Image matching using rotating filters

Darshan Venkatrayappa

Collège Doctoral du Languedoc-Roussillon

THÈSE

Pour obtenir le grade de

Docteur

Délivrée par

UNIVERSITE MONTPELLIER (France)

Préparée au sein de l'école doctorale

I2S - Information, Structures, Systèmes

Et de l'unité de recherche

LGI2P - Laboratoire de Génie Informatique et d'Ingénierie de Production de l'**école des mines d'Alès**

Spécialité : **Informatique**

Présentée par

Darshan VENKATRAYAPPA

Image matching using rotating filters

Soutenue le 04/12/2015 devant le jury composé de

Mme. Jenny BENOIS--PINEAU, Professeur *Université de Bordeaux*	Rapporteur
M. Fredric Jurie, Professeur *Université de Caen*	Rapporteur
M. Atilla BASKURT, Professeur *INSA de Lyon*	Examinateur
M. William PUECH, Professeur *Universite Montpellier*	Examinateur Président du jury
M. Philippe MONTESINOS, Maître-assistant (*directeur de thèse*) *École des mines d'Alès*	Examinateur
M. Daniel Diep, enseignant chercheur (*encadrement de thèse*) *École des mines d'Alès*	Examinateur

"Man's mind, once stretched by a new idea, never regains its original dimensions. "
-Oliver Wendell Holmes

Abstract

Nowadays computer vision algorithms can be found abundantly in applications related to video surveillance, 3D reconstruction, autonomous vehicles, medical imaging etc. Image/object matching and detection forms an integral step in many of these algorithms. The most common methods for Image/object matching and detection are based on local image descriptors, where interest points in the image are initially detected, followed by extracting the image features from the neighbourhood of the interest point and finally, constructing the image descriptor. In this thesis, contributions to the field of the image feature matching using rotating half filters are presented. Here we follow three approaches: first, by presenting a new low bit-rate descriptor and a cascade matching strategy which are integrated on a video platform. Secondly, we construct a new local image patch descriptor by embedding the response of rotating half filters in the Histogram of Oriented gradient (HoG) framework and finally by proposing a new approach for descriptor construction by using second order image statistics. All the three approaches provide interesting and promising results by outperforming the state of art descriptors.

Keywords : Rotating half filters, local image descriptor, image matching, Histogram of Orientated Gradients (HoG), Difference of Gaussian (DoG).

Résumé

De nos jours les algorithmes de vision par ordinateur abondent dans les applications de vidéo-surveillance, de reconstruction 3D, de véhicules autonomes, d'imagerie médicale, etc... La détection et la mise en correspondance d'objets dans les images constitue une étape clé dans ces algorithmes.

Les méthodes les plus communes pour la mise en correspondance d'objets ou d'images sont basées sur des descripteurs locaux, avec tout d'abord la détection de points d'intérêt, puis l'extraction de caractéristiques de voisinages des points d'intérêt, et enfin la construction des descripteurs d'image.

Dans cette thèse, nous présentons des contributions au domaine de la mise en correspondance d'images par l'utilisation de demi filtres tournants. Nous suivons ici trois approches : la première présente un nouveau descripteur à faible débit et une stratégie de mise en correspondance intégrés à une plateforme vidéo. Deuxièmement, nous construisons un nouveau descripteur local en intégrant la réponse de demi filtres tournant dans un histogramme de gradient orienté (HoG) ; enfin nous proposons une nouvelle approche pour la construction d'un descripteur utilisant des statistiques du second ordre. Toutes ces trois approches apportent des résultats intéressants et prometteurs.

Mots-clés : Demi filtres tournants, descripteur local d'image, mise en correspondance, histogramme de gradient orienté (HoG), Différence de gaussiennes (DoG).

Acknowledgements

I would like to express my deepest gratitude and appreciation to the following individuals and organization that supported and motivated me in completing this thesis.

Firstly, I would like to thank Dr. Philippe Montesinos, my supervisor, for his patience, insightful criticism, motivation, guidance, encouragement and technical arguments throughout my PhD. My constant interaction wit him on day to day basis helped me to settle down into my research quickly and his support and encouragement for helping me with the complex concepts, tools and libraries (libvision). Secondly, I would like to thank Dr. Daniel Diep, my second supervisor, for his kindness, guidance, critics, and support during the course of work. His encouragement for adapting new methods and encouragement helped me in having a good thesis. I feel privileged for being mentored by both my supervisors and look forward for future collaborations.

I would like to thank, LGI2P at Ecole Des Mines d'Ales and I2ML, for providing both material and funds for my thesis, conference and workshops. To my friend Samhitha B.S, for devoting her precious time to proofread my thesis. To my colleague, Baptiste Magnier, for the healthy discussions and to the people of LG12P lab, Dr.Yannick Vimont and head of the team Dr.Jacky montmain, Claude Badiou and Valerie Roman in the administrative section, for their help and support for making this research a possibility.

My deepest gratitude to my parents for their distant support, encouragement, care and for having belief in me throughout my years of study. Last but not the least, I would like to thank God.

Thank you.
Darshan VENKATRAYAPPA

Introduction

Feature point matching is the basic step used in most of the computer vision applications such as image matching, 3D reconstruction, object recognition, virtual reality, motion analysis and panorama generation. This method allows for the retrieval of both the position and the intensity of feature points in general. All real objects must obey the geometric and radiometric constrains imposed by the laws of physics, thus every time a new correspondence is found for a point, the derived constraint can be used to improve the information about its position in the real world [Bel11].

Given two images of the same scene/object captured under different geometric and photometric variations, the problem of image matching is to find points in an image that can be identified as same points in another image. Since, image matching based on an overall description (global methods) are sensitive to changes in backgrounds, to occlusions and to the main image transformations. Most of the matching methods typically make use of neighbouring local features(local methods) associated with those key-points.

A local image feature can be defined as a local image pattern that varies from its local neighbourhood. They find prominence in computer vision as they can provide well localized and individually identifiable key-points. The information provided by the key-points is not of much importance as long as they can be located accurately and stably over time. In multiple view geometry, more importance is given to their location (center) as they are used for estimating the model of a scene. But for other applications, local features and their descriptors can be used as a robust image representation that allows to match images, recognize objects and scenes without the need for segmentation. Here, they do not have to be localised accurately, since their statistics is more important for the analysis [TM07].

Image descriptor generation forms the crucial step in an image matching algorithm. Initially, local image features are detected and these features are used to represent an image by a well defined and informative local geometric structures extracted from the neighbourhood. The obtained informative geometrical structures are then used to form a descriptor. The task in hand is for the need of an image descriptor that is robust to geometric and photometric variations. So, image matching using local image descriptors forms the basis for this thesis.

Many local image descriptors have been proposed in the literature. They can be broadly classified into descriptors based on 1st order image statistics (SIFT, GLOH, DAISY, MROGH etc), descriptors based on filter response (SURF, local jets, Gabor filter etc), descriptors based on 2nd order image statistics, intensity based descriptors(CSLBP, LIOP) and binary descriptors(BRIEF,FREAK). In this thesis, we concentrate on image descriptors based on filter responses. In particular, we concentrate on anisotropic half filters. To the best of our knowledge, image descriptors based on any anisotropic filters

or any half filter has not been fully explored in the computer vision literature.

Most of the filter based descriptors typically make use of well designed filters to construct the descriptor. Here, the descriptor is constructed by pooling the response of the filters only at the interest points and as a result, they vaguely capture the geometry of the region around the key-point. So, they fail to outperform descriptors based on image gradients (SIFT, DAISY, GLOH). Even if they manage to outperform the gradient based descriptors by using the multi scale approach, they exhibit high dimensionality. In this thesis, the aim is to overcome this problem by proposing descriptors based on anisotropic half filters which not only outperforms the state of the art descriptors but also maintains similar dimension to that of the state of the art descriptors.

This thesis was Proposed by I2ML in partnership with LGI2P at Ecole des mines d'ales. I2ML is an institute oriented towards helping old and disabled people. The scope of the partnership is to bring positive changes in the day to day activities of old and the disabled people by providing them with *smart homes*. Many computer vision techniques such as object detection, image matching and gesture recognition can be used to improve the quality of tools/equipments used in *smart homes*. This work is the primary step in partnership between I2ML and LGI2P. In the future we would like to extend the methods presented in the thesis on applications related to object detection and gesture recognition, which we believe would bring a positive change in the day to day life of old and disabled people in *smart homes*.

Thesis Organization

The remainder of the thesis is organized as follows: Chapter.1 presents the literature review, where we discuss about the image matching pipeline and its various stages. Chapter.2 is dedicated to an overview of different isotropic and anisotropic filters. We also explore the family of anisotropic half filters. Chapter.3 introduces to the low bit-rate image descriptor and a new image matching strategy both of which are embedded in a video loop. Chapter.4 describes the new image patch descriptor RSD-HoG that exhibit a higher performance when compared with previous low bit rate and state of the art descriptors. Chapter.5 proposes an image descriptor RSD-DoG that is based on the second order image statistics. The descriptor is constructed by considering the image as a 3D surface with intensity being the third dimension. Chapter.6 concludes this thesis by highlighting the important contributions of this thesis and proposes new directions for the future.

Contents

Résumé en français

Cette thèse a été proposée par le Laboratoire de Génie Informatique et Ingénierie de Production (LGI2P) de l'Ecole des Mines d'Alès, dans le cadre d'un partenariat avec l'Institut Méditerranéen des Métiers de la Longévité (I2ML), institut orienté vers le maintien à domicile des personnes âgées. Le but de ce partenariat est d'améliorer les conditions d'activités quotidiennes des personnes âgées grâce à des "logements intelligents" (smart homes). De nombreuses techniques de vision par ordinateur telles que la détection d'objets, la mise en correspondance d'images ou la reconnaissance de gestes peuvent être utilisées pour améliorer la qualité des outils et équipements de ces logements intelligents. Le travail présenté ici est une première étape dans le partenariat I2ML-LGI2P. Dans le futur nous voudrions étendre nos méthodes à des applications de détection d'objets et de reconnaissance de gestes, qui apporteront de réelles innovations pour le maintien des personnes âgées à domicile.

La mise en correspondance de points d'intérêt constitue l'élément de base dans des applications telles que la mise en correspondance d'images, la reconstruction 3D, la reconnaissance d'objets, la réalité virtuelle, l'analyse de mouvements et la génération de panoramas. Cette méthode permet de retrouver à la fois la position d'un point d'intérêt et l'intensité de ses caractéristiques en général. Tous les objets réels obéissant à des contraintes géométriques et radiométriques imposées par les lois de la physique, à chaque nouvelle correspondance trouvée pour un point d'intérêt, les contraintes associées peuvent être utilisées pour améliorer l'information sur sa position dans le monde réel.

Etant données deux images de la même scène/objet capturées selon différentes conditions de géométrie et de photométrie, le problème de la mise en correspondance d'images est de trouver un ensemble de points dans une image qui puisse être identifié comme le même ensemble de points dans l'autre image. Les méthodes de mise en correspondance font généralement usage de caractéristiques de voisinage locales (méthodes locales) associées à ces points. Les méthodes de mise en correspondance basées sur une description d'ensemble (méthodes globales) sont sensibles aux variations d'arrière-plans, aux occlusions et aux transformations d'images.

Une zone d'intérêt locale (ou caractéristique visuelle) dans une image peut être définie comme une structure qui varie selon son voisinage local. En vision par ordinateur, les zones d'intérêt locales sont importantes en ce qu'elles fournissent des points d'intérêt bien localisés et identifiables individuellement. L'information fournie par les points d'intérêt a peu d'importance du moment que ceux-ci peuvent être localisés avec précision et stabilité au cours du temps. Dans une géométrie à vues multiples, l'accent est mis sur la localisation précise(centre) des points étant donné qu'ils servent à estimer le modèle de la scène. Pour d'autres applications, les zones d'intérêt locales et leurs descripteurs sont utilisés comme une représentation robuste d'une image, ce qui permet d'apparier des images, de

reconnaître des objets ou des scènes sans passer par une étape de segmentation. Dans ce cas, les zones d'intérêt n'ont pas besoin d'être localisées de façon précise, leurs statistiques étant suffisantes pour l'analyse.

La construction d'un descripteur d'image forme une étape cruciale dans un algorithme de mise en correspondance. Tout d'abord, les zones d'intérêt (points, régions) sont détectées et utilisées pour représenter une image par des structures locales bien définies et contenant des caractéristiques locales extraites du voisinage. Ces structures géométriques sont ensuite utilisées pour former un descripteur. Il apparaît ici le besoin d'un descripteur d'image robuste aux variations géométriques et photométriques. Ainsi, la mise en correspondance utilisant des descripteurs d'images locaux constitue la base de notre thèse.

De nombreux descripteurs d'image locaux ont été proposés dans la littérature. De façon large, on peut les classer en descripteurs basés sur les statistiques du 1er ordre (SIFT, GLOH, DAISY, MROGH, ...), descripteurs basés sur la réponse de filtres (SURF, local jets, filtre de Gabor, ...), descripteurs basés sur des statistiques du second ordre, descripteurs basés sur l'intensité (CSLBP, LIOP), et descripteurs binaires (BRIEF, FREAK). Dans notre thèse, nous nous concentrons sur les descripteurs basés sur la réponse de filtres, et en particulier, sur la réponse de demi-filtres anisotropes. A notre connaissance, les descripteurs d'image basés soit sur des filtres anisotropes, soit sur des demi-filtres n'ont pas été explorés dans la littérature.

La plupart des descripteurs basés sur des filtres font usage de filtres spécialement conçus. Ici, le descripteur est construit en rassemblant la réponse du filtre uniquement pour des points d'intérêt, et il en résulte qu'il ne capture que vaguement la géométrie de la région entourant le point d'intérêt. Par conséquent, il ne se montre pas plus performant que les descripteurs basés sur des gradients d'image (SIFT, DAISY, GLOH). Même s'il parvient à dépasser les descripteurs basés sur les gradients en utilisant une approche multi-échelle, il possède alors une grande dimension. Dans cette thèse nous cherchons à surmonter ce problème en proposant des descripteurs basés sur des demi filtres anisotropes qui non seulement dépassent les descripteurs connus, mais aussi restent de dimension similaire à celle de ces descripteurs.

Ce mémoire de thèse est organisé comme suit : le chapitre .1 présente l'état de l'art, nous y discutons du processus de mise en correspondance d'images et de ses différentes étapes. Le chapitre .2 est dédié à une présentation des différents filtres isotropes et anisotropes. Nous explorons aussi la famille des demi filtres anisotropes. Le chapitre .3 introduit le descripteur d'image à faible débit et une nouvelle stratégie de mise en correspondance, le tout étant intégré dans une boucle vidéo. Le chapitre .4 décrit le nouveau descripteur de fragments d'image RSD-HoG qui montre des performances supérieures à celles du descripteur faible débit précédent et aux descripteurs issus de l'état de l'art. Le chapitre .5 propose un descripteur RSD-DoG basé sur des statistiques du second ordre. Le descripteur est construit en considérant l'image comme une surface 3D, l'intensité étant la troisième dimension. Le chapitre .6 conclut cette thèse en mettant en avant les contributions importantes de cette thèse et propose de nouvelles directions pour le futur.

Introduction

Dans le domaine de la vision par ordinateur, la détection de zones d'intérêt ou caractéristiques d'images et leur mise en correspondance sont devenues la base d'un grand nombre d'applications. L'enjeu pour de nombreux chercheurs du domaine est d'étudier la méthode la plus efficace pour capturer les zones d'intérêt idéales dans une image, d'en exhiber les propriétés intrinsèques. Une zone d'intérêt peut être définie comme un élément d'information permettant de résoudre la tâche de calcul dans une application particulière. Etant donnée l'abondance de la littérature sur les zones d'intérêt ou caractéristiques d'images, il est impossible de détailler chaque contribution.

D'une façon large, on peut classer les zones d'intérêt dans une image en zones d'intérêt ou caractéristiques globales ou locales. Les zones d'intérêt ou caractéristiques globales tentent de représenter l'image entière dans un unique vecteur. Elles parviennent à extraire la structure d'ensemble d'une image, une version grossière des principaux contours et textures de l'image. La plupart des représentations de contours, des descripteurs de formes, des histogrammes couleur et des caractéristiques de textures peuvent être inclus dans la catégorie des caractéristiques globales [LMB+05]. Ces caractéristiques globales trouvent leur intérêt dans des zones nécessitant une segmentation grossière des objets, comme dans le cas d'une reconnaissance de scène et sa classification. Les auteurs de [OT06, MWW12, RO09] les utilisent pour la reconnaissance de scènes, et ceux de [TP91, MN95] pour la reconnaissance d'objets. Ils appliquent ici l'Analyse en Composantes Principales (ACP ou PCA) sur un ensemble d'images modèles et utilisent les premières composantes principales comme descripteurs. Cependant, comme la méthode est basée sur des caractéristiques globales, les problèmes de confusion et d'occultation subsistent. Les caractéristiques globales seules n'offrent pas une description complète et discriminante d'une image. C'est pourquoi les auteurs de [LMB+05, MTEF06] en améliorent les performances par une combinaison de caractéristiques locales et globales pour la description d'images, la classification et la représentation de scènes, etc.

Dans une image donnée, on trouve des zones d'intérêt ou caractéristiques locales en abondance, par centaines ou par milliers. Les zones d'intérêt étant extraites de différents endroits d'une image, elles collectent les principaux contours et textures de l'image en détail. De plus, étant locales à l'image, elles exploitent différentes caractérisations dans des différentes situations, montrant ainsi une robustesse aux occlusions et aux confusions. Les coins [HS88], les contours edges [VMDM15c], les gradients [Low04], les courbures [VMDM15c], les jonctions [HZWC14], les crêtes/vallées [HZWC14] et aussi les petits fragments d'images figurent parmi les zones d'intérêt locales les plus utilisées en vision par ordinateur. Ces zones d'intérêt locales possèdent la plupart des propriétés discutées précédemment, elles présentent ainsi de l'importance dans des applications de reconnaissance/appariement d'objets [VMDM15c, Low04], de suivi de mouvements [VSMM14], d'indexation et la recherche d'images par le contenu. Les caractéristiques globales et les caractéristiques locales fournissent une information différente d'une même image, puisque les régions sur laquelle ces caractéristiques sont calculées varient.

Processus de mise en correspondance

Notre travail portant sur la mise en correspondance, cette section se focalise sur les différentes étapes qui forment le processus de mise en correspondance. La mise en correspondance se compose de 4 étapes importantes. Dans la première étape, les détecteurs de zones d'intérêt sélectionnent des points ou des régions d'intérêt. Puis vient l'échantillonnage des zones d'intérêt et la construction d'un descripteur. Dans la troisième étape, le descripteur subit différentes opérations de post-processing, et enfin, dans l'étape de mise en correspondance, différentes mesures de distance et stratégies d'appariement sont utilisées pour trouver des correspondances robustes. Les différentes étapes sont détaillées ci-après.

Détecteurs de zones d'intérêt

Pour atteindre les meilleures performances en correspondance d'images, recherche d'images et autres applications, nous utilisons des points d'intérêt localisés à la fois spatialement et en échelle. D'après [MS04], les paramètres importants qui caractérisent une zone d'intérêt sont :

1. Le nombre moyen de points de correspondance détectés dans une image selon différentes transformations géométriques et photométriques.

2. La précision sur la localisation et l'estimation de la région, et

3. Le caractère distinctif de la zone d'intérêt, qui est aussi fonction du descripteur utilisé.

Le détecteur de coins de Harris [HS88] et le détecteur Hessien sont parmi les détecteurs de points d'intérêt les plus utilisés. Le détecteur de coins de Harris s'appuie sur la fonction d'auto-corrélation locale d'un signal, la fonction d'auto-corrélation mesurant les variations locales du signal sur des fragments d'image, translatés de petits déplacements dans différentes directions. Une extension du détecteur de Harris aux images couleurs a été proposée par [MGD98]. Le détecteur Hessien est basé sur une matrice de dérivées secondes. Le détecteur recherche des points dans l'image qui possèdent des dérivées importantes dans deux directions orthogonales. Les détecteurs de Harris et Hessien montrent tous deux une forte invariance à des variations en rotation, en luminosité et en présence de bruit. Pour détecter des structures de coin, tous deux utilisent des dérivées gaussiennes calculées à une certaine échelle, et ne peuvent supporter que de relativement faibles changements d'échelle. SUSAN [SB97] est un autre détecteur de coin très utilisé dans la littérature. On peut trouver dans [SMB00] une évaluation des principaux détecteurs de points d'intérêt.

La faiblesse du détecteur de coins de Harris lors de changements d'échelle ou de point de vue a motivé la communauté de vision par ordinateur à prendre en compte l'approche espace-échelle pour la détection des zones d'intérêt. Parmi les détecteurs multi-échelle et affine-invariants importants, on trouve le détecteur Harris-Laplace [MS01, MS04], Laplace Hessien [MS04], Laplacien de Gaussienne (LoG) [Lin98], différence de gaussiennes (DoG) [Low04], Harris-Affine [MS04] et Hessien-Affine [MS04], Maximally Stable Extremal Regions (régions extrêmales maximalement stables, MSER) [MCUP02], Intensity Based Regions (régions basées intensité, IBR) [TG04], et Edge Based Regions (régions basées contours, EBR) [TG04]. Tous les détecteurs multi-échelle et affine-invariants utilisent un

espace échelle linaire gaussien. L'introduction d'un flou gaussien empêche de respecter les limites naturelles des objets dans les images, et lisse à la fois les détails et le bruit, réduisant ainsi la précision sur la localisation et le caractère distinctif des zones d'intérêt. Contrairement à cette approche, les auteurs dans [ABD12] utilisent un espace échelle non linéaire au moyen d'une diffusion par filtrage non linéaire pour la détection des points/régions d'intérêt.

Pour des applications temps réel, les détecteurs de zones d'intérêt à haute vitesse deviennent une nécessité. Les détecteurs tels que DoG, Harris, SUSAN et autres ont des caractéristiques de grande qualité, mais aussi des coûts de calcul importants. Plusieurs détecteurs de zone d'intérêt sont présents dans la littérature, avec FAST (Features from Accelerated Segment Test) [RPD10], SURF (Speeded-Up Robust Features) [BTG06] et CenSurE [AKB08].

Descripteurs de zones d'intérêt

Une fois les points/régions d'intérêt obtenus, l'étape suivante dans le processus de mise en correspondance est d'extraire les caractéristiques des régions autour des points d'intérêt. Ces caractéristiques sont encodées pour former un identifiant unique ou une signature, que l'on peut ensuite utiliser pour une correspondance avec des points d'intérêt dans d'autres images. Ces identifiants ou signatures utilisés dans un but de mise en correspondance sont appelés des descripteurs d'image. La construction d'un descripteur demande de saisir les caractéristiques visuelles des pixels dans une région support autour d'un point d'intérêt. Les caractéristiques peuvent provenir de valeurs en niveaux de gris ou en couleurs de la région, texture ou géométrie de la région support. Le but est de représenter ces caractéristiques d'une façon compacte et discriminante, pour pouvoir utiliser le descripteur dans des applications variées.

Un grand nombre d'algorithmes de description d'images a été proposé dans la littérature. De façon large, on peut classer les descripteurs en 5 catégories : 1. les descripteurs basés gradient, 2. les descripteurs basés réponse de filtre, 3. les descripteurs basés sur des structures locales d'intensité, 4. les descripteurs basés sur des statistiques du second ordre et 5. les descripteurs binaires.

