REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DES FRERES MENTOURI CONSTANTINE 1

FACULTE DES SCIENCES DE LA TECHNOLOGIE

DEPARTEMENT D'ELECTRONIQUE

Année :
N° d'ordre :50/DS/2019
Série :01/Elec/2019

Thèse

En Vue de l'Obtention du Diplôme De Doctorat en Science

Spécialité : Electronique

Option : Traitement du Signal

Thème

DETECTION, RECONNAISSANCE ET SUIVI D'OBJET DANS LES IMAGES

Par : **BENAISSA Manel**

Soutenue publiquement le …/…/…… devant le jury composé de :

Présidente	**Prof. N. Mansouri**	Université Mentouri de Constantine
Rapporteur	**Prof. A. Bennia**	Université Mentouri de Constantine
Examinateur	**Pr. A. Charef**	Université Mentouri de Constantine
Examinateur	**Dr. K. Messaoudi**	Université Mohamed Cherif Messaidia
Examinateur	**Dr. H. Bourouba**	Université 8 mai 1945 de Guelma

Résumé

Dans cette thèse, nous proposons deux contributions différentes dans le domaine de la vision par ordinateur. La première contribution concerne la description des points caractéristiques et la correspondance des images. Nous avons proposé un descripteur invariant au changement de l'orientation sans étape supplémentaire dédiée à l'estimation de celle-ci. Nous avons exploité les informations fournies par deux représentations de l'image (intensité et gradient) pour une meilleure compréhension et représentation des points caractéristiques. Les informations fournies sont résumées dans deux histogrammes cumulés et utilisées dans la description et la correspondance des points clés. Dans le contexte de la détection d'objet, nous avons introduit un module basé sur la méthode de groupage k-means, pour réduire le nombre de fausses correspondances. Nous avons utilisé ce module après le processus d'appariement des points caractéristiques pour améliorer la précision de notre descripteur. Dans la seconde contribution, nous avons proposé un détecteur de bord basé sur une modélisation statistique de la surface de l'image. Nous avons utilisé deux mesures classiques et largement utilisées pour définir les propriétés de notre détecteur, à savoir la moyenne et l'écart type. Notre approche consistait à tirer parti des fluctuations de l'intensité dans une image pour une meilleure compréhension de sa surface. De plus, notre détecteur est capable de mettre en évidence les régions pertinentes dans celle-ci. Cette propriété a été exploitée dans le présent travail pour identifier les contours importants dans l'image. Outre sa nouveauté et son efficacité, le principal avantage de notre détecteur est sa simplicité, ce qui facilite son implémentation dans des terminaux à faible capacité de traitement. Il consomme également peu de mémoire et ne nécessite pas de phase d'apprentissage le rendant indépendant de la disponibilité des bases de données étiquetés. Les expériences ont montré la robustesse de notre descripteur vis-à-vis des changements d'image et une nette augmentation de la précision des descripteurs testés après la phase de pré-élimination des fausses correspondances. Notre détecteur de bord quant à lui présente de très bonnes performances de détection de contours par rapport aux détecteurs de l'état de l'art.

Mots-clés - *Compréhension des fonctionnalités; Description de fonctionnalité; Correspondance des fonctionnalités; détection d'objet; regroupement k-means; Détection de contour; Détection de bord; classification des images; vision par ordinateur.*

ملخص

في هذه الرسالة، نقترح مساهمتين مختلفتين في مجال رؤية الكمبيوتر. تتعلق المساهمة الأولى بوصف النقاط المميزة ومراسلات الصور. اقترحنا واصف ثابت لتغيير الاتجاه دون خطوة إضافية مخصصة لتقدير هذه. استخدمنا المعلومات التي قدمها تمثيلان للصورة (الشدة والتدرج) لفهم أفضل وتمثيل النقاط المميزة. يتم تلخيص المعلومات المقدمة في المدرج التكراري التراكمي وتستخدم في وصف ومراسلات النقاط الرئيسية. في سياق اكتشاف الكائنات، أدخلنا طريقة تعلم غير خاضعة للإشراف تستند على ك-مينز استخدمت في مرحلة ما قبل المطابقة لتحسين دقة واصفنا. في الإسهام الثاني، اقترحنا جهاز الكشف عن الحافة استنادًا إلى نمذجة إحصائية لسطح الصورة. استخدمنا مقياسين كلاسيكيين واسعة الاستخدام لتحديد خصائص كاشفنا، وهما المعدل والانحراف المعياري. كان نهجنا هو الاستفادة من تقلبات الشدة لفهم سطح الصورة بشكل أفضل. بالإضافة إلى ذلك، فإن كاشفنا قادر على إبراز المناطق ذات اهمية في الصورة. تم استغلال هذه الخاصية في هذا العمل لتحديد معالم الصورة المهمة. بالإضافة إلى جديتها وكفاءتها، فإن الميزة الرئيسية لجهاز الكشف لدينا هي بساطتها، مما يسهل تنفيذها في المحطات الطرفية ذات قدرة المعالجة المنخفضة. كما أنه يستهلك القليل من الذاكرة ولا يحتاج إلى مرحلة تعلم تجعله مستقلاً عن توفر قواعد البيانات المسمات. وأظهرت التجارب على متانة الواصف وجها لوجه مع التغيرات في الصورة، وزيادة في دقة الواصفات بعد مرحلة الكشف باستعمال مرحلة القضاء على المباريات الكاذبة. يتمتع كاشفنا ايظا بأداء جيد للغاية في الكشف عن الحافة مقارنة بأجهزة الكشف الحديثة.

Abstract

In this thesis, we propose two different contributions in the field of computer vision. The first contribution concerns the feature description and matching part, where we proposed an orientation invariant feature descriptor without an additional step dedicated to this task. We exploited the information provided by two representations of the image (intensity and gradient) for a better understanding and representation of the feature point distribution. The provided information is summarized in two cumulative histograms and used in the feature description and matching process. In the context of object detection, we introduced an unsupervised learning method based on k-means clustering. Which we used as an outlier pre-elimination phase after the matching process to improve our descriptor precision. In the second contribution, we proposed an edge detector based on a statistical modelization of the image surface. We used two classical and widely used metrics to constitute our detector properties, which are the mean and standard deviation. Our approach was to take advantage from the intensities fluctuation for a better understanding of the image surface. Moreover, our detector is able to highlight pertinent regions in the image. This property has been exploited in the present work, to identify important image contours. Besides its novelty and efficiency, the main advantage of our detector is its simplicity, which makes it easy to implement in terminals with low processing capacity, it's also little memory consumer and doesn't need a training phase which makes it independent from the availability of labeled datasets. Experiments shown the robustness of our descriptor to image changes and a clear increase in terms of precision of the tested descriptors after the pre-elimination phase. Our edge detector shows good performences compared to state-of-the-art detectors.

Keywords— *Feature understanding; Feature description; Feature matching; object detection; k-means clustering; Contour detection; Edge detection; image classification; computer vision.*

Dédicaces

Je dédie ce modeste travail en signe de respect et de reconnaissance

A mes parents

Mon Mari

Ma famille

Et

A tous mes amies.

Puisse Dieu les garder.

Menel Benaissa

Remerciements

Je remercie tout d'abord dieu le tout puissant (الحمد لله) pour son aide et pour le courage qu'il m'a donné afin de surmonter toutes les difficultés durant mes études.

Je remercie très particulièrement mon encadreur, **Prof. Abdelhak Bennia** pour son aide, ses conseils, et l'intérêt qu'il a su me porter, en me faisant part de sa grande expérience pour réaliser ce mémoire.

Je ne saurais oublier de remercier l'ensemble des membres de jury à savoir Prof. N. Mansouri, Dr. H. Bourouba Dr. K. Messaoudi et Prof. Charef d'avoir accepté de juger et d'évaluer mon travail.

Je remercie également tous les enseignants qui ont contribué à mon instruction durant mes années d'études.

Sans oublier tous ceux qui m'ont aidé de loin ou de près à l'élaboration de ce travail, je leur exprime ma profonde sympathie et leur souhaite beaucoup de biens.

Table des matières

List des Tableaux et Figures

Introduction générale……1

Chapitre I : Etat de l'art de la détection, la reconnaissance et suivi d'objet dans les images

I.1 Préambule..……....9

I.2 Introduction..……....12

I.3 Etat de l'art des détecteurs et descripteurs de points caractéristiques.…….……...14

I.3.1 Etat de l'art des détecteurs de points caractéristiques ……14

I.3.1.1 Détecteur Harris……...19

I.3.1.2 Détecteur MSER……..20

I.3.2 Etat de l'art des descripteurs de points caractéristiques ……..22

I.3.2.1 Descripteurs basé sur les histogrammes……...22

I.3.2.2 Descripteurs binaires……28

I.3.2.3 Descripteurs de formes…….…29

I.3.2.4 Descripteurs basé sur l'apprentissage…….....31

I.4 Etat de l'art des détecteurs de bords…… ……...32

I.4.1 Détecteurs de bords classiques……. ……...32

I.4.2 Détecteurs basées sur le filtrage des images……. ……..33

I.4.3 Détecteurs basées sur l'apprentissage……. ……33

I.5 Conclusion……...34

Chapitre II : Présentation du descripteur de points caractéristiques proposé

II.1 Introduction……....36

II.2 Présentation du descripteur……....37

II.2.1 Détection des points clés .....……38

II.2.2 Description des points clés .....…….39

II.2.2.1 Extraction et quantification des patchs.....…….40

II.2.2.2 Constitution des Histogrammes…….....……..43

II.2.2.3 Propriété d'invariance au changement d'orientation.…….45

II.2.2.4 La correspondance des points clés.....……46

II.3 Détection d'objets dans les images.……...48

II.4 Processus de pré-élimination des fausses correspondances……...51

II.4.1 K-means Clustring……..51

II.5 Conclusion ……52

Chapitre III : Présentation du détecteur de bords et d'objets pertinents dans les images

III.1 Introduction……...54

II.2 Présentation du détecteur…….……...55

II.2.1 Approche du détecteur…..…….…….55

II.2.2 Processus de classification……...……...57

II.2.3 Variabilité globale de l'image d'intensité……...58

II.2.4 Détection des régions pertinentes dans l'image……..59

II.3 Conclusion…….....66

Chapitre IV : Présentation des résultats expérimentaux du descripteur de points caractéristiques et du détecteur de bords proposés

IV.1 Résultats expérimentaux du descripteur ADOCH ……...…….……...68

IV.1.1 Correspondance d'images…..……...68

IV.1.2 Reconnaissance d'objets……...……79

IV.2 Résultats expérimentaux du détecteur de bords…….……...82

IV.3 Conclusion ……...88

Conclusion générale……89

Bibliographie……90

Liste Des Figures

Chapitre I

FIG. I.1 Exemple montrant la première et la deuxième dérivée d'un signal d'image dans une direction donnée……..9

FIG. I.2 Exemple montrant la première dérivée d'une image dans les directions x et y…10

FIG. I.3 Exemple montrant la magnitude et direction du gradient de l'image I.....10

FIG. I.4 Exemple montrant le Laplacien de l'image I.……....11

FIG. I.5 Exemple d'un filtre gaussien avec $\boldsymbol{\sigma = 1}$.……...…...11

FIG. I.6 Exemple montrant l'image I, filtré par un filtre gaussien avec $\boldsymbol{\sigma = 2}$.8…...…...12

FIG. I.7 Exemple montrant le résultat d'un détecteur de caractéristiques……....14

FIG. 1.8 Les différents changements pouvant affecter l'image d'origine……...….15

FIG. I.9 Illustration de la fonction f ……...….20

FIG. I.10 Effet de la transformation affine sur une région donnée……...…20

FIG. I.11 L'ellipse autour de la région détectée et son équivalent autour de la région transformé……21

FIG. I.12 L'ellipse autour de la région détectée et son équivalent autour de la région transformé……22

FIG. I.13 Pyramide de gradients : 3 octaves de 6 gradients……...…...23

FIG. I.14 Construction de la pyramide de différence de gaussiens (DoG) à partir de la pyramide de gradients……..24

FIG. I.15 Exemple de détection d'extremums dans l'espace des échelles……...…24

FIG. I.16 Construction de l'histogramme des orientations……...……...25

FIG. I.17 Construction d'un descripteur SIFT……...…..26

FIG. I.18 Exemple de constitution du descripteur SURF……...…..27

FIG. I.19 Exemple de constitution des descripteurs binaires……...……29

FIG. I.20 Exemple de détection des bords dans une photographie……...……32

Chapitre II

FIG. II.1 Schéma synoptique du descripteur …….……...…….....38

FIG. II.2 Processus d'extraction des points clé et de leur environnement immédiat...….40

FIG. II.3 Exemple montrant le contenue des patches extrais à partir des trois images….....41

FIG. II.4 Illustration du processus de quantification ……..……42

FIG. II.5 Illustration du processus de création des histogrammes.…...……...43

FIG. II.6 Une comparaison de deux patchs entourant le même point d'intérêt de l'image d'origine et de sa version pivotée à partir de la base de données VanGogh……...44

FIG. II.7 Histogrammes résultants de deux patchs entourant le même point d'entité dans l'image d'origine et sa version pivotée……..45

FIG. II.8 Exemple montrant le processus de correspondance entre les histogrammes obtenus à partir d'une paire de points clés extraites de l'image d'origine et sa version pivotée…….47

FIG. II.9 Quelques résultats visuel obtenues par le descripteur proposé sous diffèrent changement d'angle et ceux sans aucun recours à une estimation préalable de l'orientation…….48

FIG. II.10 Illustre le processus mis en œuvre pour le redimensionnement des patchs au niveau de l'image de référence dans le cadre de la détection d'objets……..50

FIG. II.11 Exemple illustrant le processus de pré-élimination des fausses correspondances adopté dans le cadre de la détection des objets…….....…….51

Chapitre III

FIG. III.1 Exemple illustrant la surface des intensités correspondant à une partie d'image ou la moyenne de ces intensités est représenté par le seuil vert et la variation de la std par les très orange……...55

FIG. III.2 Exemple illustrant le processus de classification utilisé par le détecteur proposé 56

FIG. III.3 Exemple montrant quelques exemples de blocs classé comme rejetée ou gardé en fonction de leurs std…….57

FIG. III.4 Exemple montrant la surface de l'image en utilisant les Std's avec des blocs de 3x3……...58

FIG. III.5 Les images résultantes, après le processus de classification, utilisent une taille de bloc de 5x5……...59

FIG. III.6 Influence de la taille du bloc sur le résultat finale du processus de classification et son effet sur la localisation des zones pertinentes dans l'image……60

FIG. III.7 Influence de la taille du bloc et du seuil de classification sur le résultat finale du processus de classification……...61

FIG. III.8 Résultats du processus de classification de deux images lisses et contrastées pour différentes tailles de blocs……62

FIG. III.9 Exemple montrant les multiples bords obtenus à partir de deux images lisses et contrastées pour différentes tailles de blocs……..64

FIG. III.10 Exemple montrant le processus de seuillage utilisé par le détecteur proposé et le résultat obtenu de ce dernier…….65

Chapitre IV

FIG. IV.1 Performances du descripteur proposé sur l'ensemble de données d'Oxford.…….71

FIG. IV.2 Les Performances du descripteur proposé sur la base de données de Salzmann pour les objets 3D déformables……73

FIG. IV.3 Exemple montrant les performances du descripteur proposé obtenu à partir de la base de données de Strecha pour le cas multiview……74

FIG. IV.4 Les Performances du descripteur proposé sur le jeu de données de Heinly pour des changements pur d'échelle et d'orientation…..……..76

FIG. IV.5 Performances quantitatives des différents descripteurs en utilisant le F-score à 50%.....77

FIG. IV.6 Quelques performances visuelles du descripteur proposé sur les quatre bases de données proposées……...78

FIG. IV.7 Autres exemples montrant les performances du descripteur proposé sur d'autres bases de données……...……..79

FIG. IV.8 Exemples d'images de la base de données d'objet de maison proposé, ou l'on voie quatre objets avec deux images de test pour chacun d'eux……..80

FIG. IV.9 Illustration visuelle de montrant quelques exemples de détection d'objet avant et après la phase de pré-élimination……..81

FIG. IV.10 Schémas bloc montrant le processus de sélection de la taille des blocs qui seront utilisé dans l'opération de classification et de détection des contours……83

FIG. IV.11 Illustration des bords obtenus en utilisant le détecteur proposé par rapport au détecteur Canny…….……...……...84

FIG. IV.12 Illustration visuel des résultats obtenus du détecteur proposé comparée à celles du détecteur HED……...86

FIG. IV.13 Les performances du détecteur proposé comparées à celles du détecteur Sketch Tokens……...……...87

FIG. IV.14 Les performances du détecteur proposé comparées à celles du détecteur DeepEdge……..……...87

FIG. IV.15 Les performances du détecteur proposé comparées à celles du détecteur

gPb…….....……...……..87

FIG. IV.16 Quelques performances visuelles du détecteur proposé sur la base de données VOC 2012……87

Liste Des Tableaux

Chapitre I

TABLEAU I.1 Travaux antérieurs sur l'évaluation des performances des détecteurs/ descripteurs de caractéristiques……30

Chapitre III

TABLEAU III.1 Variation de la moyenne et la std globale pour différentes tailles de blocs.……63

Chapitre IV

TABLEAU IV.1 Montre le taux de précision totale obtenus pour différentes bases de donnes avec et sans la phase de pré-élimination des fausses correspondances.……..……80

TABLEAU IV.2 Montre le temps de calcul des différents descripteurs (par point clé)……....81

TABLEAU IV.3 Montre les performances du détecteur proposé en termes de résultats et de temps de calcule par rapport aux détecteurs de l'état de l'art…….……..85

TABLEAU III.4 Montre les performances du détecteur proposé par rapport aux détecteurs de Mély et al. [26] et HED [37] sur la base de données Multicue [26]……...……86

Liste Des Acronymes

SIFT Scale-Invariant Feature Transform

SURF Speeded-Up Robust Features

BRIEF Binary Robust Independent Elementary Feature

DAISY Dense Descriptor Applied to Wide-Baseline Stereo

CNN Convolutional Neural Networks

HED Holistically Nested Edge Detection

MSER Maximally Stable Extremal Regions

ADOCH Absolute Difference of Cumulated Histograms.

TPO Taux de Précision d'Objet

CRC Taux de Précision Total

SCF Seuil de Contour Fixe

SVI Seuil Variable par Image

PM Précision Moyenne

IPS Image Par Seconde

Introduction générale

Un grand intérêt a été porté ces dernières années au domaine de la vision par ordinateur, ceci est expliqué par les multiples champs d'utilisation de ce dernier. De la robotique à la vision stéréo, en passant par le domaine médical et la sécurité. De plus en plus d'applications font appel à la vision par ordinateur afin de réaliser divers taches, ceci se traduit par le développement d'algorithmes sophistiqués et peu couteux en termes d'implémentation. Ces sciences du traitement de l'information et de la vision artificielle permettent de concevoir des outils toujours plus performants et permettent à l'utilisateur final d'obtenir toujours plus de leurs images : des couleurs plus belles, des images plus rayonnantes, une meilleure précision, etc. Historiquement, c'est dès les années 50, en physique des particules, que les premières images sont traitées dans le but de détecter des trajectoires issues du bombardement bilatéral des atomes, afin de scruter les composantes infimes de la matière. Dans les années 60, les chercheurs se sont intéressés à la lecture optique de caractères (OCR). Toutes ces applications sont issues de trois domaines forts : la restauration, l'amélioration et la compression d'images. A partir 70, on se concentre sur l'extraction automatique d'informations entre autres : contours, régions, et on a les premières méthodes d'interprétation d'images avec l'apparition des systèmes experts. C'est vers les années 80 que le concept de vision par ordinateur est né avec l'apparition de la première théorie formelle de la vision par ordinateur, proposée par David Marr. Il a été un des premiers à définir les bases formelles de la vision par ordinateur en intégrant des résultats issus de la psychologie, de l'intelligence artificielle et de la neurophysiologie. Marr propose un cadre pour le système de vision avec l'hypothèse qu'on peut étudier les principes de la perception visuelle en considérant que l'objectif de la vision est de décrire des scènes (appelé aussi *reconstruction de scènes*). Enfin, à partir des années 90 jusqu'à présent, des recherches basées sur les différentes représentations des images combiné à l'apprentissage machine ont été proposées.

Sujet de recherche

Dans cette thèse, nous nous intéressons à deux aspects fondamentaux dans le domaine de la vision par ordinateur. A savoir, la description et l'appareillage des points caractéristiques entre deux images. Et d'autre part, la détection des contours d'objets ou de scènes dans une image donnée.

Description et appariement de points caractéristiques

La description des caractéristiques et l'appariement d'images sont deux problèmes importants dans la vision artificielle et la robotique, leurs applications continuent à se développer dans divers domaines. Un descripteur de caractéristique idéal doit être robuste aux transformations d'image telles que l'échelle, l'illumination, la rotation, le bruit et les transformations affines. La possibilité de faire correspondre des points caractéristiques entre deux ou plusieurs images d'une scène est une composante importante dans de nombreuses tâches de la vision par ordinateur telles que la structure du mouvement [1], le SLAM [2] visuel (localisation et cartographie simultanées), la reconnaissance d'objet, la reconstruction 3D et le suivi d'objets. Une caractéristique hautement distinctive est requise, puisqu'elle peut être correctement associée avec une probabilité élevée. La précision de la correspondance d'image est fortement affectée par la précision des algorithmes utilisés à cette fin. Il est important de

rendre la description aussi unique que possible, de sorte qu'elle puisse résister à diverses transformations de l'image. Le descripteur SIFT (Scale-invariant feature transform) introduite par Lowe dans [3], est une approche réussie pour la détection et la description de points caractéristiques. Même si cela nécessite une grande complexité de calcul, ce qui est un inconvénient dans le cas des applications en temps réel. SIFT s'est avéré très efficace dans les applications de reconnaissance d'objets. Plusieurs variantes et extensions de ce dernier ont été proposées pour améliorer sa complexité de calcul comme dans [4,5]. Le descripteur SURF (speeded-up robust features) développée par Bay et al. dans [6], est basé sur le même principe que le SIFT, mais il utilise un schéma différent et fournit de meilleurs résultats plus rapidement. Ces descripteurs ont été largement utilisés pour la description des caractéristiques et ont montré leur robustesse sur les images qui ont subi des transformations telles qu'une rotation, une variation d'échelle et des changements d'éclairage.

De nos jours, le déploiement d'algorithmes de vision sur les téléphones intelligents et les dispositifs embarqués à faible complexité et mémoire de calcul a conduit à la nécessité de rendre les descripteurs plus rapides et plus compacts tout en restant robustes aux changements. Par conséquent, de nouveaux algorithmes ont été développés dans ce sens.

Le binary robust independent elementary feature (BRIEF) [7], est l'une des alternatives proposées pour le SIFT, qui est moins complexe et avec des performances de correspondance presque similaires, cependant ce dernier est sensible au changement d'orientation. Afin de remédier à cette sensibilité, Rublee et al. Proposèrent le rotated BRIEF (ORB), qui inclue le calcul et la compensation de l'orientation [8]. Le binary robust invariant scalable keypoints (BRISK) [9], et le fast retina keypoint (FREAK) [10] sont également de bons exemples. Même si les descripteurs binaires sont efficaces, leur principal inconvénient peut être résumé comme ayant une capacité distinctive limitée, puisque ces descripteurs sont généralement construits sur un ensemble de comparaisons d'intensité par paire où chaque point d'échantillonnage représente un seul pixel (par exemple, BRIEF, ORB). D'autres descripteurs telque le BRISK et FREAK utilisent un filtre gaussien sur les pixels voisins du point d'intérêt avant la phase de comparaison. Cette approche est très sensible à la perturbation des emplacements des points d'échantillonnage. De plus, les comparaisons d'intensité par paires capturent des informations très limitées d'une région d'image.

Les descripteurs d'entités basés sur la forme ont également été largement étudiés et utilisés. Tels que le DAISY [11], LIOP [12] et GSURF [13].

Plusieurs approches d'apprentissage basées sur les réseaux de neurones convolutionnels (CNN) ont été proposées. Tels que AlexNet [14], VGG [15], GoogLeNet [16] et ResNet [17]. Récemment, plusieurs combinaisons du pouvoir discriminant de CNN avec des descripteurs binaires à faible coût de calcul ont été proposées pour plusieurs applications. Tels que DeepBit [18], où les descripteurs binaires compacts sont appris de manière non supervisée et ont atteint l'état de l'art des descripteurs de caractéristiques binaires. Des approches d'optimisation ont été proposées afin d'atteindre ou de surperformer les descripteurs les plus avancés tels que dans [19], où les auteurs ont proposé un descripteur binaire adaptatif en ligne, optimisé indépendamment pour chaque patch d'image. Dans [20], les auteurs ont proposé une formulation polyvalente d'apprentissage, qui optimise les descripteurs de caractéristiques

locales pour la correspondance avec le plus proche voisin. Dans [21], une méthode d'apprentissage de fonction binaire locale sensible au contexte a été proposée pour les applications de reconnaissance faciale. Un remplacement convolutionnel supervisé de SIFT a été proposé par [22], qui est un pipeline avec détection de points caractéristiques, estimation de l'orientation et description des caractéristiques. Dans [23], les auteurs ont proposé un détecteur et un descripteur de caractéristiques auto-supervisées, fonctionnant sur des images de taille normale et calculant conjointement les emplacements des points d'intérêt au niveau des pixels et leurs descripteurs associés en un seul passage. Même si l'efficacité des méthodes d'apprentissage n'est pas discutable, la nécessité d'une phase de formation efficace et coûteuse en plus de la disponibilité obligatoire de grands ensembles de données annotées pour atteindre les performances des descripteurs de caractéristiques traditionnelles demeure un inconvénient. En outre, leur grande sensibilité aux changements d'orientation est un réel inconvénient pour une application réelle dans le domaine de la vision. Par conséquent, une étape supplémentaire dédiée à l'estimation de l'orientation est ajoutée au pipeline afin de remédier à cet inconvénient.

L'objectif principal à travers ce travail est de proposer un descripteur de caractéristiques invariant au changement d'orientation qui ne nécessite aucune étape supplémentaire dédiée à cette tâche.

Détection de contours

La détection des limites est un autre problème classique dans le domaine de vision par ordinateur. Elle est utilisée dans beaucoup de tâches telles que la segmentation d'images [24,25], la détection d'objets [26,27], et l'étiquetage sémantique. En plus de la détection des bords dans l'image, la plupart de ces tâches nécessitent une classification typique de ces derniers afin d'être utilisé dans des applications spécifiques. La plupart des méthodes de détection de contour peuvent être divisées en deux branches : les méthodes locales et globales.

Les méthodes locales effectuent la détection de contour par raisonnement en utilisant de petits patches à l'intérieur de l'image. Les méthodes globales quant à elles prédisent les contours en fonction des informations de l'image complète. Récemment, des études proposent d'appliquer des méthodes d'apprentissage approfondi à la détection des contours.

Les premières méthodes se concentraient principalement sur l'utilisation de l'intensité et des dégradés de couleurs. Robinson [28] a discuté d'une mesure quantitative dans le choix des coordonnées de couleur pour l'extraction de bords et de limites visuellement significatives. Dans [29, 30], les auteurs ont présenté des algorithmes basés sur la théorie du passage par zéro. Sobel [31] a proposé le célèbre opérateur Sobel permettant de calculer la carte de gradient d'une image, puis de générer des arêtes en appliquant un seuil à la carte. Une version étendue de Sobel, nommé Canny [32], a ajouté le lissage gaussien comme étape de prétraitement et a utilisé le seuil double pour obtenir les bords, de cette façon il est plus résistant au bruit. En fait, il est toujours très populaire dans diverses tâches en raison de son efficacité remarquable. Cependant, ces premières méthodes semblent avoir une précision médiocre et sont donc difficiles à adapter à la situation actuelle.

Plus tard, les chercheurs ont eu tendance à concevoir manuellement des détecteurs à l'aide de caractéristiques de bas niveau tels que l'intensité, le dégradé et la texture, puis à utiliser un

paradigme d'apprentissage sophistiqué pour classer les pixels entant que bord ou pas [33,34]. Konishi et al. [35] ont proposé les premières méthodes basées sur les données en apprenant les distributions de probabilité des réponses correspondant à deux ensembles de filtres de bords. Martin et al. [36] ont proposé le détecteur Pb basé sur une combinaison de différentes modifications de la luminosité, de la couleur et de la texture d'une image donnée, et à former un classificateur pour combiner les informations de ces fonctionnalités. Arbelàez et al. [37] ont amélioré le Pb en gPb en utilisant des coupes normalisées standard [38] pour combiner les indices locaux dans un cadre de globalisation. Lim [39] a proposé de nouvelles fonctionnalités, des jetons d'esquisse pouvant être utilisés pour représenter les informations de niveau intermédiaire. Dollar et al. [40] ont utilisé des forêts à décision aléatoire pour représenter la structure au niveau local. Les forêts structurées produisent des bords de haute qualité. Cependant, toutes les méthodes ci-dessus sont développées sur la base de fonctionnalités conçues à la main, ce qui a une capacité limitée à représenter des informations de haut niveau pour une détection de bord sémantiquement significative.

Avec le développement vigoureux de l'apprentissage en profondeur récemment introduit, une série d'approches fondées sur l'apprentissage ont été proposé. Ganin et al. [41] ont proposé N4-Fields qui combine les réseaux de neurones convolutionnelles CNN avec la recherche du voisin le plus proche. Shen et al. [42] ont divisé les données de contour en sous-classes et ajusté chaque sous-classe en apprenant les paramètres du modèle. Hwang et al. [43] ont considéré la détection de contour comme une classification par pixel. Ils ont utilisé DenseNet [44] pour extraire un vecteur de caractéristiques pour chaque pixel, puis le classificateur SVM a été utilisé pour classifier chaque pixel comme étant un bord ou non. Xie et al. [45] ont récemment mis au point un détecteur de bords efficace et précis dit HED, qui assure la formation et la prédiction image par image. Cette architecture imbriquée de manière globale connecte les couches composant le réseau. Ce dernier est basé sur l'architecture du VGG16 [46], il est composé de plusieurs couches successives composé chaque une d'une étape de convolution avec une taille donnée, d'une couche de déconvolution et d'une couche de softmax. Plus récemment, Liu et al. [47] utilisent des étiquettes étendues générée par des bords ascendants pour guider le processus de formation de HED, et a permis certaines améliorations. Li et al. [48] ont proposé un modèle complexe pour l'apprentissage non supervisé de la détection des contours, mais les performances sont pires que les détecteur précédents formé sur la base de données BSDS500.

L'approche proposée

Nous allons présenter dans cette section deux approches différentes dans deux domaines distincts de la vision par ordinateur ou de manière plus générale, le traitement d'images.

La première concerne la description et l'appariement des points caractéristiques entre deux images ou nous avons proposé un descripteur de points caractéristiques en tenant compte de l'information (en termes d'intensités et de gradients) fournit par entourage immédiat des points clés sélectionnée.

Dans le cadre de la détection d'objets dans l'image, nous avons ajouté au descripteur proposé un module supplémentaire dédié à une pré-élimination des fausses correspondances. Ce module est situé entre la phase d'appareillement des points clés et le consensus d'échantillon aléatoire des correspondances dit RAndom SAmple Consensus (RANSAC), qui a pour objectif de sélectionner un nombre déterminé de correspondances de façon aléatoire. Ce module basé

sur la méthode de regroupement par k-means a conduit à une nette amélioration des résultats obtenus par le descripteur proposé. D'autre part, sa modularité lui offre la possibilité d'être appliqué à d'autres descripteurs ou là également, celui-ci a montré son efficacité en termes d'élimination de fausses correspondances.

La deuxième contribution concerne la détection des bords, ou nous avons proposé un détecteur basé sur la modélisation statistique de la surface d'une image donnée en se basant sur deux mesures phare à savoir, la moyenne et l'écart type de l'intensité.

Le concept du détecteur proposé est simple, ce dernier est basé sur le découpage des images en blocs de tailles différentes en première partie. Ces derniers sont par la suite classifiés en deux catégories, contenant ou pas des bords importants en se basant sur leurs mesures locales.

A. Description et correspondance des points caractéristiques

Dans cette thèse, nous proposons un nouveau descripteur de caractéristiques, basé sur une combinaison efficace de l'intensité et du gradient des images dans une paire d'histogrammes bidimensionnelles. Le descripteur proposé ne nécessite pas une étape de calcul et de compensation de l'orientation, puisque sa structure lui offre la propriété d'invariance au changement d'angle. Il est important de noter que l'invariance à l'orientation de la plupart des descripteurs efficaces d'entités est assurée par une étape supplémentaire avant le processus de description du point caractéristique.

Nous avons remarqué que le descripteur proposé est très efficace contre le changement de l'orientation, la compression JPEG, le point de vue et les déformations 3D. Les changements d'intensité et de flou sont également supportés par le descripteur proposé. Nous estimons que le descripteur proposé est bien adapté pour les caméras stéréo Multiview ou les caméras de surveillance étant donnée sa simplicité architecturale et sa facilité de mise en œuvre. Ce qui constitue un point important dans le cas de terminaux à capacité de traitement limitée.

De plus, nous avons proposé une méthode de pré-élimination des fausses correspondances, basée sur l'apprentissage non supervisée. Cette étape a été ajoutée dans le contexte de la détection d'objet, après le processus de correspondance. Ceci a conduit à une nette amélioration des capacités du descripteur proposé en termes de précision de détection.

Nous avons évalué le descripteur proposé sur plusieurs ensembles de données et nous l'avons comparé aux descripteurs de caractéristiques les plus utilisés. Les résultats obtenus montrent l'efficacité de la méthode proposée.

B. Détection de contours

Dans cette partie, nous présentons une approche de détection de bords simple et efficace. Nous avons profité des informations fournies par l'intensité de l'image d'origine pour modéliser sa surface et détecter ces bords. Nous avons seulement utilisé deux mesures statistiques, qui sont la moyenne et l'écart type (std) des intensités pour déterminer la position du bord dans l'image.

Ces mesures ont été utilisées dans des recherches antérieures comme dans [49], où les auteurs ont utilisé la moyenne et l'écart type des images en échelle de gris pour définir un modèle de forme statistique afin d'obtenir un meilleur processus de segmentation de l'image.

Ou dans [50], où les auteurs ont utilisé des mesures d'analyse statistique locales pour construire des fonctions d'énergie pour les tâches de segmentation.

Même si l'efficacité de ces contributions est prouvée, leur charge de calcul reste lourde puisque les méthodes proposées sont basées sur le calcul de fonctions multi-paramétriques. De plus, leurs performances dépendent essentiellement des conditions initiales.

Organisation du manuscrit

Le premier chapitre de cette thèse est dédié à l'état de l'art des différents descripteurs proposé. Que ce soit les descripteurs basé sur les histogrammes de gradients, les descripteurs binaires, les descripteurs de forme ou encore ceux basé sur l'apprentissage machine. Nous présenterons également l'état de l'art des principaux détecteurs de bords proposé dans la littérature. Chaque approche aussi différente soit elle présente des avantages et des inconvénients. Ainsi, le but du chapitre I est donc de mettre en avant l'historique des diverses descripteurs de caractéristiques et détecteurs de bords dans la littérature.

Dans le chapitre II, nous développerons de façon détaillée le descripteur proposé. Basé sur une paire d'histogramme bidimensionnel contenant les intensités et les gradients du point caractéristique et de ces voisins immédiat. Nous développerons également de manière détaillée le module de pré-élimination des fausses correspondances basé sur la méthode k-means.

Le chapitre III sera dédiée à la présentation détaillée du détecteur de bord proposé. Hormis son efficacité, du détecteur de bord proposé a la capacité de fournir des informations importante sur la nature de l'image en question. Quel soit de type contrasté correspondons a une image richement fournit en termes de bords, ou lisse en référence à une image aillant peut de bords, le détecteur proposé fournies des détails cruciaux sur le nature des images utilisé.

Par ailleurs, basé sur des mesures statistiques simples de l'intensité de l'image originale, le détecteur proposé s'adapte bien sur des terminaux à faible capacité de calcule.

Nous évaluerons les performances des approchent proposé dans le chapitre IV en se fondant sur plusieurs bases de données publiquement disponible. Ainsi qu'une comparaison exhaustive à l'état de l'art des diffèrent descripteurs et détecteurs proposés dans la littérature.

Nous terminerons cette thèse par la conclusion, ou nous avons rassemblé les principaux résultats et mis en relief les techniques que nous avons élaborées en expliquant les principaux apports de celles-ci. Nous avons aussi donné les perspectives ouvertes par le travail proposé.

Chapitre I

Etat De L'art de la détection, la reconnaissance et du suivi d'objets dans les images

I.1. Définitions

Nous commençons d'abord avec quelques définitions de base. Une image définie dans le monde réel est considérée comme une fonction de deux variables réels, par exemple $I(m,n)$, où I est l'amplitude de l'image à la position (m,n).

Une image numérique $I(x,y)$ décrite dans un espace discret bidimensionnel est tout simplement le résultat de la discrétisation d'une image $I(m,n)$ dans un espace continue, au travers d'un processus d'échantillonnage.

Un traitement réalisé sur une image $I(x,y)$ est en fait une fonction qui s'applique sur un pixel, une zone dans l'image ou toute l'image. On parle alors d'une opération de type point, local ou global. La complexité de l'opération est liée à la taille de la zone sur laquelle elle est appliquée. Ainsi, dans le cas d'un point, la complexité est constante, tant dis que dans le cas d'une opération local avec une zone définie par un carré de taille PxP, la complexité est de P^2. De même pour l'opération globale, la complexité est N^2 où NxN est la taille d'image.

En générale, la détection de caractéristiques dans l'image concerne souvent des dérivées de l'image. Nous convenons $I_i(x,y)$, $I_{ii}(x,y)$ comme étant respectivement, les dérivées d'ordre 1 et 2 de l'image $I(x,y)$ selon la direction i.

$$I_i = \frac{\partial I}{\partial i} \approx I(i+1) - I(i) \qquad (I.1)$$

$$I_{ii} = \frac{\partial^2 I}{\partial i^2} \approx I(i+1) - 2 * I(i) + I(i-1) \qquad (I.2)$$

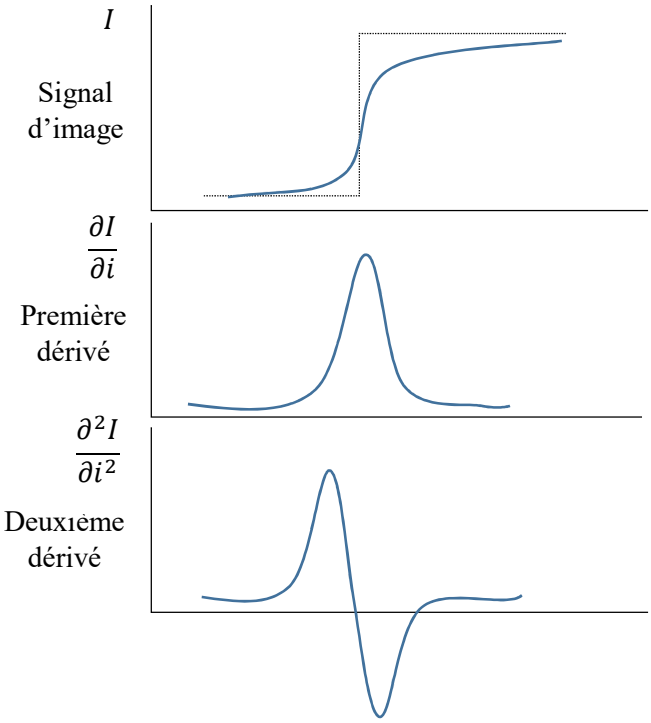


Fig. I.1. Exemple montrant la première et la deuxième dérivées d'un signal d'image dans une direction donnée.

