{n} p_i^T (\sum_{u_j \in V_i} \frac{1}{|V_i|} \alpha_{ij} P_{ij} p_j) \\ &= \sum_{i=1}^{n} p_i^T \sum_{j=1}^{n} (\frac{\Lambda_{ij}}{|V_i|} \alpha_{ij} P_{ij}) p_j \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} p_i^T B_{ij} p_j \end{aligned}$$

où $\forall \quad i,j \in (1 \dots n)^2$, $B_{ij} = \frac{\Lambda_{ij}}{|V_i|} \alpha_{ij} P_{ij}$

$$\begin{aligned} C &= \sum_{i=1}^{n} q_i^T q_i \\ &= \sum_{i=1}^{n} (\sum_{t=1}^{n} \frac{\Lambda_{it}}{|V_i|} \alpha_{it} P_{it} p_t)^T (\sum_{j=1}^{n} \frac{\Lambda_{ij}}{|V_i|} \alpha_{ij} P_{ij} p_j) \\ &= \sum_{i=1}^{n} (\sum_{t=1}^{n} B_{it} p_t)^T (\sum_{j=1}^{n} B_{ij} p_j) \\ &= \sum_{i=1}^{n} [\sum_{t=1}^{n} \sum_{j=1}^{n} p_i^T (B_{it}^T B_{ij}) p_j] \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} p_i^T C_{ij} p_j \end{aligned}$$

où $\forall \quad i,j \in (1 \dots n)^2$, $C_{ij} = \sum_{t=1}^{n} (B_{ti}^T B_{tj})$

Finalement, $C([p_1, \dots, p_n]^T) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} p_i^T H_{ij} p_j + c_3$
avec

$$\forall \quad i,j \in (1, \dots, n)^2, \quad H_{ij} = 2(c_1 - c_2) A_{ij} - 4c_1 B_{ij} + 2c_1 C_{ij}$$

# Annexe C

# Conditions de nullité des matrices $H_{ij}$

Dans l'annexe précédente, nous avons montré que le critère $C$ pouvait se mettre sous la forme :

$$C([p_1, \ldots, p_n]^T) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} p_i^T H_{ij} p_j + cte$$

avec

$$\forall \quad i, j \in (1, \ldots, n)^2, \quad H_{ij} = a_1 A_{ij} - a_2 B_{ij} + a_3 C_{ij}$$

$a_1$, $a_2$ et $a_3$ étant des constantes.

La matrice $H_{ij}$ est non-nulle, si et seulement si l'une des trois conditions suivantes est vérifiée :

$$\begin{cases} A_{ij} \neq 0 \\ B_{ij} \neq 0 \\ C_{ij} \neq 0 \end{cases}$$

Or, $A_{ij} = \delta_{ij} \mathcal{I}_m$, donc, $A_{ij} \neq 0 \Leftrightarrow i = j$

On a $B_{ij} = \frac{\Lambda_{ij}}{|V_i|} \alpha_{ij} P_{ij}$ et $P_{ij}$ est la matrice $m \times m$ des probabilités conditionnelles $p_{ij}(\lambda_k, \lambda_l)$. Par conséquent, $B_{ij} \neq 0 \Leftrightarrow j \in V_i$.

On a $C_{ij} = \sum_{k=1}^{n} (B_{ki}^T B_{kj})$. Donc, $C_{ij} \neq 0 \Leftrightarrow \exists k / (B_{ki} \neq 0 \text{ et } B_{kj} \neq 0)$.

Finalement,

$$H_{ij} \neq 0 \Leftrightarrow \begin{cases} i = j \quad \text{ou} \\ u_j \in V_i \quad \text{ou} \\ \exists k / (u_i, u_j) \in V_k \times V_k \end{cases}$$

# Annexe D

# Modèles d'objets utilisés pour la reconnaissance d'objets

Ici nous présentons les modèles d'objets utilisés dans les expériences du chapitre 5.

La figure D.1 présente les vues de face des objets de la base SOIL-47A utilisée dans le cadre de la recherche d'objets dans une base d'images (chapitre 5, section 5.3.1).

Les figures D.2, D.3, D.4, D.5, D.6 et D.7, montrent les images modélisant les objets utilisés dans les expériences de reconnaissance d'objets (chapitre 5, section 5.3.2).


Fig. D.1 – Vues de face des objets de la base SOIL-47A.


FIG. D.2 – Les objets modélisés par une seule vue.


Fig. D.3 – OVO, modélisé par 6 vues.

FIG. D.4 – Xmas, modélisé par 6 vues.


Fig. D.5 – CAR, modélisé par 8 vues.