Les descripteurs basés gradient sont obtenus en extrayant le gradient en chaque pixel de l'image. Parmi les descripteurs basés gradient connus, citons Histogram of Oriented Gradient(HoG) [DT05] et ses variantes Pyramid HoG [BZM07], Histogram Of Flow (HOF) [LMSR08] qui fait usage de l'information de mouvement, HOG3D [KMS08] utilisé pour les données volumétriques, et Compressed HoG (CHOG) [CTC+12]. Un autre descripteur très utilisé est le descripteur Scale Invariant Feature Transform (SIFT) [Low04, Low99] et ses dérivés GLOH [MS03], DAISY [TLF10], PCA-SIFT [KS04], Mirror and Inversion invariant generalization for SIFT descriptor (MI-SIFT) [MCS10], 3D-SIFT [SAS07a], OpponentSIFT [vdSGS08] et d'autres encore.

Dans le domaine de la vision par ordinateur, la réponse d'un filtre ou une banque de filtre a été utilisée abondamment. Les filtres de Gabor, les filtres orientables (steerable filters), les ondelettes de Haar ont pris de l'importance dans la mise en correspondance d'images et dans la recherche d'images par le contenu. Le très connu descripteur SURF [BTG06] et ses extensions utilise des ondelettes de Haar à la base. Parmi les descripteurs basés sur la réponse de filtres, on trouve SURF et ses variantes Upright SURF (U-SURF) [AKB08], Affine invariant SURF (ASURF) [PLYP12], N dimension

SURF (NSURF) [FZA+11], Gauge SURF (GSURF) [ABD13], Global SURF [CLB10], Top-SURF [TBL10] et d'autres.

Les descripteurs basés intensité utilisent les valeurs d'intensité autour d'un point d'intérêt pour générer le descripteur. Une des méthodes les plus simples et les plus connues est la méthode Local Binary Pattern (LBP), introduite par Ojala et al. [OPH96, OPM02a, OPM01] comme opérateur de texture. LBP génère un descripteur à partir d'un d'un ensemble d'histogrammes d'intensités d'un voisinage local autour de chaque pixel. Initialement, 8 pixels voisins sont choisis autour d'un pixel. Puis la différence d'intensité entre le pixel central et chacun des 8 voisins est calculée, et selon un test binaire, la valeur 1 est assignée au pixel voisin si la différence d'intensité est positive, la valeur 0 sinon. Les descripteurs basés sur LBP sont simples mais efficaces. Pour cette raison ils parviennent à supplanter SIFT, SURF et leurs variantes. De nombreuses variantes de LBP ont été proposées, Center Symmetric Local Binary Pattern (CSLBP) [HPS06], Multi-block LBP descriptor [ZCX+07], Three-Patch LBP descriptor (TPLBP) [WHT08], Four-Patch LBP descriptor (FPLBP) [WHT08], volume LBP (VLBP) [ZP07], Fuzzy Local Binary Pattern (FLBP) [IKM08], opponent color LBP (OCLBP) [MP04a] pour en citer quelques-uns.

Les facteurs externes tels que les variations temporelles de luminosité, de luminosité dépendant du point de vue, les ombres, les variations des paramètres de caméras, la réponse non linéaire des caméras, etc. provoquent des variations lumineuses complexes. Des descripteurs comme SIFT, SURF, DAISY qui sont invariants à un saut d'intensité ou à des changements de luminosité affines ne parviennent pas à traiter des variations lumineuses complexes. Pour pallier à cela, certains auteurs utilisent un ordre relatif des intensités plutôt que les intensités originales. Les descripteurs basés sur ce concept les plus connus sont Ordinal Spatial Intensity Distribution (OSID) [TLCT09], MROGH (Multi-Support Region Order-Based Gradient Histogram) [FWH12], MRRID (Multi-Support Region Rotation and Intensity Monotonic Invariant Descriptor) [FWH12] et Local Intensity Order Pattern (LIOP) [WFW11].

En vision par ordinateur, il existe des travaux conséquents sur la prise en compte de la courbure dans les tâches de reconnaissance d'objets, de recherche d'images, de mise en correspondance, etc. On peut trouver de tels descripteurs utilisant l'information du second ordre dans [MEO11, FB14, Zit10a, HZWC14, RBS09].

Avec l'augmentation de la taille des bases de données d'images et l'avènement d'appareils mobiles dotés de caméras, une nouvelle branche de la vision par ordinateur est apparue, équipant les appareils mobiles tels que smartphones et tablettes, ou portables tels que Google Glass, Microsoft Hololens, qui requièrent des systèmes de vision précis et efficaces en calcul. Les applications utilisant des algorithmes sur des plateformes mobiles de réalité augmentée sont confrontées à des mémoires limitées et de faibles capacités de calcul. C'est là où les descripteurs binaires deviennent une alternative aux descripteurs en nombres réels.

L'idée principale est que chaque bit est indépendant et que la distance de Hamming peut servir de mesure de similarité. Généralement, les descripteurs binaires sont construits en trois étapes. La première étape consiste à choisir un modèle d'échantillonnage. Ce modèle montre exactement où échantillonner les points dans une région autour du point considéré. Vient ensuite l'étape où une orientation est assignée au point d'intérêt, afin d'obtenir une invariance à la rotation. Dans la dernière étape, des paires d'échantillons sont choisies pour construire le descripteur final. Parmi les descripteurs binaires les plus

utilisés, on trouve BRIEF [CLSF10], ORB [RRKB11], BRISK [LCS11], FREAK [AOV12], LDB [YC12].

Post-processing

Le post-processing est une étape limitée mais importante du processus de mise en correspondance. Dans certains cas, cette étape décide de la dimension finale des descripteurs. L'un des inconvénients majeurs des descripteurs à valeurs fractionnaires réside dans leur grande dimension. Ceci limite les performances des techniques de mise en correspondance en termes de vitesse et d'extensibilité. Des méthodes telles que l'Analyse en Composantes Principales (PCA), la transformation de Walsh-Hadamard et d'autres, ont été proposées pour réduire la dimensionnalité. Des schémas de compression ont également été proposés pour réduire le débit des descripteurs en virgule flottante. Chandrashekar et al. [CTC+12] ont proposé un descripteur faible débit appelé Compressed Histogram of Gradient (CHoG).

La mise en correspondance et la reconnaissance d'objets dans des environnements non contrôlés, tels que ceux où la luminosité varie, est l'un des verrous pour les systèmes pratiques de vision par ordinateur. Certains algorithmes de vision tentent de surmonter le problème en normalisant le vecteur descripteur. Dans la phase finale de construction du descripteur SIFT, le vecteur est modifié pour réduire les effets de variations de luminosité. Au départ le vecteur est normalisé à la longueur unité, et les variations de contraste font que chaque valeur de pixel est multipliée par une constante. En conséquence, le gradient est lui aussi multiplié par la même constante. La normalisation annule ce changement de contraste. On réduit encore l'influence des grandes valeurs du gradient en seuillant les valeurs dans le vecteur unitaire à la valeur maximale de 0,2 et en renormalisant à l'unité. Presque tous les descripteurs mentionnés précédemment suivent cette procédure de normalisation avec ou sans seuillage.

Mise en correspondance de descripteurs

Le vecteur-descripteur une fois obtenu, l'étape suivante consiste à l'utiliser pour la mise en correspondance ou la recherche d'images dans une base de données. En général, les distances utilisées en vision par ordinateur sont des fonctions mathématiques connues. Dans le cas de la mise en correspondance, la distance est une mesure qui classifie un appariement correct ou non. Suivant l'application et les capacités de calcul, une métrique appropriée est choisie. Les plus courantes sont la distance euclidienne [Low04], la distance de Hausdorff [DJ94], la distance de Jaccard [Zit10b], la distance de Mahalanobis [VSMM14]. Parmi les autres mesures de distance, citons la distance de Chebyshev, la distance de Hellinger, la distance de Manhattan, la norme L1, la distance de Canberra, la distance de Bray Curtis et la distance de Kullback Leibler.

Demi filtres

Notre travail se consacrant principalement à la description d'images au moyen de réponses de filtres, nous commençons par présenter les filtres isotropes et anisotropes en général, avant de discuter des demi filtres anisotropes utilisés dans cette thèse.

Le filtre gaussien isotrope connu aussi comme l'opérateur de lissage gaussien est un opérateur de convolution 2-D. La première application du filtre gaussien est de flouter les images et de supprimer le bruit. La plupart des méthodes de mise en correspondance lissent la région autour du point d'intérêt avant de construire le descripteur. Un des autres opérateurs basés sur une gaussienne est le laplacien de gaussienne (LoG). Ce filtre a été initialement proposé par [MH80] pour détecter les contours à une échelle particulière. Pour cela, l'image subit un lissage à l'aide d'une gaussienne G, puis un filtrage laplacien. Ces deux étapes forment l'opérateur laplacien de gaussienne (LoG).

Bien que l'opérateur LoG soit précis, il est très coûteux en calculs. Une approximation de l'opérateur LoG appelée différence de gaussiennes (DoG) est proposée dans la littérature. De façon similaire au LoG, l'image est lissée par une convolution avec un noyau gaussien $G_{\sigma_1}(x, y)$ de largeur σ_1 et d'autre part avec un noyau gaussien $G_{\sigma_2}(x, y)$ of width σ_2. Le DoG est défini comme la différence des deux images filtrées par des gaussiennes. Shen et Castan ont proposé un opérateur basé sur les critères de Canny incluant la détection et la localisation. En pratique, les deux filtres sont basés sur des filtres exponentiels, et ont des comportements similaires.

Un inconvénient des filtres isotropes est leur perte de précision en matière de description de zones d'intérêt, de filtrage d'images, de détection de contours et d'autres structures géométriques. D'un autre côté, les méthodes de détection de contours utilisant des banques de filtres orientés donnent des résultats précis. Par leurs multiples orientations, ces filtres sont capables de détecter des caractéristiques visuelles telles que les arêtes, les contours, etc. Les approches par filtres de Gabor et filtres orientables (steerable filters) sont les plus utilisées. Les filtres de convolution anisotropes trouvent des applications en filtrage adaptatif en les alignant sur des structures d'image locales. On les utilise également pour la détection de structures en même temps que d'autres types de filtres.

Dans les méthodes qui emploient le filtrage anisotrope, la robustesse au bruit dépend fortement de deux paramètres, les deux écarts-types de la fonction Gaussienne à 2 dimensions. Augmenter les valeurs de ces paramètres rend la détection moins sensible au bruit, mais alors les petites structures d'image seront considérées comme du bruit et donc ignorées. Par conséquent, la précision des points détectés décroît fortement au niveau du pixel de coin et pour des objets ayant des contours non linéaires. Ce défaut peut être levé par l'utilisation de demi filtres anisotropes.

Pour résoudre les problèmes rencontrés avec le filtre anisotrope gaussien, la solution que nous proposons est de couper un filtre directionnel gaussien en deux parties et ensuite d'appliquer à l'image les filtres selon différentes orientations. Ces demi filtres ont été introduits par [MMD11a, MMD11b, MM10, MMP10, MMD11c]. Par construction, le demi filtre de lissage gaussien n'est pas symétrique dans la direction de son élongation maximale. Nous nous référons à ce filtre comme demi filtre/noyau anisotrope de lissage gaussien (Anisotropic Half Gaussian Smoothing Kernel, AHGSK). Le second filtre étudié est obtenu par différence de deux demi filtres AHGSK et est appelé différence de demi filtres de lissage (Difference of Half Smoothing Filters, DHSF). Il peut être utilisé pour l'extraction de statistiques du second ordre, et la détection de crêtes, vallées, jonctions, etc...

La non-symétrie du demi filtre de lissage rend difficile le calcul du gradient via un tenseur d'orientation. Nous utilisons pour cela un filtre dérivateur dans la direction du plus petit écart-type de manière à lisser dans une direction et à dériver dans la direction perpendiculaire. Ce filtre est appelé demi filtre anisotrope dérivateur gaussien (Anisotropic

Half Gaussian Derivative Filter, AHGDK). Le filtre de Shen Castan (première partie de ce chapitre) est modifié pour approximer le filtre AHGDK, il est appelé demi filtre anisotrope dérivateur exponentiel (Anisotropic Half Exponential Derivative Kernel AHEDK). Le filtre AHEDK affiche des caractéristiques et donne des résultats similaires au filtre AHGDK. Nous utilisons l'implémentation récursive du noyau exponentiel, qui est d'ordre 1 et est environ 5 fois plus rapide que l'implémentation récursive du noyau gaussien. Par construction, AHEDK a des caractéristiques de dérivation dans la direction X et de lissage dans la direction Y.

Mise en correspondance d'images vidéo à l'aide de filtre tournants

Nous utilisons le demi filtre anisotrope de lissage gaussien (AHGSK) expliqué précédemment afin d'obtenir une description autour d'un point d'intérêt. Ce filtre est utilisé comme un descripteur bas débit appelé descripteur de signal tournant (RSD), et est utilisé pour la mise en correspondance d'images vidéo. Ceci constitue notre première contribution de thèse.

Dans cette méthode, nous utilisons tout d'abord le détecteur Harris couleur pour trouver les points d'intérêt dans une image. Puis le filtre AHGSK fait un balayage autour de chaque point d'intérêt pour extraire les caractéristiques couleur sous la forme d'un signal signature. Nous appelons ce signal signature RSD (Rotating Signal Descriptor). Par construction, RSD ne présente pas d'invariance euclidienne. Pour cela, nous calculons la corrélation entre deux signatures par FFT. Une déformation modérée est prise en compte par une méthode de Dynamic Time Warping (DTW), puis par une vérification en cascade nous améliorons la robustesse de la mise en correspondance. Au final, notre méthode se montre invariante aux changements de luminosité, à la rotation, aux petites déformations et partiellement aux changements d'échelle. De plus, il est possible de contrôler la dimension de RSD en jouant sur le pas angulaire du filtre tournant. Notre descripteur avec une dimension limitée à 12 peut ainsi donner de bons résultats en correspondance. La faible dimension du descripteur est la principale motivation pour étendre le processus à des images vidéo. Bien que différente et nouvelle, la méthode possède des inconvénients :

- La méthode de Dynamic Time Warping modifie les signaux signatures. Si la fonction de contrainte est mal choisie, une correspondance correcte peut être identifiée comme non correcte et vice-versa. Pour définir une fonction de contrainte globale, il sera nécessaire d'introduire une phase d'apprentissage.
- La méthode proposée avec la DTW montre une invariance à des déformations affines modérées.
- La méthode proposée n'est pas invariante aux changements d'échelle.
- RSD étant obtenue par une scrutation angulaire du filtre AHGSK autour du point d'intérêt (un coin de Harris couleur), l'information extraite est pauvre. RSD utilisé seul ne peut décrire la géométrie de la région autour du point d'intérêt. Evalué sur la base d'images standard (en niveaux de gris), le descripteur RSD donne des résultats médiocres comparés à ceux obtenus avec le descripteur SIFT.

RSD-HoG

La plupart des descripteurs basés sur des réponses de filtres donnent de mauvais résultats en comparaison avec les descripteurs basés gradient. Etant obtenus à des points d'intérêts isolés, ces descripteurs ne capturent pas la géométrie de la région autour du point d'intérêt. Dans ce chapitre nous nous efforçons d'obtenir un descripteur d'image robuste à l'aide de demi filtres anisotropes. Notre descripteur peut être considéré comme une combinaison des descripteurs basés sur le gradient, la courbure et les réponses de filtres. Nous intégrons la réponse du demi filtre gaussien (RSD) dans un histogramme de gradients orientés, d'où le nom RSD-HoG. Dans ce descripteur :

- Tout d'abord nous normalisons la région autour du point d'intérêt pour obtenir un fragment d'image (patch) de taille 41×41. La normalisation apporte l'invariance à la rotation et l'invariance affine.
- Le fragment d'image est convolué avec AHGDK ou AHEDK et pour chaque pixel nous obtenons une signature RSD.
- De ce RSD, nous extrayons les angles pour lesquels le maximum et le minimum sont obtenus. Nous extrayons la réponse du filtre à ces valeurs d'angle. Cette procédure est appliquée à tous les pixels du fragment d'image.
- Enfin, comme pour HoG, nous calculons l'histogramme de ces angles pour former le descripteur RSD-HoG.
- Nous utilisons différentes combinaisons d'angles et construisons différentes variantes de RSD-HoG.
- Nous utilisons la base d'images standard et le protocole standard du groupe de recherche d'Oxford [1].
- Nous utilisons la courbe (1-précision, rappel) [MS03] pour évaluer notre méthode.

Les résultats représentés dans les Fig.4.10, Fig.4.11, Fig.4.12, Fig.4.14 and Fig.4.13 illustrent les bonnes performances de notre descripteur comparé aux descripteurs de l'état de l'art. Ces courbes montrent aussi l'avantage à utiliser des demi filtres plutôt que des filtres entiers.

RSD-DoG

En vision par ordinateur, la littérature montre que les statistiques du second ordre apportent une information importante sur l'image. En contribution finale à la thèse, nous proposons un autre nouveau descripteur basé sur des dérivées du second ordre et appelé RSD-DoG. Ici, nous traitons chaque fragment d'image comme une surface 3-D, l'intensité constituant la 3ème dimension. La surface 3-D considérée possède un riche ensemble de caractéristiques visuelles/statistiques du second ordre telles que les crêtes, les vallées, les falaises, etc... Ces statistiques du second ordre peuvent être captées aisément par la

[1] `http://www.robots.ox.ac.uk/~vgg/research/affine/`

différence de demi filtres gaussiens tournants. L'originalité de notre méthode réside dans la combinaison de réponse des filtres directionnels avec l'approche différence de gaussiennes (DoG). La méthodologie mise en oeuvre dans la construction du descripteur est la suivante :

- Tout d'abord, comme pour RSD-HoG nous normalisons la région autour du point d'intérêt pour former un patch (fragment) de 41×41.
- Le patch normalisé est considéré comme un fragment 3-D avec l'intensité comme 3ème dimension.
- Le patch est convolué avec DHSF pour obtenir les signatures à chaque pixel.
- De chaque signature, nous extrayons 2 maxima et 2 minima.
- Nous déterminons les moyennes des maxima et des minima et formons l'histogramme des angles moyens pour construire le descripteur RSD-DoG.
- Nous utilisons la base d'image standard et le protocole standard du groupe de recherche d'Oxford[2].
- Nous utilisons la courbe (1-précision, rappel) [MS03] pour évaluer notre méthode.
- Par construction notre descripteur montre de bonnes performances lors de variations de luminosité. Pour illustrer cette propriété, nous l'évaluons sur l'ensemble d'images [2].

Les résultats présentés Fig.5.8, Fig.5.9, Fig.5.10, Fig.5.11, Fig.5.12, Fig.5.13, Fig.5.14 and Fig.5.15 montrent l'avantage de notre méthode en comparaison des descripteurs de l'état de l'art. Ces courbes illustrent aussi l'avantage à prendre en compte dans les descripteurs les changements de luminosité.

Conclusion

A notre connaissance, les descripteurs d'image basés sur des filtres ou demi filtres anisotropes n'ont pas été entièrement explorés dans le domaine de la vision par ordinateur. Dans cette thèse, nous avons présenté de nouveaux descripteurs d'image basés sur une famille de demi filtres anisotropes qui dépassent en performance de nombreux descripteurs de l'état de l'art basés sur des statistiques du 1er ordre et des réponses à des filtres. Ce travail peut être utilisé dans de nombreuses application de vision par ordinateur. Dans cette thèse trois contributions nouvelles sont proposées.

Dans la première contribution, un nouveau descripteur bas débit appelé RSD a été discuté. La corrélation par FFT et la méthode DTW permettent d'obtenir l'invariance à la rotation et une invariance partielle aux déformations et transformations affines. Ce mécanisme a été étendu à la vidéo. Plusieurs défauts de la méthode ont été mis en évidence. En seconde contribution, nous avons proposé un nouveau descripteur appelé RSD-HoG, dont les performances s'avèrent supérieures à la plupart des descripteurs

[2] `http://www.robots.ox.ac.uk/~vgg/research/affine/`
[2] http://zhwang.me/publication/liop/

courants. Enfin, nous avons introduit un nouveau descripteur basé sur des statistiques d'image du second ordre. Par construction, ce descripteur se montre invariant à des changements de luminosité. Il dépasse en performance les descripteurs courants.

Dans le futur, nous aimerions utiliser les descripteurs présentés ici dans des applications de recherche d'images et de vidéos par le contenu. Par manque de temps, les descripteurs proposés n'ont pas été utilisés dans des applications telles que la détection d'objets ou la reconnaissance de gestes qui pourraient aider les personnes âgées dans leur domicile. Dans un futur proche, nous aimerions utiliser notre travail dans des applications domotiques telles que les smart homes.

Chapter 1

Literature Review

1.1 Image Features

In the field of computer vision, feature detection and matching has become the basis for many of its application. The challenge for many of the researchers within this field has been to study the most effective method that could capture the ideal features of an image, exhibiting its intrinsic properties. In this field, a feature can be defined as a piece of information required to solve the computational task belonging to a particular application. As the literature on Image features is vast, it is impossible to address every contribution in detail. Hence, we will discuss in short an overview to the image features. As in [TM07], properties of a good feature is as follows:

1. Repeatability: When we consider two images from the same scene obtained from different viewing conditions, the features detected under different viewing conditions should be present in both the images upto a greater extent.

2. Locality: The detected features should be local, so as to reduce the chances of occlusion and to allow simple model approximations of the geometric and photometric deformations between the two images taken under different viewing conditions [TM07].

3. Informativeness: The region around the detected features should exhibit a greater variation such that the features can be distinguished and matched easily.

4. Quantity: The quantity of features detected should be large, such that a good number of features are detected even if the object under consideration is small.

5. Accuracy: The detected features should be localized accurately in space, scale and shape.

6. Efficiency: It is preferable that the feature detection process is real-time in nature.

1.1.1 Global Features

Features can be broadly classified as global features and local features. Global features tries to represent the entire image in a single vector. These features succeed in extracting the overall structure of the image, thus exhibiting a poor and coarse version of the

principal contours and textures of the image. Most of the contour representations, shape descriptors, color histograms and texture features can be included under global features category [LMB+05]. These features find prominence in areas where a rough segmentation of an object of interest is involved, as in the case of scene recognition and classification. Authors of [OT06, MWW12, RO09] use them for scene recognition and [TP91, MN95] use global features for object recognition. Here, they apply Principal Component Analysis (PCA) on a set of model images and use the first few principal components as descriptors. However, as this method is based on global features, issues related to clutter and occlusion continue to exist. Global features alone fail to offer a complete and discriminative description of an image. So, the authors of [LMB+05, MTEF06] enhance the performance by combining local and global features for image description, scene classification/representation and so on.

1.1.2 Local Features

In a given image, local features are found in abundance, ranging from hundreds to thousands. As local features are extracted from different locations in an image, they capture the principal contours and textures of the image in detail. Additionally, as these features are local to an image, they exploit different types of features under different situations, thus exhibiting robustness to occlusion and clutter. Corners [HS88], edges [VMDM15c], gradients [Low04], curvatures [VMDM15c], junctions [HZWC14], ridges/valleys [HZWC14] and also tiny image patches are some of the popular local features used in the computer vision literature. As these features exhibit most of the properties that were discussed in our previous section, they find importance in applications related to object recognition/matching [VMDM15c, Low04], motion tracking [VSMM14], indexing and content based image retrieval. Global features and local features provide different information about the same image as the support region over which these features are computed, varies. Since our work is related to image matching, this chapter mainly concentrates on the different stages of an image matching pipeline where, local features play a crucial role.

1.2 Image Matching Pipeline

Image matching pipeline has a series of 4 important stages. In the first stage, key-points or regions are selected using the feature detectors, followed by the feature sampling and descriptor construction. In the third stage, the constructed descriptor undergoes various post-processing operations and finally, in the matching stage, different distance measures and matching strategies are used for finding the robust matches. In the rest of the chapter different stages are explained in detail.

1.2.1 Feature Detectors

To achieve better performance in image matching, image retrieval and other applications, we need feature points that can be localized in both location and scale. According to [MS04], important parameters that characterize the feature detector are:

1. The average number of corresponding points detected in an image under different geometric and photometric transformations.