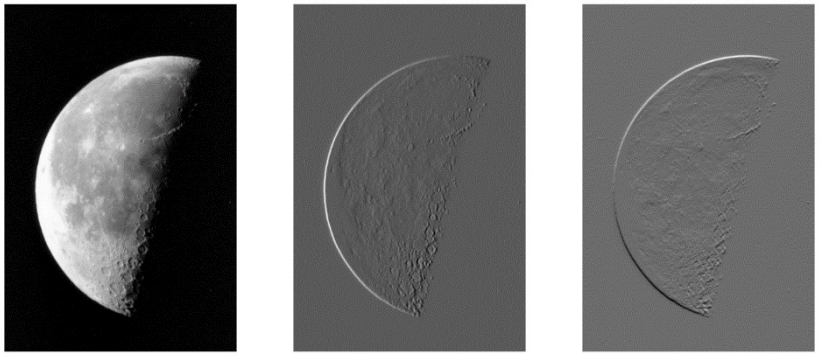


Image originale | Première dérivée dans la direction de x | Première dérivée dans la direction de y

Fig. I.2. Exemple montrant la première dérivée d'une image dans les directions x et y.

Le gradient de la fonction d'image est donné par le vecteur :

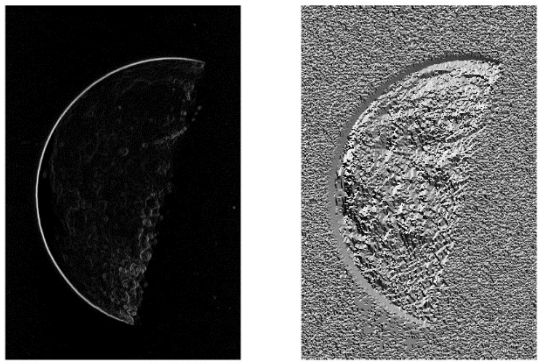
$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right] \qquad (I.3)$$

La magnitude du gradient d'une image est donnée par :

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \qquad (I.4)$$

La direction du gradient d'une image est donnée par :

$$\theta = tan^{-1}\left(\frac{\partial I}{\partial y} \Big/ \frac{\partial I}{\partial x}\right) \qquad (I.5)$$



Magnitude du gradient de I | Direction du gradient de I

Fig. I.3. Exemple montrant la magnitude et la direction du gradient de l'image I.

Le Laplacien est donné par :

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \qquad (I.6)$$

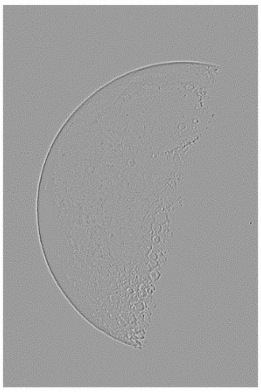
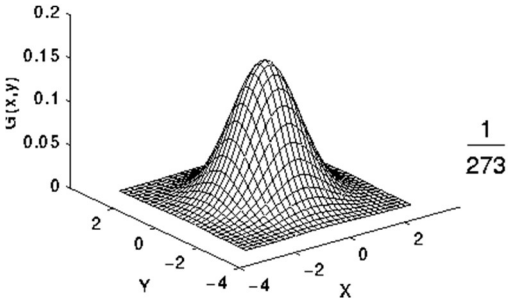


Fig. I.4. Exemple montrant le Laplacien de l'image I.

Le Filtre Gaussien et ses dérivées

En vision par ordinateur, on lisse souvent l'image originale avant de la traiter pour éviter des bruits. Le lissage est fait en convoluant l'image avec un filtre Gaussien. La fonction du filtre Gaussien avec une variance σ^2 dans un espace bidimensionnel à un point $p = [x, y]^T \in R^2$ est définie par :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (I.7)$$



$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Fig. I.5. Exemple d'un filtre gaussien avec $\boldsymbol{\sigma = 1}$.

La formule de lissage d'une image est :

$$L(x, y, \sigma) = I(x, y) \otimes G(x, y, \sigma) \qquad (I.8)$$

Fig. I.6. Exemple montrant l'image I, filtré par un filtre gaussien avec $\boldsymbol{\sigma = 2}$.

De cette manière, si on calcule la dérivée de l'image lissée, il suffit de convoluer l'image originale avec la dérivée du même ordre du filtre Gaussien. La dérivée d'ordre d d'une Gaussienne selon la direction $\theta = 0$ est définie par :

$$G_d^0(x,\sigma) = \frac{\partial^d}{\partial^d x} G(x,\sigma) \qquad (I.9)$$

Pour une direction quelconque θ, G_d^θ est définie par :

$$G_d^\theta(x,\sigma) = G_d^0(R_\theta x,\sigma) \qquad (I.10)$$

Avec la matrice de rotation :

$$R_\theta = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \qquad (I.11)$$

Il a été démontré que les caractéristiques locales sont bien adaptées à des tâches de correspondance automatique d'images et de suivi d'objets. L'extraction des caractéristiques de bas niveau à travers diverses transformations et filtrage des images est la première étape de toute procédure d'analyse d'image et est essentielle pour la performance de la vision stéréoscopique et des systèmes de reconnaissance d'objets.

La recherche concernant la détection des points d'intérêt, de bords et de traits circulaires ou ponctuels est particulièrement riche et de nombreuses procédures ont été proposées dans la littérature à ce sujet.

I.2. Introduction

Le monde moderne est entouré de masses gigantesques d'informations visuelles numériques. Afin d'analyser et organiser ces informations, le développement des techniques d'analyse d'image devient une exigence majeure. Ainsi, la proposition de méthodes qui pourraient analyser automatiquement le contenu sémantique des images ou des vidéos sont particulièrement utiles. Un aspect important du contenu de l'image concerne les objets qui s'y trouvent. Il y a donc un besoin de techniques de reconnaissance efficaces.

La reconnaissance visuelle d'objets est un sous-problème d'un problème plus général : celui de la perception visuelle. C'est autour de 1960 que les premiers essais de reconnaissance artificielle de formes, à partir d'images, sont faits afin de reconnaître des trajectoires issues de collisions entre les particules. Dès les origines, et autour des années 80, avec l'apparition de la première théorie formelle de la vision artificielle proposée par Marr [29], et jusqu'à nos jours, un grand nombre de techniques et méthodologies ont été proposées afin de résoudre le problème.

La reconnaissance d'objet est une technique de vision par ordinateur permettant d'identifier des objets dans des images ou des vidéos. C'est le résultat des algorithmes d'apprentissage en profondeur et d'apprentissage automatique dont l'objectif est d'enseigner à l'ordinateur à faire ce qui vient naturellement à l'être humain : acquérir un niveau de compréhension de ce que contient une image. Un système de reconnaissance d'objet trouve des objets dans le monde réel à partir d'une image donnée, en utilisant des modèles d'objet connus à priori. Cette tâche est étonnamment difficile. Les humains effectuent la reconnaissance d'objet sans effort et instantanément. La description algorithmique de cette tâche pour la mise en œuvre sur des machines a été très difficile. Il existe différentes étapes et techniques utilisées pour la reconnaissance d'objet dans de nombreuses applications. Un système de vision peut être amené à exécuter différents types de tâches de reconnaissance complexes en utilisant de multiples approches lors des différentes phases de la tâche de reconnaissance.

Le problème de la reconnaissance d'objets peut être défini comme un problème d'étiquetage basé sur des modèles d'objets connus. Formellement, à partir d'une image contenant un ou plusieurs objets d'intérêt (et un arrière-plan) et un ensemble d'étiquettes correspondant à un ensemble de modèles connus du système, le système doit attribuer des étiquettes correctes aux régions ou à un ensemble de régions de l'image. Le problème de reconnaissance d'objet est étroitement lié au problème de segmentation. Car sans une reconnaissance au moins partielle des objets, la segmentation est impossible et, sans segmentation, la reconnaissance d'objet n'est pas possible.

La détection d'objet et la reconnaissance d'objet sont des techniques similaires d'identification des objets, mais leur exécution varie. La détection d'objet est le processus de recherche d'instances d'objets dans des images. Dans le cas de l'apprentissage en profondeur, la détection d'objet est un sous-ensemble de la reconnaissance d'objet, dans lequel l'objet est non seulement identifié, mais également situé dans une image. Cela permet à plusieurs objets d'être identifiés et situés dans la même image.

Nous nous sommes intéressés dans le cadre de cette thèse à l'aspect détection d'objets dans les images. Il existe plusieurs approches de détection d'objets,

- Approches basées sur des modèles d'objet de type 3D.
- Méthodes basées sur les points caractéristiques de l'objet.
- Méthodes basées sur l'apparence et les contours des objets.

Dans note cas et dans le cadre de ce travail, nous avons pris en compte deux approches parmi celles citées précédemment dans le domaine de la vision par ordinateur à savoir, celles basée sur les points caractéristiques des objets dans la première contribution de notre thèse. La

deuxième contribution a été consacrée à l'approche basée sur l'apparence et les contours des objets. Ainsi, nous présenterons dans ce chapitre l'état de l'art des différentes techniques utilisées dans le domaine de la description et la correspondance des points caractéristiques entre les images. Ainsi que les différents détecteurs de bords cités dans la littérature.

I.3. Etat de l'art des détecteurs et descripteurs de caractéristiques

Une zone d'intérêt est une zone « intéressante » d'une image, et peut être utilisée comme point de départ de nombreux algorithmes de traitement d'images. De ce fait, la qualité de l'algorithme utilisé pour détecter et décrire les zones d'intérêt conditionne souvent la qualité du résultat finale de la chaîne de traitement que l'on souhaite appliquer à une image.

Nous allons présenter dans cette partie du manuscrit l'état de l'art des principaux travaux retenues dans la littérature pour le domaine de la détection et la description des points caractéristiques dans les images.

I.3.1 Etat de l'art des détecteurs de points caractéristiques

Beaucoup de détecteurs (ou opérateurs) ont été proposés dans la littérature, leur rôle étant d'extraire les caractéristiques saillantes d'une image donnée.

Fig. I.7. Exemple montrant le résultat d'un détecteur de caractéristiques.

Une caractéristique est sélectionnée si d'une part, elle se distingue de son entourage immédiat et de l'autre, elle se reproduit dans les images de correspondance d'une manière similaire. En vision par ordinateur et en traitement d'images, la **détection de zones d'intérêt** d'une image numérique (*feature detection*) consiste à mettre en évidence des zones de cette image jugées « intéressantes » pour l'analyse, c'est-à-dire présentant des propriétés locales remarquables. De telles zones peuvent apparaître, selon la méthode utilisée, sous la forme de points, de courbes continues, ou encore de régions connexes rectangulaires, ceci constitue le résultat de la détection.

Les opérateurs d'intérêt fournissent une ou plusieurs caractéristiques, qui peuvent être utilisées au cours de la correspondance des images. La signification des caractéristiques extraites dépend cependant du contexte. Ainsi, les points d'intérêt ne correspondent pas nécessairement aux coins physiques de la scène.

La *répétabilité* ou le fait que les *mêmes* zones d'intérêt (ou à peu près) puissent être détectées sur deux images (numériquement) différentes mais représentant la même scène est aussi une propriété importante et généralement exigée pour tous les algorithmes de détection de zones d'intérêt.

Il est nécessaire au départ de définir les exigences de sélection pour un opérateur d'intérêt. Les critères pour un appariement distinctif des caractéristiques sont basés sur cinq principales caractéristiques :

- *Distinction* : un point d'intérêt devrait ressortir clairement du contexte et être unique dans son voisinage.
- *Invariance* : la détermination devrait être indépendante des distorsions géométriques et radiométriques.
- *Stabilité* : La sélection des points d'intérêt devrait être robuste au bruit et aux distorsions.
- *Unicité* : en dehors de la spécificité locale, un point d'intérêt se doit également de posséder une unicité globale, afin d'améliorer la distinction des modèles répétitifs.
- *Interprétabilité* : les points d'intérêt devraient avoir un sens important, de sorte qu'ils peuvent être utilisés pour la correspondance l'analyse et l'interprétation d'une image.

Ces propriétés rendent les points d'intérêt très efficaces dans le contexte de la correspondance d'image et l'analyse temporel de séquences d'images basée sur les caractéristiques.

Ainsi, un bon détecteur présente la propriété d'invariance des points détectées à différents types de changements tels que l'illumination, la translation, la rotation, l'échelle et les transformations affines (e.g. les régions détectées dans l'image d'origine et l'image test dans le cas d'un changement de point vue sont les mêmes). La Fig.I.8 montre les différents types de changements cités ultérieurement.

Fig. I.8. Les différents changements pouvant affecter l'image d'origine.

Les changements de type 2D possibles d'une image à une autre peuvent être résumé en :

- Changements affines représenté par :
  - Changement d'échelle.
  - Changement de translation.
  - Changement de rotation et cisaillement.
- Changements projectifs représenté par :
  - Changement de projection
  - Homographies.

A. Changements affines

Une transformation affine est généralement représentée par une matrice qui détermine le type de transformation à partir d'un point P à un point P' tel que :

$$\begin{matrix} P' & & & P \\ \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} & = & \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \end{matrix} \qquad (I.12)$$

Les éléments a, b, c, d, e et f changent suivant le type de transformation.

Ainsi, nous aurons :

- Pour une translation

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad (I.13)$$

tx, ty représentent la translation dans les directions x et y.

- Pour une rotation

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & tx \\ sin\theta & cos\theta & ty \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad (I.14)$$

θ est l'angle de rotation.

- Pour un changement d'échelle

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad (I.15)$$

Sx, Sy représente la nouvelle echelle sur x et y.

- Pour un cisaillement

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & Shx & 0 \\ Shy & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad (I.16)$$

Shx, Shy sont les cisaillements sur les axes x te y.

Il est tout à fait possible de concaténer plusieurs matrices de transformations tels que :

$$A = R(rotation)S(échelle)Sh(cisaillement)\ T(translation)$$

B. Changements Projectifs

Tel que la transformation affine, la transformation projective est aussi représentée par une matrice qui détermine le type de transformation à partir d'un point P à un point P' mais avec une différence au niveau de la dernière ligne de la matrice tel que :

$$\overset{P'}{\begin{pmatrix} su \\ sv \\ 1 \end{pmatrix}} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \overset{P}{\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}} \qquad (I.17)$$

La transformation projective peut effectuer une transformation quadrilatères générale entre la source et la destination. Elle ne conserve pas les lignes ou les longueurs parallèles. Enfin, elle ne conserve pas les points équidistants. Cette transformation est particulièrement utilisée dans l'infographie.

Les premières méthodes proposées historiquement se fondent sur l'analyse des contours des arêtes, c'est-à-dire des zones où la luminance (ou la couleur) de l'image change brusquement. En d'autres termes, comporte une discontinuité.

Un des premiers opérateurs d'intérêt a été développé par Moravec [51]. Depuis lors, diverses publications sont apparues sur ce sujet. Un aperçu complet des méthodes pour l'extraction d'éléments ponctuels peuvent être trouvées par exemple dans Schmid [52], ou les approches existantes sont basées sur trois critères :

- Intensité : La présence d'une caractéristique saillante est directement définie par un changement de la valeur d'intensité.
- Contour : Ces méthodes sont basées sur l'extraction d'un contour ayant des parties de courbure maximale ou bien à partir de méthodes d'approximation de contours dans le but de détecter les intersections.
- Un modèle : Une bonne ajustation des modèles paramétriques d'intensité, permettent d'atteindre la localisation des motifs d'image avec une précision sous-pixel.

Les algorithmes de détection de points d'intérêt se focalisent en général sur des points particuliers des contours, sélectionnés selon un critère précis.

Ainsi, les coins (*corners*) sont les points de l'image où le contour (de dimension 1) change brutalement de direction, comme aux quatre sommets d'un rectangle. Il s'agit de points particulièrement stables, et donc intéressants pour la répétabilité

Un certain nombre d'expériences ont été réalisées pour évaluer les détecteurs de points d'intérêt. Schmid [52] a accompli une comparaison pratique des opérateurs d'intérêt en utilisant les implémentations originales des auteurs. Les opérateurs de Förstner [53], Cottier, Heitger, Horaud ainsi que Harris [54] ont été évalués quantitativement. Il s'est avéré que l'opérateur Harris était le plus stable de tous.

Hall et al., 2002 ont formalisé une définition de saillance sous des changements d'échelle et ont évalué la résistance des détecteurs Harris, Lindeberg [55] et Harris laplacien sur ces derniers.

Sojka [56] a utilisé les estimations Bayésien pour mesurer la probabilité qu'une zone d'image contient un point d'intérêt.

En 2004, Johansson & Söderberg [57] ont prouvé que la méthode du motif de l'étoile et le tenseur du 4ème ordre fonctionnent mieux que le détecteur Harris. Köthe [58] avait amélioré la structure de calcul du tenseur en utilisant une résolution accrue et un moyennage non linéaire pour optimiser la précision de localisation.

Mikolajczyk & Schmid, 2005 ont combiné le détecteur Harris avec une sélection d'échelle Laplacienne et étendu pour faire face aux transformations affines.

Lowe [3] a décrit la génération de caractéristiques d'image avec une transformée de caractéristique invariante à l'échelle (SIFT). Mikolajczyk & Schmid [59] ont effectué une étude comparatif assez complète autour du descripteur SIFT par rapport aux descripteurs utilisant les filtres orientables ainsi que ceux basés sur les moments invariants et différentiels. Ils l'ont également comparé aux descripteurs basé sur les filtres complexes et ceux à corrélation croisée, pour différents types de points d'intérêt. Ils ont observé que le descripteur SIFT a été le plus efficace suivi des filtres orientables. SUSAN [60], Deriche-Giraudon [61], Beaudet, Noble et Kitchen-Rosenfeld. Zuliani [62] ont proposé une description unificatrice et mathématique de comparaison entre les détecteurs de points Harris, Noble, Kanade-Lucas-Tomasi et Kenney. Cependant, la sélection d'une procédure optimale reste difficile, car les résultats dépendent essentiellement des mise en œuvre [63].

Le détecteur SURF [6] est basé sur la matrice Hessian pour trouver les points d'intérêt. Les réponses en ondelettes sont utilisées pour l'affectation de l'orientation, dans les directions horizontale et verticale, en appliquant des poids gaussiens adéquats.

Le détecteur FAST et ses variantes [8] sont les méthodes de choix pour trouver des points-clés dans des systèmes en temps réel qui correspondent à des caractéristiques visuelles, par exemple, le suivi et la cartographie parallèles. Ce dernier est assez efficace pour trouver un nombre raisonnable de points-clés, bien qu'il doit être augmentés de schémas pyramidaux pour un bon ajustement de l'échelle.

Le détecteur utilisé pour le descripteur BRISK par Leutenegger et al. dans [9] est basé sur une détection d'angle adaptative et générique à plusieurs échelles basée sur le test de segment accéléré AGAST [9]. Ce détecteur recherche les maxima è travers l'échelle de l'espace en utilisant le score du détecteur FAST comme mesure de la saillance.

Dans [64], Matas et al. proposèrent un détecteur de régions extrêmement stables (MSER). Tel-que le détecteur SIFT, le MSER extrait un certain nombre de régions covariantes, appelées MSER. Pouvant être de formes elliptiques, elles sont attachées aux MSER en ajustant des ellipses aux régions d'intérêt.

Sachant que le MSER est basé sur le détecteur Harris de coin, nous avons fait le choix de les développer de manière plus détaillé car nous avons utilisé le MSER pour la détection des

points d'intérêt du descripteur proposé. Le choix de ce dernier a été motivé par une étude comparative des détecteurs existants et qui présente le MSER comme ayant les meilleurs résultats de détection des points caractéristiques pour les transformations affines.

I.3.1.1 Détecteur Harris

Harris [54] a proposé une amélioration de l'opérateur classique Moravec, qui a par la suite été utilisé dans la résolution de problème des décalages discrets et des directions. Ceci a également eu pour effet d'augmenter la précision de la localisation des points d'intérêt à l'aide de la fonction d'autocorrélation.

En statistique, l'idée d'autocorrélation est une comparaison de similarité (corrélation) d'une fenêtre d'image décalée légèrement par rapport à l'image originale. La fenêtre est considérée comme contenant une caractéristique significative si la similitude diminue pour chaque décalage de la fenêtre dans n'importe quelle direction. La matrice d'autocorrélation A est calculée par sommation de la première dérivée de la fonction d'image I sur la zone Ω autour de chaque emplacement de l'image

$$A(x,y) = \begin{bmatrix} \sum_{i,j\ e\Omega} I_x(i,j)^2 & \sum_{i,j\ e\Omega} I_x(i,j).I_y(i,j) \\ \sum_{i,j\ e\Omega} I_x(i,j).I_y(i,j) & \sum_{i,j\ e\Omega} I_y(i,j)^2 \end{bmatrix} \qquad (I.18)$$

I_x et I_y sont les dérivées partielles de l'image I, dans les directions x et y, et sont définies comme telles :

$$\begin{cases} I_x = I * G_x \\ I_y = I * G_y \end{cases} \qquad (I.19)$$

G_x et G_y, représentent les dérivées du filtre kernel g,

$$\begin{aligned} G_x(x,y) &= \frac{\partial G_\sigma(x,y)}{\partial x} = -\frac{x}{2\pi\sigma^4} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \\ G_y(x,y) &= \frac{\partial G_\sigma(x,y)}{\partial y} = -\frac{y}{2\pi\sigma^4} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \end{aligned} \qquad (I.20)$$

Dans ce cas, la taille de Ω peut être déterminée automatiquement par l'écart type utilisé σ. La matrice A décrit enfin la structure du voisinage immédiat de chaque pixel et a les caractéristiques suivantes :

- Rank 2: Un rang complet indique un point saillant.
- Rank 1: Une matrice singulière suggère un bord droit.
- Rank 0: La matrice définit une zone homogène.

Le poids du point w à travers la matrice d'autocorrélation A est déterminé avec la fonction de réponse en coin de tels sorte que,

$$w = \det(A) - k.trace(A)^2 \qquad (I.21)$$

Afin d'avoir une séparation entre les points et les bords, le paramètre k est choisi empiriquement entre 0,04 et 0,06. Ceci donne lieu à des valeurs positives dans le cas de points et à des valeurs négatives dans le cas des bords droits. La position d'un point d'intérêt est enfin déterminée par la suppression des points non-maximaux.

I.3.1.2 Détecteur MSER

L'algorithme du détecteur MSER est le suivant :

- Commencer par les points locaux d'intensité extrême.
- Allez dans toute les directions jusqu'au point extrême d'une fonction donnée f. La courbe qui connecte tous les points environnants est la bordure qui délimite la région d'intérêt.
- Calculer les moments géométriques d'ordre 2 pour cette région.
- Remplacer la région par une éclipse.

La détection des points clés est d'abord effectuée en utilisant le détecteur Harris. Les points détectés à plusieurs échelles sont considérés comme étant les extrêmes locaux d'intensité. Cette étape est ensuite suivie par l'exploration de l'image sur des rayons autour de chaque point jusqu'à ce qu'un extremum de la fonction f soit atteint. Ou la fonction f est définie par :

$$f(t) = \frac{|I(t) - I_0|}{\frac{1}{t}\int_0^t |I(t) - I_0| dt} \qquad (I.22)$$

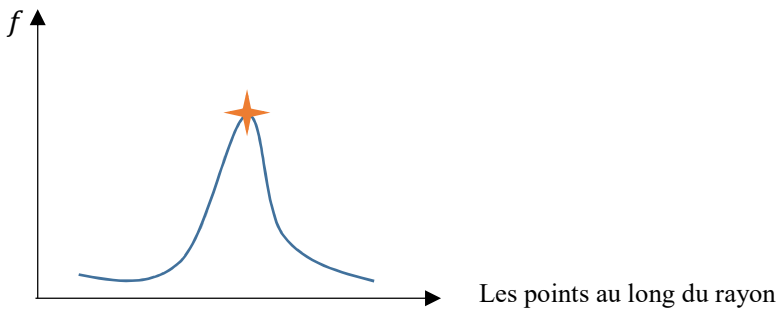


Fig. I.9. illustration de la fonction f.

Ainsi, une forme irrégulière est créée autour de chaque point, sachant que les maximas sont différents sur chaque rayon. Il est considéré que la forme de la transformation affine d'une région donnée correspond à l'originale comme le montre l'image ci-dessous.

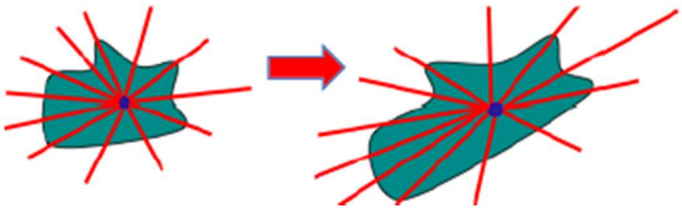


Fig. I.10. Effet de la transformation affine sur une région donnée.

Une approximation de ces régions à des régions elliptiques est alors effectué en utilisant les moments du 2^{éme} ordre, tels que

$$m_{pq} = \int x^p y^q f(x,y) dxdy \qquad (I.23)$$

$x^p y^q$, correspondent au centre de la masse. L'ellipse autour d'une région correspond également à l'ellipse autour de cette dernière après une transformation affine.

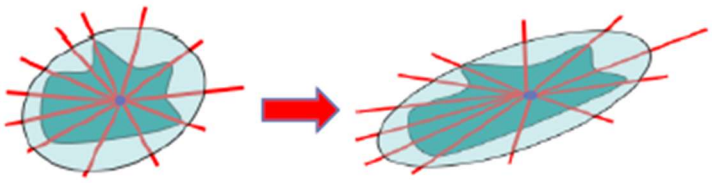


Fig. I.11. L'ellipse autour de la région détectée et son équivalent autour de la région transformé.

MSER est un détecteur de blob, ce dernier extrait de l'image un certain nombre de régions covariant, appelées MSER: un MSER est un composant stable connecté de quelques ensembles de niveaux de gris de l'image. Celui-ci est basé sur l'idée de prendre des régions qui restent presque les mêmes à travers une large gamme des seuils. Ainsi,

- Tous les pixels en dessous d'un seuil donné sont en blanc et tous ceux qui sont égalaux ou au dessus sont en noir.
- Si on nous montre une séquence d'images seuillées I_t, ou le seuil est définit par t, une image noire va donc apparaitre en premier lieu, puis des taches blanches correspondant aux minimums d'intensité apparaîtront puis grossiront.
- Ces points blancs finiront par fusionner jusqu'à ce que toute l'image soit blanche.
- L'ensemble de tous les composants connectés dans la séquence est l'ensemble de toutes les régions extrémales.

En option, les cadres elliptiques sont attachés aux MSER en insérant des ellipses dans les régions. Ces régions sont conservées en tant que caractéristiques pour les descripteurs. Le mot extrémal fait référence à la propriété que tous les pixels à l'intérieur du MSER ont des régions extrémales brillantes ou sombres.

L'extraction du MSER met en œuvre les étapes suivantes :

- Balayer le seuil d'intensité du noir au blanc, en effectuant une simple seuillage de la luminance de l'image.
- Extraire les composants connectés ("Régions Extrêmes")
- Trouver un seuil lorsqu'une région extrême est "Maximally Stable", c'est-à-dire le minimum local de l'élément.
- Approximer une région avec une ellipse (cette étape est facultative).
- Conserver les descripteurs de ces régions comme caractéristiques.

Cependant, même si une région extrême est au maximum stable, elle peut être rejetée si ce dernier est,

- Trop grand (il y a un paramètre MaxArea).

- Trop petit (il y a un paramètre MinArea).
- Trop instable (il y a un paramètre MaxVariation).
- Trop similaire à son parent MSER.

Le seuillage de l'image ce fait par :

- L'application d'une série de seuils - un pour chaque niveau de gris.
- Limitez l'image à chaque niveau pour créer une série d'images en noir et blanc.
- Un extrême sera tout blanc et un autre tout noir. Entre les deux, les taches croissent et fusionnent.

Les régions détectées présentent une importante propriété de résistance à différentes transformations affines. Une autre propriété est la stabilité de ces dernières, car seules les régions qui sont les mêmes pour plusieurs valeurs du seuil sont sélectionnées. Cependant sa résistance au changement de l'illumination reste moindre tel que le changement nuit/jour ou le changement d'ombre.

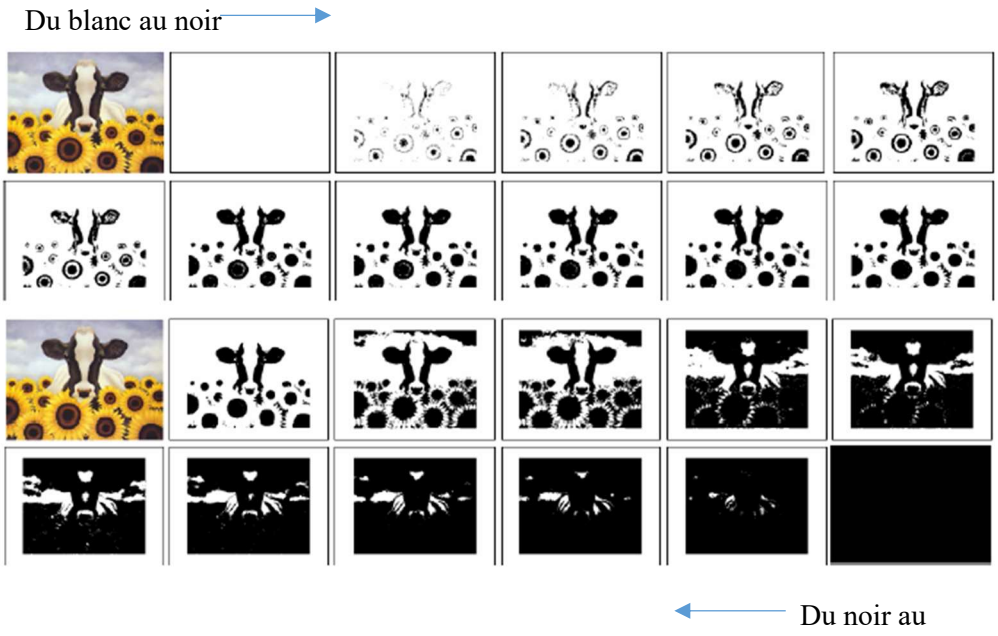


Fig. I.12. L'ellipse autour de la région détectée et son équivalent autour de la région transformé.

I.3.2 Les descripteurs de points caractéristiques

Une fois les points clés localisés, l'étape suivante consiste à les décrire et leurs environnements immédiats de la manière la plus efficace possible.

I.3.2.1 Descripteurs basé sue les histogrammes

Descripteur SIFT

Le scale-invariant feature transform (SIFT), proposé par Lowe [3] en 2004 est un descripteur d'objet local largement utilisé. Il est invariant au changement de l'échelle et l'éclairage. La première étape de l'algorithme est la détection des points d'intérêt, dits *points-*

clés. Un point-clé (x, y, σ) est défini d'une part par ses coordonnées sur l'image (x et y) et d'autre part par son facteur d'échelle caractéristique (σ).

On appelle *gradient* de facteur d'échelle σ (noté L) le résultat de la convolution d'une image I par un filtre gaussien G de parametre σ, soit

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad (I.24)$$

Cette convolution a pour effet de lisser l'image originale I de telle sorte que les détails trop petits, c'est-à-dire de rayon inférieur à σ, sont estompés.

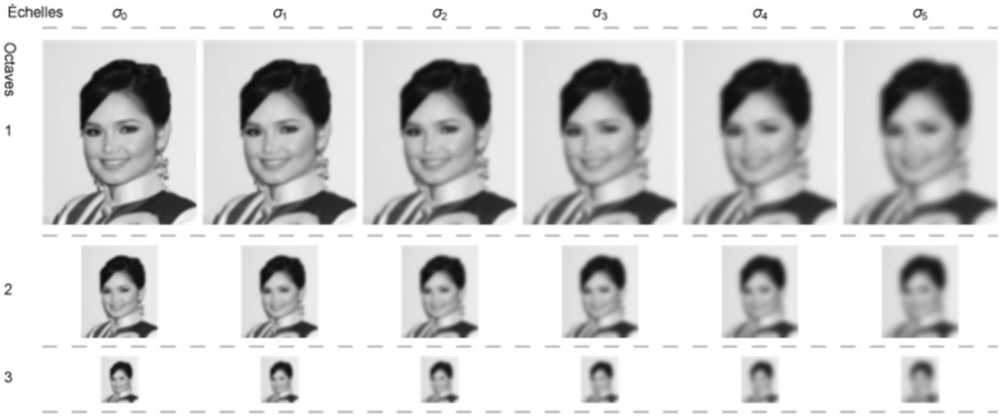


Fig. I.13. Pyramide de gradients : 3 octaves de 6 gradients.

La Figure I.13 représente une pyramide de gradients constitué de trois octaves contenant respectivement la taille de l'image originale, suivie par sa version sous échantillonné par deux et par quatre, avec pour chaque taille six gradients différents en utilisant une valeur croissante de sigma.

Par conséquent, la détection des objets de dimension approximativement égale à σ se fait en étudiant l'image appelée *différences de gaussiennes* (*difference of gaussians*, DoG) définie comme suit :

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \qquad (I.25)$$

k est un paramètre fixe de l'algorithme qui dépend de la finesse de la discrétisation de l'espace des échelles voulue.

Dans cette image ne persistent plus que les objets observables dans des facteurs d'échelle qui varient entre σ et $k\sigma$. De ce fait, un point-clé candidat (x, y, σ) est défini comme un point où un extremum du DoG est atteint par rapport à ses voisins immédiats, c'est-à-dire sur l'ensemble contenant 26 autres points défini par :

$$\left\{D\left(x + \delta_x, y + \delta_y, s\sigma\right),\ \ \delta_x \epsilon \{-1,0,1\}, \delta_y \epsilon \{-1,0,1\}, s\ \epsilon \{k^{-1}, 1, k\}\right\} \qquad (I.26)$$

Tel que l'illustre la Figure I.14, l'utilisation d'une pyramide est préconisée pour optimiser le temps de calcul des images floutées à un grand nombre d'échelles différentes. La base de la pyramide est en général l'image originale et un niveau donné – on parle d'*octave* par analogie avec la musique – est obtenu à partir du précédent en divisant la résolution de l'image par 2, ce

qui revient à doubler le facteur d'échelle. Au sein d'une même octave, le nombre de convoluées à calculer est constant.

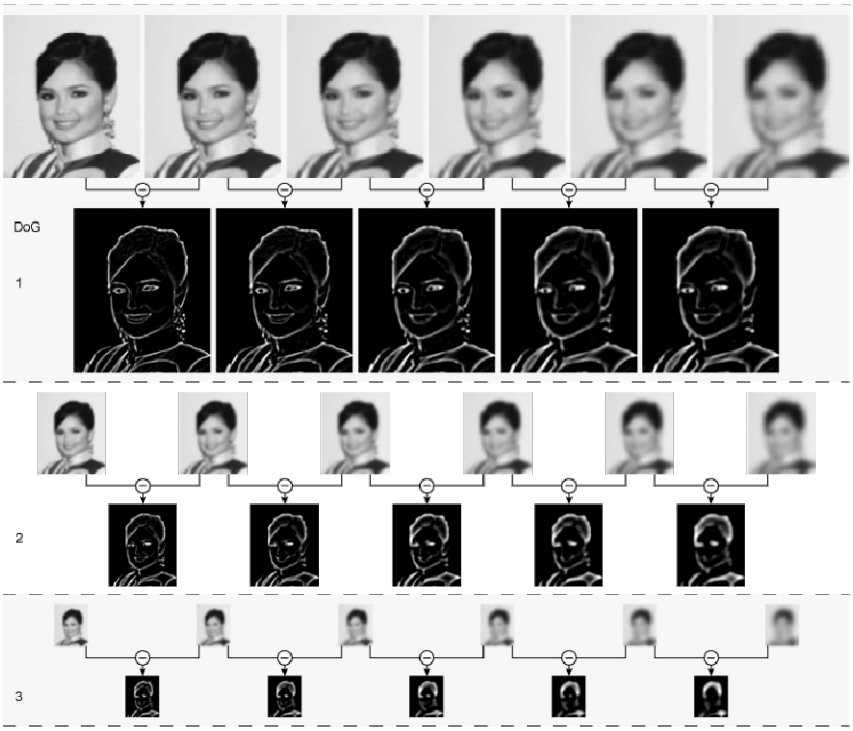


Fig. I.14. Construction de la pyramide de différence de gaussiens (DoG) à partir de la pyramide de gradients.

Le facteur fixe k dans les formules ci-dessus est calculé pour qu'au final, l'espace discrétisé des facteurs d'échelles considérés corresponde à une progression géométrique $\{\sigma_0, k\sigma_0, k^2\sigma_0, \dots\}$, avec à chaque changement d'octave une valeur $k^p\sigma_0$ qui devient égale à une quantité de la forme $2^t\sigma_0$. Ce détail, la progression géométrique des facteurs d'échelle, est important pour que les valeurs des DoG à différentes échelles soient comparables entre elles et évite d'avoir à utiliser un facteur de normalisation dans leur calcul.

L'étape de détection des points-clés candidats décrite ci-dessus est une variante de l'une des méthodes de *blob detection* (détection de zones) développée par Lindeberg qui utilise le laplacien normalisé par le facteur d'échelle au lieu des DoG. Ces derniers peuvent être considérés comme une approximation des laplaciens et présentent l'avantage d'autoriser l'utilisation d'une technique pyramidale. La Figure I.15 montre un exemple de cette procédure.