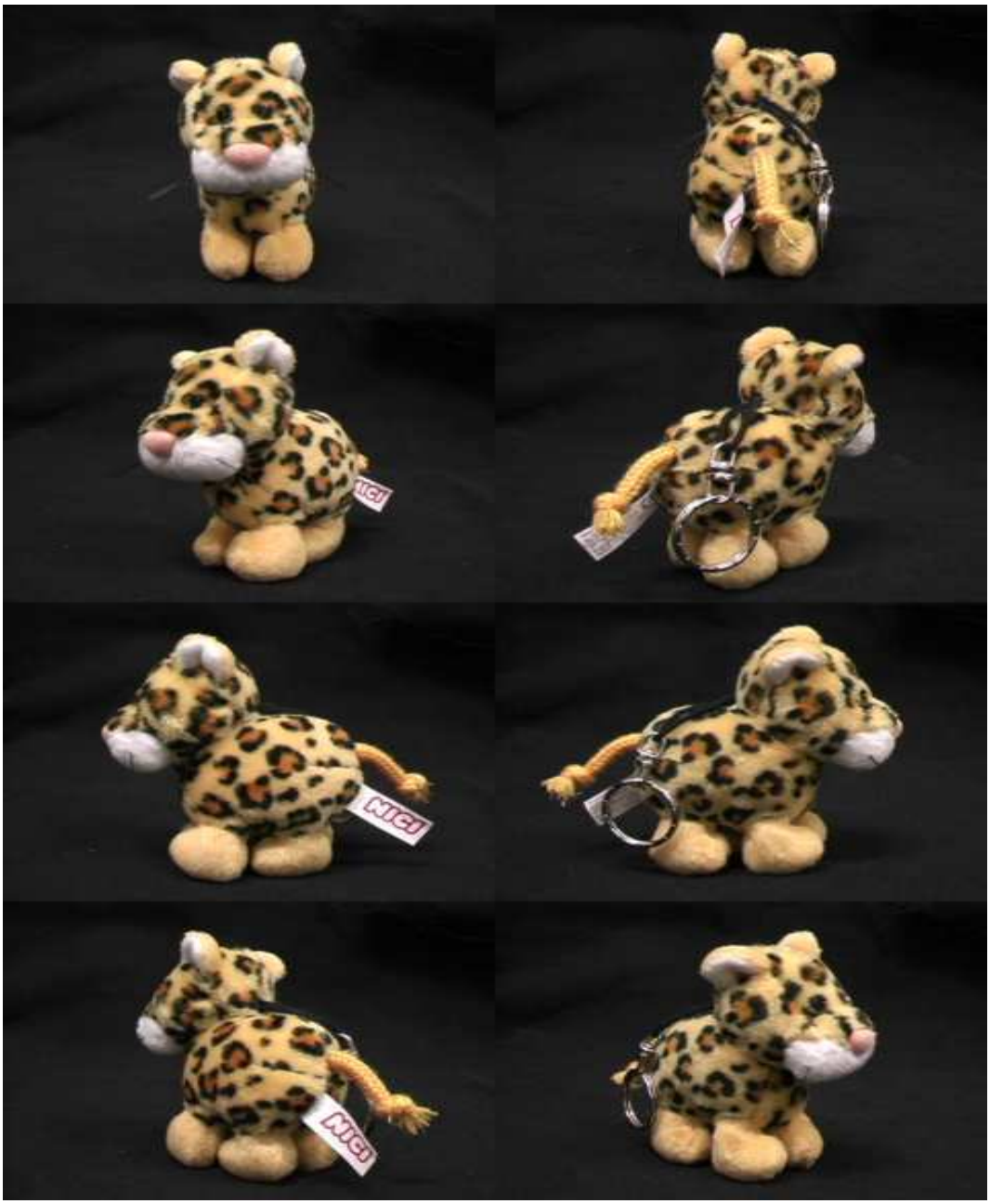



FIG. D.6 – Leo, modélisé par 8 vues.

Fig. D.7 – Suchard, modélisé par 8 vues.

# Annexe E

# Description de l'algorithme KLT

On souhaite trouver $\Delta\mathbf{p}$ telle que la quantité suivante soit minimale :

$$\epsilon(\mathbf{p}) = \sum_{\mathbf{x}\in\Omega}[J(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - I(\mathbf{x})]^2$$

En réalisant un développement de Taylor à l'ordre 1 de l'expression $J(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}))$, on a :

$$\epsilon(\mathbf{p}) = \sum_{\mathbf{x}\in\Omega}[J(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla J\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\triangle\mathbf{p} - I(\mathbf{x})]^2 \tag{E.1}$$

Dans cette expression, $\nabla J = (\frac{\partial J}{\partial x}, \frac{\partial J}{\partial y})$ est le gradient de l'image $J$ évalué en $W(\mathbf{x};\mathbf{p})$. Le terme $\frac{\partial\mathbf{W}}{\partial\mathbf{p}}$ est le Jacobien de $\mathbf{W}$.