2. The accuracy of localization and region estimation and

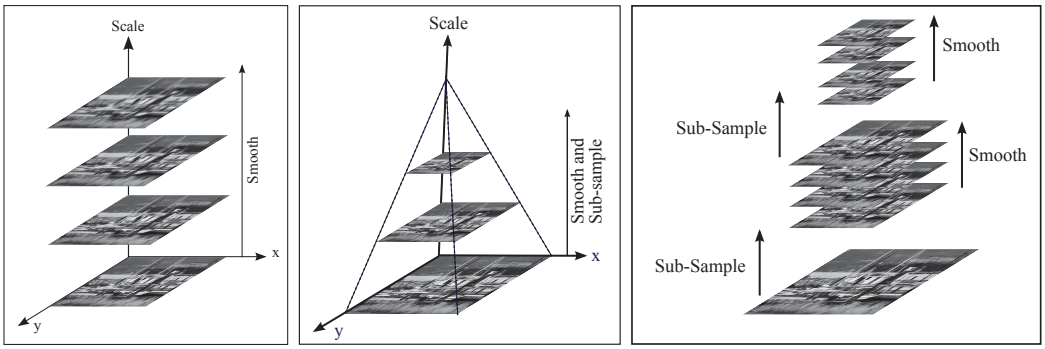
3. The distinctiveness of the feature, which is also a function of the descriptor used.

Harris corner detector [HS88] and Hessian detector [Bea78] are some of the popular interest point detectors. The Harris detector is based on the local auto-correlation function of a signal, where the auto-correlation function measures the local changes of the signal in patches, shifted by a small amount in different directions. An extension to Harris detector for corner detection in color images is proposed by [MGD98]. The Hessian detector is based on a matrix of second derivatives. The detector searches for image locations that exhibits strong derivatives in two orthogonal directions. Both Harris and Hessian detectors show strong invariance to variations in rotation, illumination and image noise. For detecting the corner structures, both these detectors use Gaussian derivatives computed at a fixed scale and hence, can be repeated only up to a relatively small scale changes. SUSAN is another popular corner detector that we come across in the literature and more information on the same can be found in [SB97]. Evaluation of some of the interest point detectors can also be found in [SMB00].

Scale Invariant Region Detectors

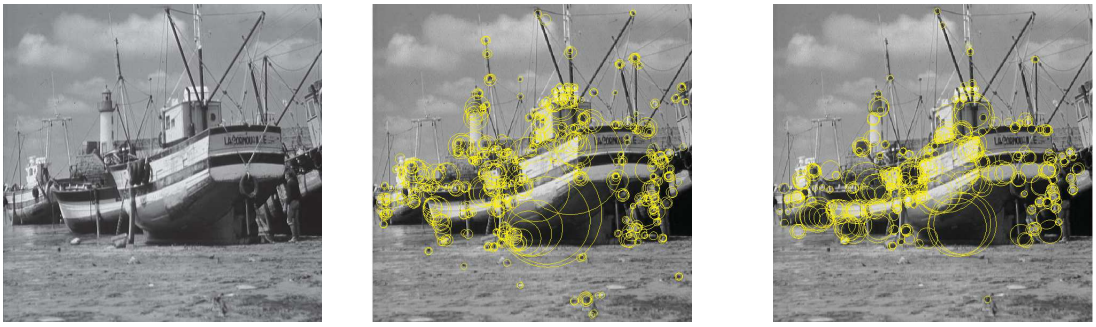
The failure of Harris corner detector to cope up with the changes in scale had prompted the computer vision community to consider the scale-space approach for feature detection. This was first introduced by [Wit83] for representing one dimensional signal at multiple scales. Scale space representation for an image is constructed by smoothing the image with different sized Gaussian kernels. This representation provides a smooth transition between different scales. But, for a coarser scale value, lot of redundant information can be found. Fig.1.1, illustrates the scale-space representation. Authors of [CP84] and [BEA83] used a multi-scale representation based on pyramids. The pyramids were constructed by successively sub-sampling the finer scale images, followed by a smoothing operation. The smoothing operation was performed to prevent the aliasing effect on the coarser scale images. In the pyramid approach, the reduced image resolution resulted in fast processing. But, matching the image structures in pyramids across different scale S was difficult. Fig.1.1(b) illustrates the pyramid representation. Recently [LB03] took advantage of both the approaches by fusing them to form a hybrid scale-space representation. Fig.1.1(c) shows an oversampled pyramid representation.

Lindeberg [Lin98] proposed a new approach for feature detection along with automatic scale selection. Automatic scale selection allows to detect interest points in an image, each with their own characteristic scale. Based on the concept of scale-space and automatic scale selection, Lindeberg further proposed the use of Laplacian-of-Gaussian (LoG) [Lin98] and several other derivative based operators for interest point detection. By construction, the LoG detector is circularly symmetric and detects blob-like structures by searching for a scale space extremum of a scale-normalized Laplacian-of-Gaussian. One of the disadvantages of LoG detector is the computational complexity. David Lowe [Low04] came up with an approximation for LoG by combining the difference-of-Gaussian DoG approach with that of the hybrid scale-space representation and called it the SIFT detector. In his approach, the input image is successively smoothed with Gaussian kernels and is sampled. Later, he subtracts the two successive smoothed images to obtain the difference-of-Gaussian representation. Thus, the DoG approach is fast and often preferred over LoG.



(a) (b) (c)

Figure 1.1: (a) Scale-space representation (b) Pyramid representation (c) hybrid scale-space representation



(a)Original image (b) Harris Laplace regions (c)Hessian Laplace regions .

Figure 1.2: Image showing the regions extracted using Harris Laplace and Hessian Laplace region detectors.

In both DoG and LoG detectors, scale coordinates are sampled only at discrete levels and hence, the accuracy of the detected key-points is not satisfactory. The accuracy of the detected key-points can be enhanced by interpolating the response at neighbouring scales.

Mikolajczyk et al. [MS01, MS04], combined the Harris operator with the scale-space mechanism of [Lin98] to form the Harris-Laplace detector. This method constructs two separate scale spaces for the Harris function as well as the Laplacian. It then uses the Harris function to localize the candidate points at each scale level and captures the points for which the Laplacian simultaneously attains an extremum over scales. Thus, the detected points are robust to variations in scale, rotation, illumination and camera noise [MS03]. This approach utilizes strict criterion for interest point detection. As a result, the number of detected points is very less when compared to that of DoG or LoG. For practical applications related to object recognition, lesser number of interest points reduces the robustness to partial occlusion. [MS04], circumvent this by relaxing the strict criterion. In [MS04], the authors use the same method by replacing the Harris operator with that of the Hessian operator to obtain the Hessian-Laplace detector. Both Hessian Laplace and Harris Laplace are also called as blob detectors. Fig.1.2 shows the key-regions detected using both Harris and Hessian Laplace detectors.

(a)Hessian Affine (b) Harris Affine (c)MSER regions (d) EBR regions.

Figure 1.3: Image showing the regions extracted using Hessian affine, Harris affine, MSER and EBR region detectors.

Affine Covariant Region Detectors

The above mentioned methods can only extract local features exhibiting invariance to changes in scale. In reality, we also have to consider changes in view point (projective transform). Some researchers used perspective correction to achieve affine invariance. They worked on the assumption that, the scene under consideration is locally planar and affine invariance is achieved by estimating and correcting for the perspective distortion that a local image patch undergoes when viewed from a different viewpoint. This procedure is highly cumbersome and produces inaccurate results. Recently, many researchers [MCUP02, SZ02, TG00] have shown that this process can be replaced by local affine approximation.

Harris-Laplace and Hessian-Laplace detectors can be extended to extract affine regions [MS04]. The extensions makes use of an iterative scheme and is initialized with a circular region returned by the original Harris-Laplace or Hessian-Laplace detector. In each of the iteration, the region's second-moment matrix is updated and the eigenvalue of this matrix is computed. This gives an elliptical shape that corresponds to local affine deformation. Finally, the image neighbourhood is transformed in such a way that, the ellipse is transformed into a circle and, the location and scale estimate is updated in the transformed image. This procedure is repeated until the eigenvalues of the second-moment matrix are approximately equal. Finally, the iterative scheme produces elliptical regions adapted to the local intensity patterns, so that the same object structures are covered despite the deformations caused by viewpoint changes. Fig.1.3 (a) and (b) illustrates the affine regions extracted using both Hessian and Harris detectors.

Matas et.al [MCUP02] approaches the problem of affine feature/region detection from a segmentation perspective. They use watershed segmentation algorithm and extract homogeneous intensity regions that are stable over a large range of thresholds. Thus, ending up with Maximally Stable Extremal Regions (MSER). These MSER are stable under different imaging conditions and can be captured under different viewpoints. In [RM03, MREM04], super-pixels obtained using normalised cuts are used as regions for feature extraction and since all the super-pixels extracted from an image have similar scales, this method is not scale invariant. Some of the other affine feature/region detectors present in the literature are Intensity Based Regions (IBR) [TG04] and Edge Based Regions (EBR) [TG04]. Authors of [KB01] proposed salient regions detector which was extended to affine covariant extraction by [KZB04].

All of the above mentioned scale and affine invariant detector use linear Gaussian scale

space for feature detection. Gaussian blurring fails to respect the natural boundaries of objects and smooths the details and noise to the same extent, reducing localization accuracy and distinctiveness. Contrary to this approach, authors in [ABD12] use non-linear scale space by means of non-linear diffusion filtering. By doing so, they make blurring locally adaptive to the image data by reducing noise and retaining the object boundaries as well as obtaining superior localization accuracy and distinctiveness. They accelerate the non-linear scale-space construction process by using efficient Additive Operator Splitting (AOS) techniques and variable conductance diffusion. Binaries for some of the above discussed detectors can be found in [1].

Real Time Feature Detectors

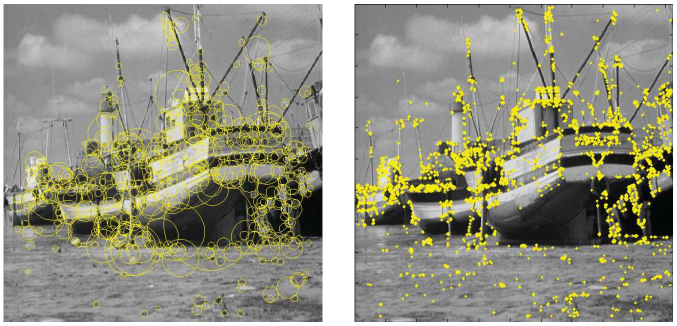
For real time applications, high speed feature detectors have become a necessity. Feature detectors such as DOG, Harris detector, SUSAN and others yield high quality features. However, they are computationally very expensive. Authors of [RPD10] use machine learning approach for corner detection and call this detector as FAST (Features from Accelerated Segment Test), as it uses accelerated segment test for feature detection. This detector exhibits real time tendencies along with high levels of repeatability. However, it is not robust to high noise levels and is dependent on a threshold. Authors of [MHB+10] enhanced the Accelerated Segment Test used in the FAST detector by making it more generic while increasing its performance. This was achieved by finding the optimal decision tree in an extended configuration space and demonstrating how specialized trees can be combined to yield an adaptive and generic accelerated segment test. The resulting method provides high performance for arbitrary environments.

Authors of [BTG06] proposed a new detector-descriptor combination called the SURF (Speeded-Up Robust Features). This detector is based on the Hessian matrix, but uses very basic approximations. It makes use of integral images to reduce the computational time and is called as Fast-Hessian detector. Fig.1.4 illustrates the key-regions obtained using FAST detector. Agrawal et al. proposed a simple but fast and efficient feature detector called CenSurE [AKB08]. CenSurE uses a simple approximation of bi-level Laplacian of Gaussian (BLoG) and exhibit better repeatability property than that of SIFT. They make use of Difference of Boxes (DoB) and Difference of Octagons (DoO) to get a better approximation of BLoG. Additionally, these filters can be implemented very efficiently using integral images. They show that their approach is 3 times faster than the SURF detector. Since filter response are computed for all pixels and at all scales, they argue that, at larger scales CenSurE is more accurate than SIFT and SURF. Fig.1.4 illustrates the key-regions obtained using SURF detector.

Performance Evaluation

Over the last few decades, many approaches have been proposed for interest point detection. To evaluate these detectors, various performance evaluation criteria and test data has been proposed. Some of the initial comparison methods can be found in [DN81, KR82, OAZ]. Among the recently proposed evaluation methodologies, the repeatability criterion as proposed by Schmid et al. [SMB00] stands out. The repeatability score for a given image pair is calculated as the ratio between the number of point-to-point

[1] `https://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html`



(a)SURF regions (b) FAST features .

Figure 1.4: Image showing the regions extracted using Hessian affine, Harris affine and MSER region detectors.

correspondences to the minimum number of points detected in the images. While calculating the repeatability score, only the points located in the portion of the scene present in both images are considered. Some of the above mentioned detectors are evaluated and explained in detail in [MS04, SLMS05]. From the comparison between the existing detectors [MS04, MS03] it is seen that Hessian based detectors exhibit improved stability and repeatability over their Harris-based counterparts. The determinant of the Hessian matrix has an advantage over the trace of the Hessian matrix, as it detects less elongated and ill-localised structures. Also, approximations like the DoG, DoB, Fast-Hessian etc. may increase the speed but reduces the detection accuracy. Features and properties of some of the above mentioned detectors are tabulated in Table.1.1.

1.2.2 Feature Descriptors

Once we have obtained the interest points/regions, the next stage in the image matching pipeline is to extract the features from the support regions around the interest points. These extracted features are encoded to form an unique identifier or a signature, which can be used to match the interest points in other images. These identifiers or signatures used for the purpose of image matching are called as image descriptors. The descriptor construction requires capturing the image characteristics of the pixels in the support region around an interest point. These characteristics could be in grey-scale or color values of the region, texture or geometry of the pooling region. The purpose of the constructed descriptor is to represent these characteristics in a compact and discriminative way, so that it can be used for various applications related to image matching [Low04], image stitching or panorama generation [BL03] , wide baseline matching [Bau00], object recognition [Low04], image indexing [SM97a], texture classification [LSP05a] and many more.

The way the image descriptor handles different geometric and photometric transformation, determines its robustness. An ideal image descriptor should be constructed in such a way that it is invariant to change in image scale and image rotation. Invariance to changes in viewpoint can be introduced whenever and wherever affine invariance is desired. On the photometric aspect, the image descriptor should exhibit invariance to

Feature Detector	*Corner*	*Blob*	*Region*	*Rotation invariant*	*Scale invariant*	*Affine invariant*	*Repeatability*	*Localization accuracy*	*Robustness*	*Efficiency*
Harris	*X*			*X*			+++	+++	+++	++
Hessian		*X*		*X*			++	++	++	+
SUSAN	*X*			*X*			++	++	++	+++
Harris-Laplace	*X*	*(X)*		*X*	*X*		+++	+++	++	+
Hessian-Laplace	*(X)*	*X*		*X*	*X*		+++	+++	+++	+
DoG	*(X)*	*X*		*X*	*X*		++	++	++	++
SURF	*(X)*	*X*		*X*	*X*		++	++	++	+++
Harris-Affine	*X*	*(X)*		*X*	*X*	*X*	+++	+++	++	++
Hessain-Affine	*(X)*	*X*		*X*	*X*	*X*	+++	+++	+++	++
Salient Regions	*(X)*	*X*		*X*	*X*	*(X)*	+	+	++	+
Edge-based	*X*			*X*	*X*	*X*	+++	+++	+	+
MSER			*X*	*X*	*X*	*X*	+++	+++	++	+++
Intensity-based			*X*	*X*	*X*	*X*	++	++	++	++
Superpixels			*X*	*X*	*(X)*	*(X)*	+	+	+	+

Table 1.1: Properties of some of the well known interest point detectors [MS04].

changes in linear as well as non-linear illumination. Additionally, the descriptor should be robust to errors in feature localization and should be able to handle partial occlusion. When all these transformations are taken into account, the effect of variations in lightening conditions and camera parameters on the image descriptor becomes negligible.

A vast number of image description algorithms has been proposed in the literature. Here, we mainly focus in detail on the image descriptors that has been introduced to characterize scale and affine invariant features. In this context, the image descriptors can be classified into floating point descriptors and binary descriptors. In the following section, we discuss some of the popular floating point and binary descriptors.

Floating-point descriptors

Floating point descriptors can be further grouped as

1. Gradient based descriptors(HoG, SIFT and its variants, DAISY, GLOH, MROGH,.....)
2. Descriptors based on filter response (SURF and its variants, RSD-HoG, Gabor filter Based Local Image Descriptors.....)
3. Descriptors based on local intensity pattern(LBP and its variants, LIOP, OSID,.....)
4. Descriptors based on second order statistic such as curvature, ridges, valleys (HSOG, Curvature histograms,...)

Gradient based descriptors

HoG

Histogram of Oriented Gradients (HoG) [DT05] was first introduced by Dalal and Triggs in the year 2005. HoG is based on a simple concept of binning the orientation of the pixel gradients over a dense grid of overlapping blocks. This was initially introduced for pedestrian detection. HoG is designed mainly to operate on the raw image data without introducing filtering artefacts that removes fine details. For the application of pedestrian detection, they use Raw RGB image with no color correction or noise filtering. There, the algorithm is based on the sliding window concept and they prefer to use a 64x128 sliding detector window. Within this detector window, a total of 8x16 or 8x8 pixel block regions are defined for computation of gradients. For each 8x8 pixel block, using the $[-1, 0, 1]$ mask in the x and y direction, 64 local gradient magnitudes and directions are computed. Separate gradients are calculated for each color channel. Local gradient magnitudes are binned into a 9-bin histogram of gradient orientations, quantizing dimensionality from 64 to 9, using bilinear interpolation. Finally, a normalized unit length value obtained from the gradient magnitude histogram is used to form the final HoG descriptor.

Many extensions of HoG has been proposed and amongst them is the well known SIFT descriptor. The SIFT descriptor construction process [Low04] is similar to that of HoG. Bosch et al [BZM07], combined both the image pyramid representation proposed by Lazebnik et al [LSP06] and the HoG representation proposed by Dalal et al [DT05] to form the PHoG descriptor. In their work, the local shape of the image patch is captured by the distribution over edge orientations within a region and spatial layout by tiling the image into regions at multiple resolutions. The descriptor is constructed by capturing the

histogram of orientation gradients over each image subregion at each resolution level and is used for image classification.

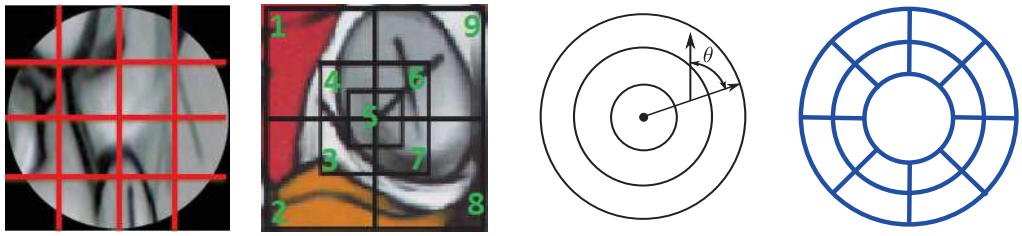
Laptev et al [LMSR08] bins the motion features obtained from optical flow to form a 90 dimension descriptor called Histogram Of Flow (HOF). Here, the authors define a 3D grid of size 3x3x2 along x,y and z(t) directions respectively, around the encompassing space-time area and compute for each cell of the grid a 5-bins histogram of optical flow. This method is used for action recognition in videos. In the same work [LMSR08], they combine the HOF with HoG to obtain an improved performance for action recognition in videos.

Klaser et al [KMS08] extended the HoG to 3D where in they proposed a new descriptor based on histograms of oriented 3D spatio-temporal gradients called HoG3D. In the initial stage, the support region around a key-point is divided into a grid of gradient orientation histograms. Each histogram is computed over a grid of mean gradients following which, each gradient orientation is quantized using regular polyhedrons and each mean gradient is computed using integral videos. These mean gradients from different regions around a key point are then concatenated to form the final HoG3D descriptor. This descriptor is used for action recognition in video sequences. There are different variants of HoG that have been proposed such as; authors of [CTC+12] compressed the HoG using different schemes to form compressed histogram of oriented gradients (CHoG) descriptor. Authors of [WHY09] combined HoG with that of local binary patterns(LBP) for human detection. Fischer et al [FB14], embed the curvature in the HoG frame work and used it for object detection and matching. However, their approach gives weak results for image matching. In our work [VMDM15c, VMDM15b] HoG is used in the descriptor construction stage.

SIFT and its variants

Scale Invariant Feature Transform (SIFT) is one of the widely used image descriptor in the field of computer vision. It was first introduced by David Lowe in [Low04, Low99] as a detector and descriptor pair that encodes the image information in a localized Histogram of Oriented Gradient (HoG) framework. Authors of [MS05] have confirmed that the SIFT descriptor can also be extracted from other region detectors as explained and have obtained good performance.

SIFT consists of 5 major stages: (1) Scale-space extrema detection; (2) feature point localization; (3) orientation assignment; (4) feature point descriptor and (5) Normalization. In the first stage, potential feature points are extracted by searching over all scales and image locations by using the above described DoG operation. The DoG is a close approximation to the scale normalized LoG, which is essential for true scale invariance. Thus, the obtained locations corresponds to the most stable features with respect to scale variances. In the second stage, candidate features are refined by eliminating the features that have low contrast and that are poorly localized along the edge. Final feature points are then selected based on the stability measures and are eliminated if unstable. In the third stage, a dominant orientation is assigned to each feature point based on local image gradient direction, where histogram of gradient orientations from the feature point neighbourhood is used to determine the dominant orientation. In the fourth stage, a regular sampling grid of 16x16 located around the key-point is used for sampling. This regular grid is split into blocks of 4x4 and the information in each block is encoded in the HoG of 8 bins as shown in Fig.1.5. Thus, each block contributes 8 dimensions to the final feature



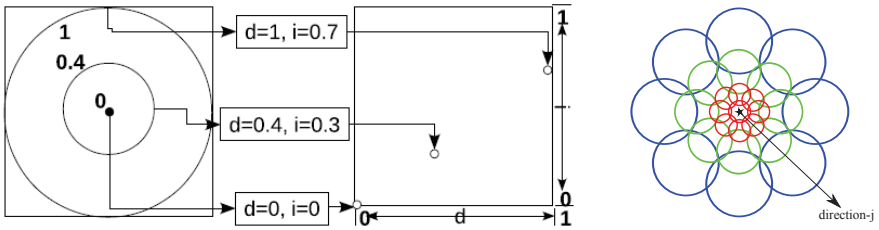
(a) (b) (c) (d).

Figure 1.5: (a)SIFT sampling grid (b)Sampling used in FIND (c)RIFT sampling pattern (d)Circular sampling used in GLOH .

vector. Then, the entries from all the blocks are concatenated to form a feature vector of dimension 4x4x8= 128 dimensions/length. In the final stage, the 128 dimension vector is normalized to unit length, thus adjusting for the changes in image contrast. Later, the complete vector is clamped to a maximum value of 0.2 and is again normalised to unit length, removing the non-linear illumination changes. Thus, the constructed SIFT descriptor is invariant to scale, rotation, illumination and partially invariant to affine changes.