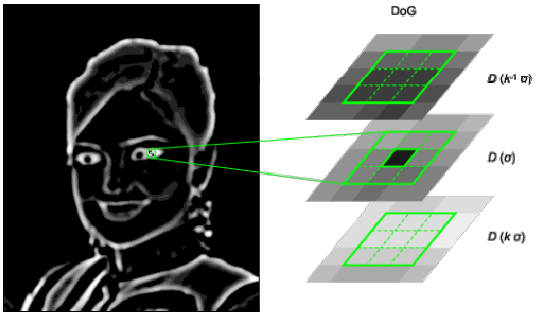


Fig. I.15. Exemple de détection d'extremums dans l'espace des échelles.

Il s'ensuit une étape de reconvergence et de filtrage qui permet d'améliorer la précision sur la localisation des points-clés et d'en éliminer un certain nombre jugés non pertinents. Chaque point-clé restant est ensuite associé à une orientation intrinsèque, c'est-à-dire ne dépendant que du contenu local de l'image autour du point clé, au facteur d'échelle considéré. Elle permet d'assurer l'invariance de la méthode à la rotation et est utilisée comme référence dans le calcul du descripteur, qui constitue la dernière étape de ce processus.

L'étape d'assignation d'orientation consiste à attribuer à chaque point-clé une ou plusieurs orientations déterminées localement sur l'image à partir de la direction des gradients dans un voisinage autour du point. Dans la mesure où les descripteurs sont calculés relativement à ces orientations, cette étape est essentielle pour garantir l'invariance de ceux-ci à la rotation : les mêmes descripteurs doivent pouvoir être obtenus à partir d'une même image, quelle qu'en soit l'orientation.

Pour un point-clé donné (x_0, y_0, σ_0) le calcul s'effectue sur $L\,(x_0, y_0, \sigma_0)$, à savoir le gradient de la pyramide dont le paramètre est le plus proche du facteur d'échelle du point. De cette façon, le calcul est également invariant à l'échelle. À chaque position dans un voisinage du point-clé, on estime le gradient par différences finies symétriques, puis son amplitude (c'est-à-dire sa norme) $m\,(x, y)$ et son orientation $\theta\,(x, y)$:

$$m\,(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}$$
$$\theta\,(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \qquad (I.27)$$

Un histogramme des orientations sur le voisinage est réalisé avec 36 intervalles, couvrant chacun 10 degrés d'angle. L'histogramme est doublement pondéré : d'une part, par une fenêtre circulaire gaussienne de paramètre égal à 1,5 fois le facteur d'échelle du point-clé σ_0; d'autre part, par l'amplitude de chaque point. Les pics dans cet histogramme correspondent aux orientations dominantes. Tel que le montre la Figure I.16.

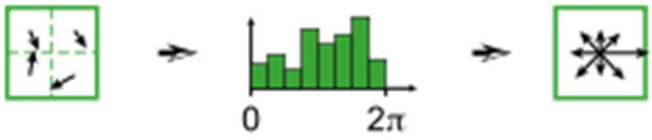


Fig. I.16. Construction de l'histogramme des orientations.

Toutes les orientations dominantes permettant d'atteindre au moins 80 % de la valeur maximale sont prises en considération, ce qui provoque si nécessaire la création de points-clés supplémentaires ne différant que par leur orientation principale.

À l'issue de cette étape, un point-clé est donc défini par quatre paramètres (x, y, σ, θ). Il est à noter qu'il est parfaitement possible qu'il y ait sur une même image plusieurs points-clés qui ne diffèrent que par un seul de ces quatre paramètres (le facteur d'échelle ou l'orientation, par exemple). Une fois les points-clés détectées et leur invariance aux changements d'échelles et aux rotations assurées, arrive l'étape de calcul des vecteurs descripteurs. À cette occasion, des traitements supplémentaires vont permettre d'augmenter le pouvoir discriminant en rendant les descripteurs invariants à d'autres transformations telles que la luminosité, le changement de

point de vue 3D, etc. Cette étape est réalisée sur l'image lissée avec le paramètre de facteur d'échelle le plus proche de celui du point-clé considéré.

Autour de ce point, on commence par modifier le système de coordonnées local pour garantir l'invariance à la rotation, en utilisant une rotation d'angle égal à l'orientation du point-clé, mais de sens opposé. On considère ensuite, toujours autour du point-clé, une région de 16 × 16 pixels, subdivisée en 4 × 4 zones de 4 × 4 pixels chacune. Sur chaque zone est calculé un histogramme des orientations comportant 8 intervalles. En chaque point de la zone, l'orientation et l'amplitude du gradient sont calculés comme précédemment. L'orientation détermine l'intervalle à incrémenter dans l'histogramme, ce qui se fait avec une double pondération – par l'amplitude et par une fenêtre gaussienne centrée sur le point clé, de paramètre égal à 1,5 fois le facteur d'échelle du point-clé.

Ensuite, tel que le montre la Figure I.17, les 16 histogrammes à 8 intervalles chacun sont concaténés et normalisés. Dans le but de diminuer la sensibilité du descripteur aux changements de luminosité, les valeurs sont plafonnées à 0,2 et l'histogramme est de nouveau normalisé, pour finalement fournir le descripteur SIFT du point-clé, de dimension 128.

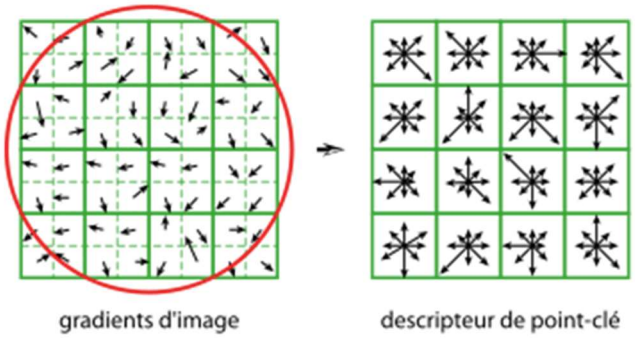


Fig I.17. Construction d'un descripteur SIFT.

Cette dimension peut paraître bien élevée, mais la plupart des descripteurs de dimension inférieure proposés dans la littérature présentent de moins bonnes performances dans les tâches de mise en correspondance pour un gain en coût de calculs bien modéré, en particulier quand la technique BBF (*Best-Bin-First*) est utilisée pour trouver le plus proche voisin. Par ailleurs, des descripteurs de plus grande dimension permettraient probablement d'améliorer les résultats, mais les gains escomptés seraient dans les faits assez limités, alors qu'à l'inverse augmenterait sensiblement le risque de sensibilité à la distorsion ou à l'occultation. La précision de recherche de correspondance de points dépasse 50 % dans les cas de changement de point de vue supérieur à 50 degrés, ce qui permet d'affirmer que les descripteurs SIFT sont invariants aux transformations affines modérées. Le pouvoir discriminant des descripteurs SIFT a pour sa part été évalué sur différentes tailles de bases de données de points-clés ; il en ressort que la précision de mise en correspondance est très marginalement impactée par l'augmentation de la taille de la base de données, ce qui constitue une bonne confirmation du pouvoir discriminant des descripteurs SIFT.

Descripteur SURF

Le Speeded Up Robust Features (SURF) que l'on peut traduire par *caractéristiques robustes accélérées*, proposé par Bay et al. [6] est un algorithme de détection de caractéristique et un descripteur. SURF est partiellement inspiré par le descripteur SIFT, qu'il surpasse en rapidité et, selon ses auteurs, plus robuste pour différentes transformations d'images. Dans cet algorithme, au lieu d'utiliser l'espace d'échelle de l'image à travers le filtrage gaussien, les auteurs configurent un espace pyramidal en changeant la taille d'un filtre passe-bas. SURF est fondé sur des sommes de réponses d'ondelettes de Haar 2D et utilise efficacement les images intégrales.

En utilisant les intégrales d'images et un changement de taille du filtre au lieu du changement de la taille d'image, le nombre de calculs et le temps de traitement est réduit par rapport à l'algorithme SIFT original. La pyramide des échelles du SURF est construite en appliquant un filtre de taille variante, suivi par la détection des point clés grâce à une suppression des non maximas.

Chaque point clé détecté en utilisant le détecteur de Harris est entouré d'un cercle dont le diamètre correspond à la valeur de l'échelle. L'orientation dominante est extraite à travers les gradients ponctuels dans des tranches de 60◦. Une fois l'orientation dominante détecté, la région d'intérêt est orienté jusqu'à cette direction.

Tel que l'illustre la Figure I.18, dans l'étape de description du SURF, la somme de l'amplitude du gradient ainsi que la valeur absolue de ce dernier est calculé dans les directions x et y dans chaque région constituant les éléments du vecteur caractéristique.

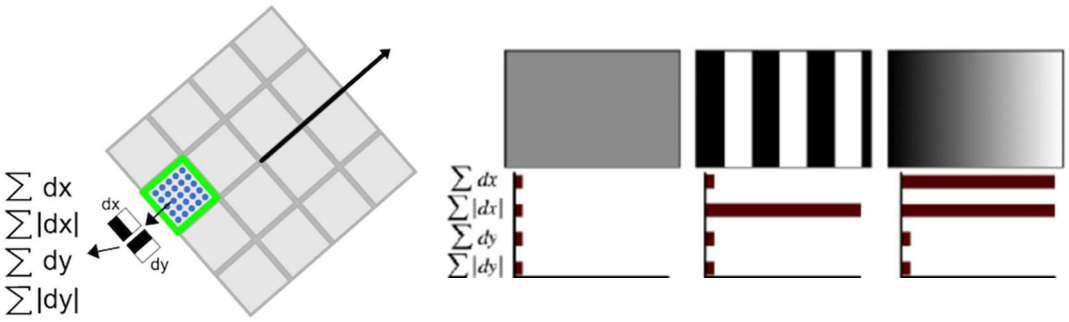


Fig I.18. Exemple de constitution du descripteur SURF.

Pour la description des caractéristiques, SURF utilise des réponses en ondelettes dans le sens horizontal et vertical (encore une fois, l'utilisation d'images intégrales rend les choses plus faciles). Un voisinage de taille $20s$ x $20s$ est pris autour du point clé où s est de l'échèle su point détecté. Il est divisé en sous-régions 4x4. Pour chaque sous-région, des réponses d'ondelettes horizontales et verticales sont prises et un vecteur est formé comme ceci, $v = (\sum dx, \sum dy, \sum|dx|, \sum|dy|)$. Lorsque le descripteur de caractéristiques SURF est représenté sous la forme d'un vecteur, ce dernier à une dimension total de 64 éléments. Pour un meilleur pouvoir discriminant, le descripteur d'entité SURF a une version étendue de 128 dimensions. Dans ce cas, les sommes de dx et $|dx|$ sont calculées séparément pour $dy < 0$ et $dy \geq 0$.

De même, les sommes de dy et $|dy|$ sont divisées selon le signe de dx, doublant ainsi le nombre de fonctionnalités. Cela n'ajoute pas beaucoup de complexité au calcul.

Une autre amélioration importante est l'utilisation du signe de Laplacien (trace de la matrice de Hesse) pour le point d'intérêt sous-jacent. Il n'ajoute aucun coût de calcul puisqu'il est déjà calculé lors de la détection. Le signe du Laplacien distingue les taches brillantes sur les fonds sombres de la situation inverse. Ainsi, dans la phase d'appariement, les caractéristiques ne sont comparées que si elles ont le même type de contraste. Cette information minimale permet une correspondance plus rapide, sans réduire les performances du descripteur.

Dans [52/65], les auteurs ont présenté un descripteur qui est calculé à partir d'un histogramme directionnel local autour d'un point clé détecté. Dans cette méthode, nommé histogramme de gradient local orienté pour Gradient Location and Orientation Histogram, **(**GLOH). L'hypothèse initiale est que le point clé est détecté et livré au descripteur avec une échelle, une orientation et un emplacement, le GLOH applique alors un motif circulaire autour du point clé. Dans cet algorithme, l'histogramme directionnel représente la densité de gradient de chaque pixel pour 17 régions autour de chaque point clé séparément, suivi du remplissage de cet histogramme, où chaque case est un élément du vecteur de caractéristiques. Le GLOH surpasse la version en composantes principales du descripteur SIFT (PCA-SIFT) et plusieurs autres algorithmes dans la correspondance d'image. Il est évident que comme SIFT et SURF, le GLOH constitue les vecteurs de caractéristiques en se basant sur le gradient local des points clés.

Dans la méthode présentée dans [66], Kang et al. conçurent le détecteur et descripteur MDGHM (modified discrete Gaussian–Hermite moment), qui peut représenter les caractéristiques plus abondamment que le détecteur de Hesse et le descripteur local SURF. Dans MDGHM-SURF, la qualité de correspondance des images est considérablement améliorée par rapport au SURF traditionnel, en particulier pour les scénarios dans lequel la transformation de l'image est liée à l'échelle, la rotation, le point de vue, la compression JPEG et les changements d'éclairage. Cependant, il convient de noter que la mise en œuvre de cette transformation spéciale induit une surconsommation en termes du temps de calcul.

I.3.2.2 Descripteurs binaires

Calonder et al. [7] ont présenté une méthode simple et directe pour construire un descripteur binaire dit BRIEF (binary robust indepentent elementary features) dans lequel chaque bit est indépendamment obtenu en comparant les intensités d'une paire de points d'échantillonnage, tel que le montre la FigureI.19.

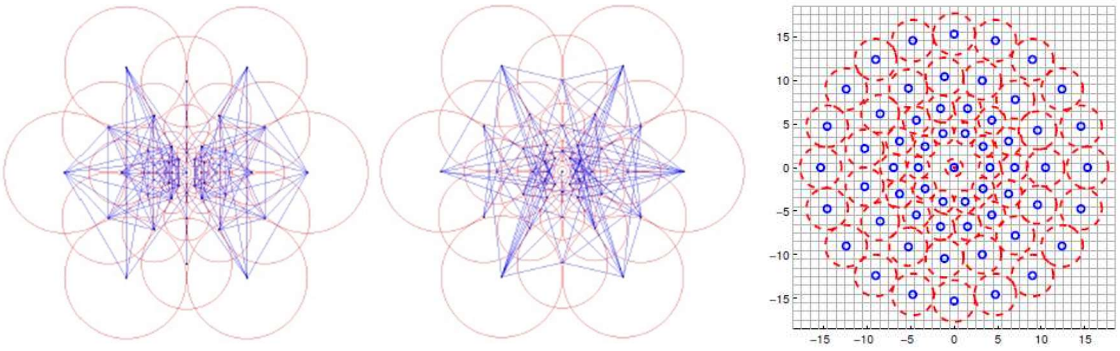


Fig. I.19. Exemple de constitution des descripteurs binaires.

BRIEF a l'inconvénient notable de l'absence d'invariance de rotation. Donc, Rublee et al. [8] ont proposé le descripteur BRIEF orienté rapide et tourné (ORB), qui est invariant aux changements de rotation et robuste au bruit. ORB choisit un bon sous-ensemble de tests binaires par une méthode d'apprentissage qui réduit la corrélation entre les tests, améliorant les performances et l'évolutivité. Leutenegger et al. [9] ont également développé un descripteur binaire appelé BRISK qui est invariant aux transformations d'échelle et de rotation. Il se détourne du modèle d'échantillonnage aléatoire de BRIEF car il utilise un modèle d'échantillonnage symétrique dont chaque point d'échantillon représente un flou gaussien des pixels environnants.

Pour obtenir des performances plus compactes et plus robustes, Alahi et al. [10] ont proposé un descripteur inspiré du système visuel humain, appelé point clé de la rétine rapide (FREAK). Une cascade de chaînes binaires est calculée en comparant efficacement les intensités d'image sur un modèle d'échantillonnage rétinien. L'ensemble des descripteurs binaires sont construits sur un ensemble de comparaisons d'intensité par paires et recours à une orientation de référence pour l'invariance de rotation.

La comparaison par paires d'intensité est très sensible aux erreurs de localisation de points d'échantillonnage. Ceci aurait tendance à fausser certaines informations du descripteur binaire et conduit à la dégradation des performances de correspondance. Les résultats expérimentaux de [8] illustre que ces descripteurs binaires montrent une puissance de description inférieure par rapport au SIFT et GLOH.

I.3.2.3 Descripteurs de formes

Le descripteur de forme est un descripteur affine-invariant conçu pour le détecteur MSER. Un patch local est utilisé pour entourer la région autour de chaque point détectée. L'arrière-plan de ce dernier est ensuite converti en blanc et noir. Similairement à SIFT, la construction du descripteur est réalisée en utilisant l'histogramme du gradient.

Le descripteur LIOP (Local Intensity Order Pattern) présenté dans [12], est invariant aux changements de la rotation et de l'intensité de l'image car l'information locale de chaque pixel est encodée et l'information globale est utilisée pour diviser la région locale en sous-régions. Chaque patch est lissé par un filtre gaussien pour réduire le bruit et les sous–régions sont

décrites par des valeurs désignées. L'histogramme de ces valeurs est utilisé comme descripteur de la sous-région, et le descripteur global est construit en accumulant les histogrammes.

DAISY [11] est conçue pour une vision stéréo à large base. Il est similaire au descripteur SIFT, mais facilite le calcul de l'histogramme de l'orientation de gradient. La description et la correspondance dans DAISY est accélérée par des convolutions multiples au lieu de la somme pondérée des normes de gradient typiques des descripteurs basés sur SIFT.

GSURF [13] est une variante du descripteur SURF basé sur les dérivées du second ordre de calibre multi-échelle. Le descripteur GSURF est rapide à calculer et plus robuste en raison de la propriété d'invariance des dérivés de calibre multi-échelle à la rotation. RFDg [30] est un descripteur binaire basé sur l'apprentissage et est construit par seuillage des réponses des champs réceptifs gaussiens.

Comparaison des performances des descripteurs dit traditionnels

Mikolazcyk et Schmidt [67] ont évalué les performances des descripteurs d'entités locales sous diverses transformations géométriques et photométriques et ont mené l'étude la plus exhaustive jusqu'à présent. Les résultats obtenus ont été résumé dans le tableau I.1 contenant une liste complète des performances des différents détecteurs et descripteurs évalué. Il a été montré que la combinaison du GLOH et SIFT ont surpassé les autres descripteurs en changement de rotation, zoom, flou, compression d'image, point de vue et éclairage.

Auteur	Type	Environnement	Meilleur résultats
Mikolajczyk [68]	Descripteur locale	Transformation géométrique et photométrique	GLOH, SIFT
Miksik [69]	Descripteur locale	Précision et rapidité	LIOP, BRIEF
Kaneva [70]	Descripteur locale	Changement d'illumination et de point de vue	DAISY
Heinly [71]	Descripteur binaire	Transformation géométrique et photométrique	BRIEF
Restrepo [72]	Descripteur de forme	Classification d'objet	FPFH
Moreels [73]	Détecteur + descripteur	Objet 3D	Hessian-affine+SIFT
Gil [74]	Détecteur + descripteur	Visual SLAM	GLOH, SURF
Dahl [75]	Détecteur + descripteur	Base de données Multi-view	MSER+SIFT
Gauglitz [76]	Détecteur + descripteur	Traking visuel	FAST Hessian +SIFT
Mikolajczyk [77]	Détecteur de région affine	Transformation géométrique et photométrique	MSER
Haja [78]	Détecteur de région	Texture+ structure	MSER
Schmid [79]	Détecteur locale	Transformation géométrique et photométrique	Harris
Dickscheid [80]	Détecteur locale	Codage d'image	MSER
Canclini [81]	Détecteur locale	Reconstitution d'image	BRISK

Tableau I.1. Travaux antérieurs sur l'évaluation des performances des détecteurs / descripteurs de caractéristiques [67].

Miksik et Mickolaczyk [69] ont évalués le compromis entre rapidité et précision pour les descripteurs locaux. Ils ont évalué la performance de plusieurs descripteurs binaires et descripteurs locaux. Leurs résultats ont montré que les descripteurs binaires ont surpassé les autres descripteurs pour les applications en temps réel avec des exigences de mémoire faible.

Kaneva et al. [70] ont comparé les performances des descripteurs locaux en termes de changement de points de vue et d'éclairage. Ils ont simulé des scènes synthétiques photoréalistes et conclu que le descripteur DAISY fonctionne mieux sous ces changements. Les performances des descripteurs de forme locaux pour la classification d'objets ont été évaluées par Restrepo et Mundy [72].

Moreels et Perona [73] ont étudié la performance des détecteurs et descripteurs populaires pour les objets 3D. Ils ont généré une base de 144 objets avec changements de point de vue et illumination. Une évaluation de plusieurs combinaisons de détecteurs de caractéristiques et des descripteurs a révélé que la combinaison des détecteur Hessian-Affine et le descripteur SIFT ont surperformé les autres détecteurs et descripteurs pour le changement de point de vue et l'éclairage dans une configuration 3D.

Gil et al. [74] ont comparé le comportement de différents détecteurs et descripteurs pour la localisation et la cartographie visuel simultanées (SLAM). Ils ont évalué la répétabilité des détecteurs ainsi que l'invariance du caractère distinctif des descripteurs. Ainsi, GLOH et SURF étaient les plus appropriés pour le SLAM visuel.

Dahl et al. [75] ont étudié des combinaisons de détecteur de caractéristique et de descripteur pour un ensemble de données multiview. Le MSER et la différence de Gauss (DoG) avec le descripteur SIFT fournit le meilleur résultat dans leur expérience.

Schmid et al. [79] ont évalués la performance de la fonctionnalité de bas niveau, ils ont introduit deux critères d'évaluation : la répétabilité et le contenu de l'information. La répétabilité compare la géométrie et la stabilité des caractéristiques sous différentes transformations, alors que le contenu de l'information mesure le caractère distinctif des caractéristiques. Il a conclu que la version améliorée de Harris a surpassé les autres détecteurs étudiés.

Dickscheid et al. [80] a mesuré l'exhaustivité des fonctionnalités locales pour le codage d'image. Ils ont proposé une mesure pour évaluer l'exhaustivité de la détection de caractéristiques en utilisant la densité d'entropie. Dans leurs expériences, le détecteur MSER a obtenu les meilleures performances.

Canclini et al. [81] ont évalué la performance des détecteurs et des descripteurs pour les applications de récupération d'image. Ils ont comparé plusieurs détecteurs et descripteurs de caractéristiques, concluant que les descripteurs binaires son meilleurs en termes de précision de correspondance et de complexité de calcul.

Les descripteurs de forme locale ont été extraits d'un modèle de la volumétrie probabiliste. Ces chercheurs ont comparé plusieurs descripteurs de forme et classer les catégories d'objets en utilisant le modèle Bag of Words de scènes urbaines à grande échelle. Dans leurs expériences, le descripteur de l'histogramme (FPFH) [82] a montré de bonnes performances.

I.3.2.4 Descripteurs basés sur l'apprentissage

De nombreux chercheurs ont tenté de remplacer les descripteurs traditionnels, les premières tentatives ont été proposées à partir d'architectures simples [83], [84] et

d’optimisation convexe [85]. Les descripteurs récents ont tendance à extraire des entités directement à partir de patchs d’images brutes avec des CNN formés sur de gros volumes de données. MatchNet [86] a formé un réseau CNN Siamese pour la représentation des points caractéristiques, suivi par un réseau entièrement connecté pour apprendre la métrique de comparaison. DeepCompare [87] a montré que sa performance pouvait être augmentée en concentrant le réseau sur le centre de l'image. DeepDesc [88] applique les Siam Networks et les métriques de distance non linéaires apprises par DeepCompare [89] pour l'appariement. Philipp-Net [90] qui apprend en adaptant les pseudo-classes. Une série de travaux récents a examiné des architectures de modèles plus avancées et des formulations d'apprentissage métriques profondes basées sur des triplets, y compris UCN [91], TFeat [92], GLoss [93], L2Net [94] et GOR [95].

I.4. Etat de l’art des détecteurs de bords

Un contour, tel que l’illustre l’image I.16 est une caractéristique structurelle importante pour la reconnaissance de cible basée sur la forme dans les images. La recherche précise de contours dans des images naturelles est un problème qui a fait l'objet de nombreuses études. Un nombre important de contributions ont été faites dans ce domaine au cours des dernières années.

Fig. I.20. Exemple de détection des bords dans une photographie

I.4.1 Détecteurs de bords classiques

Nombre de travaux detecteurs ont été effectué tels que ceux proposés par Robert [28], Sobel [31], le passage à zéro [29]. Le détecteur Canny [32] a été largement adopté, ce dernier est essentiellement basé sur le canal de niveaux de gris et de brillance. De manière similaire, la détection de bord basée sur les ondelettes a été proposée par Mallat et al. [96], ils ont prouvé que les maxima du module de la transformée en ondelettes peuvent détecter les segmentations des structures irrégulières. Dans [97], une meilleure description a été obtenue en considérant la réponse de l'image à une variété de filtres d'échelles et d'orientations différentes. Le descripteur SIFT [3], les contextes de forme [98] et l'histogramme du descripteur de l’orientation du gradient (HOG) [65] sont des détecteurs basés sur l’orientation des gradients et des entités de bord. Laptev [99] a également calculé des histogrammes de gradient en utilisant des images intégrales, résultant en des détecteurs d'objets efficaces.

I.4.2 Détecteurs basées sur le filtrage des images

Les approches les plus récentes sont évaluées à l'aide d'une comparaison entre les bords obtenues à partir ensembles de données d'images naturelles et leurs correspondants créé par l'homme, en tenant compte des informations sur les couleurs et les textures. Dans [100], Martin et al. ont défini et utilisé un opérateur de gradient pour les canaux de luminosité, de couleur et de texture en tant qu'entrée d'un classificateur de régression logistique pour prédire la force du bord. Plutôt que de compter sur de telles fonctionnalités, le Pb [36] utilise plusieurs canaux calculés par combinaison non linéaire des réponses d'une banque de filtres passe-bande, les auteurs ont combiné un ensemble de dégradés, utilisant luminosité, couleur et texture, pour surpasser le détecteur de bord de Canny sur la base de Berkeley Benchmark (BSDS).

Les méthodes spectrales formulent le problème de détection de contours comme un problème de valeur propre. Dans le travail gPb de [37], Arbelaez et. al. ont calculé les gradients en se basant sur les vecteurs propres du graphe d'affinité et les ont combiné avec des indices locaux.

I.4.3 Détecteurs basées sur l'apprentissage

Les méthodes d'apprentissage qui s'appuient sur les caractéristiques du design humain sont également exploitées. Dans [33] Dollar et al. ont utilisé des fonctions de canal intégral pour former un détecteur de bord par pixel, ils ont lancé la détection des bords en tant que problème de classification binaire et ont utilisé des arêtes étiquetées pour former un classificateur de bord de correctif binaire. Un grand nombre de canaux ont été utilisés, y compris des gradients à différentes échelles, des réponses de filtre de Gabor, des filtres gaussiens de différence de décalage, etc. Dans [34], les auteurs ont combinés des indices de bas, moyen et haut niveau, ce qui se traduit par des performances améliorées pour la détection de bord d'objet spécifique. Lim et al. [39] proposent une approche de détection des contours qui classe les correctifs de bord en jetons d'esquisse à l'aide de classificateurs de forêt aléatoires qui tentent de capturer la structure de contour locale.

Une autre approche qui a réussi est celle basée sur l'apprentissage SVM supervisé, tel que les bords structurés (SE) [40] et les gradients de code épars (SCG) [101], qui sont également considérés comme des méthodes discriminantes de détection des limites. Dans [102], les auteurs utilisaient les mêmes caractéristiques et classificateurs que les détecteurs SE et Sketch Tokens [39]. Comme pour le SCG, ils entraînaient le bord basé sur le séparateur de frontière, mais aussi la distance du centre du paramètre caractéristique qui a eu pour effet d'améliorer les performances de détection.

Avec l'apparition récente et l'amélioration des Réseaux de Neurones Convolutionnels (CNN), des méthodes basées sur un apprentissage des caractéristiques hiérarchiques ont fait leur apparition, comme DeepNet [103] ou, DeepEdge de Bertasius et al. [104], et CSCNN [105].

Une détection efficace des limites est une étape cruciale puisqu'elle constitue la base de la plupart des applications d'analyse d'images. Le choix du détecteur peut varier avec le contexte de l'application, comme dans [106], où les auteurs ont proposé une méthode de segmentation

de l'iris pour les systèmes de reconnaissance biométrique basés sur un détecteur de limite d'ondelette. Dans [107], le besoin de systèmes de détection sensibles à la catégorie influence le choix des auteurs pour les détecteurs basés sur l'apprentissage. Dans le cas de [108], où les auteurs ont utilisé un détecteur à base de patch pour la localisation de petites cibles dans des images IR. Cela dit, dans tous les cas, l'étape de détection des limites est essentielle et a une influence sur le résultat final de l'application.

Même si ces détecteurs présentent des résultats de qualité, ils restent coûteux en termes d'implémentation et de consommation de mémoire. Que ce soit des approches dites de l'état de l'art, où la plupart des travaux présentés sont basés sur l'utilisation de plusieurs canaux d'image avec plusieurs phases de transformations et de filtrage. Les méthodes basées sur l'apprentissage, où en plus de leur complexité de calcul et la nécessité d'une phase d'entraînement, restent dépendantes de la disponibilité des ensembles de données étiquetés avec des contours marqués par l'homme.

I.5. Conclusion

Ainsi, dans ce chapitre nous avons présenté un préambule de base sur le traitement des images, suivie par l'historique de la détection des caractéristiques avec une présentation des deux détecteurs les plus utilisé dans la littérature. Une présentation de l'état de l'art des diffèrent descripteurs suivie d'une étude comparative des performances de ces derniers avec l'association des détecteurs a été proposé. Enfin, nous avons présenté un état de l'art des principaux détecteurs de bords cité dans la littérature.

Chapitre II

Présentation du descripteur de points caractéristiques proposé

II.1. Introduction

Notre motivation était de trouver une autre représentation des patchs sélectionnés afin de les comparer sans aucun processus d'estimation d'orientation. La préservation de la simplicité du schéma proposé est également une priorité et une contrainte que nous nous sommes imposé. Dans cette optique, nous avons utilisé deux histogrammes bidimensionnels contenant les intensités et les gradients entourant le point caractéristique.

Nous avons choisi d'utiliser des histogrammes bidimensionnels afin de capturer l'intensité et le changement d'orientation du gradient autour des bords du patch. En fait, seule la position des bords du patch change dans le cas d'une transformation d'orientation. Considérant la magnitude du gradient comme le meilleur moyen de localiser les bords de l'image, nous l'avons utilisé à la fois dans les histogrammes d'intensité et d'orientation du gradient.

Si l'on considère le cas du SIFT, le calcul de l'orientation est réalisé à l'aide d'un histogramme contenant les directions des gradients de tous les pixels au voisinage immédiat d'un point caractéristique et ceux après le filtrage (gaussien) et l'ajustement de l'échelle de ces derniers. La direction du patch est déterminée par la valeur de l'orientation ayant reçu le plus grand nombre de votes dans l'histogramme. Ceci dit, dans le cas de deux directions dominantes, le patch en question ce verras assigner deux orientations en plus des coordonnées de position x et y et son échelle de détection.

Dans le cas du SURF, les auteurs calculent les réponses des ondelettes de Haar dans les directions x et y des voisins pondérés avec un filtre gaussien du point clé dans un radius de $6s$, et s est l'échelle de détection de ce dernier. L'orientation dominante est estimée par le calcul de la somme de toutes les réponses dans une fenêtre d'orientation coulissante avec un angle de $\frac{\pi}{3}$. Les réponses horizontales et verticales dans la fenêtre sont sommées. Les deux réponses additionnées donnent alors un nouveau vecteur. Le plus long de ces vecteurs prête son orientation au point d'intérêt. La taille de la fenêtre coulissante est un paramètre qui a été choisi expérimentalement.

Le descripteur DAISY quant à lui applique le même principe du descripteur SIFT avec une légère différence au niveau de l'application du filtre gaussien, la somme pondéré des gradients est remplacé par la convolution de ces derniers avec plusieurs filtres gaussiens dans des directions spécifiques.

Pour les descripteurs binaires, plusieurs techniques ont été utilisées pour le calcul de la direction prédominante du patch telque, la méthode des moments pour le descripteur ORB. En comparant les gradients des paires sélectionnées pour le BRISK et le FREAK.

Quelques descripteurs invariants de rotation ont été proposés dans la littérature, tels que MROGH [109], qui utilise la mise en commun des intensités avec des gradients invariants à la rotation. LIOP [12] a appliqué une autre approche de la même manière pour agréger les informations de gradient. Même si BRISK [9] et FREAK [10] prétendent être invariants aux changements de rotation, ils restent dépendants de l'estimation de l'orientation incluse dans le processus d'extraction du descripteur. Les descripteurs basés sur l'apprentissage reposent toujours sur l'estimation de l'orientation du détecteur de caractéristique utilisé. Une approche a été proposée par [110], ou les réseaux CNN ont été utilisés pour prédire les orientations stables

qui entraînent un gain significatif par rapport à l'état de l'art. Cependant, cela reste une étape supplémentaire dédiée à l'estimation indépendante de l'orientation des points caractéristiques.

Le descripteur proposé présente la particularité d'invariance au changement de l'orientation et ceux sans recourir à une étape supplémentaire dédiée à cette tâche car sa structure en elle-même lui offre cette propriété. Nous avons exploité deux composantes de l'image à savoir l'intensité et le gradient (l'orientation et la magnitude) dans le but de construire le descripteur proposé. Ce dernier est constitué de deux histogrammes bidimensionnels contenant respectivement, l'orientation et la magnitude du gradient des pixels autour du point clé et les intensités vs les magnitudes du gradient de ces derniers.

Dans le cadre du présent travail, nous avons utilisé la distance moyenne entre deux patches, la décision de correspondance ou pas des deux points est tributaire d'un seuil que nous avons fixé. Cette mesure a été proposée pour une première approche. Cela dit, elle fera l'objet d'amélioration dans des travaux futurs.

Dans le cadre de ce travail, nous avons proposé un module de pré-élimination non supervisée de point clés erronément appareillé afin d'améliorer la précision de détection du schéma proposé, basé sur la méthode de k-means clustering. Cette étape a été ajoutée dans le contexte de la détection d'objet, après le processus de correspondance et avant l'opération RANSAC. Nous présenterons le descripteur proposé dit ADOCH pour différence absolue des histogrammes cumulées de façon détaillée dans la partie qui suit.

L'objectif principal du regroupement de données est l'identification de grappes homogènes sur un ensemble d'objets. Les auteurs de [111] ont fait un travail intéressant en résumant plusieurs techniques de détection des valeurs aberrantes. Les auteurs de [112] ont proposé une méthode permettant de choisir des centres initiaux qui ne sont pas aberrants en utilisant deux méthodes d'initialisation. Le CHB-K-Means [113] a détecté des valeurs aberrantes en utilisant une matrice pondérée par des attributs. Dans [114], Yu et al. proposent le k-means OEDP où les valeurs aberrantes sont supprimées du jeu de données avant d'appliquer l'algorithme k-means.

II.2. Présentation du descripteur

Une étude exhaustive des différents descripteurs dans l'état de l'art a montré la grande efficacité de discrimination, ainsi que la forte résistance des descripteurs basés sur les histogrammes du gradient. Ceci a eu pour effet de motiver l'orientation de notre recherche vers ce genre de descripteurs.

Nous nous sommes intéressés dans la recherche proposé à effectuer une association efficace entre deux composantes de l'image à savoir l'intensité et le gradient. Ce choix a été motivé par le fait que dans le cas de terminaux à faible capacité de calcul, la génération de ces composantes est aisée et rapide à générer et à traiter.

Le principe du descripteur proposé est schématisé dans la Fig.II.1. En premier lieu, une détection des points caractéristiques est effectuée à l'aide du détecteur MSER.

Le gradient et l'intensité de l'image originale sont ensuite calculés. Cette étape est suivit par l'extraction et la description des patchs entourant les points clés. S'en suit alors l'opération de correspondance entre ces derniers, et la décision finale.

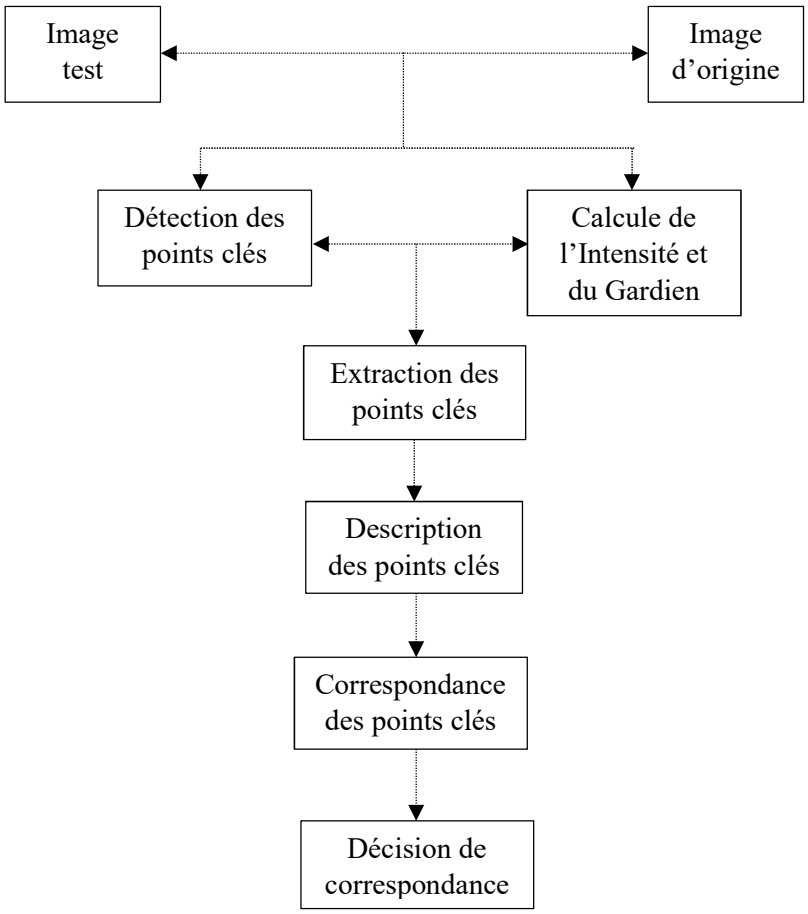


Fig. II.1. Schéma synoptique du descripteur proposé.

Nous allons développer de manière détaillée les différentes composantes du schéma précédent.

II.2.1 Détection des points clés

Dans le cadre de ce travail, nous avons utilisé le détecteur MSER dans la partie de détection des points clés. Le détecteur MSER [50/64] extrait les caractéristiques régionales qui sont stables sous diverses transformations géométriques et photométriques. L'algorithme de ce dernier est simple et se résume en quatre étapes principales à savoir :

- Partir d'un point extrême d'intensité locale.
- Aller dans tous les sens jusqu'au point extremum d'une fonction f donnée. La courbe reliant les extremums dans toutes les directions est la limite de la région d'intérêt.
- Calculer les moments géométriques du deuxième ordre pour cette région.
- Remplacer la région avec ellipse.