Si $\mathbf{W}(\mathbf{x};\mathbf{p}) = (W_x(\mathbf{x};\mathbf{p}), W_y(\mathbf{x};\mathbf{p}))^T$, alors :

$$\frac{\partial\mathbf{W}}{\partial\mathbf{p}} = \begin{pmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_n} \\ \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_n} \end{pmatrix}$$

En dérivant l'expression de l'équation (E.1) par rapprt à $\Delta\mathbf{p}$, on obtient :

$$\sum_{\mathbf{x}\in\Omega}\left[\nabla J\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^T\left[J(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla J\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\triangle\mathbf{p} - I(\mathbf{x})\right]$$

En posant enfin l'expression ci-dessus égale à zéro,on obtient une solution approchée

au sens des moindre carrés de l'équation (E.1) :

$$\Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}\in\Omega} \left[\nabla J \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [I(\mathbf{x}) - J(\mathbf{W}(\mathbf{x};\mathbf{p}))] \tag{E.2}$$

où $H$ est une approximation de la matrice Hessienne :

$$H = \sum_{\mathbf{x}\in\Omega} \left[\nabla J \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[\nabla J \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right] \tag{E.3}$$

L'algorithme KLT consiste à appliquer successivement les équations (E.2) et $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$.

Dans le cas le plus général, la méthode KLT peut se résumer par l'algorithme présenté sur la figure E.1.

Si $N$ est le nombre de pixels de l'image $I$ est si $n$ est le nombre de paramètres de la transformation $\mathbf{W}$, alors la complexité de l'algorithme KLT est de l'ordre $O(n^2N + n^3)$.

Itérer

(1) Calculer $J(\mathbf{W}(\mathbf{x};\mathbf{p}))$
(2) Calculer la différence $I(\mathbf{x}) - J(\mathbf{W}(\mathbf{x};\mathbf{p}))$
(3) Evaluer le gradient image $\nabla J$ en $\mathbf{W}(\mathbf{x};\mathbf{p})$
(4) Calculer le Jacobien $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ en $(\mathbf{x};\mathbf{p})$
(5) Evaluer la quantité $\nabla J \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
(6) Calculer la matrice Hessienne en utilisant l'équation (E.3)
(7) Calculer $\sum_{\mathbf{x}\in\Omega} \left[\nabla J \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [I(\mathbf{x}) - J(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
(8) Calculer $\Delta\mathbf{p}$ en utilisant l'équation (E.2)
(9) Mettre les paramètres à jours : $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$

Tant que $\|\Delta\mathbf{p}\| > \varepsilon$

Fig. E.1 – Algorithme KLT dans le cas d'une transformation quelconque.

# Bibliographie


[1] http ://www.ee.surrey.ac.uk/cvssp/demos/colour/soil47/.

[2] A. E. Abdel-Hakim and A. A. Farag. CSIFT : A SIFT descriptor with color invariant characteristics. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 1978–1983, 2006.

[3] A. Ahmadyfard and J. Kittler. Region-based object recognition : Pruning multiple hypothesis and representations. In *Proc. of BMCV*, pages 745–754, 2000.

[4] A. Ahmadyfard and J. Kittler. A comparative study of two object recognition methods. In *Proc. of BMCV*, pages 363–372, 2002.

[5] A. M. Alattar and S. A. Rajala. Facial features localization in front view head and shoulders images. In *IEEE Proc. of ICASSP*, volume 6, pages 3557–3560, 1999.

[6] S. Baker and I. Matthews. Lucas-Kanade 20 years on : A unifying framework. *International Journal of Computer Vision*, 56(3) :221–255, 2004.

[7] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2) :111–122, 1981.

[8] A. Baumberg. Reliable feature matching across widely separated views. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 774–781, 2000.

[9] H. Bay, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. In *Proc. of European Conference on Computer Vision*, pages 404–417, 2006.

[10] G. Bebis, M. Georgiopoulos, and N. V. Lobo. Learning geometric hashing functions for model-based object recognition. In *Proc. International Conference on Computer Vision*, pages 543–548, 1995.

[11] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans on PAMI*, 24(24) :509–522, 2002.

[12] I. Biederman. Recognition-by-components : A theory of human image understanding. *Psychological Review*, 94(2) :115–147, 1987.

[13] I. Biederman. From edges to geons to viewpoint-invariant object models : a neural net implementation. In *Proc. SPIE*, pages 570–578, 1992.

[14] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 232–237, 1998.

[15] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C.A. Sagastizábal. *Numerical Optimization : Theoretical and Practical Aspects.* Springer, 2003.