Many variants and extensions are proposed to improve on the existing SIFT descriptor and one such extension is, the Affine invariant SIFT (ASIFT) [MY09]. Affine invariant extension of SIFT exhibits complete affine invariance. While, MSER, Harris-affine, and Hessian-affine normalizes all the six affine parameters, ASIFT simulates three parameters and normalizes the rest. The scale and the changes of the camera axis orientation are the three simulated parameters and the other three, rotation and translation parameters are normalized. More specifically, ASIFT simulates the two camera axis parameters and then applies SIFT, which further simulates the scale and normalizes the rotation and the translation. The authors introduce a parameter called *transition tilt*. This parameter measures the degree of change in viewpoint from one view to another. Yang et al. [YCWQ14] presented a new affine-invariant descriptor called the Low-rank SIFT. This is an extension to the original SIFT descriptor. Unlike ASIFT, this descriptor achieves complete affine invariance without the need for simulation over affine parameter space. Low-rank SIFT is based on the observation that, local tilts that are caused by the change in camera axis orientation, could be normalized by converting local patches to standard low rank forms. Further, they achieve scale, rotation and translation invariance similar to that of the SIFT descriptor. This method is mainly applicable only on objects with regular structures.

Flip or flip-like transforms are commonly observed in real world applications due to artificial flipping or symmetric pattern of an object. In that case, SIFT results in poor matching performance. Mirror and inversion invariant generalization for SIFT descriptor (MI-SIFT) [MCS10] tries to tackle this problem by operating directly on the SIFT descriptor and transforming it into a new descriptor that is flip invariant. This can be achieved by explicitly identifying a group of feature components that are disorderly placed as a result of flip operation. They label 32 such groups and represent each group with four moments that are flip invariant. But, this descriptor is not discriminative. Additionally, from their experiments it is clear that, the MI-SIFT results in more than 10% performance degradation when compared to SIFT, its for non- flip transformations.



(a) (b) .

Figure 1.6: (a)SPIN image sampling pattern (b)DAISY pooling strategy .

Authors of [ZN13] introduce a new descriptor called F-SIFT that preserves the original properties of SIFT, while exhibiting flip invariant properties. Initially, the F-SIFT algorithm estimates the dominant curl of a local image patch and then geometrically normalizes this image patch by flipping, before computing the SIFT descriptor. They demonstrate the superiority of this descriptor on applications related to large-scale video copy detection, object recognition, and detection. FIND [GC10] is another descriptor that uses overlap partitioning and scans the 8-directional gradient histograms by following a predefined order as seen in Fig.1.5. In this method, the descriptors produced before and after a flip operation are also a mirror of one another. In other words, a descriptor generated as a result of flip can be recovered by scanning the histograms in reverse order. Some of the other flip invariant descriptors are found in the literature [XTZ14, GC12].

SIFT descriptor was initially introduced for 2D images and later, the authors of [SAS07a] have extended this to 3D, in order to address 3D images(volumetric images) and videos such as MRI data. They call this descriptor 3D-SIFT. Initially, they compute the overall orientation of the neighbourhood around the key-point. Later, they compute the sub-histogram which will encode the final 3D SIFT descriptor. Cheung et al. [CH07] extends the SIFT technique to scalar images of arbitrary dimensions. This process involves using hyper-spherical coordinates for gradients and multidimensional histograms to create the feature vectors. This is used to determine accurate point correspondence between pairs of medical images (3D) and dynamic volumetric data (3D+time).

SIFT can find matches between images that have unique local neighbourhoods. But, it fails to deal with the images that have multiple similar regions. The reason for such a failure is that it doesn't take into account, the global information around the key-point. The authors of Global SIFT [MDS05] augment the SIFT descriptor with global context vector that contains curvilinear shape information from a much larger neighbourhood. This increases the robustness to deal with 2D non-rigid transformations, since the points are more effectively matched individually at a global scale rather than constraining multiple matched points that is to be mapped via a planar homography [MDS05] .

The various SIFT extensions that are discussed in previous paragraphs, are designed mainly for grey scale images while neglecting the color information of the objects. So, when two objects of different color are matched they may be considered as same. However, by introducing color information as a feature, one would improve the matching and object recognition tasks. To leverage the benefits offered by the color components, some researchers have extended the SIFT descriptor to color images. Authors of [BZM08] compute the SIFT descriptor over Hue (H), saturation (S) and Value (V) channels. This

results in a descriptor of length 384 and 128 dimensions per channel. The properties of the H and S channels such as, scale invariance and color shift invariance are incorporated into this descriptor. However, invariance to light color changes are restricted to only the V portion of the descriptor. But, the problem faced with this is that, hue is unstable at low saturation. Van de Weijer et al. [vdWGB06] fuse the histogram of the hue channel with the SIFT descriptor. By doing so, they are able to address the instability of the hue around the grey axis. Thus, the obtained HueSIFT descriptor is scale-invariant and invariant to shift color changes where the SIFT component of the descriptor is invariant to illumination color changes. Authors of [vdSGS08] extract SIFT features from different color spaces such as Opponent color space to form OpponentSIFT descriptor and transformed color model to form the Transformed color SIFT descriptor.

Authors of [BL02] used a combination of SIFT along with the normalised RGB model, which achieves partial illumination invariance in addition to its geometrical invariance. But, the primitive color model used limits the color invariance property of the descriptor. The authors of [AF06] propose an extension for SIFT descriptor in a color invariant space, called ColoredSIFT. Thus, the obtained descriptor is more robust than the conventional SIFT descriptor for color and photometrical variations which is achieved by using the color invariance model proposed by Geusebroek et al. [GvdBSG01] and geometric invariance by constructing the SIFT descriptor in the color invariant space. Brown et al. [BS11] introduce multi-spectral information to the SIFT descriptor and use it for scene category recognition.

Large computation complexity of the SIFT detector and descriptor pair has posed great challenges for real time/embedded implementations. To deal with real time scenario, Zhong et al [ZWY+13]. introduced a new SIFT architecture, by integrating it into FPGA and DSP. Here, the FPGA architecture for the feature detection step in SIFT is optimized and further they optimize the implementation of the description stage by using a high-performance DSP. By doing this, they are able to achieve real time performance. Acharya et al. [AB13] propose a parallel implementation of SIFT using GPU from which they achieve a frame rate of 55 Frames Per Second, for an image of size 640 x 480. The reason for such an increase in frame rate is, the introduction of a novel combined kernel optimization stage. Another real time implementation of SIFT is VF-SIFT [ARG10].

SIFT and its variants have found importance in many applications such as scene recognition [FBA+06, MP04b, RLD07, VvHR05, YC07, SAS07b, GL06], image registration [ZCSS13, CSS09, GXXS13, GSSF13], image mosaic [ZR14], object recognition [Low99, NPH+13, AWRG08, PWF09, PPC12], texture recognition, image retrieval [PPC12, YYQW11], robot localization, video data mining, building panoramas, and object category recognition, face recognition and many more. As the original SIFT descriptor is closed source, a few open source libraries like OpenSIFT [2] , VLFEAT [3], and an open implementation of SIFT [4] are available on the web.

Unlike SIFT and its variants which use rectangular/square partition grid, RIFT [LSP05b] uses a partitioning scheme by dividing a region along the log-polar direction, as shown in Fig.1.5. As in SIFT, the 8-directional histograms are computed for each division and then concatenated to form the RIFT descriptor. As the partition scheme is flip and rotation

[2] http://robwhess.github.io/opensift/
[3] http://www.vlfeat.org/
[4] http://web.eecs.umich.edu/~silvio/teaching/lectures/sift.html

invariant, RIFT is not sensitive to order of scanning. The spatially loose representation of features has resulted in a RIFT descriptor that is less distinctive than that of SIFT. The authors of [LSP05b] improve on the RIFT by proposing SPIN descriptor as in Fig.1.6. The SPIN preserves the flip invariant property while encoding more spatial information from a region as a 2D histogram of pixel intensity and distance from the center of the region [LSP05b].

To increase the robustness and distinctiveness of the SIFT descriptor, Mikolajczyk et al. [MS03] propose an extension to SIFT and RIFT descriptor called GLOH (Gradient location orientation histogram). Unlike the SIFT descriptor which uses the square grid of 4x4, GLOH uses log-polar location grid with three bins in radial direction (the radius set to 6, 11, and 15) and 8 in angular direction, which results in 17 location bins [MS03]. The GLOH sampling pattern is shown in Fig.1.5. Further, the gradient orientations are quantized in 16 bins. This results in 272 bin descriptor. By using the PCA step, they further reduce the descriptor size to 128.

Another popular descriptor which is an extension of SIFT, is the shape context descriptor [BMP02]. Unlike SIFT, which is based on gradient orientation histogram, shape context descriptor is based on the 3D histogram of the edge point location and orientations. Initially, Canny edge detector is used to extract the edges. The region around the key-point is quantized into nine bins using a log-polar coordinate system as displayed, with the radius set to 6, 11, and 15 and the orientation quantized into four bins (horizontal, vertical, and two diagonals). This results in a 36 dimensional descriptor.

For dense matching, Tola et al. [TLF10] propose a variant of SIFT and GLOH descriptor called DAISY. For wide-baseline applications, DAISY yields much better results than the pixel and correlation-based algorithms that are commonly used in narrow baseline stereo matching. Similar to SIFT, DAISY descriptor is a 3D histogram of gradient locations and orientations. Unlike SIFT, which uses weighted sums of gradient norms, DAISY uses the convolutions of gradients in specific directions with several Gaussian filters. As the histograms need to be computed only once per region and it can be reused for all neighbouring pixels, this approach results in efficient computation of the DAISY descriptor. While SIFT uses rectangular grid for sampling and GLOH uses circular grid, DAISY combines the two sampling approach to form the DAISY sampling/binning strategy as shown in the Fig.1.6.

Initially, for a given input image, a certain number of orientation maps, one for each quantized direction, are computed. Each orientation map represents the image gradient norms for that direction at all pixel locations. The orientation map is then convolved several times with Gaussian kernels of different standard deviation values, to obtain the convolved orientation maps. Since the Gaussian filters are separable, the convolutions can be implemented very efficiently. Hence, the DAISY descriptor is very efficient. In this method, the region around each pixel is divided into circles of different sizes, located on a series of concentric rings as shown in Fig.1.6. The radius of each circle is in proportion to its distance from the central pixel, and the standard deviation of Gaussian kernel is in proportion to the size of the circle. For each circle, an intermediate descriptor is constructed by gathering the values of all the convolved orientation maps with the corresponding Gaussian smoothing. Finally, all the intermediate descriptors are concatenated and normalized to unity, to form the final descriptor.

Descriptors based on filter response

In the field of computer vision, response from a filter or a bank of filters has been used in abundance. Gabor filters is one of the most popular filter used in applications related to face recognition, image matching, texture retrieval, classification, medical imaging and many more. Filters such as Haar filters/wavelets have found importance in image matching and in content based image retrieval. The popular SURF descriptor and its extensions, use Haar wavelets as its basis. Isotropic Gaussian, anisotropic Gaussian, half Gaussian, Eigen-filters, wavelet transforms etc has been used in feature detection. But, in this section, we mainly concentrate on image descriptors based on filter response.

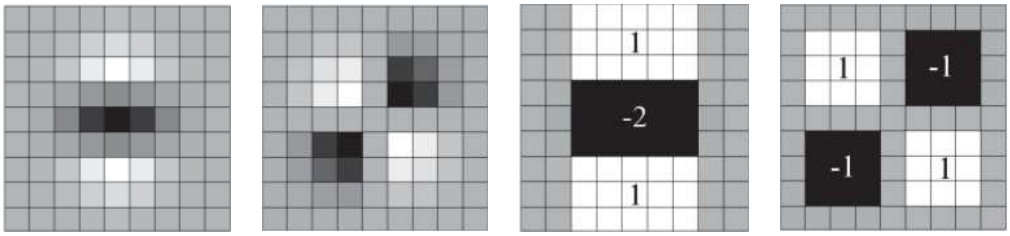
SURF and its variants

Another popular floating point detector-descriptor combination was proposed by Bay et al. [BTG06] called, the Speed Up Robust Features (SURF). Unlike SIFT, which exhibits high complexity and dimensionality, SURF exhibits low complexity with a reduced dimension of length 64. Like the SIFT algorithm, the SURF algorithm is a combination of detector and descriptor stages. The detector stage is made of the Hessian matrix approximation and an integral image which speeds up the feature detection process. The SURF detector uses four steps.

1. To speed-up the feature detection process, the detector stage uses integral images.
2. To determine the interest points, it uses an approximation of the Hessian matrix. Then, the maxima of the determinant is used for detecting the blob like structures.
3. To achieve scale-space representation, Gaussian approximated filters as shown in Fig.1.7 are adapted at each level of the filter size in scale space.
4. In the feature localization step, feature detection is performed using non-maxima suppression over three successive scales. The points that have the maxima of the determinant of the Hessian matrix are considered as feature points.

In the SURF descriptor construction, initially an orientation is assigned to the feature point. The orientation is computed by detecting, the dominant vector of the summation of the Gaussian weighted Haar wavelet responses under the sliding window split circle region by pi/3 [BTG06]. The final resulting descriptor is based on the sum of Haar wavelet responses. The region around the key-point is then split into smaller 4x4 sub-regions. From each sub-region, the horizontal Haar wavelet response d_x and the vertical Haar wavelet response d_y are extracted at 5x5 regularly spaced sample points. Then a final 64 length SURF vector $D = (\sum d_x, \sum d_y, |\sum d_x|, |\sum d_y|)$ is formed. Applying some restrictions, few extensions of the SURF descriptor such as SURF-36 and SURF-128 can be formed.

Like SIFT, SURF doesn't exhibit affine invariance properties. Pang et.al [PLYP12] propose an affine invariant version of the SURF descriptor called A-SURF. The concept of A-SURF is similar to that of ASIFT where both algorithms simulate two camera axis parameters. But, unlike A-SIFT, A-SURF algorithm uses a faster version of SURF. In applications where the camera remains more or less horizontal, the rotation invariance can be neglected. For such applications, Bay et al. [BTG06] propose a variant of SURF



(a) (b) (c) (d) .

Figure 1.7: (a),(b) Discretized Gaussians (c), (d) Approximations used in SURF .

descriptor which is not rotation invariant and calls this as Upright SURF (U-SURF). U-SURF can be considered as SURF with out rotation invariance. Since rotation invariance is absent, U-SURF is faster than SURF. Agarwal et al. [AKB08] propose a modified version of U-SURF called a modified Upright SURF (MU-SURF) descriptor. For each detected feature at a scale S, MU-SURF uses Haar wavelet of size 2S, to accumulate the responses in the horizontal and vertical direction for a $24S$ x $24S$ region. This region is further divided into $9S$ x $9S$ subregions, with an overlap of $2S$. The Haar wavelet responses in each subregion is weighted with a Gaussian centred on the subregion center. Then they follow the SURF approach for descriptor construction. Due to the overlap of the subregions and Gaussian weighting, MU-SURF handles boundaries better than U-SURF.

The original SURF descriptor is basically for 2-D images. For volumetric image data in medical imaging applications, Feulner et al. [FZA+11] extend the SURF descriptor to arbitrary number of dimensions called N-SURF. Initially, they generalize the concept of Haar-filters and rectangle filters to N dimensions. As in SURF, the image is sampled on a regular grid around an interest point. The samples are then split into b bins per dimension, resulting in b^n bins. For each sample, the gradient is approximated with N Haar-filters that are weighted with an N dimension Gaussian, centred at the interest point. From each bin, they extract a feature vector and concatenate the vector from all bins to form the final descriptor. Thus, the final descriptor has a dimension of $2Nb^N$. The standard SURF approach for assigning canonical orientation to the interest point cannot be directly generalized to more than two dimensions. To deal with this situation, gradient approximations are extracted inside a spherical region of radius ($r = 6s$) around a key-point, as in the original SURF. The final orientation is then determined using this set of gradients.

Although the SURF descriptor has succeeded in representing the nature of some underlying image patterns, it shows poor results while representing complex patterns. In P-SURF [LYH11], the authors improve on this aspect of the original SURF descriptor by introducing phase space to capture more structure information of local image patterns. In phase space, a single region represents a kind of relationship that exists between intensity changes. By building histograms on such regions, these relationships can be quantized. P-SURF consists of two stages: the feature representation for independent intensity changes and coupling description for these intensity changes. Now, they introduce phase space to model the relationship between the intensity changes and propose several statistic metrics for quantizing these relationships to meet practical demands [LYH11].

SURF, basically being a local image descriptor, fails to recognise the repetitive patterns. To help distinguish between the repetitive patterns in an image, Carmichael et al. [CLB10] proposed a framework for augmenting a SURF descriptor with a global context vector. They propose to compute the global context of a SURF feature-point with a technique that is similar to the one used in Global SIFT [MDS05] as explained in our previous section.

SURF descriptor is sensitive to rotation and viewpoint changes due to the gradient method used in the description stage of SURF. To overcome these limitations and to enhance the matching accuracy of the SURF algorithm, Kang et al. [KCL15] propose a modified SURF algorithm called MDGHM-SURF. Their approach is based on the Modified Discrete Gaussian–Hermite Moment (MDGHM), which uses a movable mask to represent the local feature information of a non-square image. Unlike SURF which uses first order derivative, MDGHM offers more feature information than SURF. In the initial stage, as in SURF, they use the integral images. In the detector step, the Hessian matrix is replaced with the MDGHM matrix that obtains more geometrical information from neighbours to detect more distinctive features. Later, in the scale-space representation step, they approximate the MDGHM matrix to reduce computation. The interest point localization step is similar to that of the conventional SURF, excepting for a replacement of the determinant of the Hessian matrix with the determinant of the MDGHM matrix. In the descriptor generation stage during the orientation assignment, they replace Haar wavelet response with MDGHM to represent the feature information more precisely. The remaining stages of the MDGHM algorithm is similar to that of the original SURF excepting that they replace the gradient magnitude and orientation of the descriptor, with the MDGHM-based magnitude and orientation.

Alcantarilla et al. [ABD13] propose a new family of multi-scale local feature descriptors called Gauge SURF (GSURF). GSURF descriptors are based on second-order multi-scale gauge derivatives and original SURF descriptor. To compute the multi-scale Gauge derivatives, the authors initially compute the derivatives in the Cartesian coordinate frame (x,y), and then compute the gradient direction for each pixel. After the computation, they obtain invariant gauge derivatives up to any order and scale, with respect to the new gauge coordinate frame [ABD13]. Similarly, they extract the first and second order Haar responses d_x, d_y, d_{xy}, d_{yy} from the region around the feature points. Using these responses, they calculate the second order Gauge derivatives d_{ww} and d_{vv}. Finally, they then calculate the four dimensional descriptor vector of length $D_G = (\sum d_{ww}, \sum d_{vv}, \sum |d_{ww}|, \sum |d_{vv}|)$. Thomee et al. [TBL10] proposed an image descriptor based on SURF, that combines interest points with visual words. The resulting descriptor which they named as TOP-SURF is compact, supports fast image matching, provides the flexibility to vary the descriptor size and exhibits superior performance than that of the original SURF descriptor. FPGA implementation of SURF is provided by svab et al. [SKFP09].

SURF, its variants and its extensions have found enormous applications in computer vision domain such as tracking [TCGP09], object recognition [TCGP09, CHYC15, CT11, SHK12], Image retrieval [JPG12], action recognition [JSFJ11], medical imaging [FZA+11] , Face recognition [DSHN09, DSC09, LWZ11], Iris recognition [MSM13], robot localization [VL07] and many more. Like SIFT, the original SURF is also a closed source. An open source version of the SURF descriptor can be found in [5].

[5] `https://code.google.com/p/opensurf1/`

Descriptors based on other Filters

SURF and its variants are based on Haar filter response. Many researchers have proposed descriptors based on response of other filters like Gabor filter, steerable filter etc. Schmid and Mohr [SM97b] use differential invariant responses to compute new local image descriptors. Differential invariant responses are obtained from a combination of Gaussian derivatives of different orders that are invariant to 2-dimensional rigid transformations. Steerable filters proposed by [FA91], steer the derivatives in a particular direction, given the components are of the local jet. Steering derivatives in the direction of the gradient, makes them invariant to rotation [MS03]. Mikolajczyk et al. [MS03] use the steerable filters to generate the image descriptors and use it for image matching. They compute the steerable filter derivatives up to fourth order, that is, the descriptor has dimension 14. Further, they use Mahalanobis distance to find the similarity between the descriptors obtained using the steerable filters. They compare the steerable filter to other low dimensional filters and conclude that the steerable filters provide the best low dimensional descriptors. Complex filter bank as proposed by [SZ02] are used in [MS03] for the generation of image descriptors. The complex filter bank is made of 15 filters and the response of these 15 filters is used as an image descriptor of size 15.

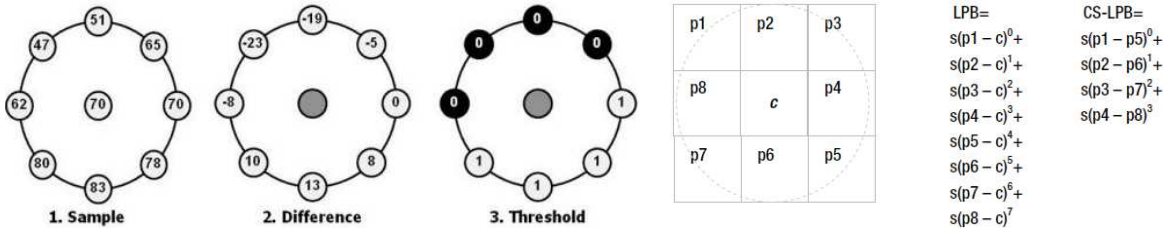
Osadchy et al. [OJL07] use oriented second derivative filters of Gaussians as an effective feature for capturing isotropic as well as anisotropic surface characteristics of an image. They restrict their method to filters of single scale. Zambanini et al. [ZK13] extends this approach of [OJL07] towards the construction of illumination invariant descriptor. In their method, they use real part of Gabor filters at multiple scales and spatial statistics to describe local image patches in an illumination invariant manner. Palomares et al. [PMD12] have come up with a local image descriptor issued from a filtering stage made of oriented anisotropic half-Gaussian smoothing convolution kernels. Other descriptors explained in chapter.3, chapter.4 and chapter.5 are also based on the same family of half filters. These half filters are explained in detail in chapter.2.

Koenderink et al. [KvD87] formulates a methodology based on Gaussian function and its derivatives, to capture the local geometry of the image. They achieve the rotation and affine invariance in the matching stage. Larsen et al. [LDDP12] follows a new approach for the construction of an image descriptor based on local k-jet, which uses filter bank responses for feature description. Lategahn et al. [LBS14] propose a new illumination robust image descriptor called the DIRD, based on the features obtained by using Haar wavelets. In their approach, the Haar features are computed for individual pixels and are normalized to L2 unit length. Thereafter, features are extracted from the pooling region. The concatenation of several such features forms the basis DIRD vector. To achieve fast matching, these features are then quantized to maximize entropy to form a binary version of DIRD.

Intensity based descriptors

The local binary patterns (LBP) [OPH96, OPM02a, OPM01] were introduced by Ojala et al. as a texture operator. LBP generates a descriptor using a set of histograms from the local intensity neighbourhood present around each pixel. Initially, 8 neighbouring pixels are chosen around each pixel. Then, the difference between the center pixel and each of the eight neighbouring pixels are considered and finally, depending on the binary test, 1 is

assigned to the neighbouring pixel if the intensity difference is greater than zero. Otherwise, a zero is assigned. This can be seen in the Fig. 1.8. Thus, the obtained descriptors based on LBP is simple but very effective. The LBP's have therefore, succeeded in replacing SIFT, SURF and its variant. Over the years many variants of LBP descriptors has been proposed. For our purpose, we concentrate only on some of the few important and popular LBP descriptors and its variants.



(a) (b)

Figure 1.8: Summary of (a) LBP (b)CS-LBP.