La détection des points caractéristiques est assurée par le détecteur de Harris développé précédemment. Il est important de noter que les points d'ancrage détectés sont des extrema locaux d'intensité à plusieurs échelles. Chaque caractéristique détectée est une région connectée qui est soit plus sombre ou plus lumineuse que son environnement immédiat. Un MSER est

basé sur l'idée de prendre des régions qui restent à peu près les mêmes à travers un large éventail de seuils. Tous les pixels en dessous d'un seuil donné sont blancs et tous les pixels supérieurs ou égaux sont noirs.

L'extraction des MSERs est effectuée en performant un balayage du seuil d'intensité en passant du noir au blanc, effectuant un seuillage de luminance simple de l'image. Cette opération est suivie par l'extraction des composants connectés ("Extremal Regions"). Ensuite, un seuil est trouvé lorsqu'une région extrémale est "Maximum Stable". En raison de la nature discrète de l'image, la région au-dessous / au-dessus peut coïncider avec la région réelle, auquel cas la région est toujours considérée comme maximale.

Le mot extremal renvoie à la propriété que tous les pixels à l'intérieur du MSER ont une intensité plus élevée (régions extrémales brillantes) ou inférieure (régions extrémales noires) que tous les pixels de la limite externe.

Si on nous montre une séquence d'images seuillées avec un seuil t, nous verrons d'abord une image noire, puis des points blancs correspondant à des minima d'intensité locaux apparaîtront puis grossiront. Ces taches blanches finiront par fusionner, jusqu'à ce que toute l'image soit blanche. Un extrême sera tout blanc, l'autre tout noir. Entre les deux, les blobs se développent et fusionnent. Les régions extrémales ont une importante propriété qui est l'invariant affine et peu importe que l'image soit déformée ou inclinée.

II.2.2 Description des points clés

Une fois détecté, le processus de description des points clés et leurs régions environnantes peut être enclenché. Le principe du descripteur proposé est d'exploiter l'information fournie par deux composantes de cette dernière à savoir, l'intensité et le gradient (orientation et magnitude), pour une meilleure description du point caractéristique.

Beaucoup d'études [54,57] ont montré l'efficacité des descripteurs basés sur l'histogramme du gradient et de l'intensité. Nous avons ainsi décidé d'apporter notre contribution sur la base de ces descripteurs en proposant un descripteur basé sur deux histogrammes bidimensionnels regroupent les deux composantes du gradient et de l'intensité.

La première étape du descripteur proposé consiste à calculer l'intensité et le gradient (magnitude et orientation) de chaque image, tel que :

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \qquad \theta = tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right) \qquad (II.1)$$

La deuxième étape du descripteur proposé consiste à identifier et sélectionner les points caractéristiques et leurs environnements immédiats.

Le paramètre délimitant les voisins sélectionner pour la description du point clé est délimité par un patch de $s \times s$, où s est la taille de ce dernier.

La Fig.II.2, montre le processus de sélection et d'extraction des points clés et de leur environnement immédiat,

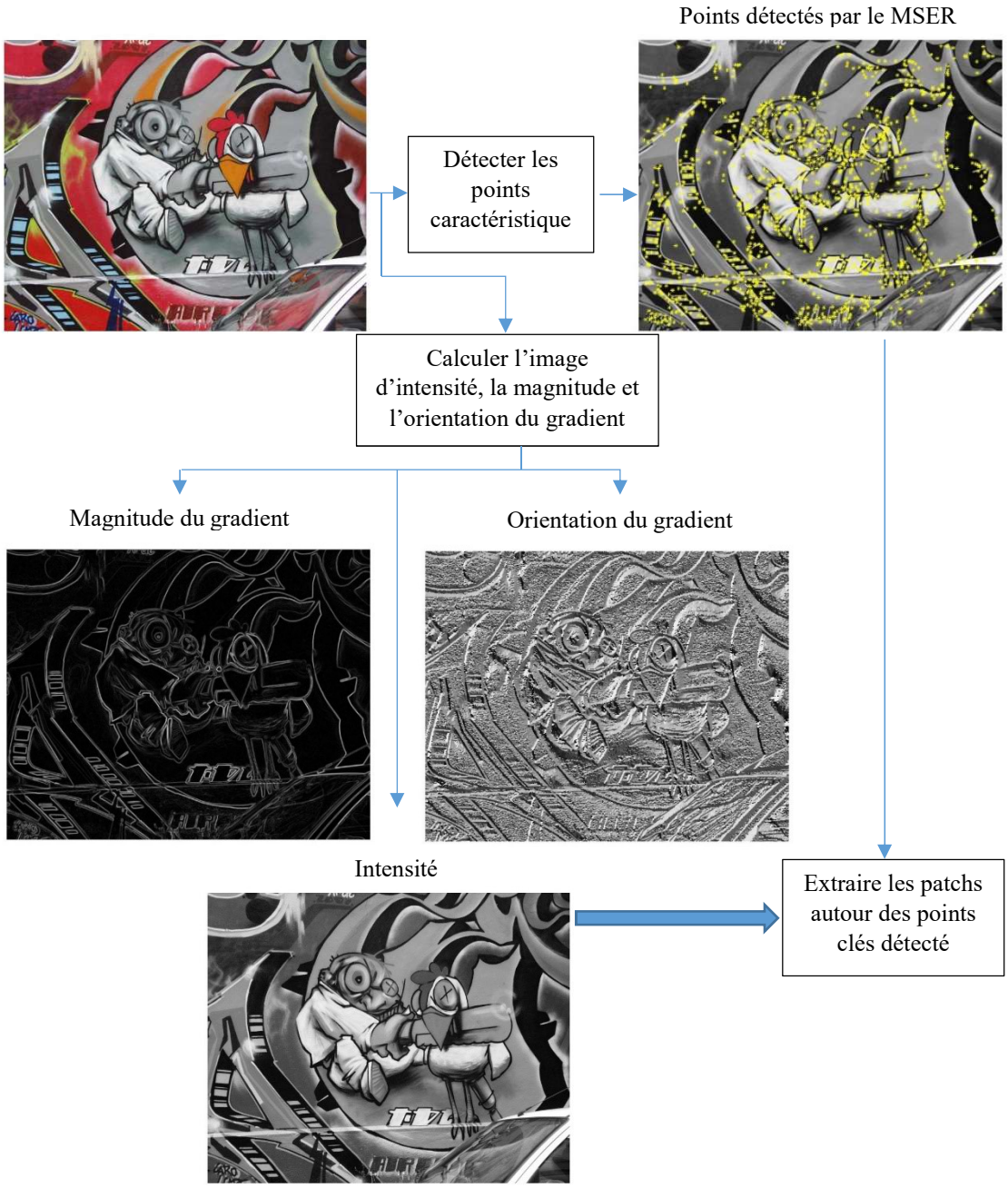


Fig. II.2. Processus d'extraction des points clé et de leur environnement immédiat.

II.2.2.1 Extraction et quantification des patchs

Une fois les points clés détectés et les images d'intensité, de l'orientation et de la magnitude du gradient générés. Un patch est extrait et quantifié à partir des trois images et ceux pour chaque point clé.

Les trois patchs son respectivement, le patch des intensités P_I, le patch des magnitudes du gradient P_m et le patch des orientations du gradient P_θ.

Un exemple montrant l'opération d'extraction des patchs est illustrée par la Fig.II.3, où on peut voir de façon détaillée le contenu des trois patchs autour d'un point caractéristique donné.

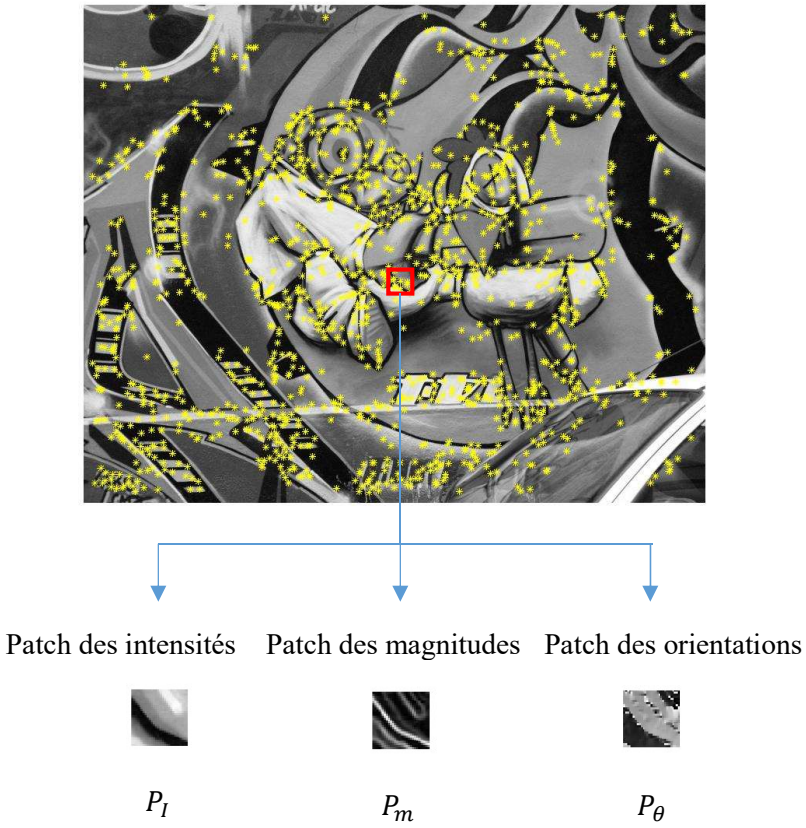


Fig. II.3. Exemple montrant le contenue des patches extrais à partir des trois images.

La prochaine étape est la quantification des éléments de chaque patch. Cette étape est importante car elle permet d'homogénéiser les valeurs contenues dans les patches dans le but de construire le descripteur proposé.

Le processus de quantification est simple, les patchs des intensités et des magnitudes du gradient son quantifié sur une échelle de [1-5], il est important de noter que le choix s'est porté sur cette échelle pour réduire la taille de notre histogramme. Cependant, il est possible d'utiliser d'autres échelles. Les étapes de quantification sont :

- Trouver le maximum dans le patch.
- Diviser tous les éléments par le maximum et multiplier par l'échelle.

L'orientation du gradient est quantifiée sur une échelle de [1-4], ou tous les ongles compris entre [1-90°] sont représentés par le chiffre un et vice-versa avec un décalage de 90°.

La Fig.II.4 représente une illustration du processus de quantification que nous avons mis en œuvre dans le cadre de cette étude.

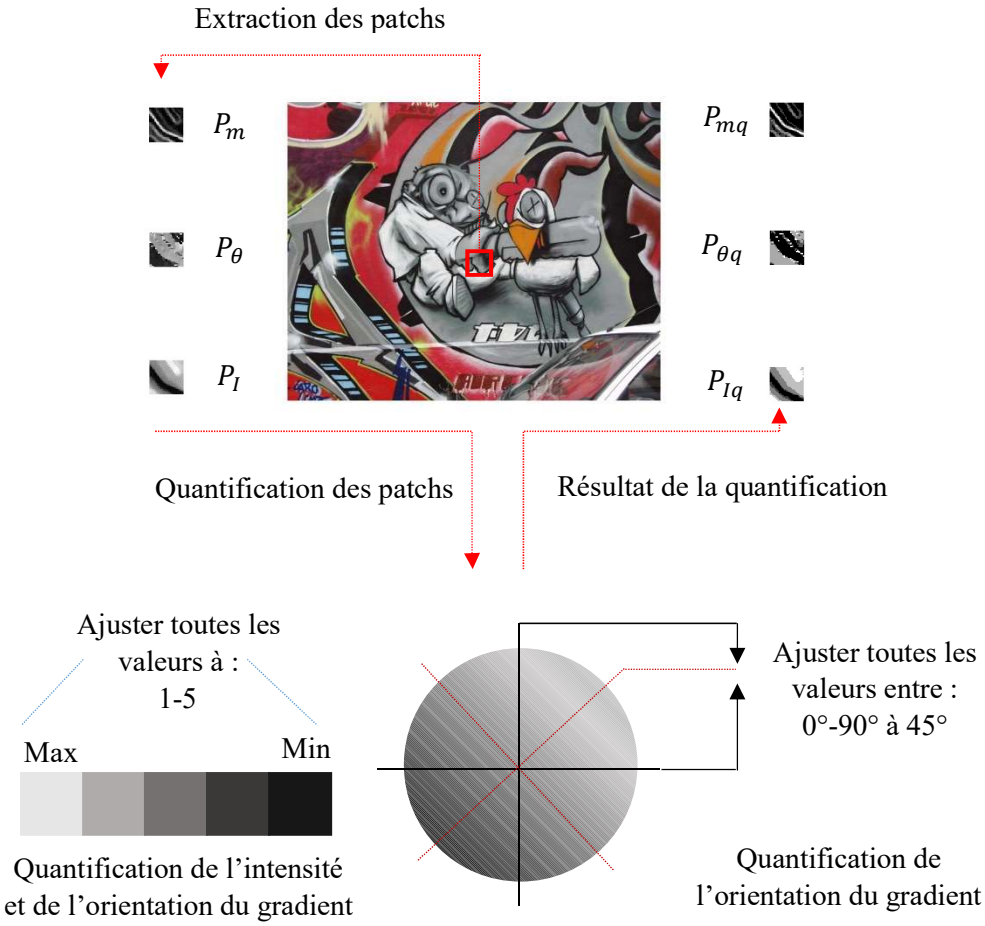


Fig. II.4. Illustration du processus de quantification.

Lorsqu'on remarque les patchs des intensités et des magnitudes du gradient, ces derniers sont constitués d'un dégradé de gris. Les plus petites valeurs sont représentées dans les deux cas par les régions foncées tandis-que les valeurs les plus élevés sont définies par les régions claires.

Dans le cas des magnitudes du gradient, il est clairement possible de constater que les bords représentés par les lignes blanches sont les régions où la magnitude est la plus élevée. Ainsi, les bords prédominants contiennent les valeurs les plus élevées et l'intensité de ces derniers diminuent au fur et à mesure que leurs magnitude est moins importante.

Le même raisonnement peut s'appliquer aux patches des intensités où dans ce cas, les régions blanches représentent les parties où l'intensité est la plus élevée. Par-ailleurs, les plus petites intensités sont définies par les régions noires. Les valeurs intermédiaires sont représentées par un dégradé de gris suivant leurs valeurs.

Enfin, les orientations du gradient sont quantifiées sur quatre espaces. Comme le montre la Fig.II.10, nous avons quantifié toutes les valeurs de l'orientation entre 0° et 90° à une seule valeur à savoir 45°. Cette opération est appliquée au reste des orientations avec un décalage de

90°. Ceci nous permet de réduire la dimension de l'histogramme proposé d'une part et de réduire les disparités entre différentes valeurs des orientations de l'autre.

II.2.2.2 Constitution des Histogrammes

Une fois la détection des points clés effectuée et après l'extraction et la quantification des trois patchs, la prochaine étape est donc la construction des histogrammes bidimensionnelles.

La Fig.II.5, illustre bien le processus de construction des histogrammes,

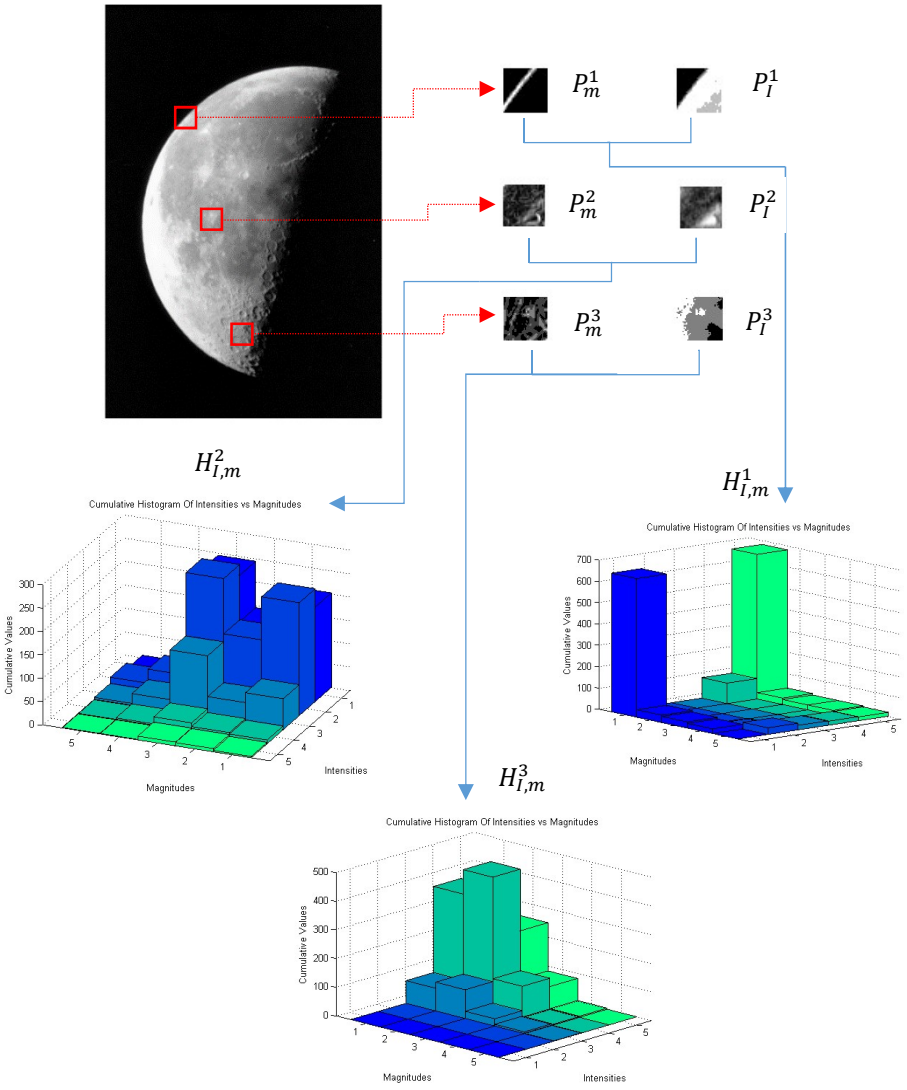


Fig. II.5. Illustration du processus de création des histogrammes.

Nous avons décidé de construire deux histogrammes bidimensionnels à partir des patchs d'intensités et des gradients, à savoir :

- L'histogramme de l'intensité vs la magnitude du gradient $H_{I,m}$.
- L'histogramme de l'orientation vs la magnitude du gradient $H_{\theta,m}$.

Pour $H_{I,m}$, les intensités et les magnitudes du gradient quantifié sont respectivement réparties sur l'axe des x et des y (de 1 à 5), l'accumulation des votes de chaque paire (Intensité, magnitude) sont quant à eux réparties sur l'axe des z. la même procédure est valable pour $H_{\theta,m}$, où les orientations du gradient sont quantifiées sur l'axe des x, les magnitudes sur l'axe des y et l'accumulation des votes sur l'axe z.

En observant le patch d'intensités P_I^1, on peut clairement voir qu'il est composé de deux régions dominantes à savoir, une partie noire et une autre blanche. Une petite région à l'extrémité est grise. Le patch des magnitudes du gradient P_m^1 contient quant à lui un bord clair entouré d'une région noire. Cette distribution se reflète parfaitement au niveau de l'histogramme $H_{I,m}^1$ où les intensités sont principalement réparties entre deux valeurs qui sont un et cinq, avec quelques votes pour la valeur quatre représentant la région grise. Les magnitudes du gradient sont quant à elles essentiellement réparties autour de la valeur de un représentant la région noire et quelques votes sont attribués à des valeurs plus élevées correspondant à l'unique bord du patch.

Dans le cas de P_I^2, les intensités fluctuent entre certaines régions noires et grises sans une séparation claire entre elles. P_m^2 est également composé de quelques bords fluctuant de différentes grandeurs. Comme dans le premier cas, les votes des intensités et des magnitudes de gradient sont répartis sur plusieurs petites valeurs au niveau de l'histogramme $H_{I,m}^2$.

Enfin, concernant le dernier patch P_I^3, nous pouvons clairement voir qu'il est essentiellement composée de trois régions grise, blanche et noire. Le patch des magnitudes de gradient P_m^3 est quant à lui composé de plusieurs petites bords. Cette description de patch est bien résumée dans le dernier histogramme $H_{I,m}^3$, où la plupart des votes sont répartis sur trois valeurs à savoir trois, quatre et cinq représentant les trois régions d'intensité précédentes. D'autre part, les magnitudes de gradient sont réparties sur différentes valeurs de faible amplitude entre un et trois.

Cet exemple illustre bien le mode de transformation patch-histogramme et par la même, l'efficacité discriminative du descripteur proposé car dans les trois cas, la distribution des intensités et des magnitudes au niveau des histogrammes reflète parfaitement leurs représentation au niveau des patches.

II.2.2.3 Propriété d'invariance au changement d'orientation

En observant les patches de l'image originale de la Fig.II.5 et sa version pivotée. Nous avons remarqué que la distribution des bords et des intensités est la même sur les deux patchs, seules leurs positions ont changé.

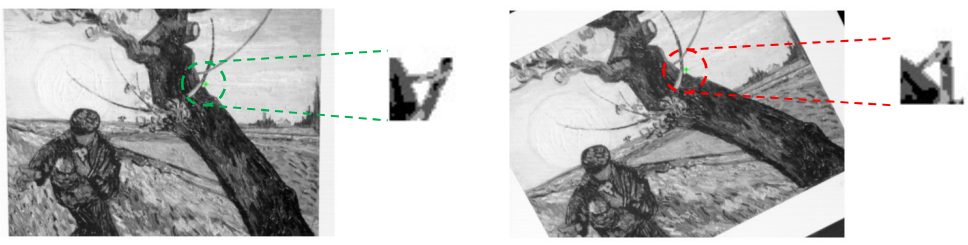


Fig. II.6. Une comparaison de deux patchs entourant le même point d'intérêt de l'image d'origine et de sa version pivotée à partir de la base de données VanGogh.

Notre intuition à travers cette observation a été de dire que, même si nous effectuons une rotation sur l'image originale. La distribution des bords restera inchangée et par conséquent, la distribution des intensités et des orientations de gradient autour de ces derniers ne changera pas non plus.

Nous avons ainsi fait un test en calculant les deux paires d'histogrammes $(H^1_{I,m}, H^1_{\theta,m})$ et $(H^2_{I,m}, H^2_{\theta,m})$ à partir des patches de de l'image originale et sa version pivoté dans la Fig.II.6.

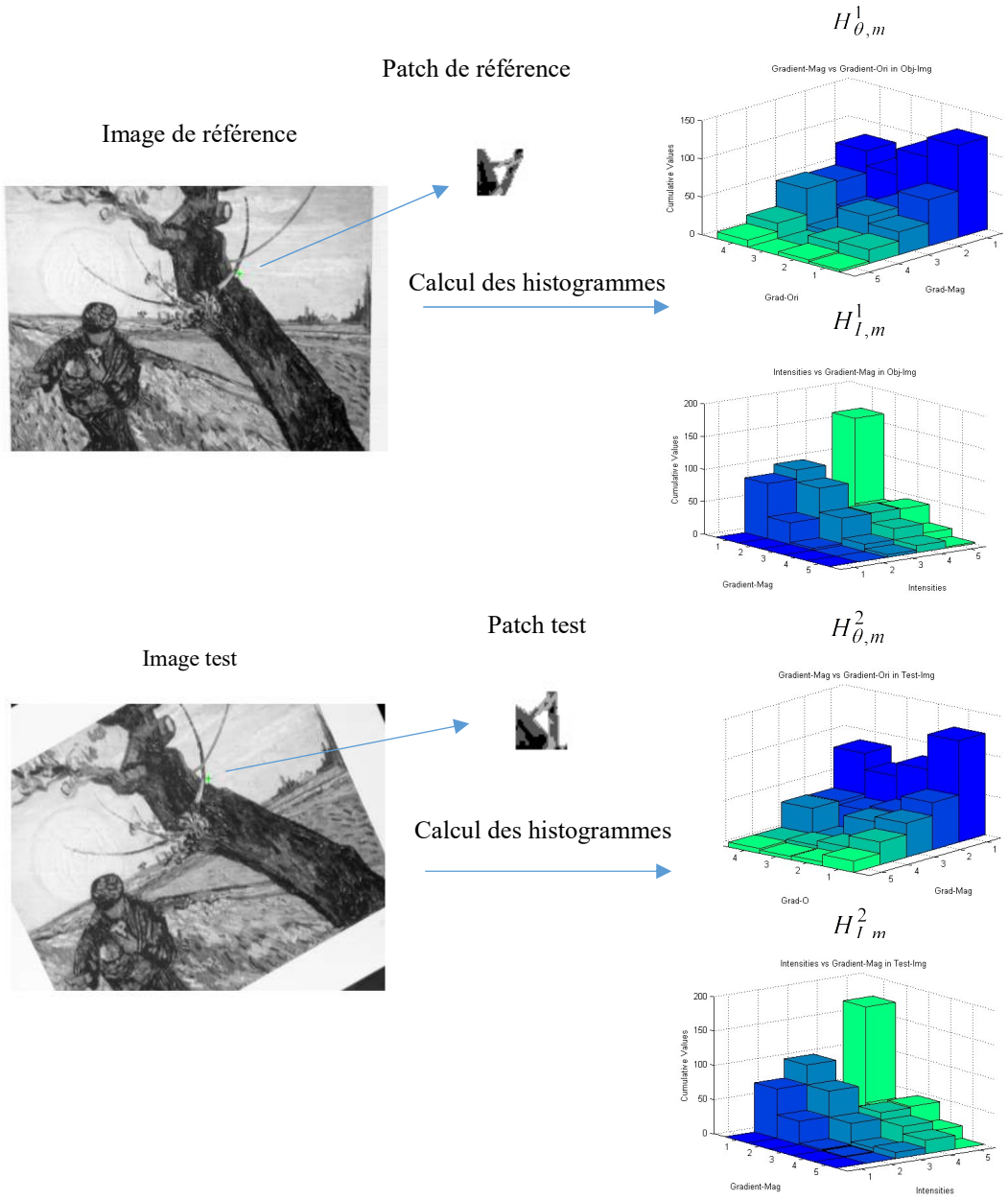


Fig. II.7. Histogrammes résultants de deux patchs entourant le même point d'entité dans l'image d'origine et sa version pivotée.

Ce test vient confirmer et conforter notre intuition car les histogrammes obtenus à partir des patches entourant les points caractéristiques dans les images originales et pivotées de la Fig.II.7, sont extrêmement similaires.

II.2.2.4 La correspondance des points clés

Dans la partie correspondance, l'objectif est d'obtenir le plus haut pourcentage de similarité entre les points clés sélectionnées à partir des deux images originale et transformée. Pour ce faire, nous allons comparer les paires d'histogrammes $(H_{I,m}^1, H_{\theta,m}^1)$ et $(H_{I,m}^2, H_{\theta,m}^2)$, obtenues respectivement à partir des patchs entourant les points caractéristique de l'image test I^1 et celle de référence I^2.

Dans le but de maintenir la faible complexité du schéma proposé, nous avons simplement effectué une soustraction entre les deux paires d'histogrammes. Cette opération débouche sur l'obtention d'une paire de leurs différences ($H_{\theta,m}$ et $H_{I,m}$).

Ainsi, nous considérons que dans le cas où un élément $h_{\theta,m}^j \in H_{\theta,m}$ ou $h_{I,m}^j \in H_{I,m}$ est inférieur à son équivalant seuillé dans les histogrammes de l'image test tel que, $h_{\theta,m}^j <^? Th * h_{\theta,m}^{1,j}$ ou $h_{I,m}^j <^? Th * h_{I,m}^{1,j}$ nous ajoutons un aux scores de correspondance (S_θ et S_I).

En d'autres termes, si l'on soustrait les histogrammes de l'image de référence $(H_{I,m}^2, H_{\theta,m}^2)$ de ceux obtenues de l'image test $(H_{I,m}^1, H_{\theta,m}^1)$. Une correspondance parfaite doit correspondre à des histogrammes ($H_{\theta,m}$ et $H_{I,m}$) dont tous les éléments sont égales à zéro. Du moins, tous les éléments résultants ($h_{\theta,m}$ et $h_{I,m}$) sont inférieurs aux éléments seuillés dans les histogrammes de l'image test, telsque : $\forall h_{\theta,m}^j < Th * h_{\theta,m}^{1,j}$ et $\forall h_{I,m}^j < Th * h_{I,m}^{1,j}$.

Ainsi, les scores de correspondances S_θ et S_I sont donnés par,

$$\begin{cases} S_I(\%) = \left[\sum_j \left(h_{I,m}^j < Th * h_{I,m}^{1,j} \right) \Big/ N_I \right] * 100 \\ S_\theta(\%) = \left[\sum_j \left(h_{\theta,m}^j < Th * h_{\theta,m}^{1,j} \right) \Big/ N_\theta \right] * 100 \end{cases} \quad II.2$$

Où N_θ et N_I représentent respectivement le nombre total des éléments dans $H_{\theta,m}$ et $H_{I,m}$. Le score final S_F d'une paire de points clés est la moyenne de S_θ te S_I, de telle sorte que

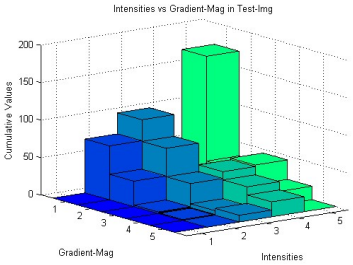
$$S_F(\%) = mean\left(S_\theta(\%), S_I(\%)\right) \quad II.3$$

Nous considérons dans le cadre de ce travail qu'une correspondance entre deux points est correcte si le score final $S_F(\%)$ est supérieur au seuil de correspondance *ThM,* que nous avons fixé à 70%. Tel que,

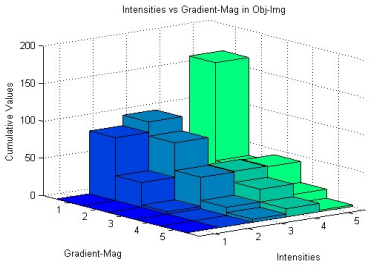
$$\begin{cases} si\ S_F(\%) \geq ThM(\%), & la\ correspondance\ est\ correcte \\ si\ S_F(\%) < ThM(\%), & la\ correspondance\ est\ incorrecte \end{cases} \quad II.4$$

Une correspondance parfaite est équivalente à un score final (ou un pourcentage de similarité) de 100%. Afin d'illustrer le processus de correspondance d'une meilleur manière, nous avons effectué une soustraction entre les histogrammes de la Fig.II.7.

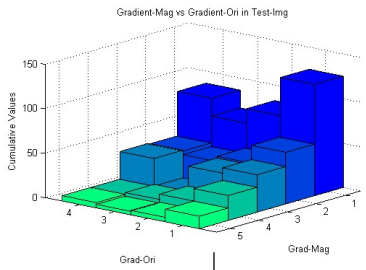
$H^1_{I,m}$



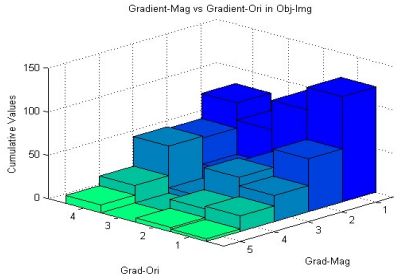
$H^2_{I,m}$



$H^1_{\theta,m}$

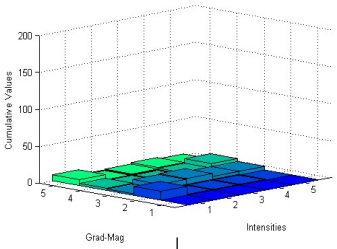


$H^2_{\theta,m}$



Le score obtenu est équivalant à une correspondance parfaite entre les points sélectionnés à partir des images originale et sa version pivoté.

$H_{I,m}$



$$S_I(\%) = \left[\sum_j \left(h^j_{I,m} < Th * h^{1,j}_{I,m}\right) \Big/ N_I\right] * 100 = 100\%$$

Calcul des scores S_θ, S_I et le score final S_F

$$S_F(\%) = mean\left(S_\theta(\%), S_I(\%)\right) = 100\%$$

$$S_\theta(\%) = \left[\sum_j \left(h^j_{\theta,m} < Th * h^{1,j}_{\theta,m}\right) \Big/ N_\theta\right] * 100 = 100\%$$

$H_{\theta,m}$

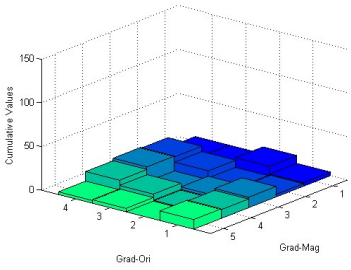


Fig. II.8. Exemple montrant le processus de correspondance entre les histogrammes obtenus à partir d'une paire de points clés extraites de l'image d'origine et sa version pivotée.

La Fig.II.8 illustre bien le processus de correspondance que nous avons mis en place. Nous pouvons clairement voir que presque tous les éléments des histogrammes résultants $h^j_{\theta,m} \in H_{\theta,m}$ et $h^j_{I,m} \in H_{I,m}$ sont égaux à zéro. Au moins, tous les éléments sont inférieurs à ceux des histogrammes test. Ainsi, nous avons une correspondance parfaite (équivalente à un score de 100%) entre les points sélectionnés. Dans le cas où un ou plusieurs points clés correspondent à un seul point dans l'image d'origine, seule la paire avec le pourcentage de similarité le plus élevé est conservé et les autres sont éliminées.

Afin de montrer l'invariance du descripteur proposé au changement d'angle, nous l'avons testé sur d'autres bases de données comprenant des séquences d'images ayant subi différentes transformation de l'orientation. La Fig.II.9 illustre quelques résultats obtenus par le descripteur proposé après le processus de correspondance.

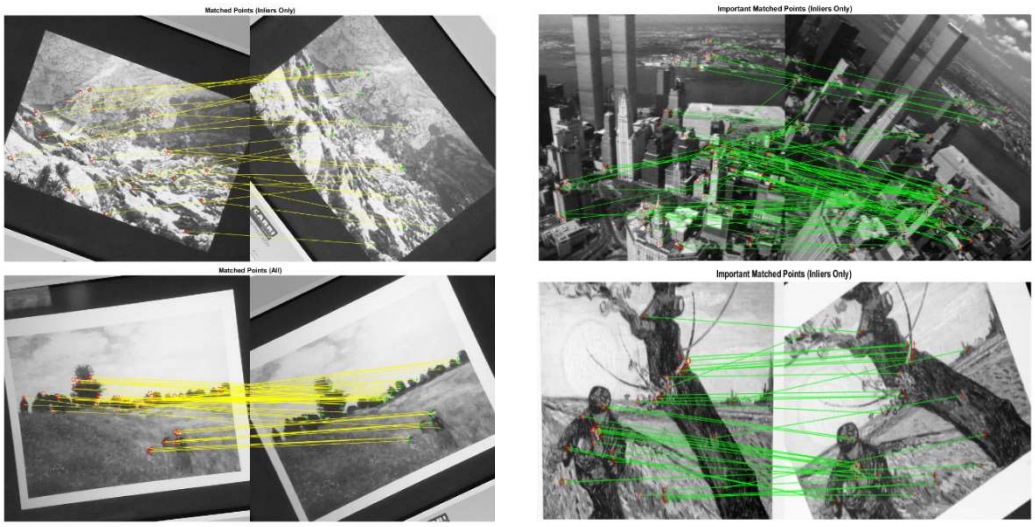


Fig. II.9. Quelques résultats visuel obtenues par le descripteur proposé sous différent changement d'angle et ceux sans aucun recours à une estimation préalable de l'orientation.

Il est important de noter que pour une première tentative, nous avons utilisé dans la partie de correspondance la différence entre les histogrammes obtenues. Ceci dit, d'autres méthodes de comparaison plus élaboré pourront donner de meilleurs résultats. Ceci pourra faire l'objet de meilleures investigations pour des travaux futurs.

II.3. Détection d'objets dans les images

Reconnaître des objets à des échelles très différentes est un élément fondamental et un défi dans le domaine de la vision par ordinateur. Les caractéristiques extraites à partir des pyramides d'images présentent une propriété d'invariance à l'échelle dans le sens où le changement d'échelle d'un objet est traduit par un niveau dans la pyramide qui le représente.

Intuitivement, cette propriété permet d'avoir un modèle pour détecter des objets sur une large gamme d'échelles à travers une opération de balayage du modèle sur les deux sens de la pyramide.

Les pyramides d'images ont été très utilisées dans la littérature [115]. Des détecteurs d'objet comme DPM [116] exigeaient un dense échantillonnage à l'échelle pour obtenir de bons résultats (par exemple, 10 échelles par octave).

Actuellement, un grand nombre de tâches liées à la reconnaissance d'objets sont basés sur l'apprentissage. En effet, les réseaux de neurones convolutionnels (ConvNets) [117,118], en plus d'être capables de représenter une sémantique de niveau supérieur, les ConvNets sont également plus robustes à la variation d'échelle et facilitent ainsi la reconnaissance des caractéristiques calculées sur une seule échelle d'entrée [119,120]. Mais même avec cette robustesse, les pyramides sont encore nécessaires pour obtenir les résultats les plus précis. Récemment, ImageNet [121] et COCO [122] ont utilisé des tests multi-échelles sur des pyramides d'images détaillées (par exemple, [123]). Le principal avantage de ces derniers est la production de caractéristiques beaucoup plus précises et tous les niveaux sont représentés, y compris les niveaux de haute résolution.

Néanmoins, en mettant en valeur chaque niveau d'une pyramide d'images, le temps d'exécution augmente considérablement (par exemple, par quatre fois [124]), rendant cette approche peu pratique pour des applications réelles. Cependant, les pyramides d'images ne sont pas le seul moyen de calculer une représentation d'entités multi-échelles. Un ConvNet calcule une hiérarchie de caractéristiques couche par couche avec un sous-échantillonnage des couches, cette méthode permet d'obtenir des caractéristiques sous forme pyramidale. Cette fonctionnalité introduit cependant de grands écarts sémantiques causés par les différentes profondeurs entre les couches.

Une correspondance entre deux points caractéristiques est un processus important dans de nombreuses applications de vision par ordinateur, telles que la reconstruction multi-vu, la reconnaissance d'objets ou la structure du mouvement. L'approche SIFT et ses variantes [124], sont basées sur la distance conventionnelle, par exemple la distance euclidienne pour mesurer la similarité entre deux patchs. Ces méthodes restent largement tributaires de l'expertise humaine et ne fournissent pas de solution optimale. Récemment, de nombreuses approches basées sur l'apprentissage ont été proposées [125] afin d'adapter les fonctions de similarité à des ensembles de données donnés.