[16] F. Bourel, C. C. Chibelushi, and A. A. Low. Robust facial feature tracking. In *Proc. of BMCV*, pages 232–241, 2000.

[17] G. R. Bradski. Computer vision face tracking as a component of a perceptual user interface. In *Proc of IEEE Workshop on Applications of Computer Vision*, pages 214–219, 1998.

[18] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, Q2(15), 1998.

[19] J. Brand and J. Masson. A comparative assessment of three approaches to pixellevel human skin-detection. In *Proc. of the Int'l Conf. on Pattern Recognition*, volume 1, pages 1056–1059, 2000.

[20] R. Brunelli and T. Poggio. Face recognition : features versus templates. *IEEE Trans. on Pattern Analysis and Machine Intel.*, 15(10) :1042–1052, 1993.

[21] T. S. Caetano, S. D. Olabarriaga, and D. A. C. Barone. Do mixture models in chromaticity space improve skin detection ? *Pattern Recognition*, 36 :3019–3021, 2003.

[22] Y. Cheng. Mean shift, mode seeking, and clustering. *in IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8) :790–799, 1995.

[23] O. Chum, J. Matas, and S. Obdrzalek. Epipolar geometry from three correspondences. In *Proc. Computer Vision Winter Workshop*, 2003.

[24] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *in Proceedings of the IEEE*, 89(10) :1456–1477, 2001.

[25] A. Colmenarez, B. Frey, and T. S. Huang. Detection and tracking of faces and facial features. In *International Conference on Image Processing*, pages 657–661, 1999.

[26] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *in IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5) :564–577, 2003.

[27] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. In *In ECCV Workshop on Statistical Learning in Computer Vision*, pages 59–74, 2004.

[28] E. Delponte, F. Isgro, F. Odone, and A. Verri. Svd-matching using sift features. *Graphical Models*, 68 :415–431, 2006.

[29] H. Deng, E. N. Mortensen, L. Shapiro, and T. G. Dietterich. Reinforcement matching using region context. In *Proc. "beyond patches" CVPR Workshop*, page 11, 2006.

[30] R. Deriche and G. Giraudon. Acurate corner detetction : An analytic study. Technical Report 1420, INRIA Sophia-Antipolis, France, 1991.

[31] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 612–618, 2000.

[32] O. D. Faugeras and M. Berthod. Improving consistency and reducing ambiguity in stochastic labeling : An optimization approach. *IEEE PAMI*, 3(4) :412–424, July 1981.

[33] R. Fergus, P. Perona, and A. Zisserman. Objects class recognition by unsupervised scale-invariants learning. In *In CVPR*, pages 264–271, 2003.

[34] V. Ferrari. *Affine Invariant Regions ++*. PhD thesis, Swiss Federal Institute of Technology, Zurich, 2004.

[35] V. Ferrari, T. Tuytelaars, and L. Van-Gool. Real-time region tracking and coplanar grouping. In *Proc of IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 226–233, 2001.

[36] V. Ferrari, T. Tuytelaars, and L. Van-Gool. Simultaneous object recognition and segmentation by image exploration. In *Proc. ECCV*, volume 1, pages 40–54, 2004.

[37] V. Ferrari, T. Tuytelaars, and L. Van-Gool. Object detection by contour segment networks. In *Proc. ECCV*, 2006.

[38] M. A. Fischler and R. C. Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395, 1981.

[39] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE PAMI*, 13(9) :891–906, 1991.

[40] D. Gabor. Theory of communication. *Journal I.E.E.*, 93(26) :429–457, 1946.

[41] T. Gevers and W.M. Smeulders. Color-based object recognition. *Pattern Recognition*, 32 :453–464, 1999.

[42] G. Gomez. On selecting components for skin detection. In *Proc of the Int'l Conf. on Pattern Recognition*, volume 2, pages 961–964, 2000.

[43] D. O. Gorodnichy. Facial recognition in video. In *Proc. of IAPR Conf. on Audio and Video-Based Biometric Person Authetication (AVBPA'03)*, pages 505–514, 2003.

[44] D. O. Gorodnichy. Seeing faces in video by computers. editorial for special issue on face processing in video sequences. *Image and Vision Computing*, 24 :551–556, 2006.

[45] V. Gouet. *Mise en Correspondance d'Images en Couleur : Application à la synthèse de vues intermédiaires.* PhD thesis, Université Montpellier II, 2000.

[46] V. Gouet, P. Montesinos, and D. Pele. A fast matching method for color uncalibrated images using differential invariants. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 367–376. Southampton, UK, 1998.

[47] H.P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. In *Proc First Int'l Workshop Automatic Face and Resture Recognition*, pages 41–46, 1995.

[48] H. Greenspan, J. Goldberger, and I. Eshet. Mixture model for face-color modeling and segmentation. *Pattern Recognition Letters*, 22 :1525–1536, 2001.