Heikkila et al. [HPS06] proposes a new descriptor called a Center Symmetric Local Binary Pattern (CSLBP), which combines the advantages of SIFT descriptor and the LBP operator. CSLBP is constructed similarly to that of SIFT, but the gradient features used in the SIFT descriptor is replaced with features extracted by a center symmetric local binary pattern operator, the binary pattern being similar to LBP operator. Initially, they enhance the region around the key-point with an edge-preserving adaptive noise-removal filter. Later, a feature for each pixel in the region is extracted using a CSLBP operator. Eight intensities around each pixel are chosen for the same. These intensities are then spread evenly around the pixel at every 45°. Each intensity is compared with the intensity in the symmetric position to form a four bit feature vector. Unlike SIFT, they weigh the features with a simple uniform weighting scheme. Finally, a 4x4 Cartesian grid is used to build the descriptor and then, a CSLBP histogram is built for each cell. The final descriptor is built by concatenating the descriptors from all the cells and normalizing it. Thus, the obtained descriptor has a length of 256.

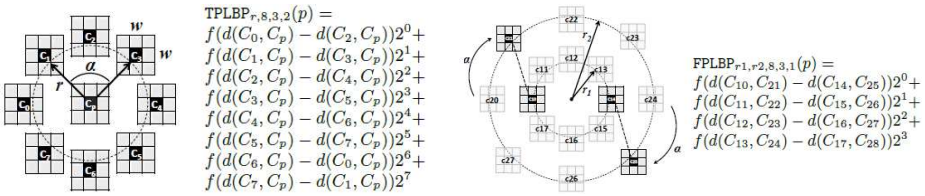
Authors of [ZCX+07] encode the rectangular regions by local binary pattern operator to form the Multi-block LBP descriptor. They make use of integral images for descriptor construction. Unlike LBP which uses intensity values in its computation, MBLBP uses the mean intensity value of image blocks. Thus, the constructed descriptor captures more information about the image structure than say a descriptor based on Haar like features. They use this descriptor for face detection.

Wolf et al. [WHT08] proposed two novel patch based LBP descriptors to improve on the performance of the original LBP descriptor and they are:

1. Three-Patch LBP descriptor (TPLBP): Here, for each pixel in the region around a key-point, they consider a wxw patch centred on the pixel and S additional patches distributed uniformly in a ring of radius r around it. Then, they consider a pair of patches, α-patches apart along a circle and the obtained values are compared with that of the central patch. The resulting code has S bits per pixel [WHT08]. This results in each pixel having S bits per pixel. Hence, the processes generates a

descriptor similar to that of CSLBP. The feature computation process is shown in the Fig.1.9.

2. Four-Patch LBP descriptor (FPLBP): For each pixel in the region around the keypoint they consider two circles of radii r_1 and r_2 with the same pixel as its centre. As in TPLBP, they consider S patches of size wxw evenly placed on each ring. To generate the features, they compare the two central symmetric patches in the inner ring with the two central symmetric patches in the outer ring, positioned α patches away along the circle [WHT08]. Thus, each pixel has a feature of length S. Then, the same procedure as in TPLBP is used to construct the descriptor. The feature computation process is shown in the Fig.1.9.



(a) (b)

Figure 1.9: Summary of (a) Three Patch LBP [WHT08] . (b) Four patch LBP [WHT08]

Although LBP captures the local structures effectively, it is not rotation invariant. Many approaches have been used to achieve rotation invariant LBP descriptor. In one of the approaches, the authors of [PHZA11] propose a new rotation invariant LBP descriptor in which the local LBP is shifted bitwise circularly to obtain the minimum binary value. This minimum value LBP is used as a rotation invariant descriptor and is captured in the histogram. Thus, the obtained RILBP is computationally very efficient. In another method, the authors of [AMHP09] combine LBP with Fourier features to obtain rotation invariant LBP. Like SIFT and SURF, LBP is not a global descriptor. In order to potentially use the local and global information in texture images, authors of [GZZ10] propose a global rotation invariant matching with local variant LBP features. The authors of [IKM08] extend the LBP approach by incorporating Fuzzy logic in the LBP mechanism and call it, Fuzzy Local Binary Pattern (FLBP). Introducing the fuzzy logic concepts in the descriptor construction, allows the FLBP to contribute more to the feature descriptor. To deal with volumetric data such as medical images, dynamic textures etc, authors in [ZP07] have come up with a new descriptors called volume LBP (VLBP) and TOP-LBP.

LBP and some of its variants were originally developed for gray scale images. Latter, opponent color LBP (OCLBP) [MP04a] was introduced to deal with color and texture jointly. In OCLBP, LBP is extracted for each independent channels and LBP features are extracted for different channel pairs such that the center pixel is taken from one channel and the neighbouring pixels from the other. In total, 3 pairs of channels are used and the rest are ignored as they provide no information. Finally, a total of 6 histograms are used which are concatenated to form the OCLBP descriptor whose dimension is six times larger than the original LBP descriptor. For the application of visual object class recognition, authors of [ZBC10] use multi-scale color LBP descriptor. For color constant

image indexing, the authors of [CF06] use 3D histograms of LBP values computed from LBP images of three channels.

LBP and its variants has been used in abundance, in applications related to face recognition. To name a few LB variants, Local Gabor Binary Patterns (LGBP) [ZSG+05], Heat Kernel Local Binary Pattern (HKLBP) [LHZW10], Elliptical Binary Patterns (EBP) [LC07], Local Line Binary Patterns (LLBP) [AP09], Three-Patch Local Binary Patterns(TPLBP) [WHT08], Four-Patch Local Binary Patterns (FPLBP) [WHT08], Improved LBP (ILBP) [JLLT04], Local Ternary Patterns (LTP) [THL09], Probabilistic LBP [TT07], Local Derivative Patterns (LDP) [ZGZL10] are some of the LBP and its variants. In Medical image analysis, LBP variants such as Elongated Quinary Patterns (EQP) [NLB10], Elongated Ternary Patterns (ELTP) [OP99] and volumetric LBP [ZP07] has been used. LBP and its variants are the most sought after descriptors when it comes to texture classification, analysis and retrieval. Some of the variants of LBP used for applications related to texture are Local Edge Patterns (LEP) [YC03], Median Binary Patterns (MBP) [HSZ07], Fuzzy Local Binary Patterns [IKM08], Bayesian Local Binary Patterns (BLBP) [HAP08], Adaptive LBP (ALBP) [GZZZ10], LBP variance (LBPV) [GZZ10] etc.

External factors such as temporal changes in illumination, viewpoint dependent illumination, shadowing, variations in camera parameters, non-linear camera response etc results in complex brightness changes. Descriptors such as SIFT, SURF, DAISY etc are invariant to intensity shift or affine brightness changes and fail to handle the complex brightness changes. To overcome this issue, some of the authors use relative ordering of the pixel intensities rather than the original intensities. This is based on the observation that, although the pixel intensities in the corresponding locations may vary due to the variations in illumination or other camera parameters, the relative ordering of the pixel intensities in the region remains unchanged if the brightness change function is monotonically increasing [TLCT09].

Tang et al. [TLCT09] propose a new image descriptor called ordinal spatial intensity distribution (OSID), which exhibits invariance to any monotonic increase in brightness change. Unlike the above described descriptors, where the gradient orientations or filter responses are compressed in a 2D histogram using the raster scan order, OSID is generated based on intensity ordering and spacial subdivision spaces. Initially, they pre-process the *dxd* patch around the key-point by smoothing it with a Gaussian filter. This is followed by generating the ordinal distribution, where the pixels in the patch are grouped into N bins where each bin has pixels with similar ordinal pixel intensities. For spacial distribution, the pixels in the *dxd* patch are labelled based on npies spatial subdivisions. First, the ordinal-spatial 2-D histogram is constructed and is followed by spacial 2-D histogram. Finally, the two histograms are concatenated to form the final OSID descriptor.

Most of the patch based descriptor such as SIFT, GLOH etc use dominant orientation to achieve rotation invariance. In some cases, the patch may have 2 or 3 dominant orientation and ambiguity arises as to which is the correct orientation. So, the dominant orientation introduces an error. Fan et al. [FWH12] tries to obtain a true rotation invariant descriptor by generating gradients in a rotation invariant way and adaptively pooling these gradients based on their intensity order to capture the spatial information. Further, they use multiple support regions to enhance the discriminative ability of the descriptor. This is approached in two ways: 1) By using gradient based local features to generate the MROGH (Multi-Support Region Order-Based Gradient Histogram) descriptor and 2) By

using intensity-based local features to generate MRRID (Multi-Support Region Rotation and Intensity Monotonic Invariant Descriptor) descriptor.

Wang et.al [WFW11] propose a new method for feature description based on intensity order called Local Intensity Order Pattern (LIOP). In their descriptor, they use ordinal information in a novel way. Initially, they smooth the region around the key-point using a Gaussian. Next, an overall intensity order is used to divide the local patch into its sub-regions, called ordinal bins. Later, by considering the intensity relationship between the neighbouring sample pixels, the local intensity order pattern for each pixel is calculated. Additionally, in this step, they propose a permutation based encoding scheme, which compresses the dimension of the LIOP of the pixel. The intermediate LIOP descriptor is constructed by accumulating the LIOPs of pixels in each ordinal bin respectively. Finally, the intermediate descriptors from all the ordinal bins are concatenated to form the final LIOP descriptor.

Descriptors based on second order Image statistic such as curvature, ridges, valleys

In the field of computer vision, there exists an extensive body of work on the importance of curvature for the tasks related to object recognition, image retrieval, image matching etc. Several methods have been proposed to estimate the curvature of a planar curve in images. Lewiner et al. [LJLC04] propose a new method to estimate the curvature based on weighted least square fitting and local arc length approximation. Mokhtarian et al. [MM86] calculated the curvature of a planar curve by representing boundary as a parametric function of an arc length. Later, they convolve the image with Gaussian filters at different scales and detect the inflection points as stable zero-crossing points. Han et al. [HP01] proposed a more stable method for calculating the discrete curvature of planar digital boundaries by accumulating the distance from a point in the boundary to a chord specified by moving end points. Depending on the boundary shape, positive or negative distances are obtained and the values are then accumulated as the chord is moved [HP01]. Only few authors have been successful in constructing a image descriptor by directly using the curvature information.

Monroy et al. [MEO11] show that by integrating the curvature information would substantially improve the detection results over descriptors that solely rely upon histograms of orientated gradients. In their method, they directly encode curvature from shape and use this information along with the orientation of gradients to perform image matching and object detection. Initially, they extract edges using the Berkeley edge detector [MFM04]. Then, they approximate the curvature for planar boundaries using the chord-to-point distance accumulation [HP01] and later, this curvature information is embedded in the histogram of orientation framework to form the final descriptor. Fischer et al. [FB14] propose a way to capture the details described by the local curvature. They extend the idea of orientation histogram to curvature and propose a descriptor made of direction and magnitude of curvature. Instead of using the parametric curve segments as in [MEO11], they compute the curvature using the per-pixel filter. Additionally, to increase the matching performance, they include the sign of curvature which is different from the sign of gradient.

Zitnick [Zit10a] proposes an image patch descriptor based on edge position, orientation and local linear length. He names this descriptor Binary Coherent Edge Descriptor

(BiCE). It is based on the hypothesis that the presence and not the magnitude of edges provides an informative measure of patch similarity that is robust not only to illumination and pose changes but also to intra-category appearance variation [Zit10a]. BiCE binarizes the edge histogram to encode the edge locations, orientations and lengths. Initially, they locally normalizes the image patch gradients to remove relative gradient information. This is followed by binning the normalized gradients using position, orientation and local linear length of an edge. Finally, the normalized gradient histogram is binarized to encode the presence of edges.

Ram et al [RBS09] propose a new method for detection and matching of vascular landmarks in retinal images, using histograms of curvature. In this method, they use Hessian filter for vessel enhancement. Based on the local curvature computed at multiple scales, they are able to localize vessel junctions as landmarks. They extract a curvature orientation histogram over a patch around every vessel point and further, the entropy of this histogram is calculated which is used to determine vessel junctions. Another advantage of this method is that, the curvature orientation histogram implicitly captures the vessel branching information at a landmark point, including the angles between them. In retinal images, this information remains invariant to rigid transformation [RBS09]. They use these informations to establish correspondence between sets of landmark points obtained from images related by rigid transformations.

Recent studies in psychophysics and physiology on human vision have shown that the first order gradient information is far from being sufficient and accurate in capturing the perceived visual features of human beings. Additionally, some studies on human vision further suggest that neural image is made of surface, consisting of second order image properties such as cliffs, ridges, summits, valleys, or basins [HZWC14]. Huang et al. [HZWC14] capture these information by local curvatures of differential geometry and constructs a second order descriptor called histogram of second order gradients (HSOG). Initially, they compute a set of first order gradient maps for each quantized direction. They then use these oriented gradient maps to extract the histogram of second order gradients and finally, the histogram from all the oriented gradient maps is concatenated to form the HSOG descriptor. In addition to this, they explore the concept of scale space to further reinforce the descriptive completeness of local shape changes and thereby, increasing the discriminative power and performance. Thus, the obtained descriptor has a very high dimension in the range of hundreds. They further use PCA to reduce the dimensionality and resulting is a HSOG descriptor of length 256.

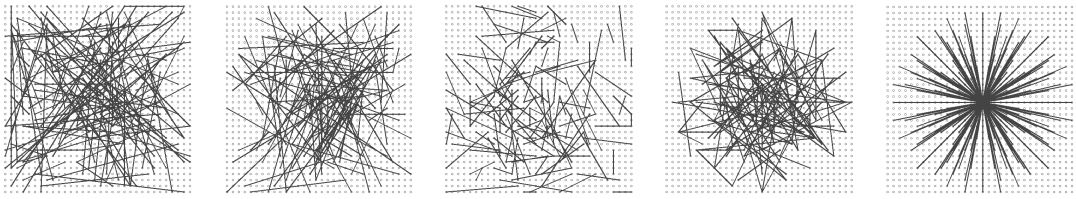
Local Binary Image Descriptors

With an increase in the image database size and the advent of camera enabled mobile devise, a new branch of computer vision has come into existence that has enabled mobile devices such as smartphones, tablets etc and wearable devices such as google glass, Microsoft hololens etc, which requires vision systems that are accurate and computationally very efficient. Feature point descriptors forms the basis for many computer vision applications such as augmented reality, panorama stitching, 3D reconstruction, camera localization and many more. Applications utilizing these algorithms on mobile and augmented reality platforms have to deal with limited storage and poor computational capabilities. This is where the binary descriptors becomes an alternative option for floating point descriptors. The initial approach was to first compute a floating-point descriptor

then binarize it. Since this approach is computationally expensive, the new approach is focused on directly computing the binary descriptors from local image patches.

The main idea behind binary descriptor is that, each bit is independent and the Hamming distance can be used as similarity measure. Generally, binary descriptors are made of three stages. In the initial stage, a sampling pattern is chosen. The sampling pattern shows where exactly to sample the points in a region around the descriptor. This is followed by the orientation assignment stage, where the key-point is assigned to an orientation, to achieve orientation invariance. In the final stage, sampling pairs are chosen to construct the final descriptor. Some of the popular and widely used binary descriptors are BRIEF [CLSF10], ORB [RRKB11], BRISK [LCS11], FREAK [AOV12], LDB [YC12].

Calonder et al. [CLSF10] propose to use binary strings as an efficient feature point descriptor called, the Binary Robust Independent Elementary Features (BRIEF). The proposed descriptor is computed using simple intensity difference tests and exhibits high discriminative property using fewer number of bits. Further, they use the Hamming distance to evaluate the descriptor similarity. Initially, the image patch or the region around the key-point is smoothened by a Gaussian function and is followed by pooling/sampling stage. As in some of the other binary descriptors, a fixed sampling pattern has not been used. The sampling pairs are chosen randomly from the image patch. Some of the sampling pairs are shown in the Fig.1.10. In the next stage, they perform the binary tests on these sampling pairs to obtain the descriptor. This is followed by the matching stage which uses the Hamming distance for fast matching. One of the disadvantage of BRIEF is that, it is not rotation invariant.



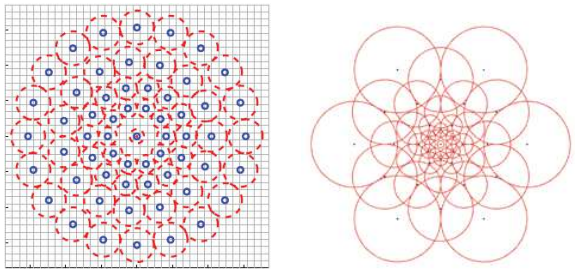
(a) (b) (c) (d) (e)

Figure 1.10: Different sampling pattern used in the construction of BRIEF descriptor [CLSF10]

Rublee et al. [RRKB11] propose a new binary descriptor based on FAST detector [RPD10] and BRIEF descriptor [CLSF10] called Oriented FAST and Rotated BRIEF (ORB). Unlike BRISK, ORB uses an orientation compensation mechanism to achieve rotation invariance. ORB also learns the optimal sampling pairs is resistant to noise and is two orders of magnitude faster than SIFT [RRKB11]. In the initial stage, ORB uses the FAST detector to detect the key-points and is followed by assigning the orientation using the intensity centroid. Ideally, the sampling pairs should have less correlation and exhibit high variance. But, the sampling pairs will be uncorrelated such that each new pair will bring new information to the descriptor, thus maximizing the amount of information the descriptor carries. Lesser correlations of the sampling pairs is required so that, each new pair will bring new information to the descriptor and maximizes the information carried by the descriptor. Whereas, high variance makes the feature more discriminative. In the

next stage, the ORB uses a learning algorithm to learn the sampling pairs to make sure that the sampling pairs exhibit these two properties. This learning stage produces a set of 256 relatively uncorrelated sampling pairs with high variance. Binary intensity tests are conducted on these sampling pairs to form the final ORB descriptor.

Leutenegger et al [LCS11] propose Binary Robust Invariant Scalable Key-points (BRISK) which is a new method for key-point detection, description and matching. In the initial stage, a new scale space key-point detection method is proposed. Here, points of interest are identified across both the image and the scale using a saliency criterion. To further enhance the efficiency, key-points are detected in octave layers of the image pyramid as well as in-between the layers. For sampling the pixels, unlike the BRIEF and ORB, BRISK uses a predefined sampling pattern. The sampling pattern is made of scaled concentric circles as shown in figure 1.11. This sampling pattern is then applied around the key-point and the dominant orientation to the key-point is assigned. Finally, the oriented BRISK sampling pattern is used to obtain pairwise brightness comparison results that are assembled into the binary BRISK descriptor [LCS11].



(a) (b)

Figure 1.11: sampling pattern used in the construction of (a) BRISK (b) FREAK

Alahi et al. [AOV12] propose a new key-point descriptor called Fast Retina Key-point (FREAK). It is based on the retina of the human visual system. Similar to BRISK, FREAK uses a hand crafted sampling pattern that is, by using the circular retinal sampling grid with the difference of having higher density of points near the center. By doing so, the density of the points drops exponentially. The main difference between BRISK sampling pattern and the FREAK sampling pattern is the exponential change in size and overlapping receptive fields. Similar to ORB, FREAK uses machine learning technique to learn the optimal set of sampling pairs. To achieve rotation invariance, FREAK measures the orientation of the key-point and rotates the sampling pairs by the measured angle. FREAK's approach for orientation assignment is similar to that of BRISK. Another advantage of FREAK is that it is an open source software.

For augmented reality applications, Yang at al. [YC12] propose a highly efficient, robust and distinctive binary descriptor called, the Local Difference Binary pattern (LDB). LDB makes use of simple intensity and gradient difference tests on pairwise grids within the patch to generate a binary string. Additionally, it uses a multi-level grid pattern to capture the distinct patterns of the patch at different spatial granularities. Like many floating point descriptors, above explained binary descriptors also suffer from orientation estimation errors and limited description abilities.

To address these problems, Xu et al. [XTFZ14] proposed a new binary descriptor based

on the ordinal and spatial information of regional invariants called OSRI. This descriptor is generated over a rotation invariant sampling pattern. The OSRI descriptor is obtained by performing difference tests of regional invariants over pairwise sampling-regions, instead of difference tests of pixel intensities. This approach enhances the discriminative ability of the descriptor. As in [WFW11], here also the pixels are re-ordered in accordance to their intensities and gradient orientations to achieve rotation and illumination changes. Additionally, OSRI uses a cascade stage to reduce the matching time. This cascade stage rejects the non-matching descriptors at an early stage, by comparing just a small portion of the whole descriptor.

Some of the other binary descriptors in the literature are MOBIL [BOZB14] a binary descriptor based on moments, Ultra short binary descriptor (USB) [ZTH+14], Discrete Robust INvariant Keypoints (DRINK) [GC14], EDGESIFT [ZTL+13] and many more. Detailed evaluation of some of the binary descriptors is provided in [MM12, HDF12].

Learning Image descriptors

Almost all of the above discussed image descriptors rely on parameters that need to be hand tuned. These parameters include Gaussian parameters, Number of orientation bins, descriptor size and many more. Some of the authors in [Low04], [ZK13] have tried to manually optimize the descriptor performance by varying the parameters. This approach works only when there are limited number of parameters and it also proves to be tedious and unrealistic in the presence of large number of parameters. To overcome this issue, the authors of [WB07] tries to improve on the state of art in local descriptor matching by learning optimal low-level image operations using a large and realistic training dataset.

To learn the descriptors, authors of [WB07] have generated their own data set. The data set consists of 3 different set of images. Each set has more than 400,000 image patches of size 64x64. The dataset is built using multiple images of a 3D scene, where the camera matrices and 3D point correspondences are accurately recovered. As a result, the dataset captures the 3D appearance variations around each key-point. Additionally, they have come up with the ground truth dataset for testing and optimizing descriptor performance. For this particular dataset, they have precise matching and non-matching information. Here, they split up the descriptor extraction process into separate stages, and further they test the descriptor extraction process by rearranging the stages in different combinations. By doing so, they are able to test many untested combinations and examine each building block in detail. Finally, they learn the most appropriate parameter values using Powell's multidimensional direction set method, to maximize the ROC area.

Unlike [WB07], which uses parametric models for descriptor learning. [BHW11] follows the same procedure as in [WB07] by using non-parametric dimensionality reduction techniques. They describe a set of building blocks for constructing descriptors which can be combined together and jointly optimized, so as to minimize the error of a nearest-neighbour classifier. They consider both linear and non-linear transforms with dimensionality reduction, and make use of discriminant learning techniques such as Linear Discriminant Analysis (LDA) and Powell minimization to solve for the parameters [BHW11]. [HBW07] tries to learn the image descriptors using linear discriminative embedding.

Ylioinas et al. [YKHP14] propose a novel framework for learning binary image descriptors extracted by considering the local pixel neighbourhood. Here, the descriptors

are constructed using binary decision trees which are learnt from a set of training image patches. The proposed framework can utilize both labelled or unlabelled training data and hence fits into both supervised and unsupervised learning scenarios. Trzcinski et al. [TCFL13] propose a framework to learn an extremely compact binary descriptor called BinBoost where the learnt descriptor used is robust to illumination and viewpoint changes and each bit of this BinBoost is computed using a boosted binary hash function. Trzcinski et al. [TL12] also propose a binary image descriptor, which depends on the discriminative projections that is trained to be computed efficiently from a small set of simple filters. They call this descriptor as Discriminative BRIEF (D-BRIEF).

1.2.3 Post-processing

Post-processing is one of the small, but important stage in the image matching pipeline. In some cases, post-processing stage decides the final dimension of the descriptors. One of the major disadvantage of the above proposed floating point descriptors is the high dimensionality. This limits the performance of feature matching techniques in terms of speed and scalability. Many methods such as Principal Component Analysis (PCA), Walsh-Hadamard transform and many more, have been proposed to reduce dimensionality. Here, we look into some of the important methods used in the computer vision literature. PCA, was first used in computer vision for face representation by Sirovich et al. [SK87]. In PCA, the basis of the subspace is obtained from the eigenvectors of the sample covariance matrix of the input (facial images). When the eigenvectors corresponding to the largest eigenvalues are used as a basis, the resulting projection simultaneously maximizes the variance of the projected data or minimizes the average projection cost [BN07].