Dans le cadre de ce travail, nous allons également utiliser les pyramides. Cependant, ces dernières ne sont pas appliques à l'échèle des images mais directement aux patches entourant les points caractéristiques sélectionnées.

Car le changement d'échelle est évident dans le contexte de la détection d'objet. Nous avons choisi d'utiliser différentes échelles pour les patchs sélectionnés au niveau de l'image contenant l'objet. Tel que l'illustre la Fig.II.9, nous avons augmenté la taille du patch entourant le point clé dans l'image de référence. Ceci nous a permis de couvrir un plus grand espace dans l'image. Nous avons ensuite redimensionné la taille de ce dernier de manière à ce qu'il

corresponde de nouveau à la taille du patch dans l'image test en utilisant un sous-échantillonnage.

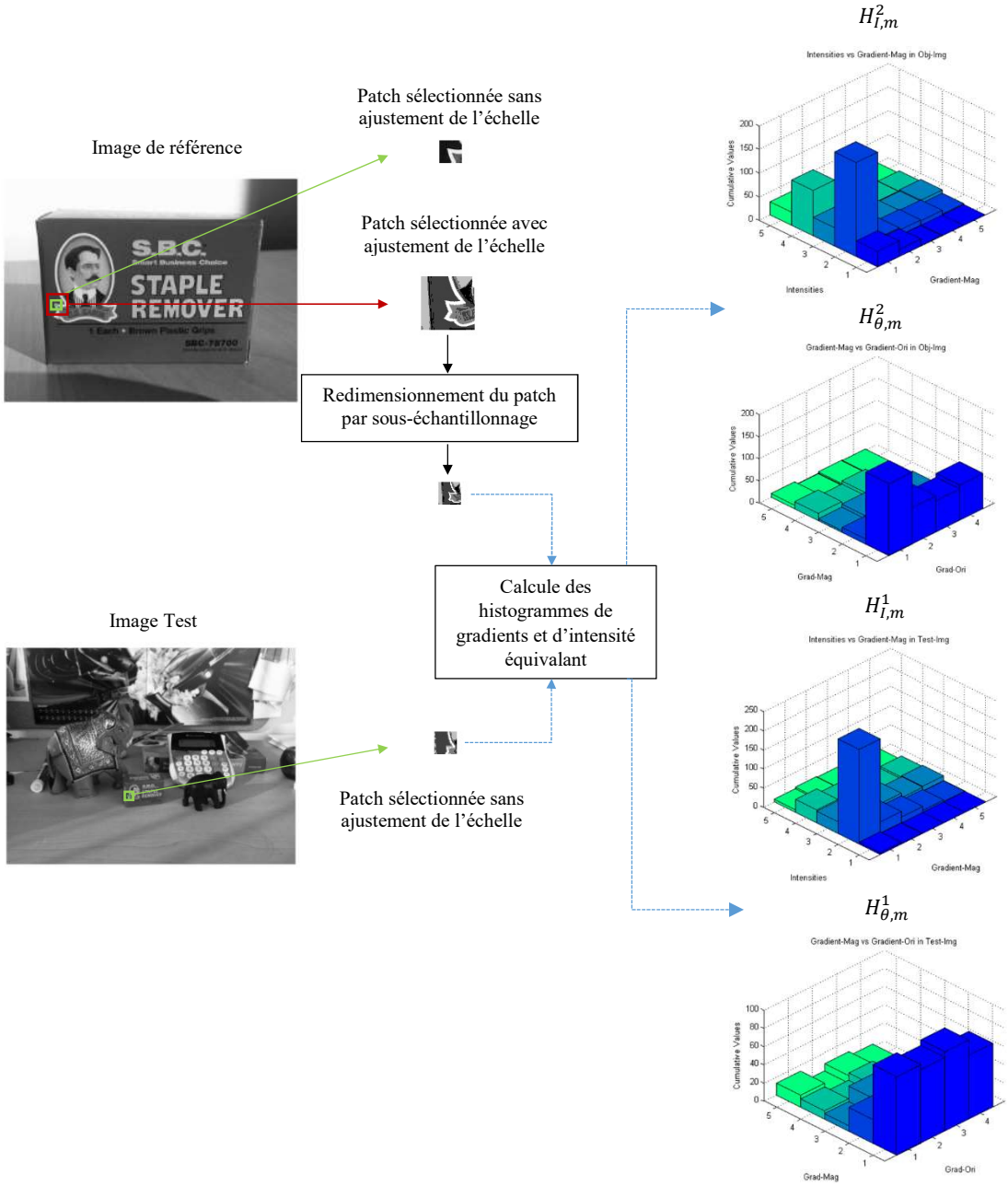


Fig. II.10. Illustre le processus mis en œuvre pour le redimensionnement des patchs au niveau de l'image de référence dans le cadre de la détection d'objets.

Cette méthode nous permet d'obtenir de meilleurs résultats dans la partie correspondante, car si l'on observe l'exemple de la Fig.II.10, nous pouvons clairement constater qu'il y a une importante similarité entre les histogrammes de la paire de points-clés sélectionnés.

Pour chaque paire, nous avons appliqué trois échelles différentes au patch correspondant à l'image de référence $[Scale_1, Scale_2, Scale_3]$ afin de couvrir trois différentes surfaces autour du point clé. Nous conservons alors le meilleur score de correspondance avec le patch test et nous suppriment les autres.

II.4. Processus de pré-élimination des fausses correspondances

Afin d'obtenir de meilleurs résultats de détection des objets, tout en préservant la simplicité du système proposé. Nous avons ajouté une phase de pré-élimination des fausses correspondances, basée sur la méthode d'apprentissage non supervisé de K-means.

II.4.1 K-means Clustring

La methode de regroupement K-means clustering [111] est une méthode couramment utilisé pour partitionner automatiquement un ensemble de données en k groupes. Il procède en sélectionnant k groupes initiaux, ensuite ces derniers sont affiné de manière itérative comme suit :

1. Chaque groupe est constitué de plusieurs données d_i.

2. Le centre de chaque groupe C_j est mis à jour pour devenir la moyenne de ces données constituantes.

L'algorithme converge lorsqu'il n'y a plus de changement dans l'attribution d'instances à des clusters.

Dans le cadre de ce travail, l'objectif principal de cette opération est de segmenter les correspondances obtenues au niveau de l'image test en k régions (k groupes de correspondances) avec une probabilité pour chaque région de contenir l'objet recherché. Cette étape est effectuée après le processus de mise en correspondance et avant le test RANSAC entre les deux images. La Fig.II.11 montre un exemple réel de mise en œuvre de ce processus, pour $k = 3$.

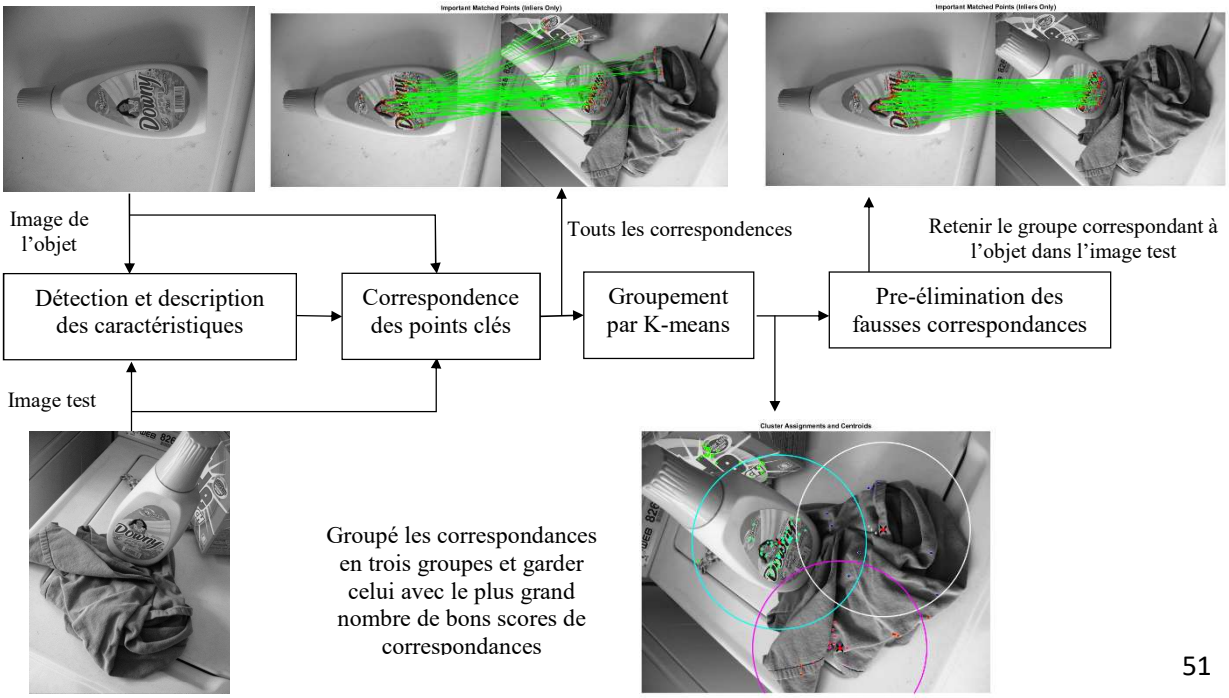


Fig. II.11. Exemple illustrant le processus de pré-élimination des fausses correspondances adopté dans le cadre de la détection des objets.

Nous estimons que le groupe avec le plus grand nombre de bon scores en correspondances définit la région la plus probable contenant l'objet dans l'image de test. Le reste des groupes est par la suite automatiquement éliminé. Cela aide à évincer une quantité considérable de fausses correspondances.

Cette étape est ignorée dans le cas d'un alignement d'image. En effet, la modularité de cette étape permet de ne l'activer qu'en cas de détection d'objets. Les résultats expérimentaux ont montré une nette augmentation de la précision en utilisant ce module.

II.5. Conclusions

Dans ce chapitre, nous avons proposé une contribution dans le domaine de la description des caractéristiques et de l'appariement des images.

La propriété principale du descripteur proposé est son invariance au changement de la rotation et ceux sans avoir recours à une étape additionnel d'estimation et de compensation de l'orientation.

De plus, nous avons ajouté une étape de pré-élimination des fausses correspondances, basée sur la méthode de regroupement k-means afin d'améliorer la précision de détection du descripteur proposé dans le contexte de la détection d'objets.

La facilité de mise en œuvre du descripteur proposé et sa résistance aux différents types de changements d'images le rend attrayant pour diverses applications, telles que les caméras stéréo ou les caméras de surveillance.

Les résultats obtenus à travers les différentes expériences réalisés montrent une nette augmentation de la précision avec le module de pré-élimination des fausses correspondances dans le contexte de la détection d'objets.

Chapitre III

Présentation du détecteur de bords et d'objets pertinents dans les images

III.1. Introduction

En traitement d'image et en vision par ordinateur, on appelle détection de contours les procédés permettant de repérer les points d'une image matricielle qui correspondent à un changement brutal de l'intensité lumineuse. Ces changements de propriétés de l'image numérique indiquent en général des éléments importants de structure dans l'objet représenté. Ces éléments incluent des discontinuités dans la profondeur, dans l'orientation d'une surface, dans les propriétés d'un matériau et dans l'éclairage d'une scène.

La détection des contours dans une image réduit de manière significative la quantité de données en conservant des informations qu'on peut juger plus pertinentes. Il existe un grand nombre de méthodes de détection de l'image mais la plupart d'entre elles peuvent être regroupées en deux catégories. La première recherche les extremums de la dérivée première, en général les maximums locaux de l'intensité du gradient. La seconde recherche les annulations de la dérivée seconde, en général les annulations du laplacien ou d'une expression différentielle non linéaire.

Dans cette partie, nous proposons un nouveau détecteur de bords basé sur une modélisation statistique de la surface de l'image. Nous avons utilisé deux mesures classiques et largement utilisées dans le domaine de traitement des données, à savoir l'écart moyen et l'écart type afin de modéliser du détecteur proposé.

Ces mesures ont été utilisées dans des recherches précédentes, comme dans [49], où les auteurs ont utilisé la moyenne et l'écart type des images en niveaux de gris pour définir un modèle statistique de forme afin d'obtenir un meilleur processus de segmentation d'image.

Ou encore dans [50], où les auteurs ont utilisé des mesures d'analyse statistique locale pour construire des fonctions d'énergie pour les tâches de segmentation.

Même si l'efficacité de ces contributions est prouvée, leur charge de calcul reste lourde puisque les méthodes proposées sont basées sur le calcul de fonctions multi-paramétriques. De plus, leurs performances dépendent de manière cruciale des conditions initiales

Notre approche consiste à mieux comprendre la surface de l'image en prenant en considération les différentes fluctuations de l'intensité de cette dernière.

En plus de la détection des bords, le détecteur proposé est capable de mettre en évidence les régions les plus pertinentes dans l'image. Cette propriété a été exploitée dans le présent travail pour identifier des contours d'image importants.

Outre sa nouveauté et son efficacité, le principal avantage du détecteur proposé est sa simplicité, ce qui facilite son implémentation au niveau des terminaux à faible capacité de calcul.

Il consomme également peu de mémoire et ne nécessite pas de phase préalable pour l'apprentissage le rendant ainsi indépendant de la disponibilité de bases de données avec annotations humaines.

Les expériences ont montré que le détecteur proposé surpasse certains détecteurs de l'état de l'art en termes de résultats et de temps de calcul.

III.2. Présentation du détecteur proposé

Lorsqu'on observe la Fig.III.1, nous pouvons constater que la surface de l'image est essentiellement composée de zones planes séparées par de multiples valeurs faible et élevée d'intensité qui représentent des contours.

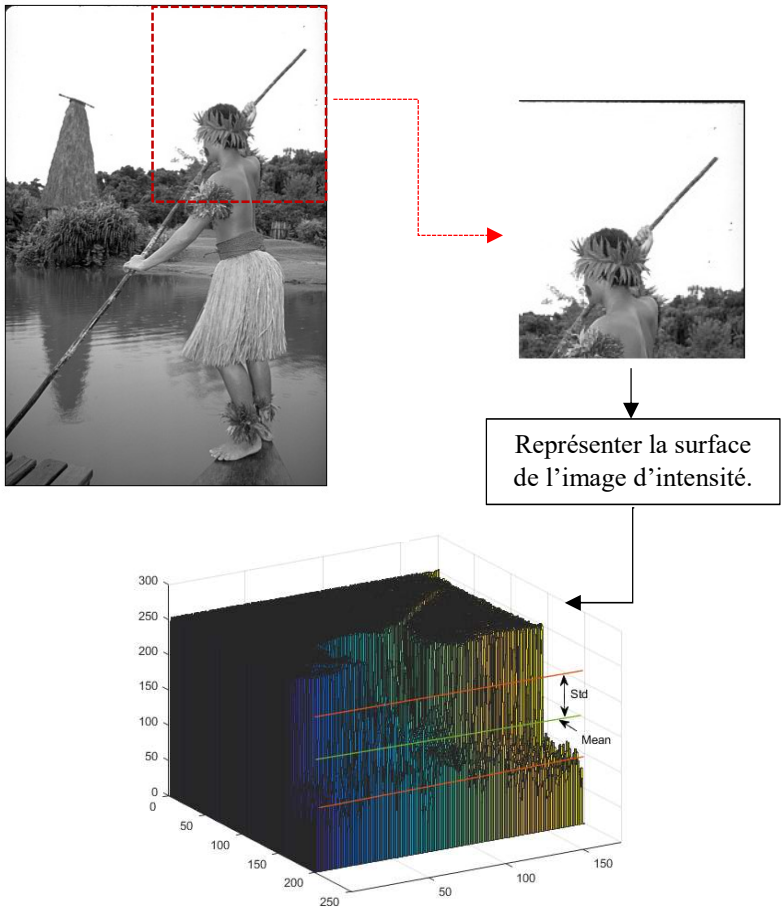


Fig. III.1. Exemple illustrant la surface des intensités correspondant à une partie d'image ou la moyenne de ces intensités est représenté par le seuil vert et la variation de la std par les très orange.

Notre première intuition a été d'essayer d'utiliser le seuillage sur l'image d'intensité afin d'identifier la position des bords dans l'image, tel que le montre la Fig.III.1.

Cependant nous avons vite face à la problématique suivante, si nous choisissons une faible valeur du seuil, plusieurs petits bords seront détectés. En revanche, une valeur élevée nous conduira à ignorer certains bords importants.

III.2.1 Approche du détecteur proposé

Ainsi, nous avons décidé d'adopter l'approche suivante :

- En premier lieu, nous avons appliqué un filtrage gaussien afin de lisser l'image
- Nous avons divisé l'image d'intensité en plusieurs blocs. Nous avons ainsi considéré chaque surface de bloc comme une variable aléatoire.
- Enfin, nous avons calculé la moyenne et de l'écart-type (std) de chaque bloc.

Afin de mieux comprendre la variabilité globale de l'image, nous avons d'abord calculé la valeur std de chaque bloc de $18x18$ pixels dans l'image de la Fig.III.1 et les résultats obtenus sont illustrés par la Fig.III.2, où on peut voir la différence entre deux blocs sélectionnés et classés comme arborant ou pas des bords importants.

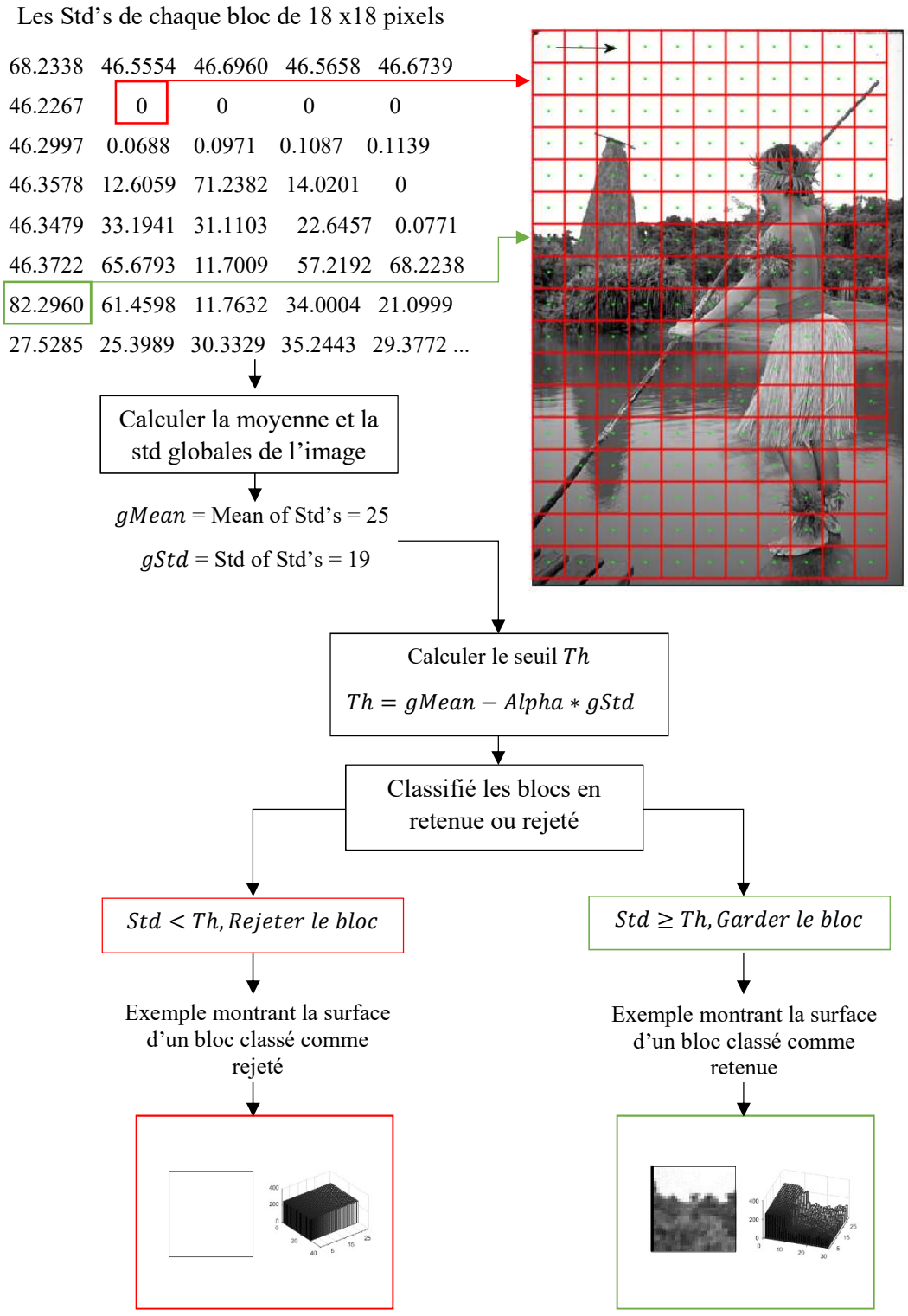


Fig. III.2. Exemple illustrant le processus de classification utilisé par le détecteur proposé.

Nous avons remarqué que les blocs avec des valeurs std élevées sont ceux qui contiennent des changements d'intensité importants et par la même, ceux qui présentent la probabilité la plus élevée d'abriter des bords. En revanche, des valeurs std faibles ou nulles indiquent que les blocs concernés ne contiennent pas d'importantes fluctuations d'intensité, ce qui correspond à une région plate.

III.2.2 Processus de classification

Le processus de classification est réalisé par une comparaison du changement dans l'image au niveau local (bloc) au changement globale de cette dernière. Ce changement est quantifié par une mesure principale à savoir, l'écart type.

En effet, si l'on considère l'exemple de la Fig.III.2. Nous remarquons que la std varie d'un bloc à un autre en fonction des changements en termes d'intensité dans ce dernier. Ainsi, l'idée de calculer la moyenne des différentes standards std's et l'écart type de ces dernières afin d'avoir une idée plus globale du changement dans l'image entière est apparue. Nous avons par la suite décidé de comparer chaque bloc dans l'image à un seuil basée sur ces deux mesures.

Lorsqu'on considère le bloc classé comme rejeté dans la Fig.III.2, on peut constater que sa std est nulle, ce qui vaut à un bloc où il n'existe aucun changement. Ceci est vraie lorsqu'on regarde la surface de ce dernier. Dans le cas du bloc retenu, la std locale est très élevé traduisant un changement important de l'intensité. Ceci est également vrai en vue de la surface contrasté de ce dernier.

La Fig.III.3 montre quelques exemples de blocs classés comme rejetés ou gardés dans l'image précédente.

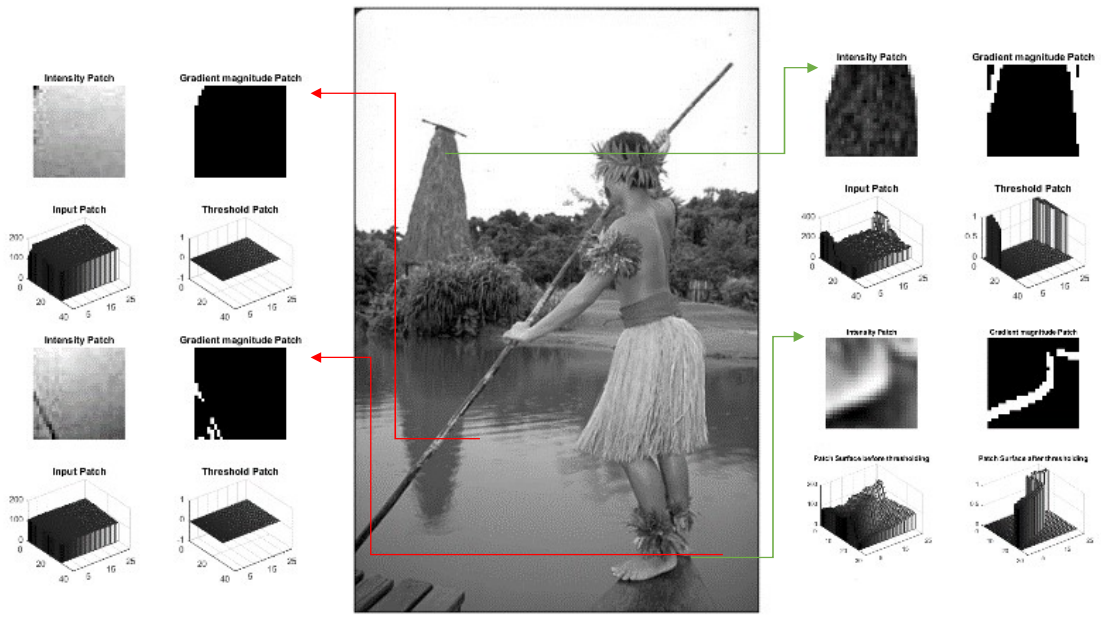


Fig. III.3. Exemple montrant quelques exemples de blocs classés comme rejetés ou gardés en fonction de leurs std.

III.2.3 Variabilité globale de l'image d'intensité

Pour obtenir le champ de variabilité globale de la surface de l'image, nous avons calculé la moyenne globale ($gMean$) et la std globale (gStd) de tous les std's résultants des blocs d'image tels que :

$$\mu = \frac{1}{M}\sum_{l=1}^{M} x_l \qquad Std = \sqrt{\frac{1}{M}\sum_{j=1}^{M}[x_j - \mu]}$$
$$gMean = \frac{1}{N} * \sum_{i=1}^{N} Std_i \quad gStd = \sqrt{\frac{1}{N}\sum_{j=1}^{N}[Std_j - gMean]} \qquad III.1$$

où M est le nombre d'éléments dans le bloc, μ est la moyenne locale du bloc. Std est l'écart type local des éléments du bloc et N , le nombre de blocs dans l'image. $gMean$ est la moyenne de toutes les std's et $gSdt$ est simplement leur écart type.

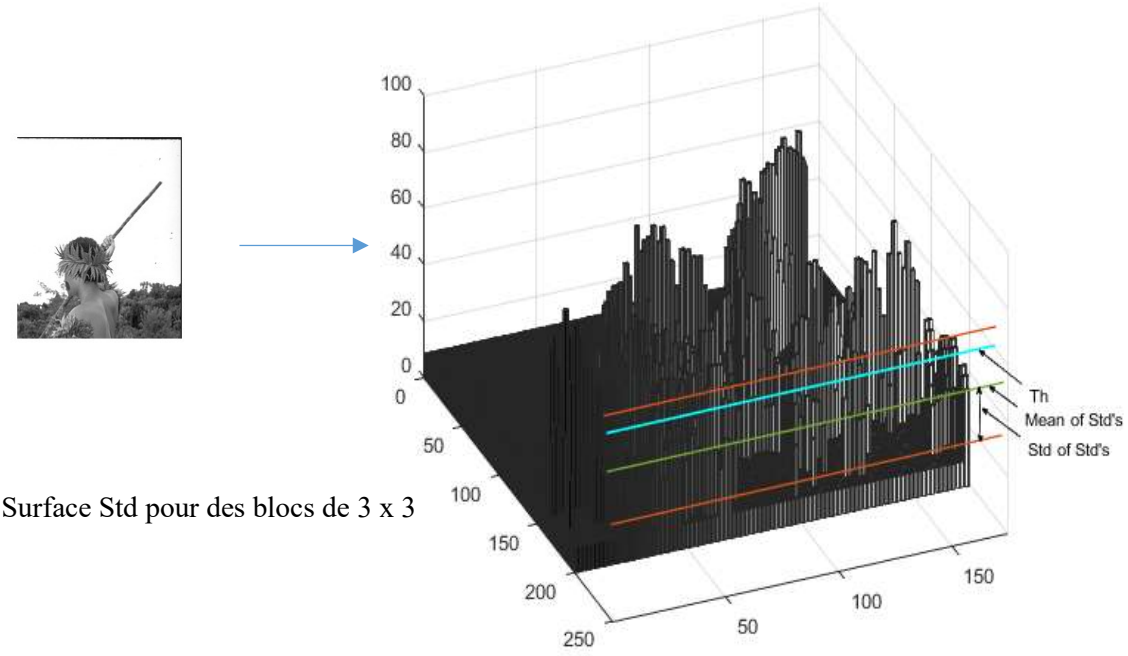
Nous avons dans le cadre de ce travail fixé le seuil de classification à :

$$Th = gMean - Alpha * gStd \qquad III.2$$

où $|Alpha| < 1$, est un coefficient positif ou négatif servant à varier le seuil Th.

Dans la partie classification, nous avons considéré que chaque bloc dont la valeur std est inférieure au seuil fixé, il sera ignoré et remplacé par un bloc contenant uniquement des zéros.

Les blocs conservés seront remplacés par leurs amplitudes de gradient quantifiées, afin d'obtenir la position des bords.



Surface Std pour des blocs de 3 x 3

Fig. III.4. Exemple montrant la surface de l'image en utilisant les Std's avec des blocs de 3x3.

Comme le montre la Fig III.4, nous pouvons clairement voir qu'il est plus évident de seuiller la surface de l'image représenter par les std's des blocs la composant que de le faire directement sur l'intensité de l'image. Dans ce cas, chaque barre représente un bloc de 3x3 pixels, où la valeur de la magnitude est égale à la valeur la std correspondante.

III.2.4 Détection des régions pertinentes dans l'image

Nous avons également remarqué que nous pourrions utiliser cette mesure pour trouver les régions pertinentes dans l'image et l'ampleur de leur pertinence.

Comme le montre la Fig III.5, les images résultantes (en noir et blanc) du processus de classification sont constituées de plusieurs blocs s × s, où s est la taille du bloc et tous les éléments de celui-ci sont égaux à sa valeur std.

Ainsi, les zones sombres dans l'image correspondent aux régions dans l'image qui ne contiennent pas d'importantes fluctuations de l'intensité. Ces derniers ont alors une valeur nulle de la std.

Par ailleurs, les régions lumineuses correspondent aux endroits ou le changement d'intensité est important. Par conséquent leurs std sont élevées.

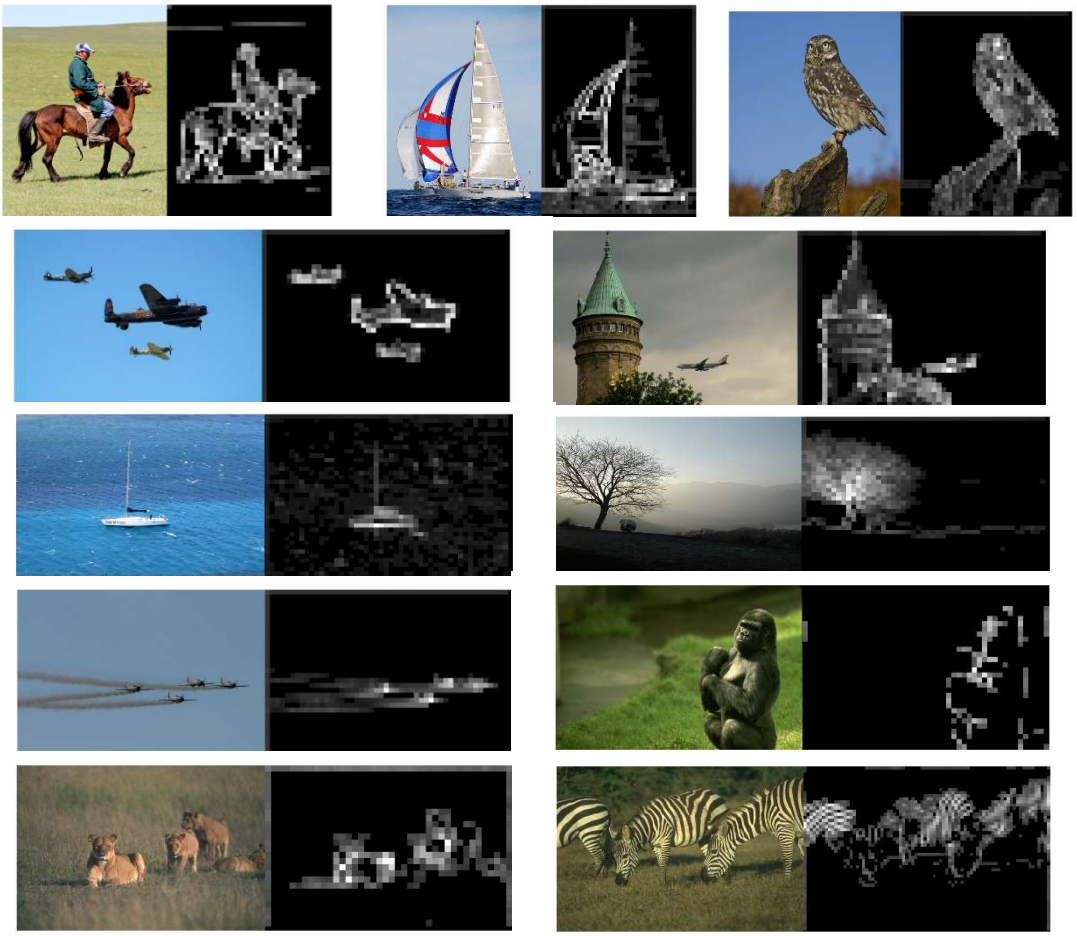


Fig. III.5. Les images résultantes, après le processus de classification, utilisent une taille de bloc de 5x5.

La taille des blocs est importante, car la précision de la détection des régions pertinentes en dépend. Car si l'on prend l'exemple de la Fig.III.6, les objets pertinents (les animaux dans ce cas) sont localisés plus précisément dans l'image (b) avec une taille de bloc de 3x3, par rapport aux blocs de 18x18 dans (a).

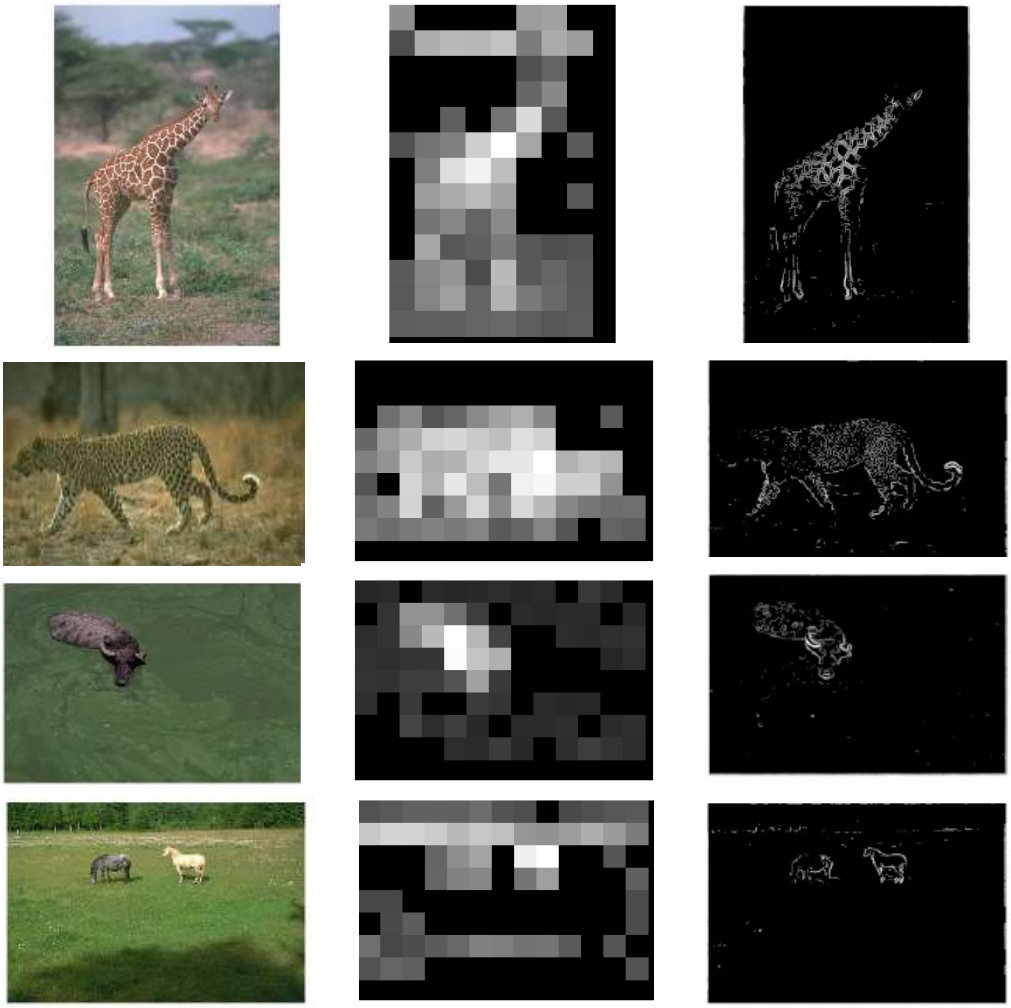


Image d'origine (a) Bloc de 18x18 (b) Bloc de 3x3

Fig. III.6. Influence de la taille du bloc sur le résultat finale du processus de classification et son effet sur la localisation des zones pertinentes dans l'image.

Cette propriété est très importante et peut être exploité dans divers domaines du traitement d'image tel que la segmentation ou encore la suppression de l'arrière-plan dans une image et ceux en n'utilisant que l'image d'intensité, sans qu'il ne soit nécessaire de recourir à des étapes de filtrage et de soustraction. Dans notre cas, nous nous sommes intéressés à un autre aspect important du traitement d'images, à savoir la détection des bords.

En plus de la taille du bloc, un autre paramètre influe sur le résultat final, à savoir le seuil de classification.

Nous avons constaté au cours de nos expérimentations que la taille du bloc a une influence sur la qualité des bords détectés puisqu'avec des blocs de grandes tailles, nous obtenons les bords les plus dominants. Cela dit d'autres bords importants peuvent être ignorés. À l'opposé, de petits blocs donnent plusieurs bords insignifiants.

Le choix du seuil a quant à lui une influence sur le nombre de blocs sélectionnés. Une valeur élevée conduira à ignorer des bords importants. En revanche, une valeur faible donnera trop de détails inutiles.

Un exemple montrant l'influence de ces deux paramètres est illustré par la Fig.III.7.