[49] C. C. Han, H. Y. M. Liao, G. J. Yu, and L. H. Chen. Fast face detection via morphology-based pre-processing. *Pattern Recognition*, 33 :1701–1712, 2000.

[50] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen. Computer vision for driver assistance systems. *in Proceedings of SPIE*, 3364 :136–147, 1998.

[51] B. B. Hansen and B. S. Morse. Multiscale image registration using scale trace correlation. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 202–208, 1999.

[52] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[53] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge university press, 2002.

[54] S. Helmer and D. G. Lowe. Object class recognition with many local features. In *Workshop on Generative Model Based Vision (GMVB).* Washington DC, july 2004.

[55] R. Horaud and T. Skordas. Stereo matching through feature grouping and maximal cliques. *PAMI*, 11(11) :1168–1180, 1989.

[56] Paul V. C. Hough. Method and means for recognising complex patterns. Technical Report 3069654, U.S. Patent, 1962.

[57] R.L Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5) :696–706, may 2002.

[58] R. A. Hummel and S. W. Zucker. On the fundations of relaxation labeling processes. *IEEE PAMI*, 5(3) :267–287, May 1983.

[59] K. Iwata, H. Hongo, K. Yamamoto, and Y. Niwa. *LNCS/KES*, volume 2774/2003, chapter Robust Facial Parts Detection by Using Four Directional Features and Relaxation Matching. Springer-Verlag Berlin / Heidelberg, 2003.

[60] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. In *Proc. of the Third Int'l Conf. on Audio and Video-based Biometric Person Authentication*, pages 90–95. Halmstad, Sweden, 2001.

[61] A. Johnson and M. Hebert. Object recognition by matching oriented points. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 684–689, 1997.

[62] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. In *Proc of the CVPR*, volume 1, pages 274–280, 1999.

[63] F. Jurie and W. Triggs. Creating efficient codebooks for visual recognition. In *In ICCV*, pages 604–610, 2005.

[64] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proc. 8th European Conference on Computer Vision*, pages 404–416, 2004.

[65] T. Kawaguchi and M. Rizon. Iris detection using intensity and edge information. *Pattern Recognition*, 36 :549–562, 2003.

[66] Y. Ke and R. Sukthankar. PCA-SIFT : A more distinctive representation for local image descriptors. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 511–517, 2004.

[67] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Proc of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 253–259, 1999.

[68] L. Kitchen and A. Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, pages 95–102, 1982.

[69] J. Koenderink. The structure of images. *Biological Cybernetics*, 50 :363–396, 1984.

[70] J. Koenderink and A. Van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55 :367–375, 1987.

[71] S. G. Kong, J. Heo, B. R. Abidi, J. Paik, and M. A. Abidi. Recent advances in visual and infrared face recognition : a review. *Computer Vision and Image Understanding*, 97 :103–135, 2005.

[72] D. Koubaroulis, J. Matas, and J. Kittler. Evaluating colour-based object recognition algorithms using the SOIL-47 database. In *Proc. of ACCV*, 2002.

[73] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *Proc of IEEE Intl. Workshop on Visual Surveillance*, pages 3–10, 2000.

[74] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans on PAMI*, 27(8) :1265–1278, 2005.

[75] T. Lindeberg. Scale-space theory : A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2) :224–270, 1994.

[76] T. Lindeberg. Feature detection with automatic scale selection. *Int. J. of Computer Vision (IJCV)*, 30(2) :79–116, 1998.

[77] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31 :455–395, 1987.

[78] D. G. Lowe. Fitting parameterized three-dimentional models to images. *IEEE PAMI*, 13(5) :441–450, 1991.

[79] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157. Corfu, Greece, september 1999.

[80] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[81] D.G. Lowe. Local feature view clustering for 3d object recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 682–688, 2001.

[82] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. IJCIA*, 1981.

[83] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[84] David Marr. *Vision*. Fremann and Company, 1982.

[85] A. M. Martinez and R. Benavente. The AR face database. Technical Report 24, CVC, june 1998.

[86] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. 13th British Machine Vision Conference*, pages 384–393, 2002.

[87] J. Matas, D. Koubaroulis, and J. Kittler. Colour image retrieval and object recognition using the multimodal neighbourhood signature. In *In proc. ECCV*, pages 48–64, 2000.

[88] G. J. McLachlan. *The EM Algorithm*. Wiley, New York, 1997.

[89] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. of the 8th International Conference on Computer Vision*. Vancouver, Canada, 2001.

[90] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142. Copenhag, Denmark, may 2002.

[91] K. Mikolajczyk and C. Schmid. Sacle & affine invariant interest point detectors. *Internationl Journal of Computer Vision*, 60(1) :63–86, 2004.