Ke et al. proposed to use PCA for dimensionality reduction of a SIFT descriptor and named it as, PCA-SIFT [KS04]. PCA-SIFT, uses an alternative feature vector derived using PCA based on normalized gradient patches rather than using the weighted and smoothed histograms of gradient, that is used in the construction of the SIFT descriptor. Additionally, SIFT-PCA can also have a dimension as small as 20 or 36 and this small sized feature vector, results in faster matching speed. But, as per the descriptor evaluation proposed by [MS03], PCA-SIFT performs slightly worse than the standard SIFT descriptors. PCA-SIFT descriptor uses 3 main steps. In the first step, an eigenspace is constructed based on the gradients obtained from the local 41x41 image patches, resulting in a 3042 element vector. In the second step, the local image gradients are computed for patches and finally generates the reduced-size feature vector from the eigenspace, using PCA on the covariance matrix of each feature vector. Hua et al. (Hua-LDA) proposed a dimensionality reduction scheme that uses Linear Discriminant Analysis (LDA). Winder et al. [WB07] tries to reduce this dimensionality by combining PCA with that of the optimization of gradient and spatial binning parameters as part of the training step.

Bo et al. [BRF10] propose kernel descriptors for visual recognition. In their work, they highlight the kernel view of orientation histograms and propose a new method to design and learn, low-level image feature. Their framework consists of three main stages. In the initial stage, they design the match kernels using the pixel attributes. This is followed by learning compact basis vectors using Kernel Principle Component Analysis (KPCA) and finally, they construct the kernel descriptors by projecting the infinite-dimensional feature vectors on to the learned basis vectors. They apply the same frame work on color, gradient and shape attributes to generate three effective kernel descriptors. The

dimensionality reduction techniques based on PCA, requires an off-line training stage to estimate the covariance matrix that is used for PCA projection. Usually, in the off-line stage, large and diverse collection of images are trained prior to its use. Thus reducing the benefits obtained by using the dimensionality reduction [TWY08].

The authors of [TWY08] propose a new descriptor called the Compact Descriptor through Invariant Kernel Projection (CDIKP), which completely bypasses the off-line training stage. In the initial step, they use the DoG scale-space approach as described in [Low04], to generate the key-points. In the second stage, they obtain the scale, view-point and illumination normalized canonical square patch and finally, they construct the descriptor by projecting the normalized kxk patch on to a k^2 Walsh-Hadamard Kernel. Thus, the obtained descriptor is highly compact, having a dimension of 20. However, one of the disadvantage of CDIKP is that it only considers the first order derivatives of pixel intensity along the horizontal and vertical directions and as in most cases, PCA-SIFT outperforms CDIKP [ZCCY10]. Inspired by [TWY08], Zhao et al. [ZCCY10] propose a new descriptor called Kernel Projection Based SIFT (KPB-SIFT). Like SIFT, KPB-SIFT encodes the salient aspects of image information in the feature point's neighbourhood. However, instead of using SIFT's smoothed weighted histograms, it uses Walsh-Hadamard Kernel projection on orientation gradient patches, to obtain the descriptor of size 36.

Several compression schemes has been proposed to reduce the bit-rate of the floating point descriptors. Chandrashekar et al. [CTC+12] proposed a low bit rate descriptor called Compressed Histogram of Gradient (CHoG). In the initial step, they split the image patch into soft log polar spatial bins using DAISY configurations [TLF10] and latter, they capture the gradient histogram from each of the spatial bin directly into the descriptor. Finally, CHoG retains the information in each spatial bin as a distribution. In this approach, they make use of 9 to 13 spatial bins and 3 to 9 gradient bins, resulting in 27 to 117 dimensional descriptors. For compressing the descriptor, CHoG quantize the gradient histogram in each cell individually and maps it to an index. The fixed length or entropy coding is used to encode the indices and the bit-stream is concatenated together to form the final descriptor.

Yeo et al. [YAR08] propose the use of coarsely quantized random projections of descriptors to build binary hashes and use the Hamming distance between binary hashes as their matching criterion. Torralba et al. [TFW08] propose a new approach using machine learning techniques to convert the GIST descriptor [OT01] into a compact binary code, having a few hundred bits per image. In [CTC+09], the authors use transform coding to efficiently store and transmit SIFT and SURF image descriptors. By using this approach, the authors claim that the image and feature matching algorithms are robust towards significant compressed features. But, Jegou et al. [JDS11] use vector quantization technique to compress the descriptor. In this approach, they decompose the SIFT descriptor directly into smaller blocks and perform the vector quantization on each block. Many hashing schemes such as Locality Sensitive Hashing (LSH), Similarity Sensitive Coding (SSC) or Spectral Hashing(SH) has been proposed by many authors to compress the descriptors. But according to [CTC+12], these hashing schemes do not perform well at low bit-rates.

Image matching and object recognition in uncontrolled environments such as varying illumination is one of the most important bottlenecks for practical computer vision systems. Some vision algorithms try to overcome this problem by normalizing the descriptor

vector. In SIFT descriptor construction [Low04], in the final stage, the feature vector is modified to reduce the effects of illumination change. Initially the vector is normalized to unit length and the variations in the image contrast results in each pixel values being multiplied by a constant. As a result, the gradient is also multiplied by the same constant. Normalization nullifies this change in contrast. They further reduce the influence of large gradient magnitudes by thresholding the values in the unit feature vector to no larger than 0.2 and later renormalizes it to unit length. Almost all of the descriptors explained above follow this normalization procedure with or without clipping.

1.2.4 Descriptor Matching

Distance Measure

Once we have the feature descriptor/vector, the next step is to use the descriptor for image matching or for retrieving images from the database. The comparison between a feature vector v_1 obtained from a key-point i belonging to an image $I1$ and another feature vector v_2 obtained form a point j from another image $I2$ is given by a distance function/metric d. In general, distance metrics are well-known mathematical functions used for different applications in computer vision. In the case of image matching application, distance metric is a measure that classifies a good or a bad match. Depending on the computational capability and application used for a specific vision task, an appropriate distance metric is chosen. For applications related to image matching, Euclidean distance and Hamming distance are better suited, whereas, for applications related to image retrieval other distance metrics are preferred. Many distance metric methods has been proposed in the computer vision literature for applications related to image matching and retrieval. Some of them are tabulated in Table1.3. In this section, we discuss briefly about some of the well known distance measures.

Euclidean distance is one of the most widely used distance metric in applications related to image matching. Euclidean distance, shows a good trade-off between computational complexity and matching performance. The authors of SIFT [Low04], use Euclidean distance for key-point matching where as, Mikolajczyk et al [MS03] propose a frame work for evaluating the image descriptors. Their evaluation protocol use euclidean distance as the performance metric. Most of the descriptors explained above use this performance evaluation protocol. Earth Mover's Distance, is another distance metric used for descriptor matching [LSP05b] and fast contour matching [GD04]. Correlation, is widely used as an effective similarity measure for image matching applications. Palomares et al. [PMD12, WL08] uses correlation distance for image matching.

The Hausdorff distance, measures the extent to which each point on a model set lies close to some points of an image set and vice versa. This can be used for object detection and matching. Dubuisson et al. [DJ94], on the other hand, has used a modified version of Hausdorff distance for object matching and Rotter et al. [AK09] has used Hausdorff distance for word image matching. Jaccard distance (also known as Jaccard coefficient (JC) or Tanimoto coefficient) is another similarity measure used for image matching where a higher JC indicates a better correspondence between the images. A value of 1 indicates complete correspondence and a value of 0 means that there is no correspondence at all. Zitnick [Zit10b] uses Jaccard similarity for fast image patch matching and retrieval. Jain et al. [JZ97] has used Jaccard distance for recognition of handwritten digits. Venkatrayappa

et al. [VSMM14] use Mahalanobis Distance as a metric for tracking with particle filters. Bo et al. [BZlCl15] proposes an algorithm for Image Matching Based on Mahalanobis Distance and Weighted KNN Graph. Most of the binary descriptors such as BRIEF [CLSF10], BRISK [LCS11], FREAK [AOV12], ORB [RRKB11] uses hamming distance for fast image matching. Some of the other distance metrics used in computer vision are Chebyshev distance, Hellinger distance, Manhattan distance, L1 Norm, L2 Norm, Canberra distance, Bray Curtis distance and Kullback Leibler distance.

Name	Distance between $\vec{v}_{1,i}$ and $\vec{v}_{2,j}$
Manhattan Distance :	$\sum_{k=1}^{n} \lvert v_{1,i}^k - v_{2,j}^k \rvert$
Euclidienne Distance :	$\sqrt{\sum_{k=1}^{n} \left(v_{1,i}^k - v_{2,j}^k\right)^2}$
Chebychev Distance:	$\max_{k=1,.,n} \lvert v_{1,i}^k - v_{2,j}^k \rvert$
Kullback-Leibler Divergence:	$\sum_{k=1}^{n} v_{1,i}^k \log \frac{v_{1,i}^k}{v_{2,j}^k}$
Jeffrey Divergence:	$\sum_{k=1}^{n} v_{1,i}^k \log \frac{v_{1,i}^k}{\frac{v_{1,i}^k + v_{2,j}^k}{2}} + v_{2,j}^k \log \frac{v_{2,j}^k}{\frac{v_{1,i}^k + v_{2,j}^k}{2}}$
Quadratic Distance (A is a similarity matrix):	$\sqrt{t(v_{1,i}^k - v_{2,j}^k).A.(v_{1,i}^k - v_{2,j}^k)}$
Mahalanobis Distance(C is a covariance matrix):	$\sqrt{t(v_{1,i}^k - v_{2,j}^k).C^{-1}.(v_{1,i}^k - v_{2,j}^k)}$

Table 1.3: Example of distances calculation between feature vectors $v1$ and $v2$ of two key-points i and j each taken from two different images.

Matching process

Descriptor Matching process involves computing the distance between all possible pairs of detected features and selecting a matching pair for those features whose nearest-neighbour is closer than some threshold. The number of matches found depends on the matching strategy. For example, SIFT [Low04] algorithm achieves matching under realistic conditions with the help of special data structures or approximate nearest-neighbour algorithms. Some of the well-known matching strategies used in the literature are Distance Threshold matching (DT), Nearest-Neighbour with Ratio Test matching (NNRT) and Nearest-Neighbour with Distance Threshold matching (NNDT).

In the distance threshold matching strategy (DT), two image features are said to be a match if the distance between their descriptors lies below a fixed distance threshold Td. This is one of the simplest matching strategies. In this strategy, each query descriptor can match several descriptors in the database. In NNRT, a match is found if it satisfies the nearest-neighbour ratio-test. Let A and B be two images, A_1 be a descriptor in image A, and B_1 and B_2 be first nearest and second nearest descriptors in image B. Let $D_1 =\parallel A_1 - B_1 \parallel$ and $D_2 =\parallel A_1 - B_2 \parallel$ be the distance to the first and the second nearest neighbours, respectively. Let D_r be a predefined threshold. If the ratio $\frac{D_1}{D_2} < D_r$, then B_1 is said to be a match of A_1. NNDT can be considered as a fusion of DT and NNRT methods. For A_1 to be a match with B_1, it must be its nearest neighbour and also it must satisfy the distance threshold criterion, i.e, distance to nearest neighbour $D_1 =\parallel A_1 - B_1 \parallel< Td$.

The present day image matching and retrieval systems should be able to handle a large database. When there are millions of such descriptors, the above mentioned methods becomes very expensive even after dimensionality reduction. To overcome this limitation to a certain extent, algorithms based on approximate nearest neighbour strategies can be used. Arya et al. [AMN+98] propose an optimal algorithm for approximate nearest neighbour searching in Fixed Dimensions. Beis et al. [BL97] propose a modified k-d tree based approximate nearest neighbour algorithm for shape indexing. Anan et al. [SH08] propose approximate nearest neighbour algorithm based on multiple randomized kd-trees for indexing a large number of SIFT and other types of image descriptors. Similarly, hierarchical k-means trees [FN75], spill trees [LMGY04], vantage-point trees [Yia93] and others have been used to accelerate the approximate nearest neighbour strategy for image matching.

Some researchers have chosen to speed up the nearest-neighbour search by, binarizing the real-valued descriptors using techniques such as Boosting, hashing [AI08, KD09], Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) based methods [RL09,SBBF12] and quantization [GL11]. Binarization of real-valued descriptors leads to binary vectors which can be speedily evaluated using Hamming distance. But, there has been very little progress made on improving the performance of ANN algorithms on binary descriptors. Some of the algorithms which involve PCA decomposition such as spectral Hashing are not adaptable for binary descriptors. Other methods treat the binary vector as vectors of zeros and the ones encoded as floating-point numbers. But, this approach results in weak performance and the encoding negates the advantages of binary descriptors over floating point descriptors. There are few algorithms such as vantage-point trees and HKM that can be modified to mainly deal with binary vectors and to use the Hamming distance as a similarity measure.

1.3 Summary

This chapter mainly deals with the state of the art in image matching, wherein we give an overview of the image matching pipeline. Since the literature on image matching/image descriptors is abundant and research continues to be very active, we restrict ourselves to methods that are most used or most promising.

In the initial stage of the image matching pipeline, we spoke about interest point/region

detectors. Among these interest point detectors, Harris corner detector and its extensions such as Harris-laplace and Harris-Affine are popular. Some of the other popular detectors are SIFT detector, SURF detector and MSER. Each of the interest point detector has its own advantage and Table 1.1 gives qualitative information about these detectors. In the second stage of the pipeline, we saw different categories of image descriptor based on gradient information, filter response, intensity patterns etc. Finally, we explained about the different post-processing steps, distance measures and matching protocols used for image matching. In the next chapter we concentrate on the filters and in particular about the Anisotropic half rotating filters, which forms the basis of our work.

Chapter 2

Half Filters

2.1 Introduction

In the chapter.1 we spoke about the global and local image features. This was followed by the image matching pipeline where we explained in detail about different feature detectors, feature descriptors, post processing steps and different matching methodologies involved. Since our work mainly concentrates on the description of an image or image descriptor using the filter responses, in the initial part of this chapter, we concentrate on the isotropic and anisotropic filters in general. The later part is devoted to the anisotropic half filters that is used in our work.

2.2 Gaussian isotropic filter

(a) Original Image (b) $\sigma = 1$ (c) $\sigma = 2$

Figure 2.1: Figure illustrating the effect of σ.

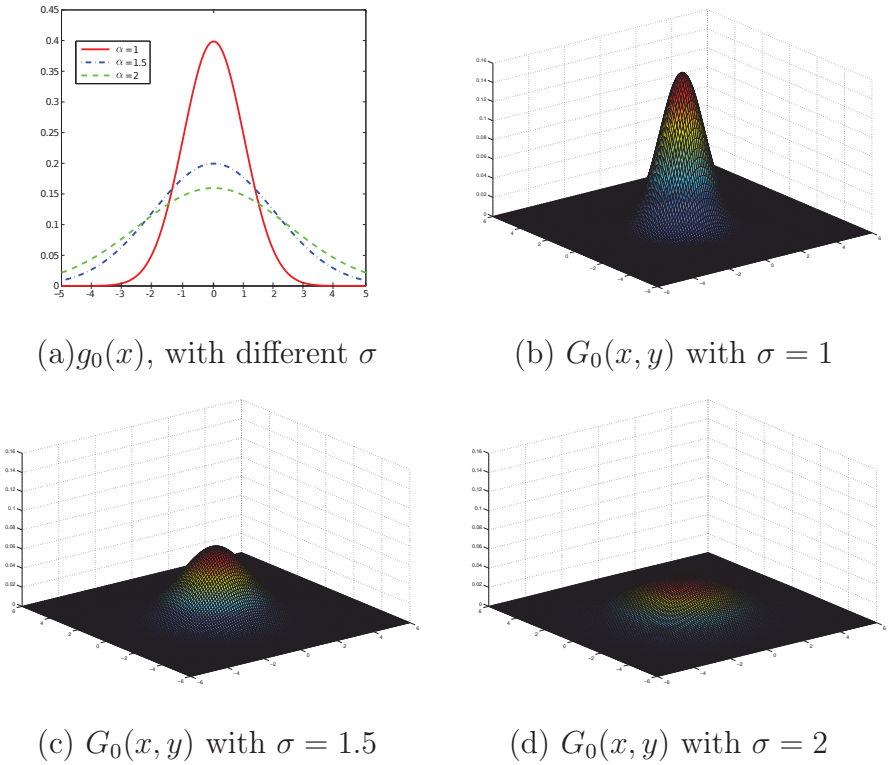
Gaussian isotropic filter also known as the Gaussian smoothing operator is a 2-D convolution operator. The primary application of Gaussian filter is to blur the images and to remove details and noise as illustrated in Fig.2.1. The Gaussian filter uses a kernel that represents the shape of a bell. The Gaussian distribution in 1-D takes form as shown

in Eq.2.1. In 2-D, an isotropic Gaussian is represented as in the Eq.2.2.

$$g_0(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x^2)}{2\sigma^2}} \tag{2.1}$$

$$G_0(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{2.2}$$

where x and y represents the pixel coordinates. σ represents the standard deviation or smoothing factor. An ideal Gaussian distribution is non zero everywhere, but in practice it is effectively zero for more than three standard deviations from the mean and hence we can truncate the kernel at this point. Fig.2.2 illustrates the Gaussian distributions with varying values of variance σ.



(a)$g_0(x)$, with different σ (b) $G_0(x, y)$ with $\sigma = 1$

(c) $G_0(x, y)$ with $\sigma = 1.5$ (d) $G_0(x, y)$ with $\sigma = 2$

Figure 2.2: Gaussian distribution with different values of variance σ.

Some of the important properties of Gaussian filters are:

1. Gaussian filtering an image, removes the noise from the image and blurs the edges.
2. Larger the σ is, the more details are removed.
3. Another advantage of Gaussian is that it is separable i,e. the Gaussian as shown above can be separated into x and y components. Because of this separability property, the 2-D convolution can be performed by first convolving with a 1-D Gaussian in the x direction and then convolving with another 1-D Gaussian in the y direction. Thus, increasing the speed at which the image is convolved with the Gaussian.

One of the popular application of Gaussian isotropic filters is for noise removal by image smoothing. It is also used in the preprocessing step during the descriptor construction as in [Low04, FWH12, KS04] as well as in scale space analysis [Lin98, Low04].

2.2.1 Gaussian Derivative Filters

When we take the spatial derivative of a Gaussian function repeatedly, we obtain a pattern of a polynomial in increasing order multiplied with the original (normalized) Gaussian function again. Some of the graphs of the Gaussian derivative functions are shown in the Fig.2.3.

The first order derivative g_1 of a 1D Gaussian filter g_0 is given by the Eq.2.3. A first order 2D Gaussian filter is obtained by the product of $g_1(x)$ and $g_0(y)$ as in Eq.2.4, where C_1 is the normalization constant.

$$g_1(x) = \frac{-x}{2\pi \cdot \sigma^3} \cdot x \cdot e^{\frac{-x^2}{2\sigma^2}}. \tag{2.3}$$

$$G_1(x,y) = g_1(x) \cdot g_0(y) = C_1 \cdot x \cdot e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2.4}$$

For an image I, the gradient $||\nabla I||$ is approximated by first calculating the image derivatives G_x and G_y. G_x and G_y are obtained by convolving the image with the first order gaussian $G_x(x,y) = g_0(x).g_1(y) * I(x,y)$ and $G_y(x,y) = g_0(y).g_1(x) * I(x,y)$. Finally, the gradient $||\nabla I||$ and the associated direction η are calculated as in equation 2.5 and 2.6 respectively. Some of the examples of gradient and orientations at different values of σ are illustrated in the fig.2.5.

$$||\nabla I(x,y)|| = \sqrt{G_x^2(x,y) * I(x,y) + G_y^2(x,y) * I(x,y)} \tag{2.5}$$

$$\eta(x,y) = \arctan\left(\frac{G_y(x,y) * I(x,y)}{G_x(x,y) * I(x,y)}\right) \tag{2.6}$$

Gaussian derivative filters find application in edge detection, image restoration and many other applications.

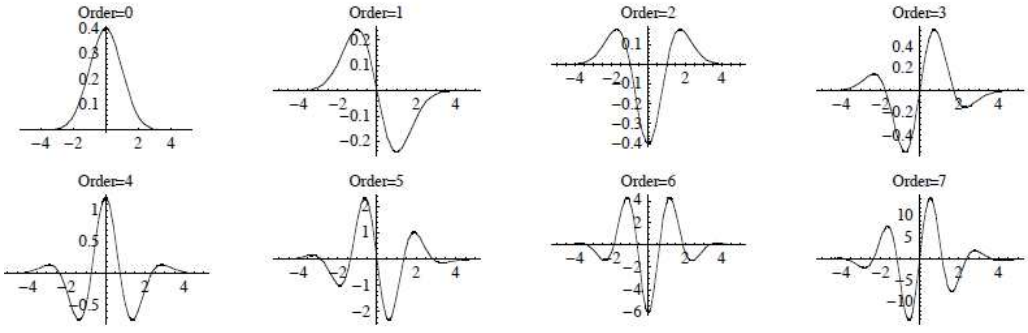
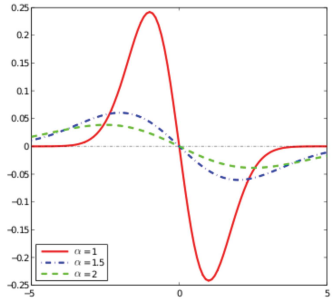
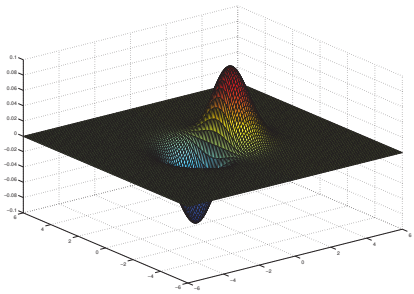


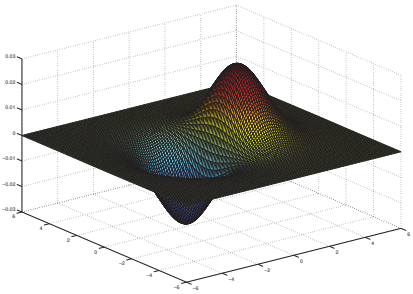
Figure 2.3: Plots of the 1D Gaussian derivative function up to order 7.



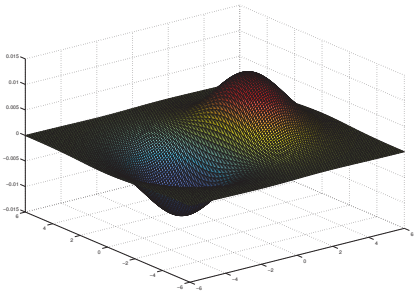
(a) $g_1(x)$, with different σ



(b) $G_1(x, y)$ with $\sigma = 1$



(c) $G_1(x, y)$ with $\sigma = 1.5$



(d) $G_1(x, y)$ with $\sigma = 2$

Figure 2.4: Gaussian derivative filters with different values of σ.

2.2.2 Laplacian of Gaussian (LoG)

Maar et al. [MH80] proposed an operator that can be tuned to detect edges at a particular scale. For this purpose, they propose to initially smooth the image with a Gaussian G, followed by the laplacian operation. These two steps form the Laplacian of Gaussian (LoG) operator. Laplacian filters are basically second order derivative filters used to find areas of an edge in the images. Since derivative filters are sensitive to noise, the common procedure used is to smooth the image before applying the Laplacian filters. The procedure to localize the edges is to detect the zero crossing of the lapalcian of an image pre-filtered by a Gaussian. However, a larger value of σ degrades the image and fails to detect the fine structure. Thus, creating false negatives. Conversely, if σ is very low, the noise is insufficiently filtered and will be localized as contours, thus creating false positive.