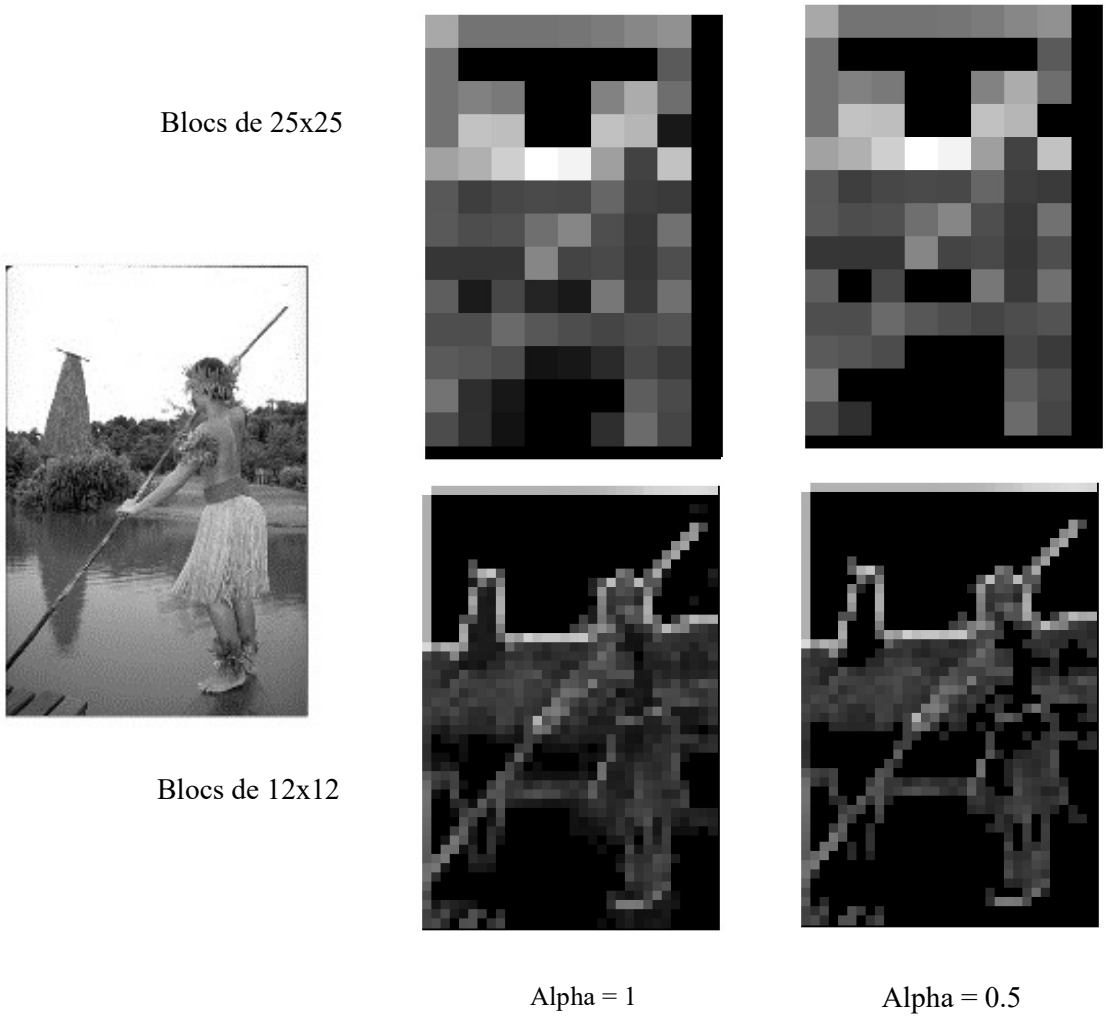


Fig. III.7. Influence de la taille du bloc et du seuil de classification sur le résultat finale du processus de classification.

La moyenne et l'écart type globales $gMean$ et $gStd$ fournissent une autre information importante concernant la texture de l'image.

Puisque $gMean$ représente le changement d'intensité à l'intérieur des blocs. Ainsi, une valeur importante correspond à une image localement contrastée. À l'inverse, une valeur faible correspond à une surface d'image plate.

La $gStd$ est quant à elle relative au changement à l'intérieur du bloc, donc une valeur élevée signifie que l'image est composée de différents éléments et l'inverse indique une image uniforme.

La Fig.III.8 montre un exemple de deux images, l'une est lisse et l'autre est contrastée. Nous avons utilisé trois tailles de bloc différentes avec un seuil fixe où Alpha = 0.25 et le paramètre de filtrage sigma a été défini sur 3.0.

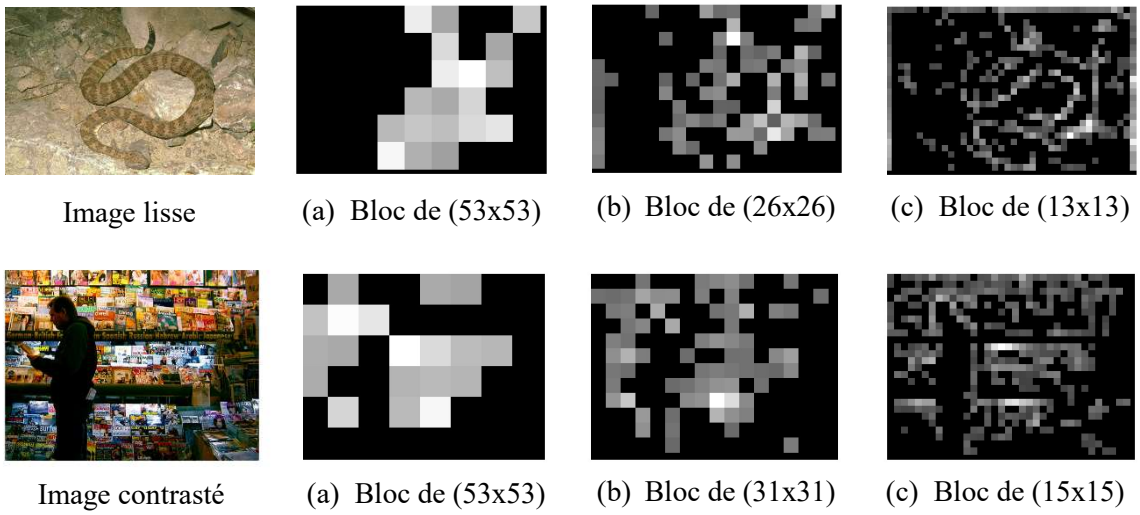


Fig. III.8. Résultats du processus de classification de deux images lisses et contrastées pour différentes tailles de blocs.

Nous pouvons clairement constater que pour les deux images, les régions pertinentes sélectionnées varient en fonction de la taille du bloc utilisé.

Ainsi, nous pouvons observer qu'une détection grossière est effectuée par les blocs de tailles importantes.

Une description plus détaillé du contenue de l'image est réalisé par de plus petit blocs sont utilisé.

Le Tableau III.1, montre la relation entre la nature de l'image et la variabilité de $gMean$ et $gStd$ qui indique respectivement le changement intra et inter des intensités au niveau des blocs d'image.

Nature de l'image \ Taille du bloc			(a)	(b)	(c)
Image lisse	Sigma=3.0	*gMean*	18.9565	14.0509	9.4152
		gStd	6.4719	6.7302	6.0650
	Sigma=1.0	*gMean*	22.8528	18.7073	14.6744
		gStd	7.2535	7.9328	8.0217
Image contrasté	Sigma=3.0	*gMean*	43.3588	31.5346	20.3562
		gStd	14.5350	14.0979	13.0465
	Sigma=1.0	*gMean*	52.0741	41.8821	31.4957
		gStd	15.4563	16.9470	18.2768

Tableau III.1 Variation de la moyenne et la std globale pour différentes tailles de blocs.

Nous remarquons d'après les résultats obtenus que la variation des intensités au niveau des blocs (*gMean*) est importante dans le cas d'une image contrastée. Et ceux aussi bien pour les blocs de taille importante que pour de plus petits blocs. Ceci est caractéristique d'une image localement contrastée.

Nous remarquons également que l'inter-variabilité des blocs d'intensité (*gStd*) est conséquente dans le cas des blocs de petites tailles et diminue au fur et à mesure que ces derniers deviennent plus grands. Ceci est évident puisque le changement est moins important entre des gros blocs que pour les petits.

Enfin, dans le cas d'une image lisse, nous pouvons voir que la variabilité de l'intensité à l'intérieur comme à l'extérieur des blocs est faible. Ceci est expliqué par la nature non contrasté de l'image.

Le paramètre de lissage Sigma est également un facteur important, car nous pouvons constater qu'une valeur élevée de celui-ci tend à lisser considérablement l'image et ainsi à réduire les disparités d'intensité à l'intérieur ($gMean$) et entre ($gStd$) les blocs.

Les résultats montrent que dans le cas de sigma = 1, les petits blocs ont tendance à donner de petites valeurs pour $gMean$ et des valeurs plus élevées pour $gStd$. En effet, ceci est expliqué par le fait que généralement, les petites surfaces des blocs ne changent pas considérablement. Mais les changements entre les blocs sont importants.

À l'inverse, les blocs de grande dimension fournissent des valeurs plus élevées de $gMean$et plus petites de $gStd$, car les plus grands blocs contiennent des d'importants changement d'intensité, mais la différence entre eux est moins importante.

Dans le cas de sigma = 3, le raisonnement reste inchangé pour la variabilité de la moyenne globale par rapport à la taille du bloc mais la std globale est maintenant plus stable en raison de l'effet du lissage gaussien. Ce dernier réduit les faibles disparités d'intensité dans l'image, tout en préservant les différences les plus significatifs. Nous avons choisi de fixer le paramètre sigma à 3.0 pour le reste de nos expériences.

Puisque différentes tailles de blocs donnent des limites différentes, nous avons fait le choix d'utiliser plusieurs tailles de blocs et avons combiné toutes les limites détectées. La Fig.III.9 montre les limites détecté pour deux types d'images avec différentes tailles de blocs. Nous définissons les paramètres comme suit: sigma = 3.0 et alpha = 0.25.

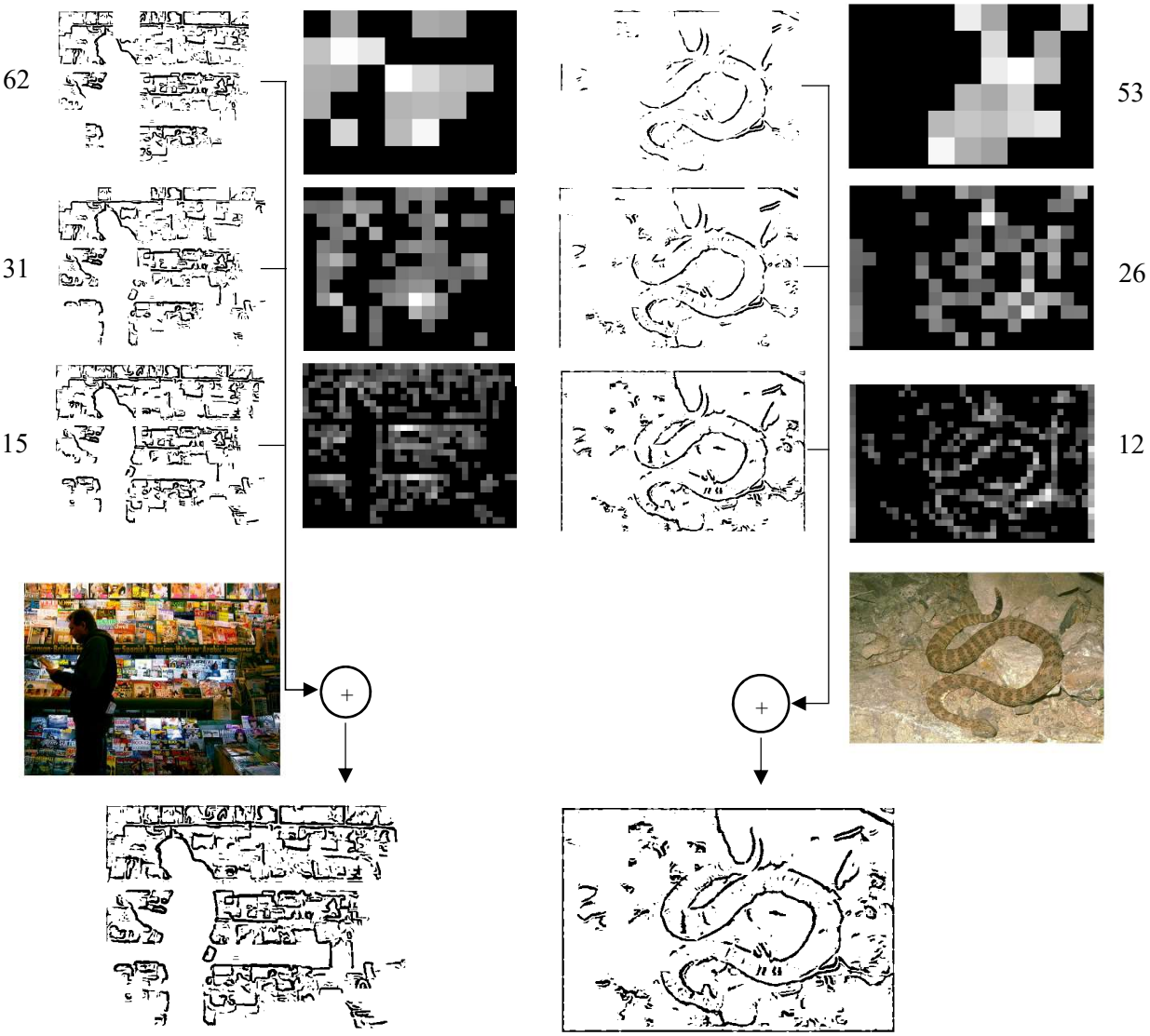


Fig. III.9. Exemple montrant les multiples bords obtenus à partir de deux images lisses et contrastées pour différentes tailles de blocs.

Nous pouvons voir à travers les résultats obtenus que pour les gros blocs, seuls certains bords dominants sont détectés. D'autre part, des bords plus détaillés sont détectés à l'aide de blocs plus petits.

Ainsi, l'idée de combiner les bords détectés pour différentes tailles de blocs est apparue comme une évidence car cela permet d'avoir dans le résultat final la plus part des bords jugé important sans perte de qualité.

Nous avons fixé le seuil de classification Th en fonction de $gMean$, gStd et Alpha. Nous avons fait ce choix car, comme indiqué dans le Tabeau III.1 et comme expliqué précédemment, nous avons considéré la surface de l'image comme une variable aléatoire. Ainsi, nous avons jugé que la moyenne et l'écart sont des mesures représentatives du changement de cette dernière.

La Fig.III.10 montre le processus de seuillage du détecteur proposé. Chaque barre de l'image (b) représente un bloc de 3x3 pixels où toutes les valeurs sont égales au std local de ses intensités. Nous pouvons voir dans l'image résultante (c) que les régions les plus lumineuses correspondent aux valeurs std's les plus élevées et par la même, aux blocs contenant le plus de changement en termes d'intensité.

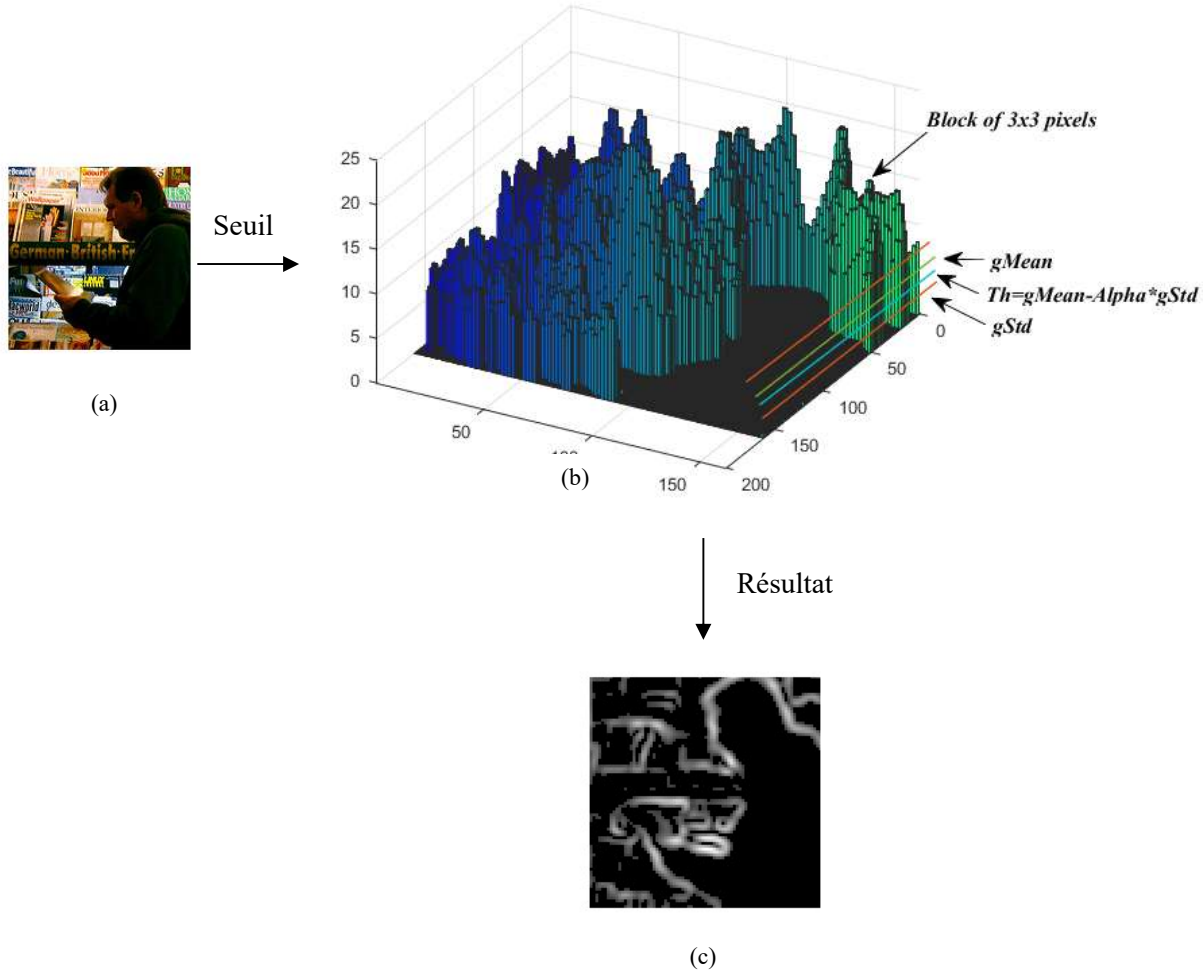


Fig. III.10. Exemple montrant le processus de seuillage utilisé par le détecteur proposé et le résultat obtenu de ce dernier.

Les résultats obtenus montrent d'une part, la performance de détection du schéma proposé et de l'autre à comprendre la nature de l'image.

Ceci le rend très efficace pour d'autres tâches de vision par ordinateur telles que la segmentation ou la compréhension du contenu de l'image.

II.3. Conclusions

Dans ce chapitre, nous avons proposé une méthode de détection des contours complètement différente car nous n'utilisons pas de fonction multiparamétrique qui sera utilisée pour modéliser des formes dans l'image en fonction de la moyenne et l'écart type au niveau pixel.

Dans notre cas, nous comparons directement le changement au niveau local (moyenne, standard) du bloc sélectionné à l'image globale (gMean et gSdt). En effet, la notion de variabilité locale et globale de l'image, associée au concept de classification par blocs dans le domaine de la détection des contours, n'a pas été proposée auparavant. Cela représente l'idée de base de ce travail.

La nouveauté, l'efficacité et la simplicité du détecteur proposé, associées à sa facilité de mise en œuvre, le rendent très attrayant pour des applications simples et étendues. Les expériences montrent que le détecteur proposé est très efficace car, en plus de la détection de bord, il fournit des informations riches sur la nature de l'image.

Il est également capable de détecter et de mettre en évidence des régions pertinentes dans l'image, qui peuvent être exploitées pour d'autres expériences dans le domaine de la segmentation. L'objectif principal de cette contribution était de présenter une méthode efficace et différente de détection des contours.

Pour les travaux futurs, nous avons l'intention d'optimiser la taille des blocs et la sélection du seuil. Nous prévoyons également de réduire le temps de calcul du détecteur proposé en le mettant en œuvre sur un processeur parallèle.

Chapitre IV

Présentation des résultats expérimentaux du descripteur de points caractéristiques et du détecteur de bords proposés

IV.1. Résultats expérimentaux du descripteur ADOCH

IV.1.1 Correspondance d'images

Nous avons comparé le schéma proposé à l'état de l'art des descripteurs, à savoir le SIFT et SURF. Nous l'avons également comparé au descripteur BRIEF, qui est exempt de l'etape de calcul et de compensation de l'orientation. Sachant que comme le nôtre, le descripteur DAISY utilise les histogrammes, nous l'avons ajouté au processus de comparaison. Enfin, nous avons comparé le descripteur proposé à un descripteur basé sur l'apprentissage [124]. Pour simplifier, nous avons nommé le descripteur proposé ADOCH, pour la différence absolue des histogrammes cumulés. Nos tests ont été réalisés sous Matlab R2015a.

Nous avons sélectionné quatre bases de données populaires pour tester le descripteur proposé. Celles-ci sont constituées de plusieurs séquences ou un nombre croissant de transformations connues sont effectuées entre la première image et le reste des images.

La première base de données est celle d'Oxford, introduite par Mikolajczyk et Schmid [125]. Elle contient des séquences d'images comprenant six à neuf images avec des changements d'orientation, d'éclairage, d'échelle et de flou. La base de données de Salzmann [126] a été utilisée pour évaluer les performances du descripteur proposé pour les objets 3D déformables. La base de données Strecha [127] a été utilisée pour le changement de vue au niveau des caméras stéréo. Enfin, nous avons testé le descripteur proposé pour le changement pur d'éclairage, la rotation de la caméra et le changement d'échelle sur la base de données de Heinly [128].

Nous avons également calculé le temps de génération du descripteur proposé en le comparant au reste des descripteurs.

Les performances des descripteurs sont fortement liées à la combinaison détecteur / descripteur. Néanmoins, le classement global de leurs performances reste le même quel que soit le détecteur sélectionné.

Dans [129], [130], les auteurs ont montré que le MSER est le meilleur détecteur de région affine-invariante en termes de précision et de répétabilité. Nous l'avons ainsi utilisé en combinaison avec le descripteur proposé. Environ 500 à 1 000 points clés ont été détectées pour tous les tests.

Nous avons choisi empiriquement la taille optimale du patch après plusieurs tests, nous avons fixés ce dernier à $s = 34$ pixels. Cela dit, une optimisation de ce paramètre pourrait faire l'objet d'autres investigations dans les travaux futurs.

Dans la partie correspondante, nous avons fixé le seuil Th à $Th = 0.25$, puisque nous estimons que les éléments résultants de l'opération de soustraction $h^j_{\theta,m} \in H_{\theta,m}$ et $h^j_{I,m} \in H_{I,m}$ qui sont inférieurs de 75% par rapport aux éléments correspondants dans les histogrammes de test $h^{1,j}_{\theta,m}$ et $h^{1,j}_{I,m}$ sont considérés comme proches de zéro. Par conséquent, nous considérons que si $h^j_{\theta,m} < 0{,}25 * h^{1,j}_{\theta,m}$ ou $h^j_{I,m} < 0{,}25 * h^{1,j}_{I,m}$, cela équivaut à une très petite différence entre les histogrammes de test et de référence. Ainsi, nous ajoutons un aux scores correspondants S_θ et S_I.

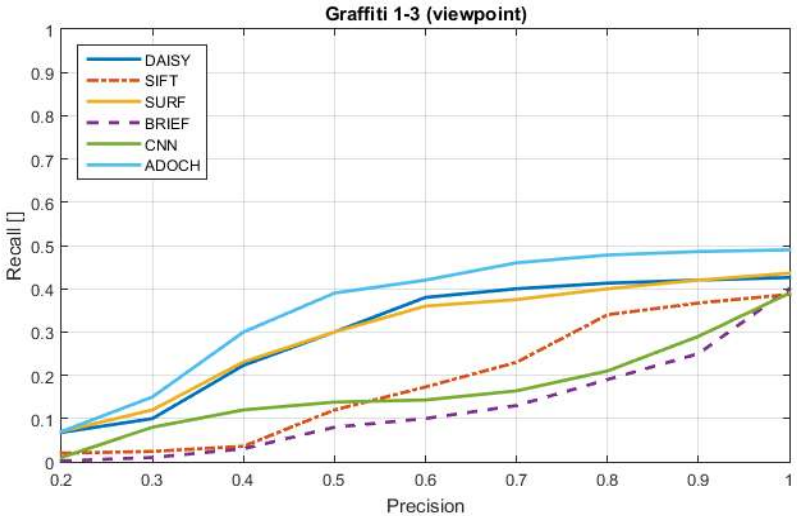
Nous avons utilisé les courbes de Recall vs 1-précision pour évaluer les performances du descripteur proposé sous différents changements telles que le flou, la luminosité, la rotation et le changement d'échelle. Tel que,

$$Recall = \frac{nombre\ de\ correspondances\ correctes}{nombre\ totale\ des\ correspondances} \qquad IV.1$$

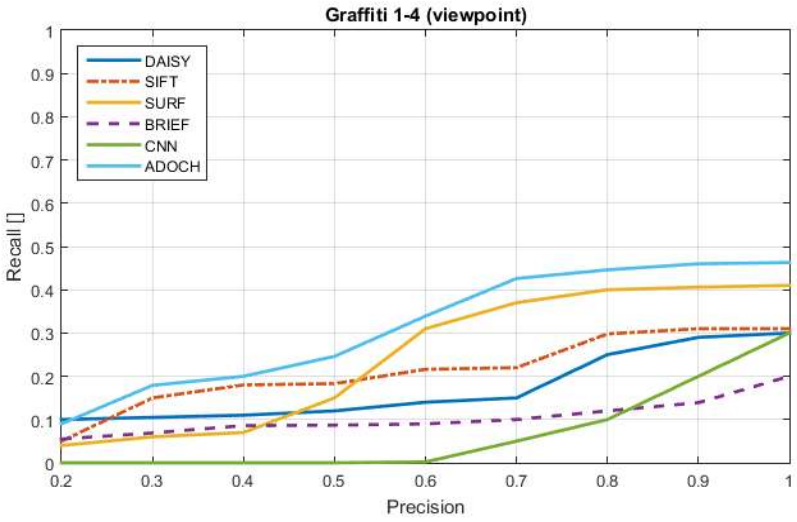
$$1 - precision = \frac{nombre\ de\ fausses\ correspondances}{nombre\ de\ correspondances\ correctes + nombre\ de\ fausses\ correspondances} \qquad IV.2$$

Nous estimons qu'une correspondance est correcte si le score de correspondance final est supérieur au seuil de similarité, que nous avons fixé a $ThM(\%) = 70\%$. Seul un petit nombre de correspondances incorrectes ont un score de similarité supérieur au seuil fixé. Néanmoins, celles-ci ne sont pas quantifiées comme de véritables correspondances.

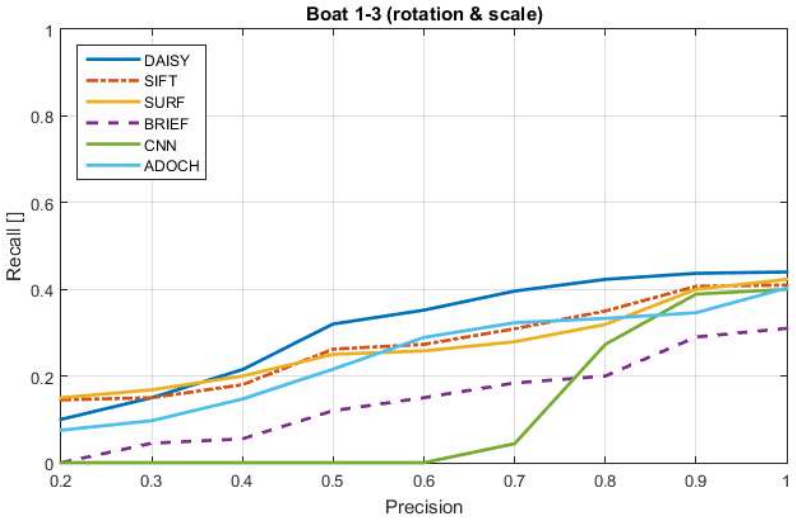
La figure 10 montres les performances du descripteur proposé sur la base de données (Oxford), les résultats obtenus illustrent la résistance du descripteur proposé à différents types de changements.



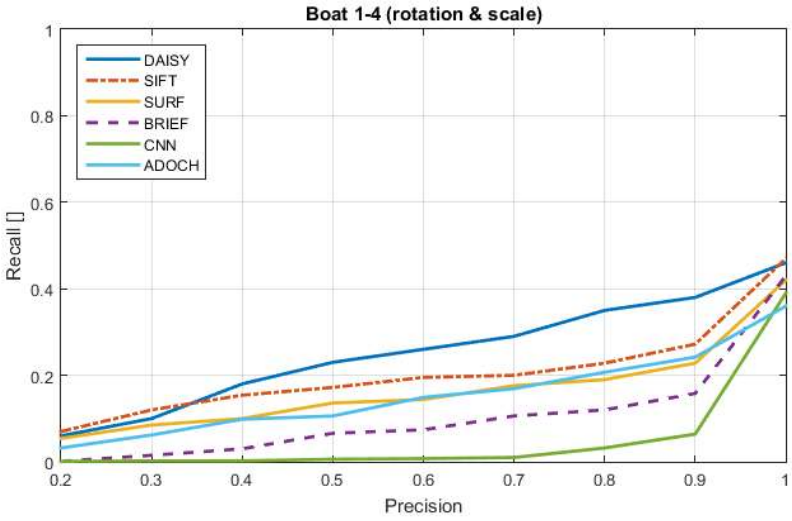
(a)



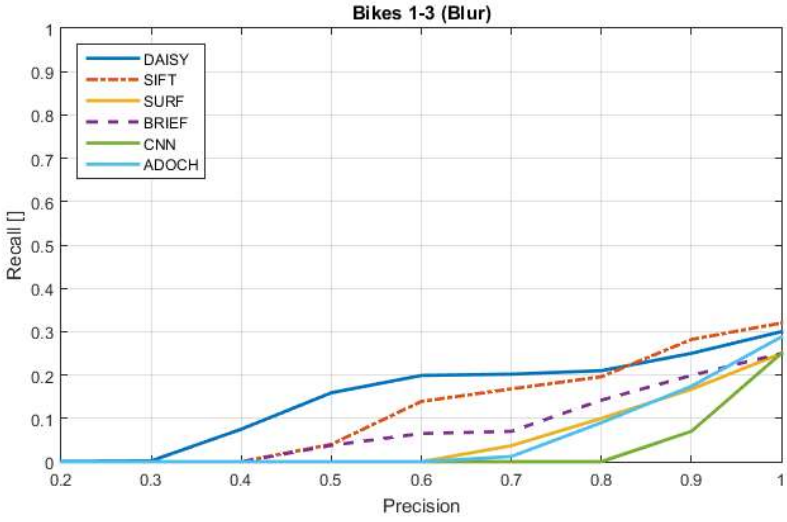
(b)


Boat 1-3 (rotation & scale)
DAISY
SIFT
SURF
BRIEF
CNN
ADOCH
Recall []
Precision
0
0.2
0.4
0.6
0.8
1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

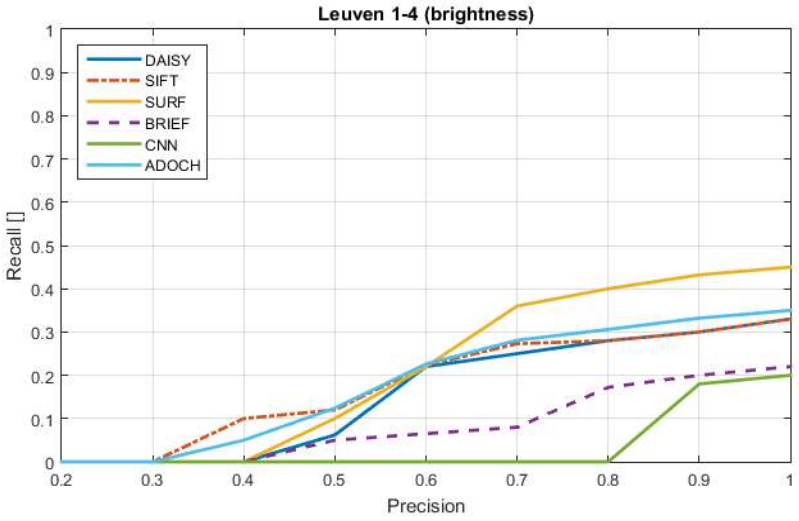
(c)


Boat 1-4 (rotation & scale)
DAISY
SIFT
SURF
BRIEF
CNN
ADOCH
Recall []
Precision
0
0.2
0.4
0.6
0.8
1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

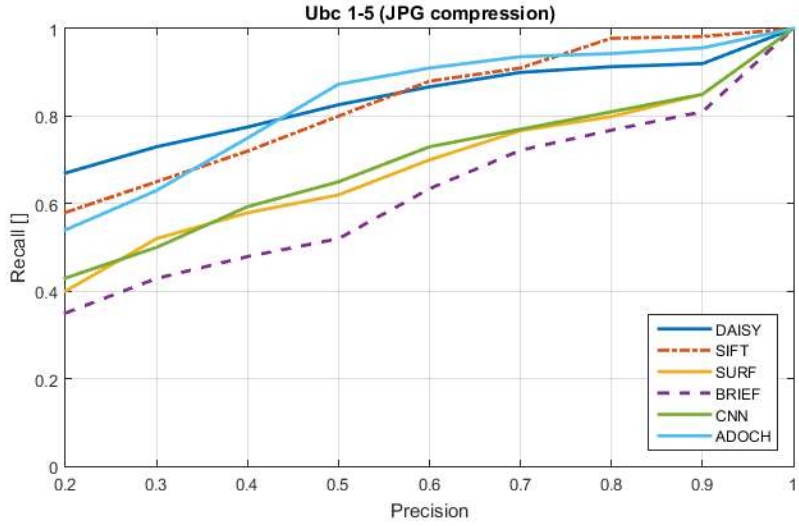
(d)


Bikes 1-3 (Blur)
DAISY
SIFT
SURF
BRIEF
CNN
ADOCH
Recall []
Precision
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(e)



(f)

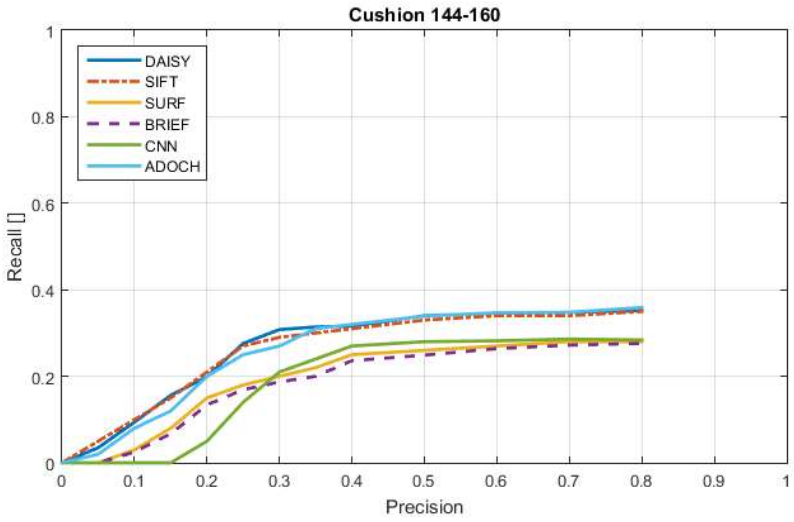


(g)

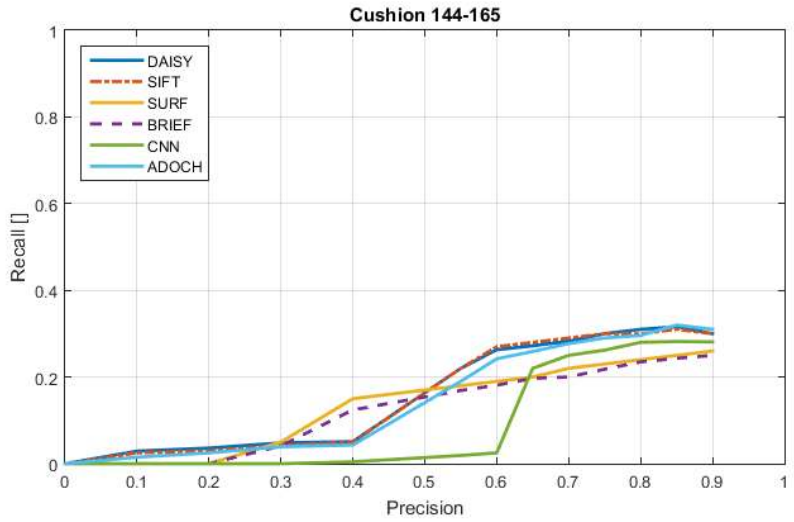
Fig. IV.1 Performances du descripteur proposé sur l'ensemble de données d'Oxford pour des changements de points de vue (a-b), rotation et échelle (c-d), flou (e), luminance (f) et compression JPEG (g).

Même si la sensibilité de l'amplitude du gradient au changement du flou a tendance à affecter négativement les performances du descripteur proposé. Pour les changements de point de vue et de compression JPEG, les performances du descripteur proposé sont élevées. Les performances de ce dernier sont également bien pour des changements de l'illumination ainsi que de la rotation et de l'échelle.

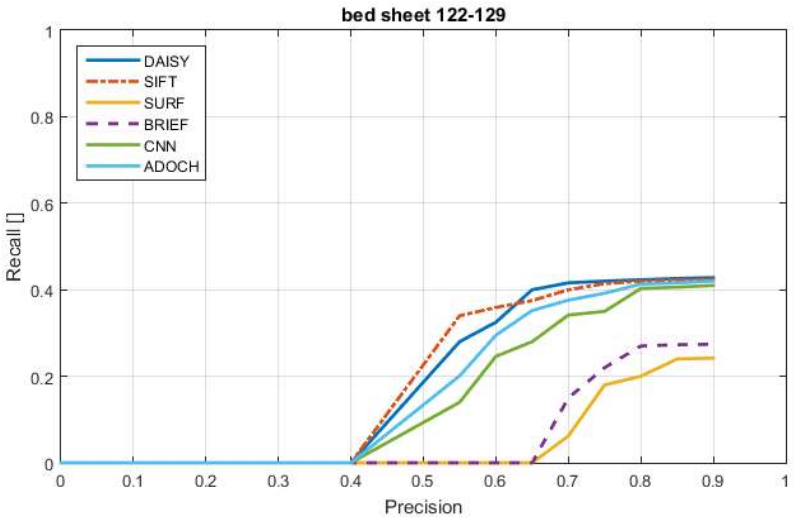
La Fig.IV.2 montre les performances du descripteur proposé dans le cas d'objets déformables en 3D.