[92] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans on PAMI*, 27(10) :1615–1630, 2005.

[93] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *Internationl Journal of Computer Vision*, 65(1/2) :43–72, 2005.

[94] F. Mindru, T. Moons, and L. Van Gool. Recognizing color patterns irrespective of viewpoint and illumination. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 368–373, 1999.

[95] P. Montesinos, V. Gouet, and R. Deriche. Differential invariants for color images. In *Proceedings of 14th Inter. Conference on Pattern Recognition*. Brisbane, Australia, 1998.

[96] P. Montesinos, V. Gouet, R. Deriche, and D. Pele. Matching color uncalibrated images using differential invariants. *Image and Vision Computing*, 18 :659–671, 2000.

[97] H. Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th Int. Joint Conference on Artificial Intelligence*, page 584. Cambridge, Massachusetts, USA, 1977.

[98] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE PAMI*, 27(11) :1832–1837, 2005.

[99] E. N. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. In *Proc. Computer Vision and Pattern Recognition*, pages 184–190, 2005.

[100] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14 (1) :5–24, 1995.

[101] J. A. Noble. Finding corners. *Image and Vision Computing*, 6 :121–128, 1988.

[102] S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *In Proceedings British Machine Vision Conference*, pages 113–122, 2002.

[103] Y. I. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13 :222–241, 1980.

[104] N. Oliver, A. Pentland, and F. Berard. Lafter : Lips and face real-time tracker with facial expression recognition. In *Proc of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 123–130, 1997.

[105] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *In Proc. ECCV*, 2006.

[106] E. Osuna, R. Freund, and F. Girosi. Training support vector machines : An application to face detection. In *Proc of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[107] A. Pentland, B. Moghanddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE Proc. of Int. Conf. on CVPR*, pages 84–91. Seattle, Washington, USA, 1994.

[108] P. Perez, Jaco Vermaak, and Andrew Blake. Data fusion for visual tracking with particles. *in Proceedings of the IEEE*, 92(3) :495–513, 2004.

[109] Cambridge University Press, editor. *Commission Internationale de l'Eclairage Proceedings*, 1931.

[110] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.

[111] K. E. Price. Relaxation matching techniques - a comparison. *IEEE PAMI*, 7(5) :617–623, 1985.

[112] A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri. Finding faces in photographs. In *Proc. of ICCV*, pages 640–645, 1998.

[113] K. Rohr. Modelling and identification of characteristic intensity variations. *Image and vision Computing*, 10 :66–76, 1992.

[114] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems. Man Cybernetics*, 6 :420–433, 1976.

[115] Azriel Rosenfeld. From image analysis to computer vision : An annotated bibliography, 1955-1979. *Computer Vision and Image Understanding*, 84 :298–324, 2001.

[116] H. Rowley, S. Baluja, and T. Kanade. Neural networks-based face detection. *IEEE Trans on PAMI*, 20(1) :23–38, 1998.

[117] E. Saber and A. M. Tekalp. Frontal-view face detection and facial feature extraction using color, shape and symetry based cost functions. *Pattern Recognition Letters*, 19 :669–680, 1997.

[118] G. Saporta. *Probabilités, Analyse de données et Statistique*. Technip, 1990.

[119] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proc. 8th ICCV*, pages 636–643, 2001.

[120] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets. In *Proc. 7th European Conference on Computer Vision*, pages 414–431, 2002.

[121] C. Schmid. *Appariement d'images par invariants locaux de niveaux de gris*. PhD thesis, INP Grenoble, GRAVIR- IMAG- INRIA Rhône Alpes, 1996.

[122] C. Schmid, G. Dorko, S. Lazebnik, and K. Mikolajczyk andJ. Ponce. *Handbook of Pattern Recognition and Computer Vision*, chapter Pattern Recognition with Local Invariants Features. World Scientific Publishing Co., 2004.

[123] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 19(5) :530–534, 1997.

[124] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2) :151–172, 2000.

[125] K. Schwerdt and J. L. Crowley. Robust face tracking using color. In *4th Int. Conf. Automatic Face and Gesture Recognition*, 2002.

[126] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. In *Proceedings of Royal Society London B244*, pages 21–26, 1991.

[127] S. Se, D. Lowe, and J. Little. Local and global localization for mobile robots using visual landmarks. In *Proc. of IEEE/RSJ Conf. on Intelligent Robots and Sysytems*, pages 414–420, 2001.

[128] J. Shi and C. Tomasi. Good features to track. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[129] I. Shimshoni and J. Ponce. Probabilistic 3d object recognition. In *Proc. of International Conference on Computer Vision*, pages 448–493, 1995.