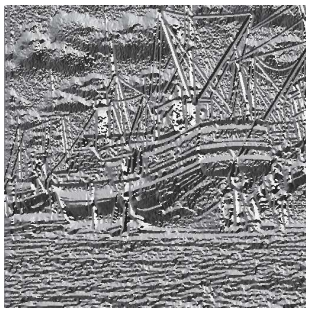
In 1D, $G_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$, for a data signal s, The LoG operation is :

$$\Delta(s * G_\sigma)(x) = (\Delta s) * G_\sigma(x) = s * (\Delta G_\sigma(x)), \tag{2.7}$$

OR

$$\Delta G_\sigma(x) = \frac{\partial^2}{\partial^2 x} G_\sigma(x) = \frac{x^2 - \sigma^2}{\sqrt{2\pi} \cdot \sigma^5} e^{\frac{-x^2}{2\sigma^2}}.$$

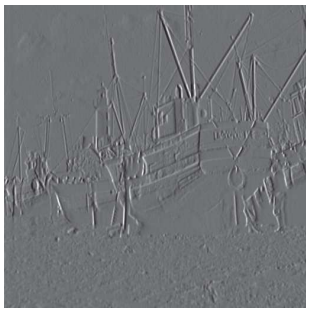
(a) Original Image



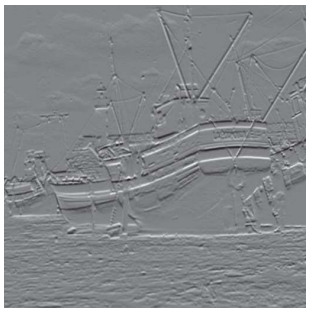
(b) η with $\sigma = 1$

(c) η with $\sigma = 4$

(d) $||\nabla I||$ with $\sigma = 1$,

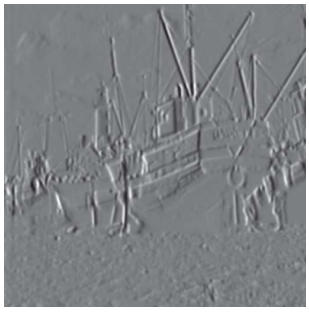


(e) Image G_x with $\sigma = 1$,

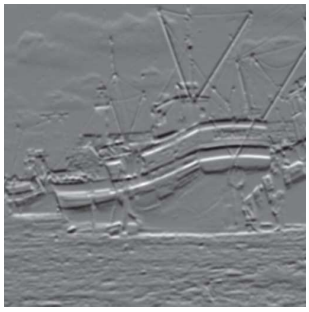


(f) Image G_y with $\sigma = 1$,

(g) $||\nabla I||$ with $\sigma = 2$,

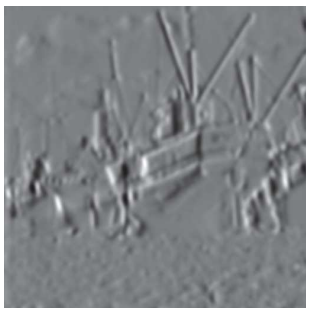


(h) Image G_x with $\sigma = 2$,

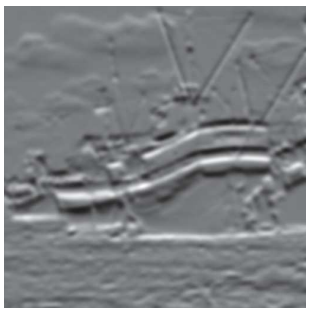


(i) Image G_y with $\sigma = 2$,

(j) $||\nabla I||$ with $\sigma = 4$,

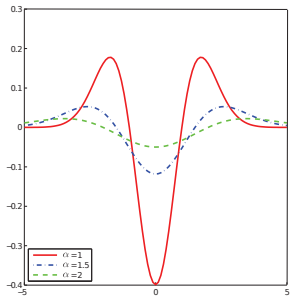


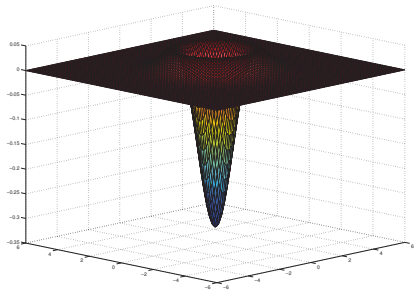
(k) Image G_x with $\sigma = 4$,



(l) Image G_y with $\sigma = 4$.

Figure 2.5: Approximation of gradient $||\nabla I||$ with different σ.





(a) 1D Laplacian distribution with different σ (b) 2D Laplacian distribution with $\sigma = 1$

Figure 2.6: Laplacian of gaussian.

(a) Input Image (b) $\sigma = 1.6$ (c) $\sigma = 2.5$

Figure 2.7: Edge detection using LoG.

In case of 2D, Gaussian $G_\sigma(x, y)$ is represented as :

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

and the LoG operation is represented as :

$$\Delta G_\sigma(x, y) = \frac{\partial^2}{\partial^2 x} G_\sigma(x, y) + \frac{\partial^2}{\partial^2 y} G_\sigma(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sqrt{2\pi} \cdot \sigma^5} e^{-\frac{x^2+y^2}{2\cdot\sigma^2}} \tag{2.8}$$

LoG can be used for scale selection in corner/region detection, blob detector, scale invariant feature transform and in other image descriptors for image matching and object recognition. The zero crossing of the LoG can be used to detect edge points.

2.2.3 Difference of Gaussian (DoG)

Though LoG is accurate, it is computationally very expensive. An approximation of the LoG called Difference of Gaussian (DoG)is proposed in the literature. Similar to LoG, the image is smoothed by convolving the image with Gaussian kernel $G_{\sigma_1}(x, y)$ of width σ_1 :

$$G_{\sigma_1}(x, y) = \frac{1}{\sqrt{2\pi{\sigma_1}^2}} e^{-\frac{x^2+y^2}{2{\sigma_1}^2}} \tag{2.9}$$

to get:

$$g_1(x, y) = G_{\sigma_1}(x, y) * f(x, y) \tag{2.10}$$

A second image is smoothed with Gaussian $G_{\sigma_2}(x, y)$ of width σ_2 to obtain a second smoothed image :

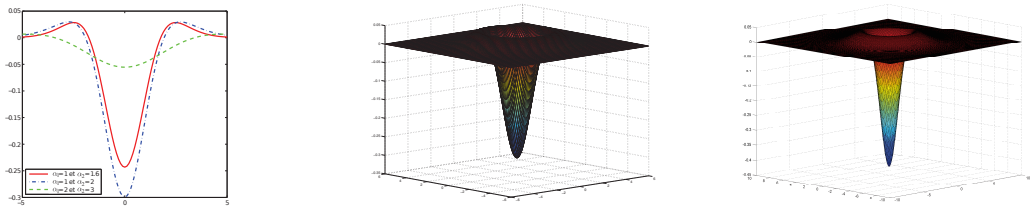
$$g_2(x, y) = G_{\sigma_2}(x, y) * f(x, y) \tag{2.11}$$

Now, DoG can be defined as the difference of these two Gaussian smoothed images :

$$g_1(x, y) - g_2(x, y) = G_{\sigma_1}(x, y) * f(x, y) - G_{\sigma_2}(x, y) * f(x, y) = (G_{\sigma_1} - G_{\sigma_2}) * f(x, y) = DoG * f(x, y) \tag{2.12}$$

Finally, DoG as an operator or convolution kernel is defined as:

$$DoG \triangleq G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}} \left(\frac{1}{\sigma_1} e^{-\frac{x^2+y^2}{2{\sigma_1}^2}} - \frac{1}{\sigma_2} e^{-\frac{x^2+y^2}{2{\sigma_2}^2}} \right) \tag{2.13}$$



(a) 1D DoG with σ_1 and σ_2 (b) $\sigma_1 = 1$ and $\sigma_2 = 1.6$ (c) $\sigma_1 = 1$ and $\sigma_2 = 2.5$

Figure 2.8: Distribution of DoG in 1D and 2D.

(a) Input Image (b) $\sigma_1 = 1$ and $\sigma_2 = 1.6$ (c) $\sigma_1 = 1$ and $\sigma_2 = 2.5$

Figure 2.9: Edge detection using DoG.

Difference of Gaussian (DoG) is used for blob detection in scale space [Low04]. It is also used as a feature enhancement and image enhancement algorithm.

2.2.4 Shen Castan Filter

Shen and Castan proposed an operator based on the criteria similar to the one proposed by Canny including detection and localization. In practice, both the filters are based on

exponential filters and hence, the behaviour is similar to each other. The smoothing Shen filter is given by :

$$F(x) = C_0.e^{-\alpha|x|} \tag{2.14}$$

Where C_0 is the normalization factor:

$$C_0 = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}} \tag{2.15}$$

The associated derivative filter is given by:

$$\mathcal{F}'(x) = \begin{cases} C_1\, e^{-\alpha x} & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -C_1\, e^{\alpha x} & \text{if } x < 0 \end{cases} \tag{2.16}$$

C_1 is chosen in a manner so as to obtain a normalized derivative filter F' :

$$\sum_{n=-\infty}^{+\infty} n \cdot \mathcal{F}'(n) = 1 \tag{2.17}$$

Where,

$$C_1 = \frac{(1 - e^{-\alpha})^2}{2 \cdot e^{-\alpha}}.$$

The parameter α determines the filter width. As α approaches 0, more smoothing is performed by the filter. This can be seen in the Fig.2.10. The Fig.2.11, illustrates the gradient estimation of an image using different α parameters.

Shen Castan Filters are mainly used for edge detection.

2.3 Isotropic Orientation Filters

One of the disadvantage of isotropic filters is that they are less accurate in feature description, image filtering, detecting edges, contours and other geometrical structures in the image. Whereas, edges or contour detection methods using orientation filter bank, estimates the edges and contours accurately. Due to multiple orientations, these filters are able to detect several image features such as edges, contours etc. One of the most popular filter bank is, the Gabor filter bank [MM96], which is made of a set of Gabor filters at different scale and orientations. Another example of orientation filter bank is the steerable filters. As a solution to the above stated problem, Freeman et al. [FA91, JU04] introduced steerable filters that can be directed at specific orientations using a linear combination of isotropic filters.

Steerable filters are constructed by synthesizing steered or oriented linear combinations of chosen basis functions such as quadrature pairs of Gaussian filters and oriented versions of each function in a simple transform. According to [JU04, LM01a], many different types of filter can be used as the basis for steerable filters. The construction of steerable filters is shown in Fig.2.12. Initially, the filter transform is generated by combining the basis

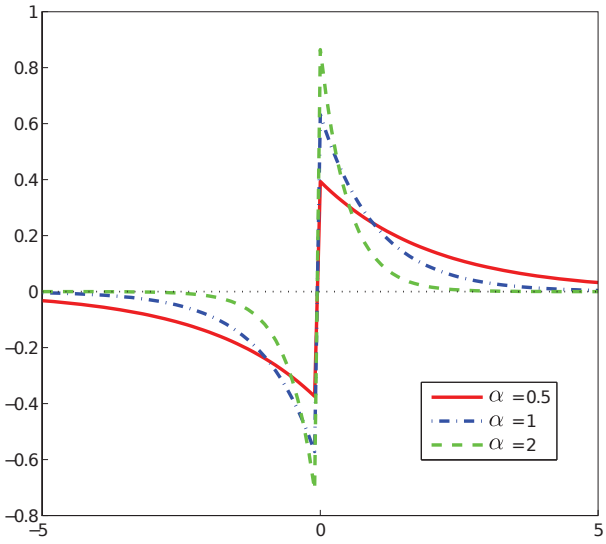


Figure 2.10: Shen casten derivative filter with different α values.

functions in a filter bank and is followed by gain selection for each function. Finally, all filters in the bank are summed and adaptively applied to the image.

Similar techniques using Gaussian anisotropic filtering were introduced by Perona [Per92] and implemented recursively by Geusebroek et al. [GSvdW02] and extended to color Images by [KvdWHR06]. These methods are able to detect the linear structures correctly.

According to Freeman et al [FA91], the general definition of a orientation filter or steerable filter is :

$$f(x,y) * h_\theta(x,y) = \sum_{k=1}^{N} \sum_{i=0}^{k} b_{k,i}(\theta) f(x,y) * \left(\frac{\partial^{k-i}}{\partial x^{k-i}} \frac{\partial^i}{\partial y^i} g(x,y) \right), \tag{2.18}$$

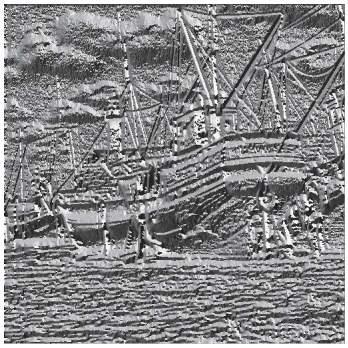
Where, h is a steerable filter with N number of angles, $f(x,y)$ is an image or a 2D function, $b_{k,i}(\theta)$ is an interpolation function and $g(x,y)$ is an isotropic window function which can be a Gaussian.

In their work, Freeman et al [FA91] used 2 partial derivatives of 2D Gaussian G_σ with variance σ in any direction θ. Let $G_1^{0^\circ}(x,y)$ be the x derivative of the function G_σ and $G_1^{90^\circ}(x,y)$ be the derivative in the y direction :

$$\begin{cases} G_1^{0^\circ}(x,y) & = \dfrac{\partial G_\sigma(x,y)}{\partial x} \\ G_1^{90^\circ}(x,y) & = \dfrac{\partial G_\sigma(x,y)}{\partial y}. \end{cases} \tag{2.19}$$

So, G_1^θ is the derivative of the function G_σ in the direction θ :

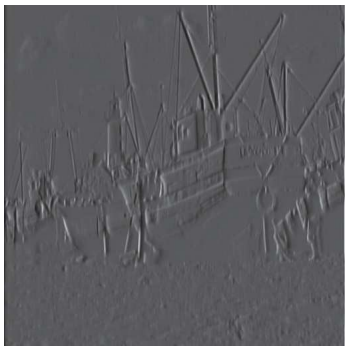
(a) Original Image



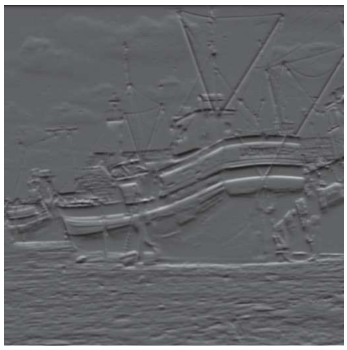
(b) η with $\alpha = 1$

(c) η with $\alpha = 2$

(d) $||\nabla I||$ with $\alpha = 0.5$,



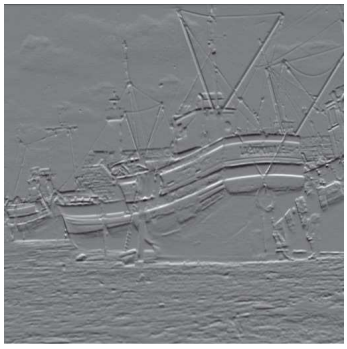
(e) Image G_x with $\alpha = 0.5$,



(f) Image G_y with $\alpha = 0.5$,

(g) $||\nabla I||$ with $\alpha = 1$,

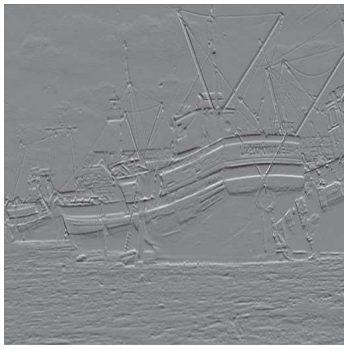
(h) Image G_x with $\alpha = 1$,



(i) Image G_y with $\alpha = 1$,

(j) $||\nabla I||$ with $\alpha = 2$,

(k) Image G_x with $\alpha = 2$,



(l) Image G_y with $\alpha = 2$.

Figure 2.11: Shen gradient approximation with different values of α.

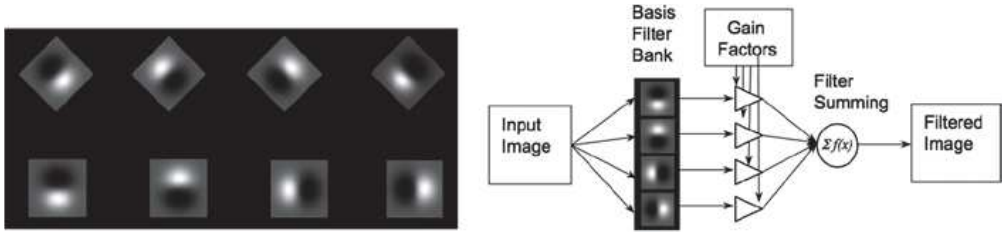
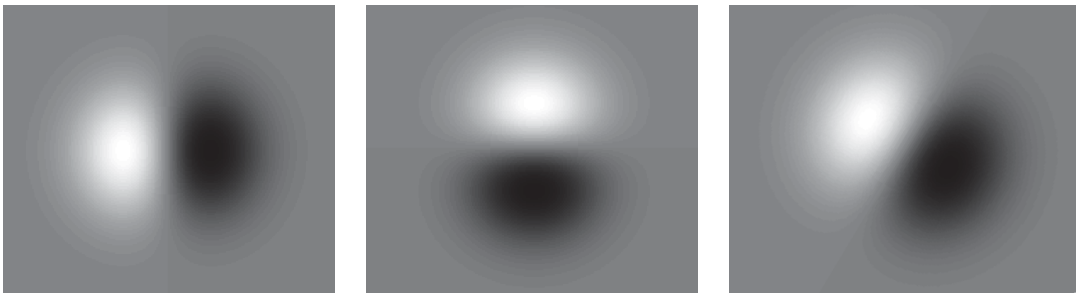


Figure 2.12: (Left) Set of steerable filters in 8 different directions. (Right) Methadology involved in image filtering using steerable filters. Figure obtained from [sj14]



(a) G_1^0 , (b) G_1^{90} , (c) G_1^{30}.

Figure 2.13: Example of Gaussian oriented filters G_1^θ.

$$G_1^\theta(x,y) = \frac{\partial G_\sigma R_\theta}{\partial x},$$

Where, R_θ is the rotation matrix.

In [FA91], Freeman and Adelson have shown that the directional derivative G_1^θ at orientation θ can be generated by a linear combination of a rotation of the basic filters $G_1^{0^\circ}$ and $G_1^{90^\circ}$:

$$G_1^\theta(x,y) = \cos(\theta)\, G_1^{0^\circ}(x,y) + \sin(\theta)\, G_1^{90^\circ}(x,y). \tag{2.20}$$

Some of the isotropic oriented Gaussian kernels are shown in the Fig.2.13. The convolution of an image I by the isotropic oriented gaussian kernels is given by:

$$\left(I * G_1^\theta\right)(x,y) = \cos(\theta)\, \left(I * G_1^{0^\circ}\right)(x,y) + \sin(\theta)\, \left(I * G_1^{90^\circ}\right)(x,y). \tag{2.21}$$

The magnitude of the gradient $\|\nabla I(x,y)\|$ and the orientation G_1^θ at each pixel in the image is obtained by convolving the image I with the filters G_1^θ and the one with the maximum absolute value is selected as the gradient and angle as shown below:

$$\|\nabla I(x,y)\| = \max_{\theta\in[0,360[} \left|\left(I * G_1^\theta\right)(x,y)\right|, \tag{2.22}$$

$$\theta_m = \arg\max_{\theta\in[0,360[} \left(\left|\left(I * G_1^\theta\right)(x,y)\right|\right) . \tag{2.23}$$

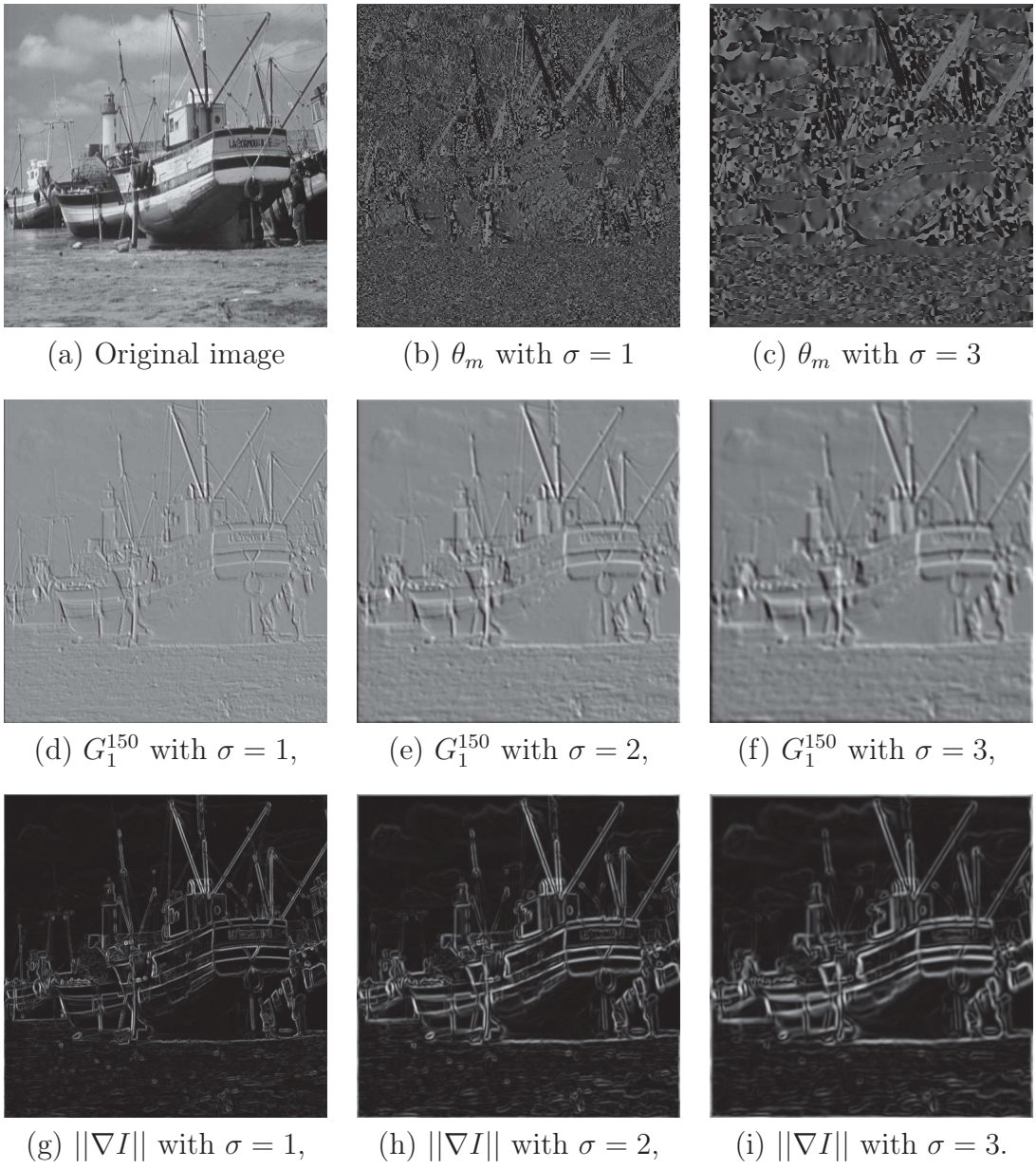
Isotropic filter banks have been used for content based image retrieval [GSvdW02, Sch01, LM01b]. Steerable filters have found application in the field of medical imaging and biometrics.