Cushion 144-160
DAISY
SIFT
SURF
BRIEF
CNN
ADOCH
Recall []
Precision
0
0.2
0.4
0.6
0.8
1
0.1
0.3
0.5
0.7
0.9

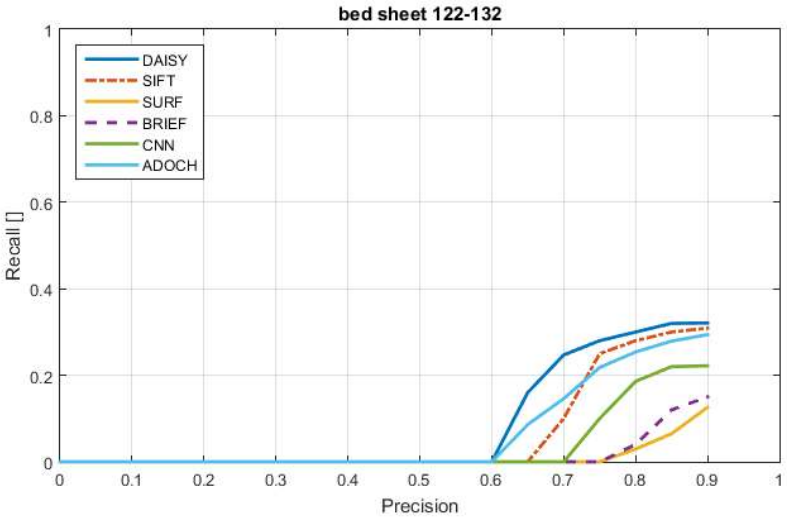
(a)


Cushion 144-165
DAISY
SIFT
SURF
BRIEF
CNN
ADOCH
Recall []
Precision
0
0.2
0.4
0.6
0.8
1
0.1
0.3
0.5
0.7
0.9

(b)


bed sheet 122-129
DAISY
SIFT
SURF
BRIEF
CNN
ADOCH
Recall []
Precision
0
0.2
0.4
0.6
0.8
1
0.1
0.3
0.5
0.7
0.9

(c)



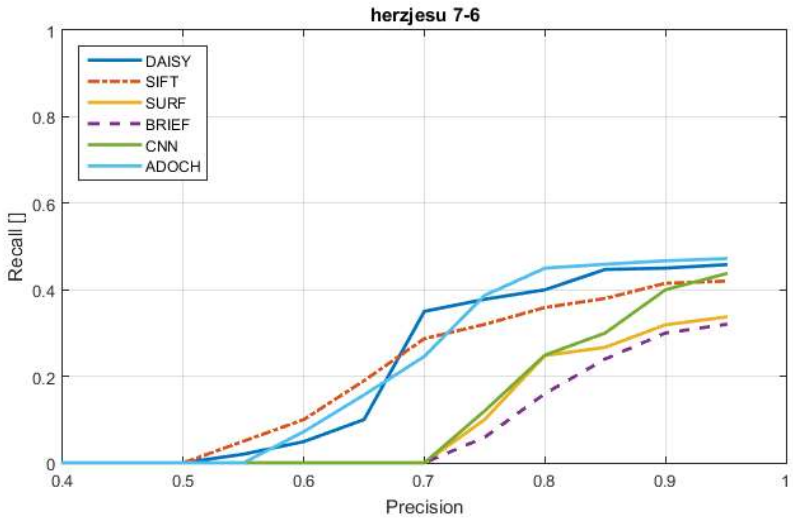
(d)

Fig.IV.2 Les Performances du descripteur proposé sur la base de données de Salzmann pour les objets 3D déformables.

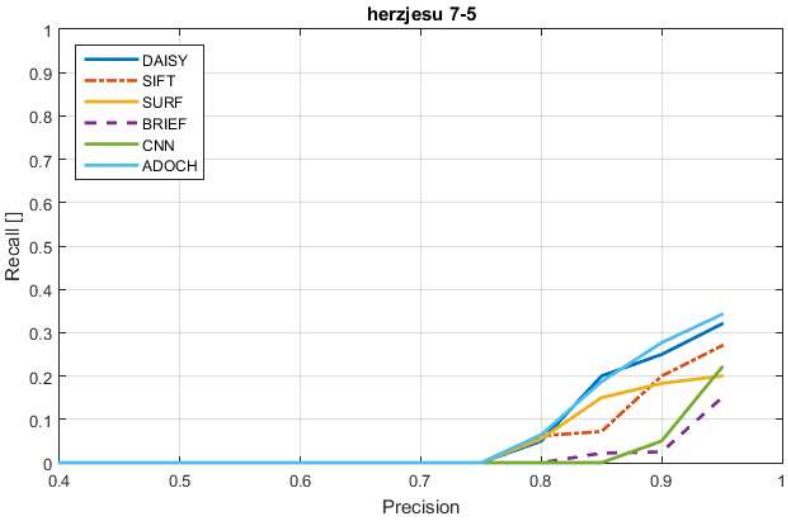
Dans le cas d'objets déformables 3D, la distribution des bords au niveau du patch n'est pas trop affectée et l'aspect fort du descripteur proposé est précisément sa capacité de résister à ce genre de changements car celui-ci est basé sur des histogrammes contenant toutes les informations importantes autour des bords du patch. Cette propriété le rend très résistant à ce genre de changements.

Il en va de même pour la Fig.IV.3, qui montre les résultats du descripteur proposé pour le changement de vue multiple. Les résultats obtenus sont extrêmement satisfaisants et même meilleurs que certains descripteurs de l'état de l'art.

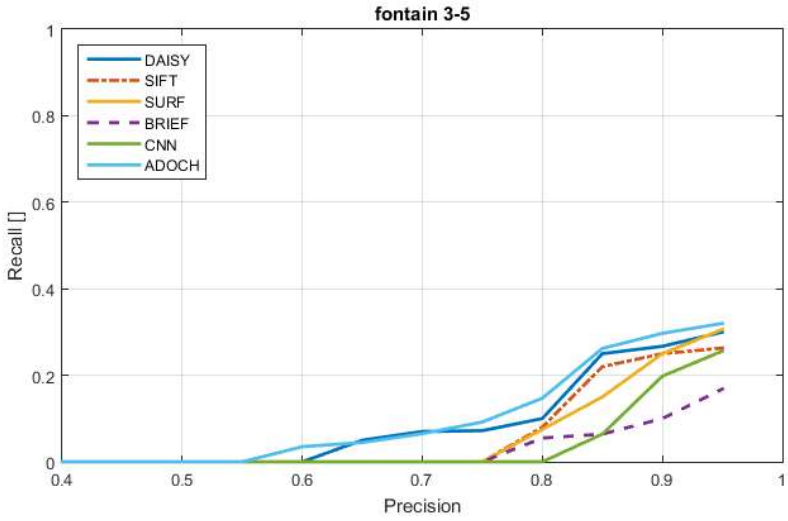
De plus, il convient de noter dans ce cas que le descripteur proposé n'a besoin d'aucune phase de prétraitement, telle que la création de modèles dans le cas de descripteurs binaires, la phase d'apprentissage pour les descripteurs CNN ou le calcul et la compensation d'orientation pour les descripteurs basés sur les histogrammes.



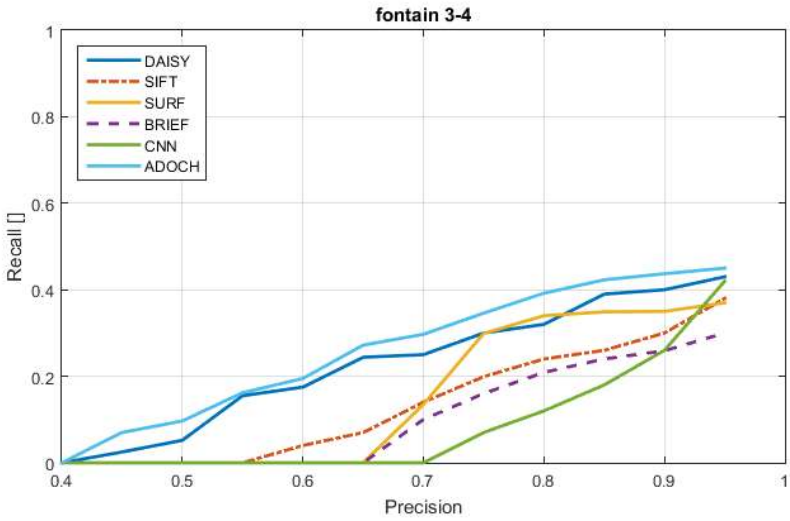
(a)



(b)



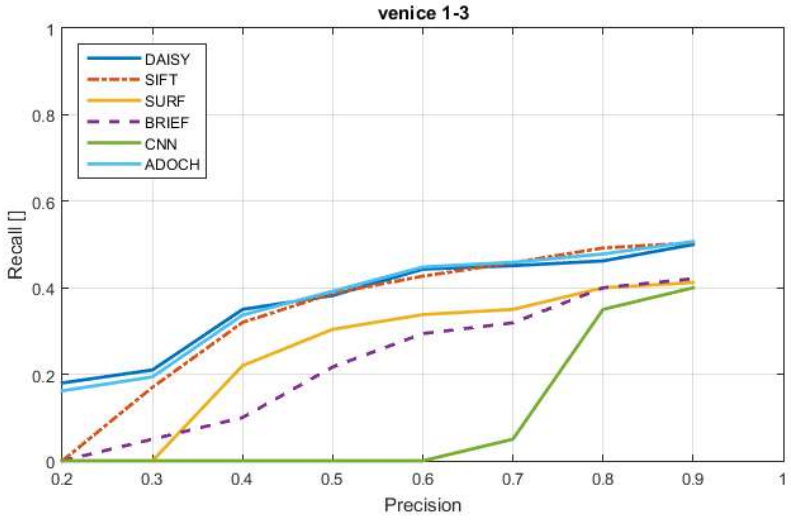
(c)



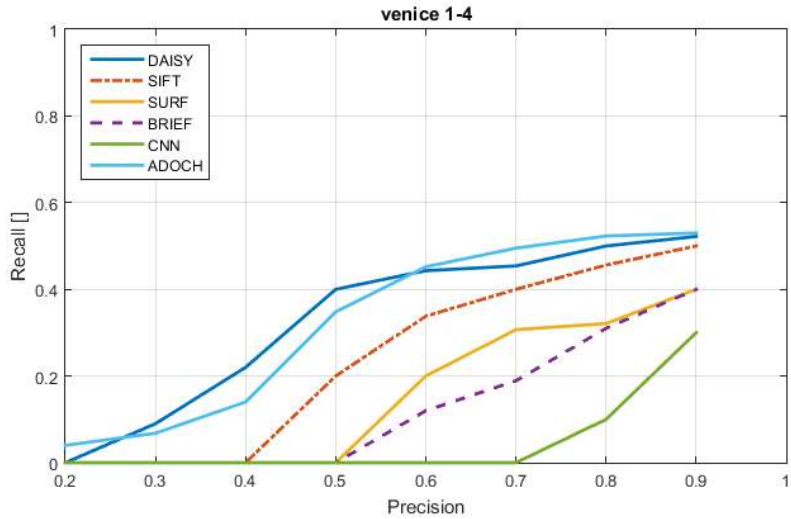
(d)

Fig.IV.3 Exemple montrant les performances du descripteur proposé obtenu à partir de la base de données de Strecha pour le cas multiview

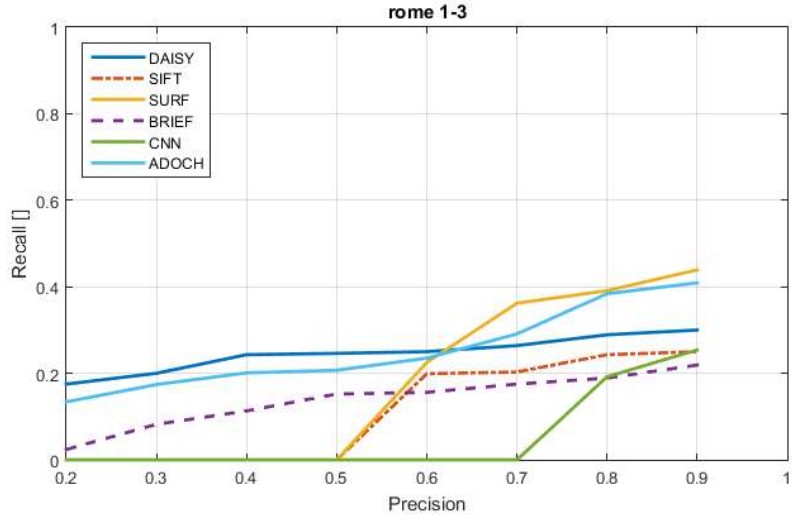
Enfin, nous avons testé le descripteur proposé pour les changements pur d'échelle et de l'orientation sur la base de données de Heinly, comme le montre la Fig.IV.4. Ou nous constatons que le descripteur proposé fonctionne extrêmement bien.



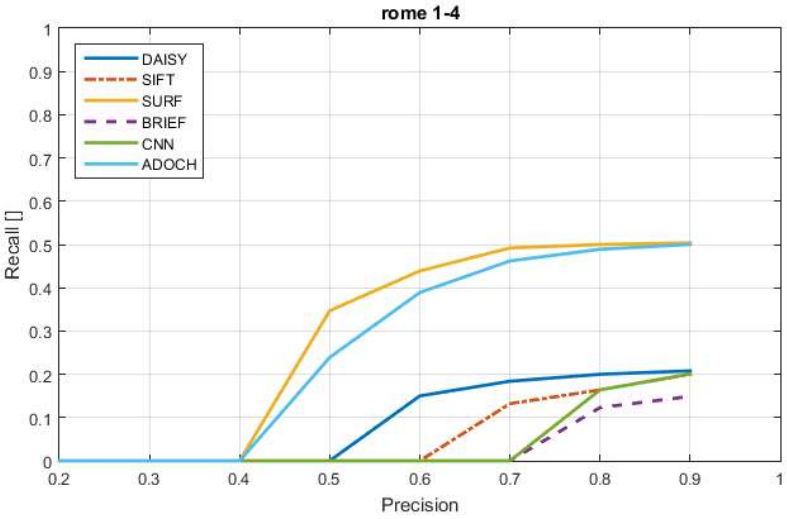
(a)



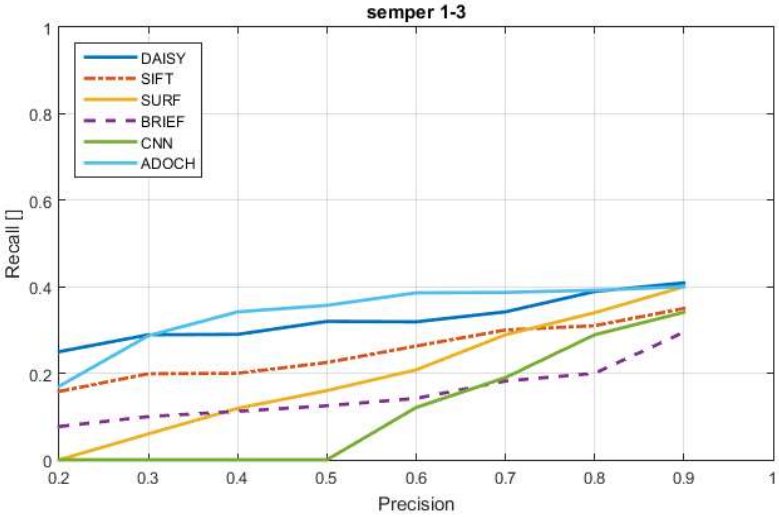
(b)



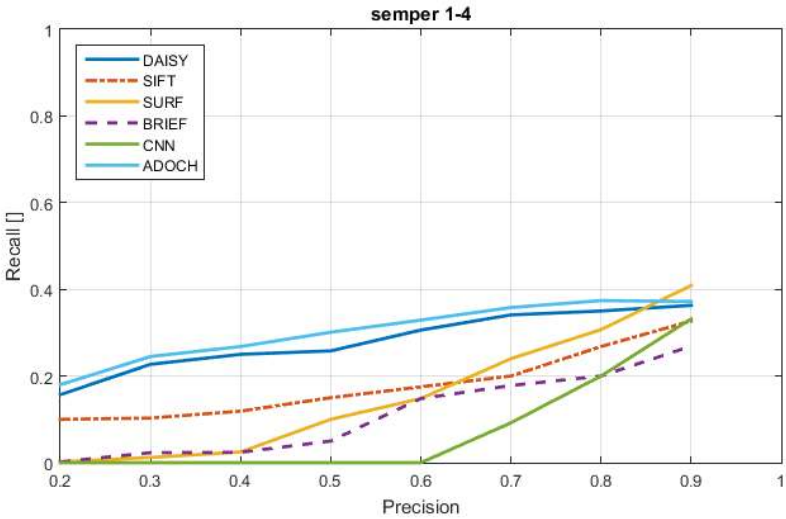
(c)



(d)



(e)



(f)

Fig.IV.4 Les Performances du descripteur proposé sur le jeu de données de Heinly pour des changements pur d'échelle (a-b) et d'orientation (c-d-e-f) pur.

Cela confirme non seulement notre première intuition pour la propriété d'invariance du descripteur proposé au changement de la rotation, mais montre également sa résistance au changement d'échelle pur.

Dans la Fig.IV.5, nous avons résumé les performances quantitatives de tous les descripteurs en utilisant le score F (F_β) à partir des courbes de précision vs rappel, comme dans [131], nous avons utilisés dans notre cas $F_{0.5}$.

Nous pouvons clairement voir que le descripteur proposé surpasse le reste des descripteurs dans pour les changements de points de vue et de vues multiples. Ses résultats sont également élevés pour les objets déformables.

Pour le changement de rotation et d'échelle, le descripteur proposé est moins compétitif. Cela dit, notre descripteur est extrêmement résistant au changement d'échelle et de la rotation pur.

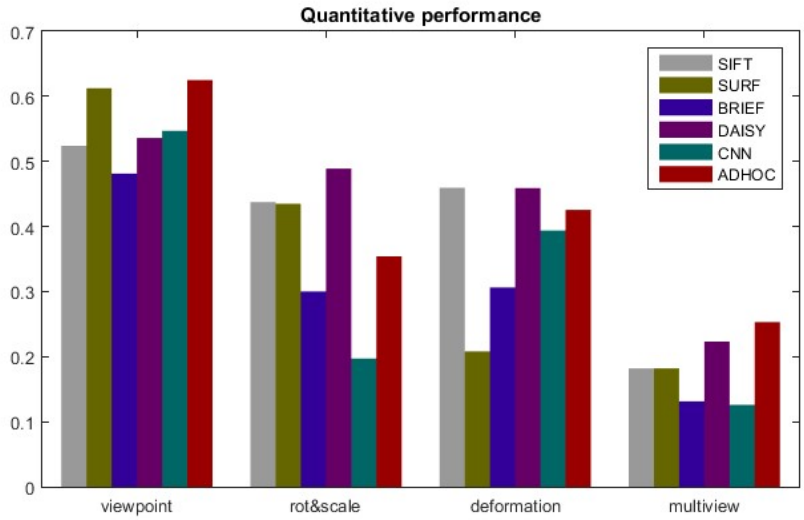


Fig.IV.5 Performances quantitatives des différents descripteurs en utilisant le F-score à 50%.

Les figures IV.6 et IV.7 montrent quelques résultats visuels du descripteur proposé sur les quatre bases de données proposés précédemment.

Nous pouvons clairement voir les performances du descripteur proposé sous différents types de changements.

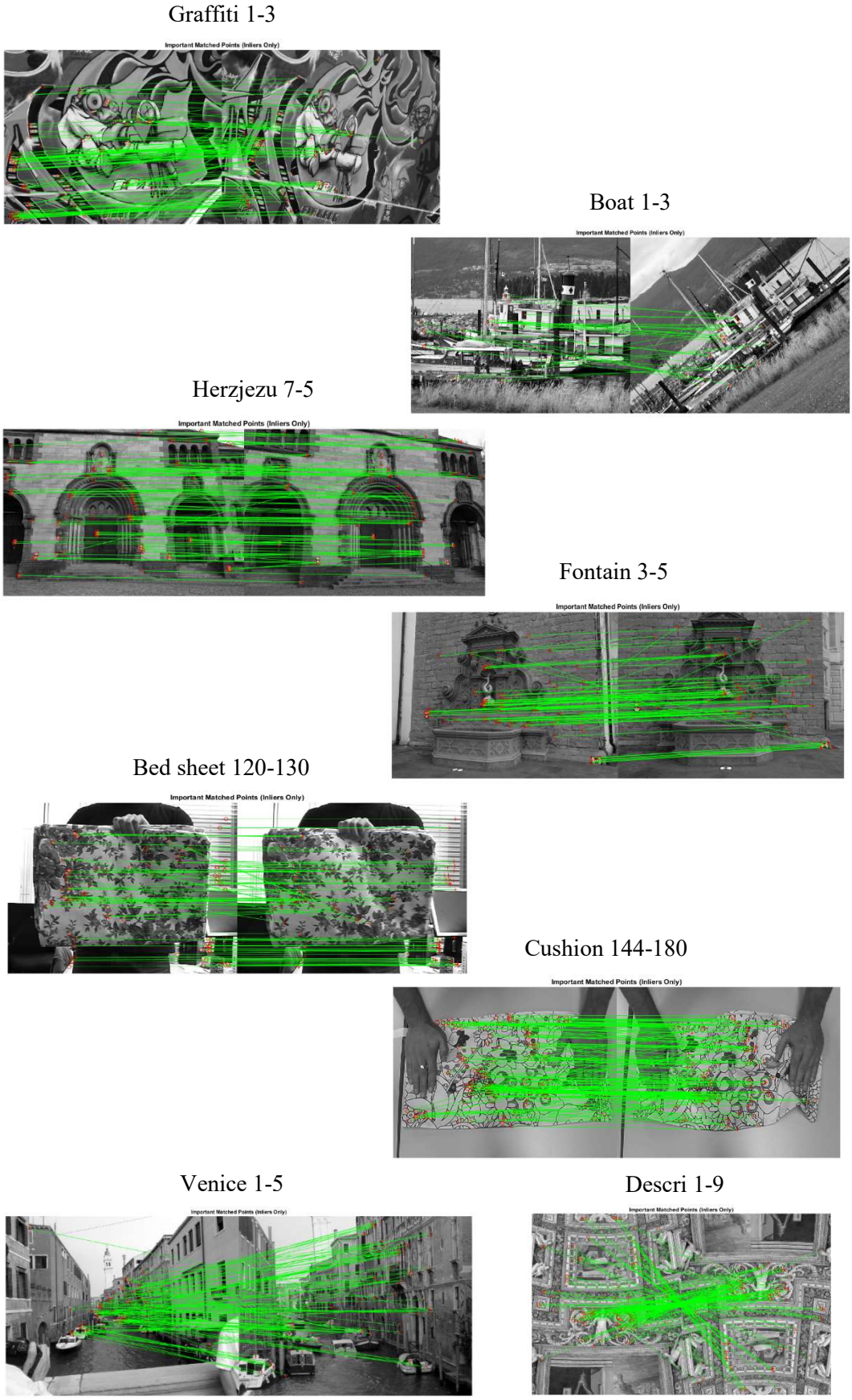


Fig. IV.6 Quelques performances visuelles du descripteur proposé sur les quatre bases de données proposés**.**

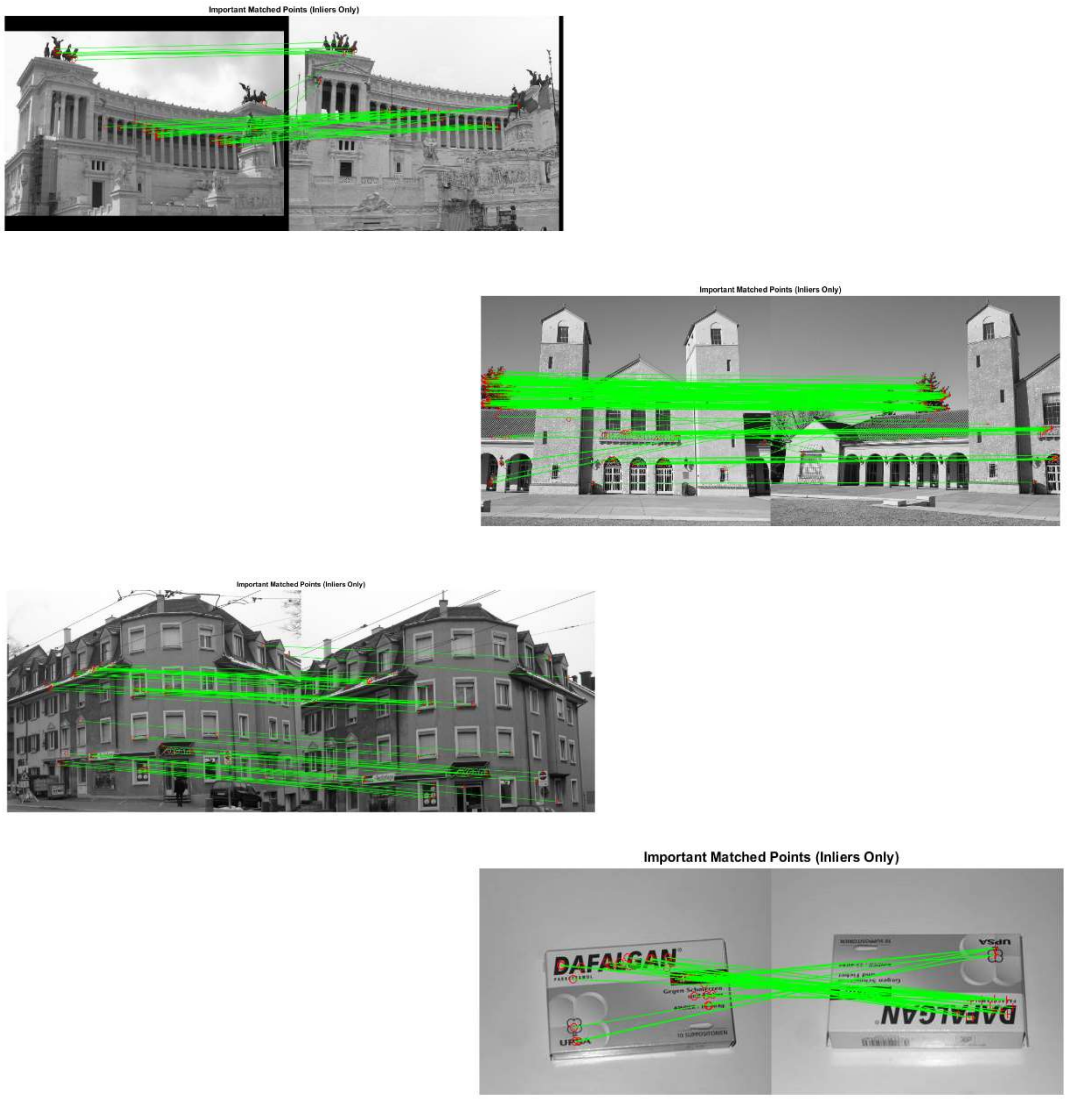


Fig. IV.7 Autres exemples montrant les performances du descripteur proposé sur des images du monde réel**.**

IV.1.2 Reconnaissance d'objets

Pour notre dernière investigation, nous avons testé le descripteur proposé sous le contexte de la détection d'objets en utilisant l'étape de pré-élimination des fausses correspondances.

Nous avons testé le descripteur proposé sur deux ensembles de données disponibles publiquement, à savoir la base de données 53 Objects [132] et Home Objects 06 [133] composés de plusieurs objets avec des déclinaisons différentes pour chacun d'entre eux.

Nous avons également testé le descripteur proposé sur notre base de données, constitué d'images du monde réel avec une résolution d'environ huit millions de pixels par image, ou nous avons sélectionnée dix-neuf objets que nous avons soumis à différentes modifications de la luminance, de l'échelle et de l'orientation. Des exemples d'images de celui-ci sont illustrés à la Fig.IV.8.

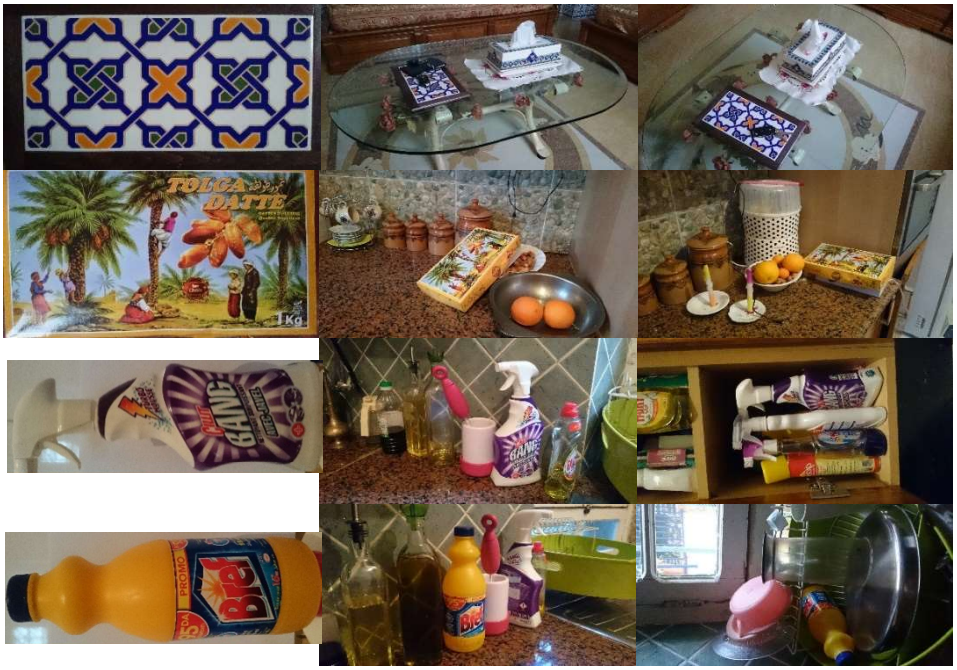


Fig.IV.8 Exemples d'images de la base de données d'objet de maison proposé où l'on voit quatre objets avec deux images de test pour chacun d'eux.

Afin de tester l'efficacité de la phase de pré-élimination, nous avons calculé le taux de précision du descripteur proposé, avec et sans cette étape :

$$\begin{cases} Taux\ de\ Precision\ Par\ Objet = \dfrac{Nombre\ d'objets\ correctement\ reconnus}{Nombre\ Totale\ d'objets} \\ Taux\ de\ Precision\ Total\ (\%) = Moyenne\ (Precision\ Par\ Objet) * 100 \end{cases} \qquad IV.3$$

Le taux de précision d'objet (TPO) reflète tous les cas où l'objet est correctement détecté sous ses différentes déclinaisons. Le taux de précision total (TPT) correspond à la moyenne de tous les taux obtenus pour tous les objets de l'ensemble de données.

Ces taux sont calculés avec et sans la phase de pré-élimination. Les résultats obtenus sont résumés dans le Tableau IV.1, montré ci-dessous.

Base de données	TPT sans pré-élimination	TPT avec pré-elimination
Home Objects	46.425 %	**52.361 %**
53 Objects	32.913 %	**43.785 %**
Our dataset	45.325%	**53.784%**

Tableau IV.1 Le taux de précision totale obtenus pour différentes bases de données avec et sans la phase de pré-élimination des fausses correspondances.

Les résultats montrent une amélioration d'environ 10% du taux de précision total avec l'utilisation de la phase de pré-élimination, ce qui est considérable en termes de précision. La modularité de cette étape est son principal avantage, car elle peut être ajoutée à n'importe quel descripteur et peut être activée pour la détection d'objet et désactivée ailleurs.

La Fig.IV.9 illustre quelques résultats visuels du descripteur proposé avant et après la phase de pré-élimination,

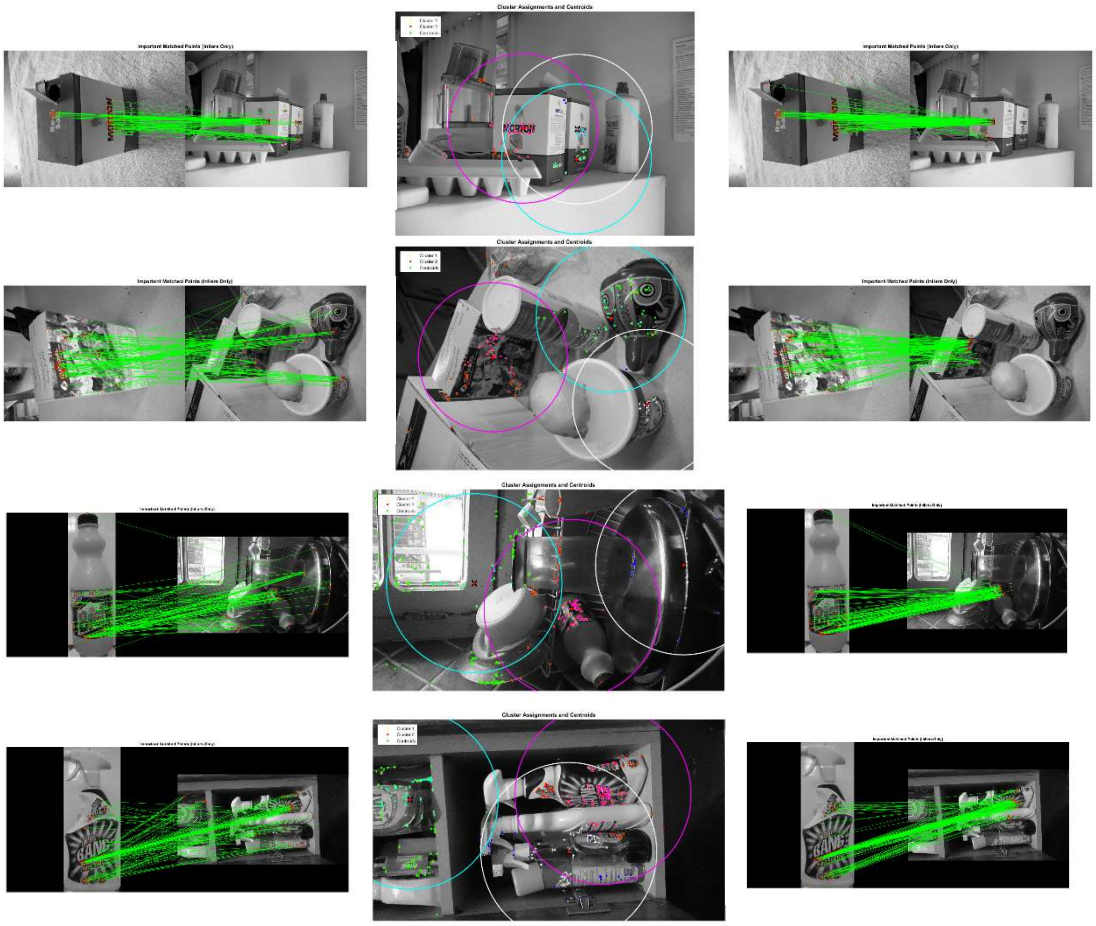


Fig. IV.9 Illustration visuelle montrant quelques exemples de détection d'objet avant et après la phase de pré-élimination.

Nous pouvons clairement voir qu'une réduction considérable des fausses correspondances a été réalisée après l'opération de regroupement.

Le Tableau IV.2 présente une comparaison du temps de calcul des descripteurs. À titre de référence, nous avons pris le temps de calcul des descripteurs traités sur des machines similaires à celle utilisé pour l'obtention de nos résultats. Le tableau contient le temps de calcul par point clé. Toutes les expériences présentées ont été exécutées sur Intel Core i7-2720QM à 2,2 GHz, 16 Go de RAM.

Nom	Le temps de génération du descripteur (ms)
ADOCH	**0.86**
SIFT	6.156 [11], 2.5 [12], 2.071 [46]
SURF	1.4 [12], 0.67 [27], 0.81 [46]
BRIEF	0.046 [46]
DAISY	0.012 [46]

Tableau IV.2 Le temps de calcul des différents descripteurs (par point clé).

Les résultats obtenus montrent que la durée de génération du descripteur proposé est inférieure à SIFT et proche du descripteur SURF. Le descripteur BRIEF est plus rapide, car le temps de création du modèle d'échantillonnage n'est pas inclus dans ce cas. Le descripteur DAISY est également plus rapide que le nôtre.

Il est important de noter que le calcul et de compensation de l'orientation ne sont pas inclus dans le cas des descripteurs SIFT, SURF et DAISY, alors que cette étape est ignorée pour le descripteur proposé. Si nous examinons les étapes de constitution du descripteur proposé, nous pouvons clairement conclure qu'elles sont principalement indépendantes. Par exemple, les histogrammes peuvent être traités indépendamment, ce qui rend le descripteur facile à mettre en parallèle. Par conséquent, d'autres améliorations dans la réduction du temps de calcul sont attendues.

Les expériences ont montré que le descripteur proposé est efficace pour les changements élevés de l'orientation, du point de vue, des déformations 3D et la compression JPEG. Il fonctionne également bien en cas du changement de flou et de l'échelle. Ces propriétés le rendent très attrayant pour les applications à grande utilisation telles que les caméras de vision stéréoscopique Multiview ou de surveillance, compte tenu de sa facilité de mise en œuvre.

IV.2. Résultats expérimentaux du détecteur de bords proposé

Nous avons testé le détecteur proposé sur la base de données largement utilisé BSDS500 [35], celle-ci est en fait la base la plus complète et la plus utilisé dans le domaine de la détection de bords. Elle est composée de 200 images dédiées à l'apprentissage, 100 images de validation et 200 images de test.

Nous avons également testé le détecteur proposé sur la base de données Multicue, proposé dans [134] par Mély et al. Elle consiste en 100 courtes séquences de scènes naturelles binoculaires, obtenues à partir d'une caméra stéréo. Chaque séquence contient 10 images de gauche à droite de la scène. Les sujets humains étiquettent la dernière image de gauche dans chaque séquence.

Enfin, nous avons appliqué le détecteur proposé à l'ensemble de données PASCAL-Context VOC 2012 [135], qui est également une base données très utilisé dans le domaine de la vision par ordinateur.

Pour évaluer le détecteur proposé, nous avons utilisé la métrique de rappel de précision (precision-recall), car elle est la plus utilisée pour la segmentation d'images [129/135] :

$$\begin{cases} Precision = \dfrac{L'ensembl\ de\ bords\ correctement\ détecté}{L'ensemble\ de\ bords} = \dfrac{\left|S \cap (\cup_{i=1}^{M} G_i)\right|}{|S|} \\ Rappel = \dfrac{L'ensemblede\ bords\ correctement\ détecté}{L'ensemble\ de\ bords\ annoté\ par\ l'humain} = \dfrac{\sum_{i=1}^{M}|S \cap G_i|}{\sum_{i=1}^{M}|G_i|} \\ F = \dfrac{2PR}{R+P} \end{cases} \qquad \boldsymbol{IV.4}$$

où S représente les limites générées par la machine et l'ensemble M les limites d'annotation humaines. ∩ est simplement l'opérateur d'intersection entre les ensembles de limites.