[130] M. C. Shin, K. I. Chang, and L. V. Tsap. Does colorspace transformation make any difference on skin detection ? In *In Proc. 6th IEEE Workshop on Application of Computer Vision*, page 275, 2002.

[131] D. Sidibe, P. Montesinos, and S. Janaqi. A simple and efficient eye detection method in color images. In *Proc. 21th International Conference Image and Vision Computing New Zealand*, pages 385–389, 2006.

[132] D. Sidibe, P. Montesinos, and S. Janaqi. Fast and robust image matching using contextual information and relaxation. In *Proc. 2nd International Conference on Computer Vision Theory and Applications*, pages 68–75, 2007.

[133] D. Sidibe, P. Montesinos, and S. Janaqi. Matching local invariant features : How can contextual information help ? In *Proc. EC-SIPMCS 07 - 6th EURASIP Conference Focused on Speech and Image Processing, Multimedia Communication and Services*, 2007.

[134] D. Sidibe, P. Montesinos, and S. Janaqi. Mise en correspondance d'invariants locaux par relaxation. In *Proc. ORASIS 07 - 11eme congrès francophone des jeunes chercheurs en vision par ordinateur*, 2007.

[135] D. Sidibe, P. Montesinos, and S. Janaqi. On matching local invariant features with context : An experimental evaluation. *Soumis à Electronic Letters on Computer Vision and Image Analysis*, 2007.

[136] M. Singh, M. Mandal, and A. Basu. Robust KLT tracking with gaussian and laplacian of gaussian weighting functions. In *Proc of 17th International Conference on Pattern Recognition*, 2004.

[137] J. Sivic and A. Zisserman. Video google : A text retrieval approach to object matching in videos. In *In ICCV*, pages 1470–1477, 2003.

[138] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE PAMI*, 22(12) :1349–1380, 2000.

[139] S. M. Smith and J. M. Brady. Susan, a new approach to low level image processing. *International Journal of Computer Vision*, 23 (1) :45–78, 1997.

[140] J. Song, Z. Chi, and J. Liu. A robust eye detection method using combined binary edge and intensity information. *Pattern Recognition*, 39 :1110–1125, 2006.

[141] S. Spors and R. Rabenstein. A real-time face tracker for color video. In *in IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[142] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7 (1) :11–32, 1991.

[143] K. Talmi, L. P. Bala, and J. Liu. Automatic detection and tracking of faces and facial features in video sequences. In *Picture Coding Symposium 1997*, 1997.

[144] D. S. Tauban and W. Marcellin. *JPEG2000 : Image Compression Fundamentals, Standards and Practice.* Kluwer Academic Publishers, 2001.

[145] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. In *Proc of the 7th ECCV*, pages 68–81, 2002.

[146] J.C. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proc. IEEE Int'l Conf. on Face and Gesture Recognition*, pages 54–61, 2000.

[147] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report 91-132, Carnegie Mellon University - Robotics Institue, 1991.

[148] K. Toyama. 'Look, Ma — No Hands!' hands-free cursor control with real-time 3d face tracking. In *Proc. Workshop on Perceptual User Interfaces (PUI'98)*, 1998.

[149] A. Tremeau, C. Fernandez-Maloigne, and P. Bonton. *Image Numérique Couleur : de l'acquisition au traitement.* Dunod, 2004.

[150] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1) :71–86, 1991.

[151] T. Tuytelaars. *Local, Invariant Features for Registration and Recognition.* PhD thesis, Katholieke Universiteit Leuven, Faculteit Toegepaste Wetenschappen, december 2000.

[152] T. Tuytelaars and L. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *Proc. International conference on visual Information Systems*, pages 493–500, 1999.

[153] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1) :61–85, 2004.

[154] T. Tuytelaars, L. Van Gool, L. Dhaene, and R. Koch. Matching affinely invariant regions for visual servoing. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 1601–1606, 1999.

[155] J. Van de Weijer and C. Schmid. Coloring local feature extraction. In *Proc. European Conference on Computer Vision*, pages 334–348, 2006.

[156] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *13th Int'l Conf. on Computer Graphics and Vision*, September 2003.

[157] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2) :137–154, 2004.

[158] A.P. Witkin. Scale-space filtering. In A. Bundy, editor, *Proceedings of the 8th Inter. Joint Conference on Artificial Intelligence*, pages 1019–1022. Karlsruhe, West Germany. William Kaufmann, 1983.

[159] J. Wu and Z. H. Zhou. Efficient face candidates selector for face detection. *Pattern Recognition*, 36 :1175–1186, 2003.

[160] X. Wu and B. Bhanu. Gabor wavelets for 3d objetct recognition. In *Proc. of 5th International Conference on Computer Vision*, pages 537–542, 1995.

[161] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. In *In Proc. ACCV*, pages 687–694, 1998.