La mesure F sert à évaluer les performances générales du détecteur proposé pour un seuil de contour fixe (SCF), qui correspond à $Th = gMean - 0.25 * gStd$ et un seuil variable par image (SVI) où Alpha peut varier afin d'obtenir le meilleur résultat de détection d'une image à une autre.

Enfin, la précision moyenne (PM) est simplement la moyenne des précisions maximales mesurées pour les valeurs de rappel allant de 0 à 1.0 :

$$AP = \frac{1}{11}\sum_{Rappel} Precision\ (Rappel_i) \qquad \textbf{\textit{IV.5}}$$

Nous avons également calculé le temps d'exécution du détecteur proposé afin d'évaluer son efficacité en termes de rapidité.

Veuillez noter que pour nos expériences, nous avons appliqué le détecteur proposé aux images de test fournies dans la base de données de référence BSDS. Nous avons également utilisé les annotations faites par les humains de celle-ci.

Nous avons considéré tous les pixels détectés qui se croisent ou ont une distance de plus ou moins un pixel dans les deux directions x et y avec les limites de vérité comme étant correcte.

La taille des blocs "s" est fonction de la résolution de l'image, nous avons utilisé quatre tailles de bloc dans le processus de détection. Le bloc initial est égal au tiers de la valeur minimale entre les lignes et les colonnes de l'image. Les blocs suivants sont simplement le demi, le quart, le sixième et le huitième de la taille du premier bloc, comme indiqué ci-dessous.

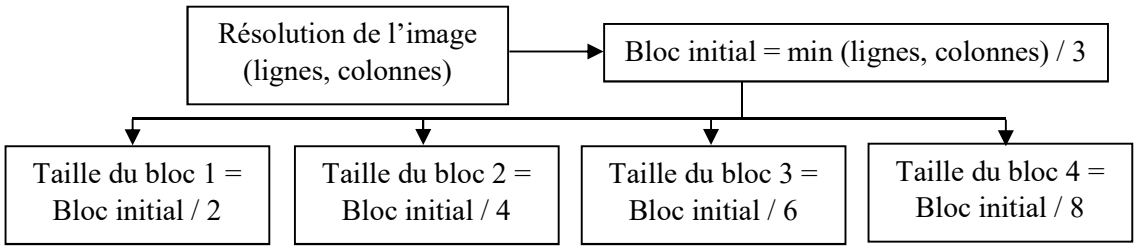


Fig. IV.10 Schémas bloc montrant le processus de sélection de la taille des blocs qui seront utilisé dans l'opération de classification et de détection des contours.

Il est important de noter que le nombre de blocs dans l'image dépend de leur taille. Ainsi, le nombre des blocs de grande taille est moins élevé que dans le cas de petits blocs. Le choix de quatre tailles de blocs a été fait de manière empirique, car nous avons effectué de nombreux tests sur les bases de données citées précédemment et les meilleurs résultats ont été obtenus à partir de ces valeurs. Cela dit, le choix de la taille des blocs pourrait faire l'objet d'investigations plus approfondi dans les travaux futurs, car il s'agit d'un paramètre important qui a une influence significative sur le résultat final du détecteur proposé.

Tel que nous l'avons expliqué dans le chapitre III, la valeur du seuil est fonction de $gMean$ et $gStd$. Si on considère que la partie la plus importante de la variabilité du signal est délimitée dans l'intervalle [$gMean$ - $gStd$, $gMean$ + $gStd$], nous avons choisi de fixer le paramètre Alpha à 0,25 pour tous les blocs et résolutions d'image. Ainsi, nous obtenant un seuil $Th = gMean - 0.25 * gStd$, ceci signifie que nous conservons 62,5% de la variabilité du

signal dans chaque bloc et le reste est ignoré. Nous estimons que ce pourcentage est suffisamment représentatif du changement de la surface du bloc. De même, nous garantissons que les bords les plus importants dans le bloc sont sélectionnés. Enfin, le paramètre de flou gaussien a été fixé pour toutes nos expériences à Sigma = 3,0. Nous avons considéré cette valeur comme appropriée car elle lisse suffisamment la surface de l'image tout en préservant ses bords importants.

La Fig. IV.11 montre quelques résultats visuels du détecteur proposé sur la base de données BSDS500 par rapport à ceux obtenus avec le détecteur Canny, nous avons respectivement fixé les paramètres de ce dernier à sigma= 3.0 et un seuil égale à 0.1. Nous pouvons effectivement constater que les bords produits en utilisant l'approche proposé sont quantitativement moins mais qualitativement plus importants que ceux du détecteur Canny.

Fig. IV.11 Illustration des bords obtenus en utilisant la méthode proposée par rapport au détecteur Canny. La première colonne contient l'image originale, la deuxième colonne contient les contours crée par l'humain suivis par les bords détecté par Canny et les résultats du détecteur proposé.

Nous avons résumé les résultats obtenus avec le détecteur proposé par rapport aux détecteurs de l'état de l'art sur la base de données BSDS500 dans le Tableau IV.3. Les résultats obtenus pour un seuil de contour fixe (SCF), un seuil variable par image (SVI) et la moyenne de précision (MP) montrent que le détecteur proposé est à la fois très performant. Nous avons également mesuré le temps de calcul du détecteur proposé en image par seconde (IPS), les résultats obtenus confirment la rapidité du détecteur proposé. Ceci est expliqué par le fait que le détecteur proposé est peu complexe.

	SCF	**SVI**	**MP**	**FPS**
Human	.80	.80	-	-
Our	.765	.782	.807	2.0
Canny	.600	.640	.580	15
Felz-Hutt [137]	.610	.640	.560	10
BEL [33]	.660	-	-	0.1
Mean Shift [136]	.598	.645	.565	0.2
gPb-owt-ucm [38]	.726	.757	.696	$4.16.10^{-3}$
Sketch Tokens [39]	.727	.746	.780	1
SCG [101]	.739	.758	.773	$3.57.10^{-3}$
SE-Var [40]	.746	.767	.803	2.5
DeepNet [103]	.738	.759	.758	1/5
DeepEdge [104]	.753	.772	.807	10^{-3}
HDE [45]	.788	.808	.840	2.5
OEF [23]	.749	.772	.817	-
DeepContour [42]	.756	.776	.797	$3.33.10^{-2}$

Tableau IV.3 Les performances du détecteur proposé en termes de résultats et de temps de calcule par rapport aux détecteurs de l'état de l'art.

Pour une première tentative, les résultats expérimentaux montrent l'efficacité du schéma proposé par rapport aux détecteurs de l'état de l'art. Nous pouvons constater la similarité des résultats obtenus par rapport à certains détecteurs tels que SE [40], OEF [23] ou encore DeepEdge [104], le détecteur HDE [45] est plus efficace.

Les bords et les limites dans leur sens strict sont différents en termes de perception. Dans la mesure où une limite est considérée comme des pixels entourant des objets significatifs, les arêtes sont représentées par des pixels au niveau desquels ce produises des changements de couleur ou de luminance. Ainsi, pour le test du détecteur proposé sur la base de données Multicue, nous suivrons le même raisonnement que ces auteurs Mély et al. [134] ainsi que les auteurs de HDE [45], où nous avons divisées de manière aléatoire les images marquées par l'homme en 80 échantillons dédié à la phase d'apprentissage et 20 échantillons de test. Le résultat final est considéré comme la moyenne des scores obtenus après trois essais indépendants.

Nous avons conservé les mêmes paramètres pour le détecteur proposé que dans le cas de la base de données BSDS500. Les résultats obtenus sont indiqués dans le Tableau IV.4 et

certains résultats visuels du détecteur proposé comparés au détecteur HED sont illustrés par la Fig.IV.12.

	SCF	SVI	PM
Human-Boundary [134]	.760 (.017)	-	-
Multicue-Boundary [134]	.720 (.014)	-	-
HED-Boundary [45]	.756 (.011)	.822 (.008)	.869 (.015)
Our-Boundary	.724 (.008)	.798 (.004)	.814 (.012)
Human-Edge [134]	.750 (.024)	-	-
Multicue-Edge [134]	.830 (.002)	-	-
HED-Edge [45]	.851 (.014)	.864 (.004)	.890 (.007)
Our-Edge	.832 (.007)	.857 (.005)	.853 (.004)

Tableau IV.4 Les performances du détecteur proposé par rapport aux détecteurs de Mély et al. [130] et HED [45] sur la base de données Multicue [134].

Comme pour le cas de la base de données BSDS500, le détecteur HED présente de meilleurs résultats en termes de bords détecté. Ceci dit, le détecteur proposé présente une fluctuation peu importante sur les résultats de la précision moyenne par rapport au détecteur HED.

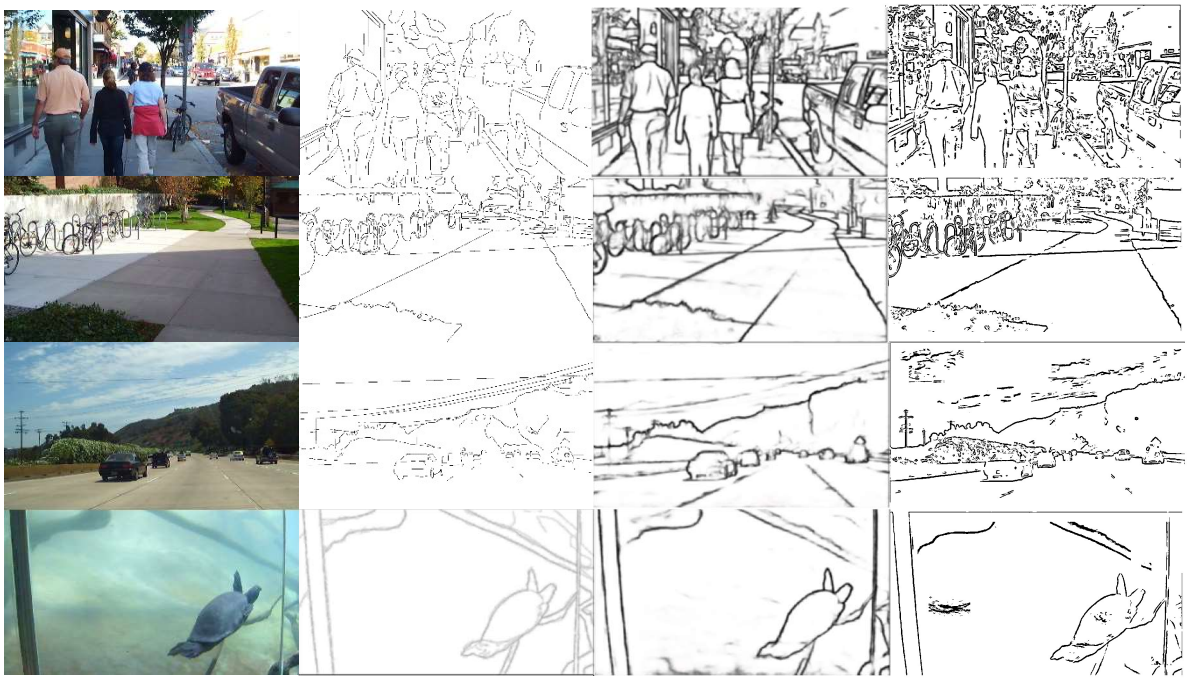


Fig. IV.12 Illustration visuel des résultats du détecteur proposé comparée à celles du détecteur HED. Les première et deuxième colonnes représentent les images originales et les bords annotés par l'humain. La troisième colonne contient les bords détectés par le HED [45], enfin la dernière colonne contient les résultats de détection.

Les Fig.IV.13, 14 et 15 montrent quelques résultats visuels du détecteur proposé comparés aux détecteurs Sketch Tokens, DeepEdge et gPb, où nous constatons que le détecteur proposé est très efficace.

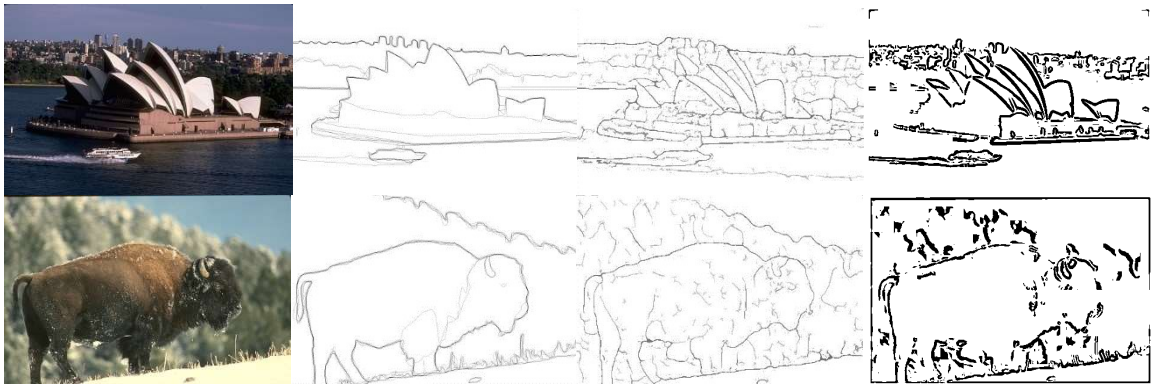


Fig. IV.13 Les performances du détecteur proposé comparées à celles du détecteur Sketch Tokens. Les première et deuxième colonnes représentent les images d'origine et les bords annotés par l'humain. La troisième colonne contient les résultats de Sketch Tokens [39] suivis des résultats du détecteur proposé.

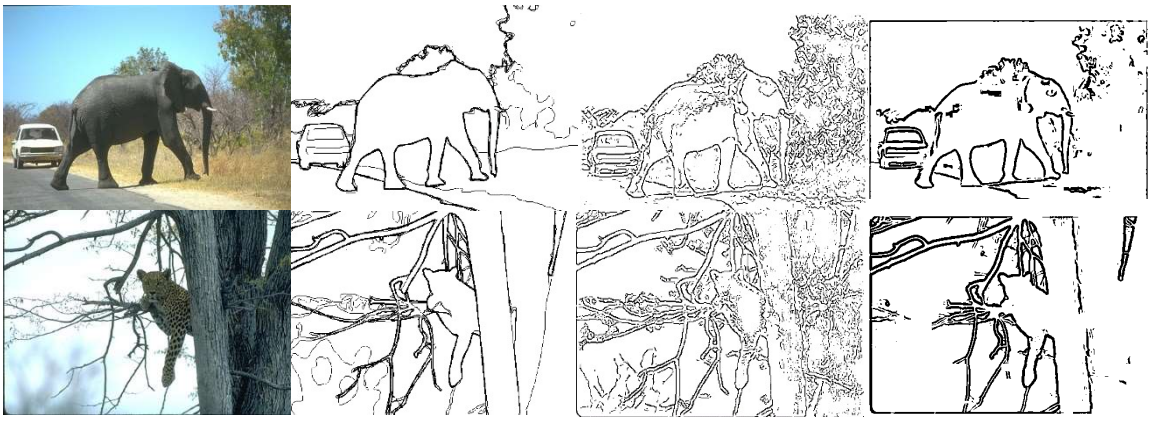


Fig. IV.14 Les performances du détecteur proposé comparées à celles du détecteur DeepEdge. Les première et deuxième colonnes représentent les images d'origine et les bords annotés par l'humain. La troisième colonne contient les résultats de DeepEdge [104] suivis des résultats du détecteur proposé.

Fig. IV.15 Les performances du détecteur proposé comparées à celles du détecteur gPb. Les première et deuxième colonnes représentent les images d'origine et les bords annotés par l'humain. La troisième colonne contient les résultats de gPb [38] suivis des résultats du détecteur proposé.

Nous avons finalement testé le détecteur proposé sur la base de données PASCAL-Context VOC 2012 [36], cette dernière est populaire dans le domaine de la vision par ordinateur. Les résultats visuels du détecteur proposé sont illustrés à la Fig. IV.16.

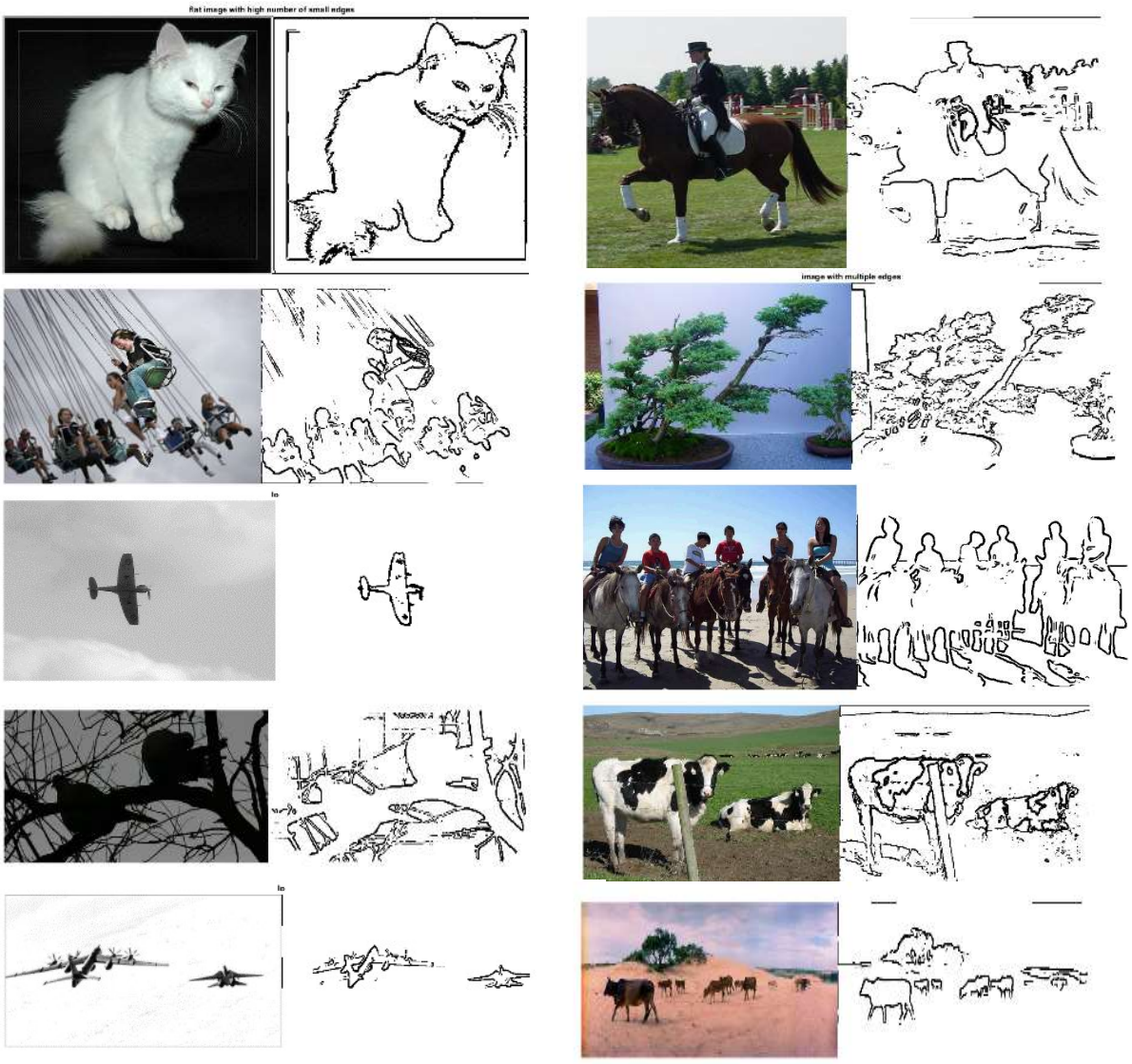


Fig. IV.16 Quelques performances visuelles du détecteur proposé sue la base de données VOC 2012.

Nous n'avons pas de résultats quantitatives pour cette base de données car elle ne contient pas d'images de bords annoté par l'humain, ce qui est nécessaire pour le processus de calcul des résultats expérimentaux.

IV.3. Conclusions

Dans ce chapitre, nous avons montré les résultats expérimentaux du descripteur proposé de caractéristiques avec la phase de pré-élimination des fausses correspondances. Ainsi que le détecteur de bords proposé avec sa caractéristique de localisation d'objets pertinent dans l'image. Les résultats montrent l'efficacité des schémas proposés même si ces derniers sont encore sujet à d'éventuels améliorations afin d'atteindre de meilleurs performances.

Conclusion générale

Nous avons présenté dans cette thèse deux différentes contributions dans le domaine de la vision par ordinateur. Ainsi, nous nous sommes intéressés en premier lieu à un sujet très étudié et qui s'applique à plusieurs domaines tels que la reconstitution de scènes à point de vue multiples à savoir la correspondance des images. Nous nous sommes également intéressés à la détection des objets dans les images sous différentes contraintes et transformations. Le deuxième sujet que nous avons étudié n'est pas en reste de ce qui est de l'importance qu'il tient dans le domaine de la vision par ordinateur, celui-ci concerne la détection de bords dans les images. De même que pour la première contribution, nous nous sommes intéressés à la partie de détection et rehaussement d'objets pertinent dans les images.

Ainsi, dans la première contribution nous avons proposé un descripteur de caractéristiques basé sur les histogrammes d'intensité et de gradient indépendant de la phase de calcul et de compensation de l'orientation et invariant au changement d'orientation.

Il est important de noter que l'invariance à l'orientation de la plupart des descripteurs de l'état de l'art est assurée par une étape supplémentaire avant le processus de description du point caractéristique et de son environnement.

Les résultats expérimentaux ont montré l'efficacité du descripteur proposé contre différents types de changement d'images et de manière plus précise contre le changement d'orientation, la compression JPEG, le changement du point de vue et les déformations 3D. Les changements d'intensité et de flou sont également supportés par le descripteur proposé.

La simplicité du descripteur proposé et sa facilité de mise en œuvre le rend attrayant pour différant terminaux à faible capacité de calcul tels que les caméras stéréo Multiview ou les caméras de surveillance.

Dans le contexte de la détection d'objet, nous avons proposé une étape de pré-élimination de fausses correspondances basée sur la méthode de groupage k-means. Cette a permis d'augmenter les capacités du descripteur proposé en termes de précision de détection. Aussi, la modularité de cette étape la rend facile à mettre en œuvre sur différant descripteurs en l'activant dans le cas de détection d'objets ou en la désactivant ailleurs.

Les résultats expérimentaux ont montré une augmentation des performances des descripteurs en termes de précision de détection avec le module de pré-élimination des fausses correspondances d'environ 10%. Ce qui représente un gain de précision important.

Pour la deuxième contribution, nous nous sommes intéressés à la détection de bords et de contours dans les images. Nous avons proposé un schéma qui présente en plus de la fonction de détection de bords, la capacité de rehausser les objets pertinents dans les images en se basant exclusivement sur une bonne compréhension de la distribution de la surface de l'image intensité.

Ainsi, nous avons pris en compte les informations fournies par l'intensité de l'image d'origine pour modéliser sa surface et détecter ces bords. Nous avons seulement utilisé deux mesures statistiques, qui sont la moyenne et l'écart type (std) des intensités pour déterminer la position du bord dans l'image.

Les expériences ont montré l'efficacité du détecteur proposé en termes de résultats qualitatifs et de rapidité de calcul comparativement à l'état de l'art des détecteurs existant. En outre de sa capacité de fournir des informations riches sur la nature de l'image, il est également capable de détecter et de mettre en évidence des régions pertinentes dans l'image, qui peuvent être exploitées pour d'autres expériences dans le domaine de la segmentation telle que l'élimination d'arrière plans dans les images.

Perspectives et travaux futures

L'objectif principal à travers ces contributions était de présenter deux méthodes différentes dans deux domaines distincts à savoir, la correspondance et la détection d'objets dans les images. Ainsi que la détection des contours et le rehaussement d'objets pertinents dans les images.

Pour les travaux futurs, nous avons pour projets d'optimiser la partie de correspondance des points clés entre les images en utilisant des méthodes plus élaboré basés sur l'apprentissage. Nous avons également l'intention d'améliorer le temps de calcul du descripteur proposé en s'appuyant sur les méthodes de parallélisme existante.

Pour le détecteur de bords proposé, nous avons pour objectifs futurs d'optimiser la méthode de sélection de la taille des blocs, ainsi que la sélection du seuil de classification de ces derniers. Nous prévoyons également de réduire le temps de calcul du détecteur proposé en le mettant en œuvre sur un processeur parallèle.

BIBLIOGRAPHIE

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[2] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2005.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, 2004.

[4] M. Güzel, "A Hybrid Feature Extractor using Fast Hessian Detector and SIFT," *Technologies*, 2015.

[5] Y. K. Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004.*, 2004.

[6] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006.

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[9] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[10] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012.

[11] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.

[12] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[13] P. F. Alcantarilla, L. M. Bergasa, and A. J. Davison, "Gauge-SURF descriptors," *Image Vis. Comput.*,

2013.
[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, 2012.
[15] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in *Procedings of the British Machine Vision Conference 2015*, 2015.
[16] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[18] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[19] V. Balntas, L. Tang, and K. Mikolajczyk, "Binary Online Learned Descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 8, pp. 1–1, 2017.
[20] K. He, Y. Lu, and S. Sclaroff, "Local Descriptors Optimized for Average Precision," 2018.
[21] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Context-Aware Local Binary Feature Learning for Face Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.
[22] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.
[23] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," 2017.
[24] P. Arbel´aez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *IEEE CVPR*, pages 2294–2301. IEEE, 2009.
[25] P. Arbel´aez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *IEEE CVPR*, pages 328–335, 2014.
[26] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE TPAMI*, 30(1):36–51, 2008.
[27] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE TPAMI*, 13(10):992–1006, 1991.
[28] G. S. Robinson. Color edge detection. *Optical Engineering*, 16(5):165479–165479, 1977.
[29] D. Marr and E. Hildreth . Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences*, 207(1167) : 187–217, 1980.
[30] V. Torre and T. A. Poggio. On edge detection. *IEEE TPAMI*, 8(2):147–163, 1986.
[31] I. Sobel. Camera models and machine perception. Technical report, DTIC Document, 1970.
[32] G. Bertasius, J. Shi, and L. Torresani. High-for-low and lowforhigh: Efficient boundary detection from deep object features and its applications to high-level vision. In *IEEE ICCV*, pages 504–512, 2015.
[33] P. Doll´ar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *IEEE CVPR*, volume 2, pages 1964–1971. IEEE, 2006.
[34] X. Ren. Multi-scale improves boundary detection in natural images. In *ECCV*, pages 533–545. Springer, 2008.
[35] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu. Statistical edge detection: Learning and evaluating edge cues. *IEEE TPAMI*, 25(1):57–74, 2003.
[36] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TPAMI*, 26(5):530–549, 2004
[37] P. Arbel´aez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, 2011.
[38] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.
[39] J. J. Lim, C. L. Zitnick, and P. Doll´ar. Sketch tokens: A learned mid-level representation for contour and object detection. In *IEEE CVPR*, pages 3158–3165, 2013.
[40] P. Doll´ar and C. L. Zitnick. Fast edge detection using structured forests. *IEEE TPAMI*, 37(8):1558–1570, 2015
[41] Y. Ganin and V. Lempitsky. N4-Fields: Neural network nearest neighbor fields for image transforms. In *ACCV*, pages 536–551. Springer, 2014.
[42] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deep-Contour: A deep convolutional feature learned by positive sharing loss for contour detection. In *IEEE CVPR*, pages 3982–3991, 2015.
[43] J.-J. Hwang and T.-L. Liu. Pixel-wise deep learning for contour detection. *arXiv preprint arXiv:1504.01989*, 2015
[44] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. Densenet: Implementing efficient convent descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.
[45] S. Xie and Z. Tu. Holistically-nested edge detection. In *IJCV*. Springer, 2017.

[46] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[47] Y. Liu and M. S. Lew. Learning relaxed deep supervision for better edge detection. In *IEEE CVPR*, pages 231–240, 2016.
[48] Y. Li, M. Paluri, J. M. Rehg, and P. Doll´ar. Unsupervised learning of edges. In *IEEE CVPR*, pages 1619–1627, 2016.
[49] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," *Proc. IEEE Conf. Comput. Vis. Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, 2000.
[50] C. Li, C. Y. Kao, J. C. Gore, and Z. Ding, "Implicit active contours driven by local binary fitting energy," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 200
[51] H. P. Moravec. "Towards Automatic Visual Obstacle Avoidance". In *Proc. 5th International Joint Conference on Artificial Intelligence, pp. 584, 1977.*
[52] Schmid, C., Mohr, R. and Bauckhage, C. "Evaluation of interest point detectors", *Int. Journal of Computer Vision*, Vol.37, No. 2, pp. 151-172, 2000.
[53] Förstner, W.: "A framework for low level feature extraction", In: Ecklundh (Eds.): *Proc. 3rd European Conf. on Computer Vision*, LNCS 800, Springer, pp. 383-394, 1994.
[54] Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference. (1988) 147 – 151.
[55] Lindeberg T. "Feature detection with automatic scale selection", *Int. Journal of Computer Vision*, Vol. 30, No. 2, pp.79-116, 1998.
[56] Sojka, E.: "A new approach to detecting the corners in digital images", *Proc. Int. Conf. Image Processing*, Vol. III, pp.445-448, 2003.
[57] Johansson, B. and Söderberg, R.: "A repeatability test for two orientation based interest point detectors", *TR LiTH-ISY-R-2606, Dept. of Electrical Engineering*, Linköping University, Sweden, 18 p, 2004.
[58] Köthe, U.: "Integrated Edge and Junction Detection with the Boundary Tensor", *Proc. of 9th Int. Conf. on Computer Vision*, pp. 424-431, 2003.
[59] Lee, M.H., Park, I.K., 2017. Performance Evaluation of Local Descriptors for Maximally Stable Extremal Regions. J. Vis. Commun. Image Represent.
[60] Smith, S.M. and Brady, J.M.: "SUSAN – a new approach to low level image processing", *Int. Journal of Computer Vision*, Vol. 23, No. 1, pp. 45-78, 1997.
[61] Deriche, R. and Giraudon, G.: "A computational approach for corner and vertex detection", *Int. Journal of Computer Vision*, Vol. 10, No. 2, pp. 101-124, 1993.
[62] Zuliani, M, Kenney, C. and Manjunath, B.S.: "A mathematical comparison of point detectors", *Conf. on Computer Vision and Pattern Recognition Workshop*, Volume 11, pp. 172-178, 1993.
[63] Rodehorst, V.: "Photogrammetrische 3D-Rekonstruktion im Nahbereich durch Auto-Kalibrierung mit projektiver Geometrie*"*, PhD. thesis, Wissenschaftlicher Verlag Berlin.
[64] P. E. Forssén and D. G. Lowe, "Shape descriptors for maximally stable extremal regions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.
[65] T. Yamasaki, "Histogram of Oriented Gradients (HOG)," *J. Inst. Image Inf. Telev. Eng.*, 2010.
[66] T. K. Kang, I. H. In-HwanChoi, M. T. Lim, MDGHM-SURF : arobust local image descriptor based on modified discrete Gaussian–Hermite moment, Pattern Recognit.48(3)(2015)670–684.
[67] Lee, M.H., Park, I.K., 2017. Performance Evaluation of Local Descriptors for Maximally Stable Extremal Regions. J. Vis. Commun. Image Represent.
[68] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Machine Intell. 27 (10) (2005) 1615–1630.
[69] O. Miksik, K. Mikolajczyk, Evaluation of local detectors and descriptors for fast feature matching, in: Proc. of International Conference on Pattern Recognition, 2012, pp. 2681–2684.
[70] B. Kaneva, A. Torralba, W.T. Freeman, Evaluation of image features using a photorealistic virtual world, in: Proc. of IEEE International Conference on Computer Vision, 2011, pp. 2282–2289.
[71] J. Heinly, E. Dunn, J.M. Frahm, Comparative evaluation of binary features, in: Proc. Of European Conference on Computer Vision, 2012, pp. 759–773.
[72] M.I. Restrepo, J.L. Mundy, An evaluation of local shape descriptors in probabilistic volumetric scenes, in: Proc. of the British Machine Vision Conference, 2012, pp. 46.1–46.11.
[73] P. Moreels, P. Perona, Evaluation of features detectors and descriptors based on 3D objects, Int. J. Comput. Vision 73 (3) (2007) 263–284.
[74] A. Gil, O.M. Mozos, M. Ballesta, O. Reinoso, A comparative evaluation of interest point detectors and local descriptors for visual SLAM, Machine Vision Appl. 21 (6) (2010) 905–920.

[75] A.L. Dahl, H. Aanaes, K.S. Pedersen, Finding the best feature detector-descriptor combination, in: Proc. of International Conference on 3D Imaging, Modeling, Processing Visualization and Transmission, 2011, pp. 318–325.
[76] S. Gauglitz, T. Hollerer, M. Turk, Evaluation of interest point detectors and feature descriptors for visual tracking, Int. J. Comput. Vision 94 (3) (2011) 335–360.
[77] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L.V. Gool, A comparison of affine region detectors, Int. J. Comput. Vision 65 (1–2) (2005) 43–72.
[78] A. Haja, B. Jahne, S. Abraham, Localization accuracy of region detectors, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[79] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, Int. J. Comput. Vision 37 (2) (2000) 151–172.
[80] T. Dickscheid, F. Schindler, W. Forstner, Coding images with local features, Int. J. Comput. Vision 94 (2) (2011) 154–174.
[81] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, R. Cilla, Evaluation of low-complexity visual feature detectors and descriptors, in: Proc. of International Conference on Digital Signal Processing, 2013, pp. 1–7.
[82] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3D registration, in: Proc. of IEEE International Conference on Robotics and Automation, 2009, pp. 3212–3217.
[83] J. Donahue *et al.*, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," 2013.
[84] M. S. Extremal, J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from," *Br. Mach. Vis. Conf.*, 2002.
[85] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," *... Sci. Dep. Univ. Toronto, Tech. ...*, 2009.
[86] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
[87] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
[88] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, 1998.
[89] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT," pp. 1–10, 2014.
[90] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
[91] C. B. Choy, S. Savarese, and M. Chandraker, "Universal Correspondence Network," pp. 1–17.
[92] V. Balntas, "Learning local feature descriptors with triplets and shallow convolutional neural networks," *Bmvc*, 2016.
[93] V. K. B. G, G. Carneiro, and I. Reid, "Learning Local Image Descriptors with Deep Siamese and Triplet Convolutional Networks by Minimizing Global Loss Functions," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[94] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
[95] X. Zhang, F. X. Yu, S. Kumar, and S. F. Chang, "Learning Spread-Out Local Feature Descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
[96] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," *Proc. IEEE Conf. Comput. Vis. Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, 2000.
[97] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Inf. Theory*, 1992.
[98] W. T. Freeman and E. H. Adelson, "The Design and Use of Steerable Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9. pp. 891–906, 1991.
[99] S. Belongie, J. Malik, and J. Puzicha, "Matching shapes," *Proc. IEEE Int. Conf. Comput. Vis.*, 2001.
[100] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005.
[101] R. Xiaofeng and L. Bo, "Discriminatively trained sparse code gradients for contour detection," *Neural Inf. Process. Syst.*, 2012.
[102] S. Hallman and C. C. Fowlkes, "Oriented edge forests for boundary detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[103] J. J. Kivinen, C. K. I. Williams, N. Heess, and D. Technologies, "Visual boundary prediction: {A} deep neural prediction network and quality dissection," *Proc. {AISTATS}*, 2014.
[104] G. Bertasius, J. Shi, and L. Torresani, "DeepEdge: A multi-scale bifurcated deep network for top-down contour detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
[105] J.-J. Hwang and T.-L. Liu, "Pixel-wise Deep Learning for Contour Detection," vol. 1, pp. 4–5, 2015.
[106] J. Huang, X. You, Y. Y. Tang, L. Du, and Y. Yuan, "A novel iris segmentation using radial-suppression edge detection," *Signal Processing*, 2009.
[107] Z. Yu, C. Feng, M. Y. Liu, and S. Ramalingam, "CASENet: Deep category-aware semantic edge detection," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
[108] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *Tpami*, 2011.
[109] K. Mikolajczyk, C. Schmid, K. Mikolajczyk, C. Schmid, I. Computer, and C. Schmid, "Indexing based on scale invariant interest points To cite this version :," 2010.
[110] Yi, K., Verdie, Y., Lepetit, V., Fua, P.: Learning to Assign Orientations to Feature Points. In: CVPR. (2016).
[111] Q. Yu, Y. Luo, C. Chen, and X. Ding, "Outlier-eliminated k-means clustering algorithm based on differential privacy preservation," *Appl. Intell.*, 2016.
[112] F. Jiang, G. Liu, J. Du, and Y. Sui, "Initialization of K-modes clustering using outlier detection techniques," *Inf. Sci. (Ny).*, 2016.
[113] Aparna, K., Nair, M.K., 2016. Computational Intelligence in Data Mining. Springer. volume 2. chapter Effect of Outlier Detection on Clustering Accuracy and Computation Time of CHB K-Means Algorithm. pp. 25–35.
[114] Yu, Q., Luo, Y., Chen, C., Ding, X., 2016. Outlier-eliminated k-means clustering algorithm based on differential privacy preservation. Applied Intelligence, 1–13.
[115] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.
[116] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained partbased models. *TPAMI*, 2010.
[117] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.
[118] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989.
[119] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*. 2014.
[120] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.
[121] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.
[122] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.
[123] A. Shrivastava, A. Gupta, and R. Girshick. Training regionbased object detectors with online hard example mining. In *CVPR*, 2016.
[124] P. Mély, A. Dosovitskiy, and T. Brox, "Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT," pp. 1–10, 2014.
[125] K. Mikolajczyk, K. Mikolajczyk, C. Schmid, and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005.
[126] M. Salzmann and F. Moreno-Noguer, "Closed-Form Solution to Non-Rigid 3D Surface," *Eccv*, 2008.
[127] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," *2008 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008.
[128] J. Heinly, E. Dunn, and J. M. Frahm, "Comparative evaluation of binary features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012.
[129] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, 2005.
[130] A. Haja, B. Jähne, and S. Abraham, "Localization accuracy of region detectors," in *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.
[131] M. H. Lee and I. K. Park, "Performance evaluation of local descriptors for maximally stable extremal regions," *J. Vis. Commun. Image Represent.*, vol. 47, pp. 62–72, 2017.
[132] 53 Objects dataset: Zurich Buildings Database. http://www.vision.ee.ethz.ch/en/datasets., 2003.

[133] Pierre Moreels. Home Object dataset. http://www.vision.caltech.edu/pmoreels/Datasets/Home_Objects_06/., 2006.
[134] D. A. Mély, J. Kim, M. McGill, Y. Guo, and T. Serre, "A systematic comparison between visual cues for boundary detection," *Vision Res.*, 2016.
[135] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," *httpwwwpascalnetworkorgchallengesVOCvoc2009workshopindexhtml*, 2012.
[136] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002.
[137] P. Dollár, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.