[162] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1) :34–58, january 2002.

[163] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable template. *Int. J. Computer Vision*, 8(2) :99–111, 1992.

[164] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *AI Journal*, 78 :87–119, 1995.

[165] W. Zhao, R. Chellappa, A. Ronsenfeld, and P. J. Phillips. Face recognition : A literature survey. *ACM Computing Surveys*, pages 399–458, 2003.

[166] Z. Zivkovic and B. Kröse. On matching interest regions using local descriptors - can an information theoretic approach help ? In *Proc. BMCV*, pages 50–58, 2005.

**Titre :** Une technique de relaxation pour la mise en correspondance d'images
Application à la reconnaissance d'objets et au suivi du visage

**Résumé :** Le principal intérêt de l'utilisation des invariants locaux pour la mise en correspondance de différentes vues d'une même scène est le caractère local qui les rend robustes aux occultations et aux changements de point de vue et d'échelle. Néanmoins, cette localité limite le pouvoir discriminant des descripteurs locaux qui échouent dans les cas difficiles où l'ambiguité est élevée. Dans une première partie, nous proposons une méthode de mise en correspondance basée sur la relaxation qui prend en compte une information plus globale, dite *contextuelle*, afin de garantir des résultats corrects même dans les cas les plus difficiles. Nous présentons une application dans le cadre de la reconnaissance d'objets dans des scènes complexes.
Dans une seconde partie, nous abordons le problème de la détection et du suivi du visage dans une séquence d'image. Nous proposons une méthode simple et efficace pour la détection du visage dans une image couleur, et nous montrons comment l'algorithme de mise en correspondance peut être utilisé pour suivre efficacement le visage dans une séquence d'images.

**Mots clés :** Mise en correspondance d'images, reconnaissance d'objets, relaxation, détection de la peau, détection du visage, suivi du visage.

---

**Title:** A relaxation method for matching images with local invariant features :
Application to object recognition and face tracking

**Abstract:** Local invariant features are a powerful tool for finding correspondences between images since they are robust to cluttered background, occlusion and viewpoint changes. However, they suffer the lack of global information and fail to resolve ambiguities that can occur when an image has multiple similar regions. In the first part of this thesis, we describe a matching algorithm based on a relaxation scheme, which makes use of contextual information for better performances. We show how the relaxation scheme can be made robust and fast, and we apply it in the case of object recognition.
In the second part of this thesis, we tackle the problem of face detection and tracking in video sequences. We propose a simple and efficient face detection method in color images, and show how the matching method described in the first part can be used for tracking faces in video sequences.

**Keywords:** Matching, Relaxation, Local invariant features, object recognition, skin detection, face detection, face tracking.

**Titre :** Une technique de relaxation pour la mise en correspondance d'images
Application à la reconnaissance d'objets et au suivi du visage

**Résumé :** Le principal intérêt de l'utilisation des invariants locaux pour la mise en correspondance de différentes vues d'une même scène est le caractère local qui les rend robustes aux occultations et aux changements de point de vue et d'échelle. Néanmoins, cette localité limite le pouvoir discriminant des descripteurs locaux qui échouent dans les cas difficiles où l'ambiguité est élevée. Dans une première partie, nous proposons une méthode de mise en correspondance basée sur la relaxation qui prend en compte une information plus globale, dite *contextuelle*, afin de garantir des résultats corrects même dans les cas les plus difficiles. Nous présentons une application dans le cadre de la reconnaissance d'objets dans des scènes complexes.
Dans une seconde partie, nous abordons le problème de la détection et du suivi du visage dans une séquence d'image. Nous proposons une méthode simple et efficace pour la détection du visage dans une image couleur, et nous montrons comment l'algorithme de mise en correspondance peut être utilisé pour suivre efficacement le visage dans une séquence d'images.

**Mots clés :** Mise en correspondance d'images, reconnaissance d'objets, relaxation, détection de la peau, détection du visage, suivi du visage.

---

**Title:** A relaxation method for matching images with local invariant features :
Application to object recognition and face tracking

**Abstract:** Local invariant features are a powerful tool for finding correspondences between images since they are robust to cluttered background, occlusion and viewpoint changes. However, they suffer the lack of global information and fail to resolve ambiguities that can occur when an image has multiple similar regions. In the first part of this thesis, we describe a matching algorithm based on a relaxation scheme, which makes use of contextual information for better performances. We show how the relaxation scheme can be made robust and fast, and we apply it in the case of object recognition.
In the second part of this thesis, we tackle the problem of face detection and tracking in video sequences. We propose a simple and efficient face detection method in color images, and show how the matching method described in the first part can be used for tracking faces in video sequences.

**Keywords:** Matching, Relaxation, Local invariant features, object recognition, skin detection, face detection, face tracking.