2.4 Anisotropic Orientation Filters

Anisotropic convolution filters have found applications in adaptive image smoothing by aligning them to local image structures. They can also be used to detect image structures where different types of filters can be used for the same. Here, we mainly concentrate on Gaussian anisotropic orientation filters. The anisotropic Gaussian kernel $G_{\sigma_x\sigma_y}(x, y)$ is given by :

$$G_{\sigma_x\sigma_y}(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\frac{-x^2}{2\sigma_x^2}} e^{\frac{-y^2}{2\sigma_y^2}}. \tag{2.24}$$

To achieve more smoothing in the direction of the filter, Perona et al [Per95], in their



(a) Original image (b) θ_m with $\sigma = 1$ (c) θ_m with $\sigma = 3$

(d) G_1^{150} with $\sigma = 1$, (e) G_1^{150} with $\sigma = 2$, (f) G_1^{150} with $\sigma = 3$,

(g) $||\nabla I||$ with $\sigma = 1$, (h) $||\nabla I||$ with $\sigma = 2$, (i) $||\nabla I||$ with $\sigma = 3$.

Figure 2.14: Approximation of the gradient using isotropic oriented filters.

work use the function $G_{\sigma_x\sigma_y}(x,y)$ with the ratio $\frac{\sigma_x}{\sigma_y} = 3$. Fig.2.15 illustrates the oriented anisotropic Gaussian filter with $\frac{\sigma_x}{\sigma_y} = 3$. Further, the edges and contours are detected by calculating the first derivative of the filter as in Eq. 2.25 and by rotating the filter in different orientations and retaining the orientation which produces the maximum energy.

$$G'_{\sigma_x\sigma_y}(x,y) = \frac{\partial G_{\sigma_x\sigma_y}}{\partial x}(x,y) = \frac{-x}{\pi\sigma_x\sigma_y} e^{\frac{-x^2}{2\sigma_x^2}} e^{\frac{-y^2}{2\sigma_y^2}}, \tag{2.25}$$

(a) (b) (c)

Figure 2.15: $G_{\sigma_x\sigma_y}(x,y)$ and $G'_{\sigma_x\sigma_y}(x,y)$ kernels with $\sigma_x/\sigma_y = 3$. (a) Deformable kernel. (b) original $G'_{\sigma_x\sigma_y}(x,y)$ (c) $G'_{\sigma_x\sigma_y}(x,y)$ oriented by 30^o.

The anisotropic filters cannot be decomposed as a finite sum basic filters as demonstrated by Freeman et al in the case of isotropic filters. Perona [Per92] followed the technique similar to the one proposed by Freeman et al. [FA91] and replaced the Gaussian isotropic filter with that of Gaussian anisotropic filter. In addition, the anisotropic filters are computationally expensive. Guesebroek et al. [GSvdW02] proposed the recursive implementation of Gaussian anisotropic filter which was extended to detect edges in color Images by [KvdWHR06]. These filters are able to detect the linear structures correctly. The Fig.2.16 illustrates the estimation of gradient with anisotropic Gaussian filters.

It finds application in detecting long, discontinuous and blurred edges. As in the isotropic case, anisotropic orientation filter can be used in content based image retrieval [GSvdW02, Sch01, LM01b]. They are abundantly used in the field of medical imaging and biometrics.

2.5 Anisotropic Half Filters

In most of the computer vision applications, edge detection forms the basis for geometric feature extraction. Many of the successful edge detection methods depends on the amount of smoothing applied. They also depend on the type of filter used for approximating the gradient. The detected edges should be informative and should also have the least false positive detection rate. Conventional edge detectors [Can86, SC92, Der87, Der93, SB97] and the detectors based on the above explained filters doesn't guarantee proper localization of edges. Moreover, junctions and corners are not well identified and is poorly localized by these methods.

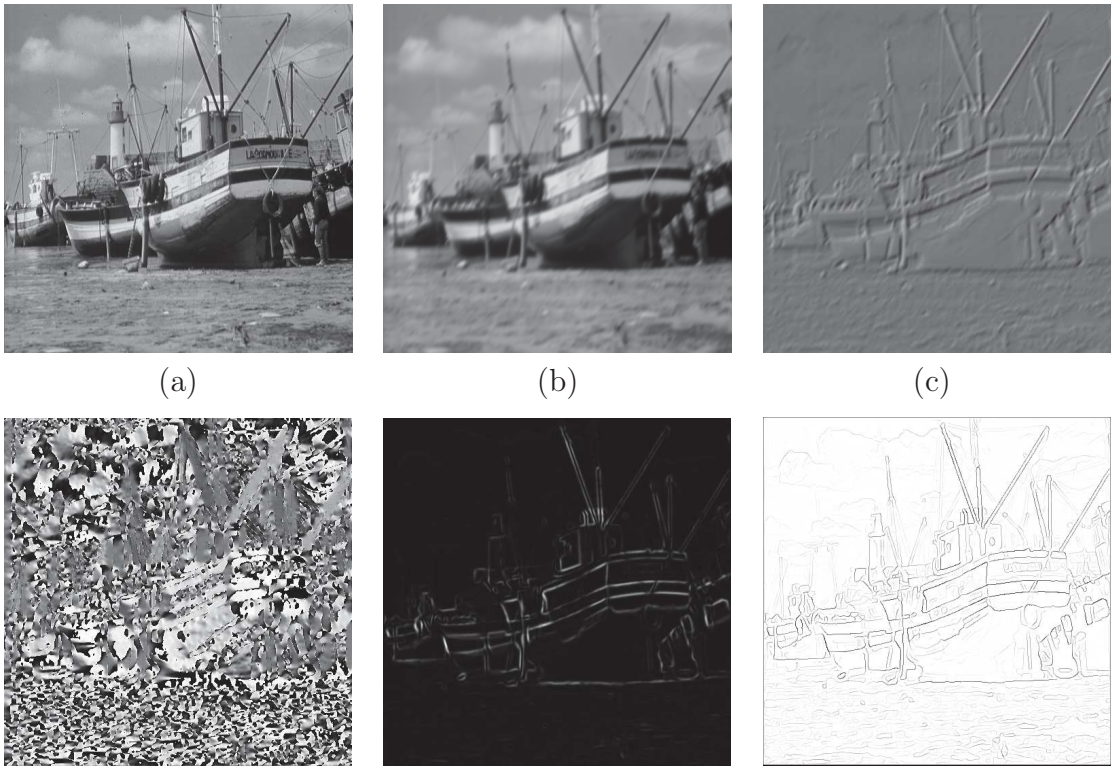
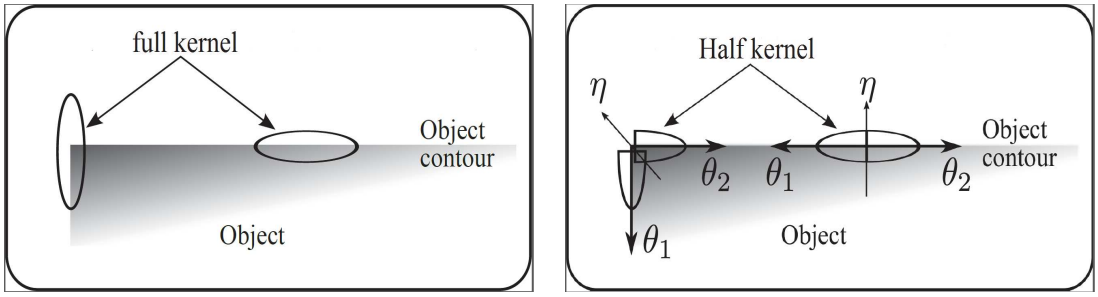


Figure 2.16: Smoothed and derivative image with orientation $\theta = 35°$ and gradient estimation using the oriented Gaussian anisotropic parameters $\sigma_x = 3$, $\sigma_y = 1$ and $\Delta\theta = 5°$. (a) original Image. (b) Smoothed image at 35°. (c) Gradient approximation at 35° with the first order derivative anisotropic filter (Eq.2.25). (d) η image. (e) Estimated gradient image. (f) Maxima in the direction of the gradient η. The images (c), (d), (e) and (f) are normalized.

In methods that employ anisotropic filtering, robustness to noise strongly depends on two filter parameters. These parameters correspond to the two standard deviations of the Gaussian function in two dimensions. The increase in the value of these parameters makes the detection less sensitive to noise, as small structures will be regarded as noise and thus, are ignored. Therefore, the accuracy of the points detected decreases sharply at the pixel corner and for objects having non-linear edges. Fig.2.17(a) presents the application of anisotropic Gaussian filters on the outline of an object. The filter is applied at two locations, one at the linear portion of the edge where the response of the anisotropic filter is high and secondly, it is applied at the corner. Here, only a part of the filter takes into account the information. Therefore, the filter response is greatly reduced where the results are particularly affected by noise.

Oriented corners filters provide an improved accuracy at corners and junctions [SF96, YDS01]. However, these filters are too flared or the central pixels have no effect. Also, as the implementation of this filter is slow, they cannot be implemented recursively.

The filter discussed and presented in the following section is motivated by the need to overcome the drawbacks of the above mentioned filters and also, the need for a good filter for detecting the edges in general and minute edges in particular. In this section, we present a new family of anisotropic filters capable of detecting minute edges and other



(a) (b)

Figure 2.17: (a)Application of Anisotropic full Gaussian kernel at the contour of an object. (b)Application of Anisotropic half Gaussian kernel at the contour of an object.

geometrical image properties such as junctions, corners etc. These filters were introduced by [MMD11a, MMD11b, MM10, MMP10, MMD11c]. The following filters are basically an elongated Anisotropic half Gaussian and exponential filters, cut in the middle to form a half filter. The image features are detected by rotating this filter at a pixel or a key-point in an image.

Anisotropic half filters have been used in the construction of local image descriptors [VMD15, VMDM15c, VMDM15a]. It has also found applications in edge detection and preservation [MM10], texture removal [MMD11a, MMD11b], anisotropic diffusion [MM14], image de-blurring and Regularization [MXM13] and in medical imaging applications [MMD13].

2.5.1 Anisotropic Half Gaussian Smoothing Filter /Kernel (AHGSK)

To address the problems with the anisotropic Gaussian filter as described above, the proposed solution was to cut a directional Gaussian in two parts as presented in Fig.2.17(b) and then to apply the filters at several orientations on the image. By construction, the half Gaussian smoothing filter is not symmetrical in the direction of its maximum elongation (towards the largest standard deviation ξ). The smoothing filter is given by the Eq.2.26:

$$g_{(\sigma_\xi,\sigma_\eta)}(x,y,\theta) \;=\; C \cdot S_y\left(R_\theta \begin{pmatrix} x \\ y \end{pmatrix}\right) \cdot e^{-\begin{pmatrix} x & y \end{pmatrix}.Z} \tag{2.26}$$

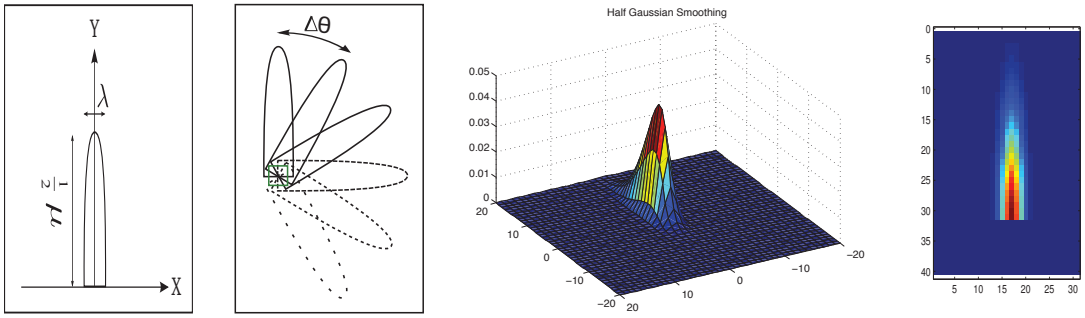
on a considered pixel point at (x, y) with:

$$Z \;=\; R_\theta^{-1} \begin{pmatrix} 1/(2\sigma_\xi^2) & 0 \\ 0 & 1/(2\sigma_\eta^2) \end{pmatrix} .R_\theta \begin{pmatrix} x \\ y \end{pmatrix},$$

where:

σ_ξ and σ_η controls the size of the Gaussian along the two orthogonal directions, radial and axial. Since, we are only interested in the causal part of the filter, we cut the filter in the middle along Y axis using a sigmoid function S_y. Sigmoid function S_y can be replaced by a Heavyside function H_y. H_y also performs the similar operation as that of S_y, but it helps in the recursive implementation of the half Gaussian. C as normalization coefficient. R_θ is a 2D rotation matrix given by:

$$R_\theta = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}.$$



(a) AHGSK (b) AHGSK at different orientations (c) AHGSK in 3D (d) $\mu = 10$ $\lambda = 1$

Figure 2.18: AHGSK with parameters μ and λ. (c) example of 3D AHGSK(d) example of discrete AHGSK.

By convolving the image with these oriented kernels, we obtain a set of smoothed images in different directions $I_\theta = I * G_{(\mu,\lambda)}(\theta)$ as shown in Fig.2.19.

2.5.2 Difference of Half Smoothing Filters (DHSF)

As shown in Fig.5.2, we want to estimate at every pixel a smoothed second derivative of the image along a curve passing through that pixel. In one dimension, the second derivative of a signal can be estimated using the Difference of Gaussian operator (DoG). As shown in Fig.2.21, by convolution with a DoG, the pulses of a 1D signal always appear as peaks while carrier signals will be completely deformed with zero mean. When it comes to two dimensions, we need to apply two half smoothing filters(AHGSK) having two different lambda (λ height) but with same width (μ mu) to get the directional derivatives. Then, we calculate the difference of the response of these two filters to get second derivative information of desired smoothness. We refer to this filter as DHSF and is illustrated in Fig.5.2(a).

Let $D(x, y, \theta)$ be the response of the DHSF obtained at pixel P located at (x, y). $D(x, y, \theta)$ is a function of the direction θ such that:

$$D(x, y, \theta) = G_{(\mu,\lambda_1)}(x, y, \theta) - G_{(\mu,\lambda_2)}(x, y, \theta) \tag{2.27}$$

μ, λ_1 and λ_2 correspond to the standard-deviations of the half Gaussians. Some of the examples of the response of the DHSF on a synthetic image at different orientations are shown in Fig.2.22.

2.5.3 Anisotropic Half Gaussian Derivative Filter /Kernel (AHGDK)

The non-symmetry of the smoothing filter makes it difficult to calculate gradient via an orientation tensor. Therefore, a derivative filter is used directly in the direction of the lowest standard deviation η, such that we have a smoothing filter in the half ξ direction and

(a) Original Image

(b) $\theta = 34°$, $\mu = 10, \lambda = 1$

(c) $\theta = 270°$, $\mu = 10, \lambda = 1$

(d) $\theta = 34°$, $\mu = 10, \lambda = 1.5$

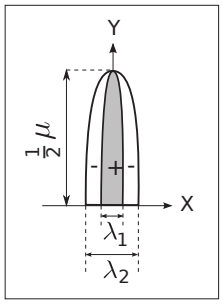
(e) $\theta = 270°$, $\mu = 10, \lambda = 1.5$

Figure 2.19: Smoothed images at different orientations and different images.

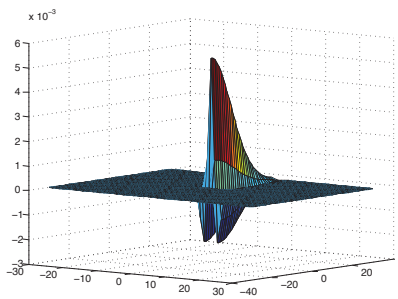
a derivative filter in the direction perpendicular to η. This filter is called the Anisotropic Half Gaussian Derivative Filter and is referred to as AHGDK in the rest of the chapters. It is then possible to estimate an anisotropic gradient, simply by the differences in response to the maximum and minimum directional filter. Mathematically, this filter is described by the equation:

$$\mathcal{G}_{(\mu,\lambda)}(x,y) \quad = \quad C_1 \cdot H\left(-y\right) \cdot x \cdot e^{-\left(\frac{x^2}{2\lambda^2} + \frac{y^2}{2\mu^2} \right)} \tag{2.28}$$

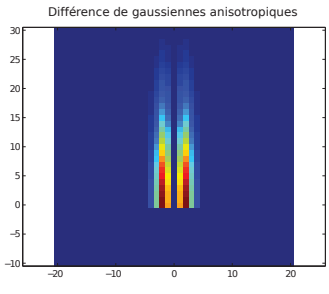
Where:



(a) DHSF



(b) DHSF in 3D



(c) discrete DHSF

Figure 2.20: Description of DHSF. for (c) $\mu = 10$, $\lambda = 1$ et $\lambda = 1.5$.

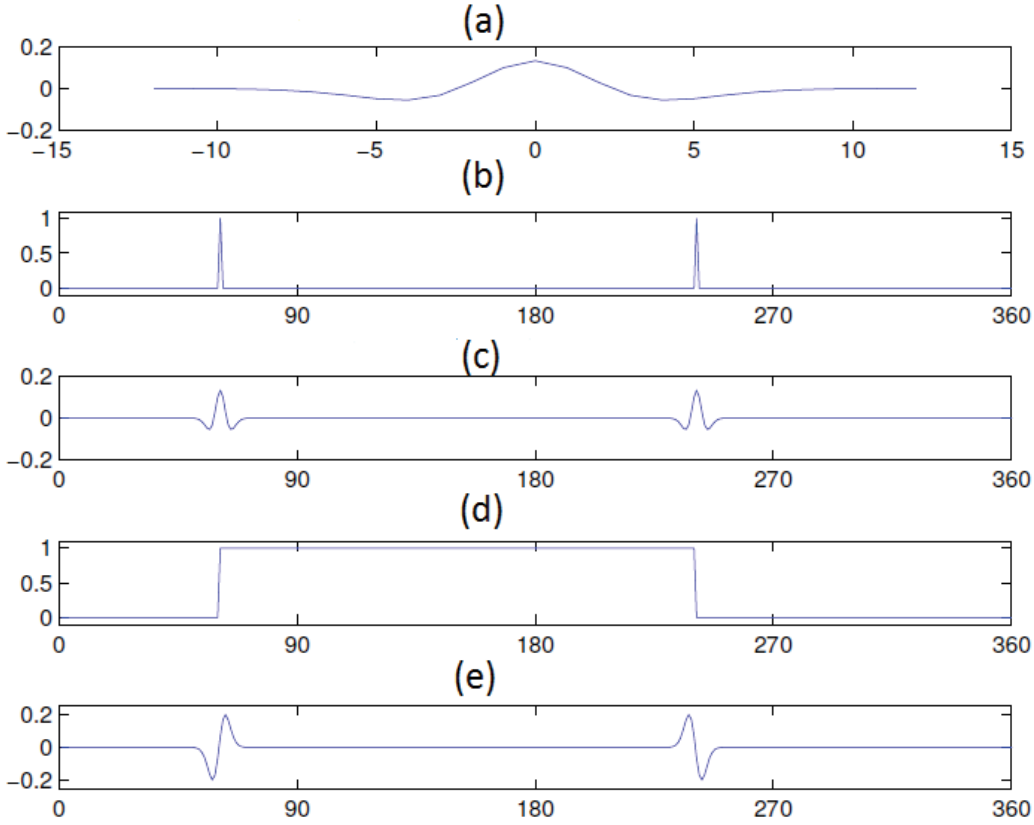


Figure 2.21: Difference of Gaussian in one dimension obtained by convolution of 1D signals. (a) Difference of Gaussians, $\sigma_1 = 1$, $\sigma_2 = 1.5$. (b) 1D signal with two pulses. (c) convolution of (a) & (b). (d) Test signal with two edge. (e) convolution of (a) & (b).

C_1 is the normalization factor. μ and λ controls the size of the Gaussian along the two orthogonal directions, radial and axial. Here, we are only interested in the causal part of the filter(As shown in Fig.2.23), and hence we cut the filter in the middle along the Y axis using a heaviside function H. By convolving the image with these derivative oriented kernels, we obtain a set of derivative images I_θ in different directions, θ ($I_\theta = I * \mathcal{G}_{(\mu,\lambda)}(\theta)$) as shown in Fig.2.26.

(a) $\theta = 34°$ (b) $\theta = 270°$

Figure 2.22: Response of DHSF at different orientation θ with parameters : $\mu = 10, \lambda_1 = 1$ and $\lambda_2 = 1.5$.

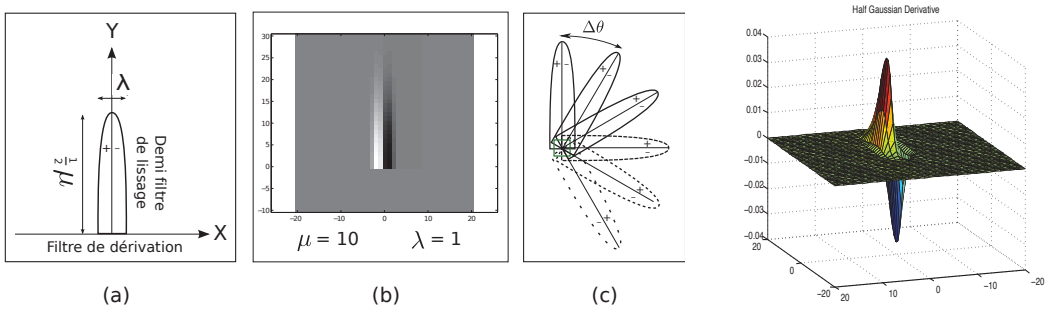
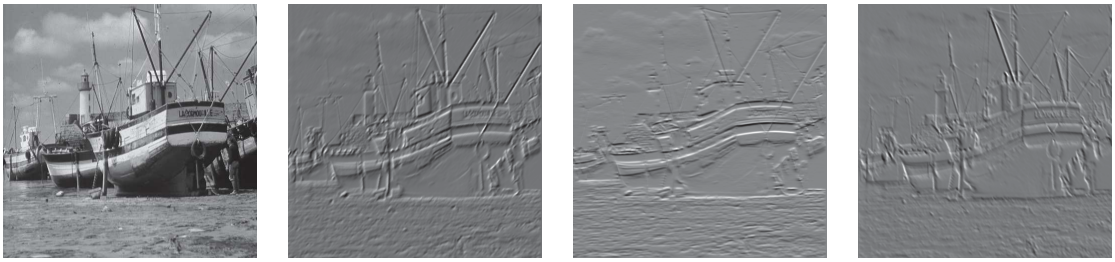


Figure 2.23: (a)An AHGDK (b) discreteized AHGDK. (c) AHGDK with orientation angle $\Delta\theta$. (d) AHGDK in 3D



(a) Originale (b) $\theta = 40°$ (c) $\theta = 90°$ (d) $\theta = 320°$

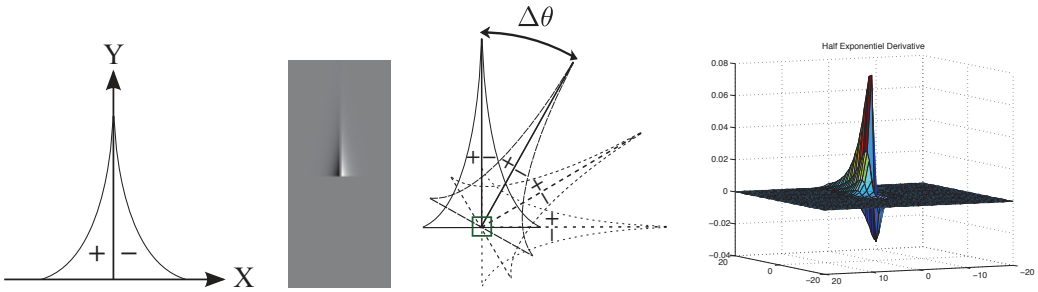
Figure 2.24: Image and its derivatives using AHGDK at different orientations θ, with $\mu = 5$ et $\lambda = 1$.

2.5.4 Anisotropic Half Exponential Derivative Filter /Kernel (AHEDK)

The Shen Castan filter (shown in the first part of this chapter) is modified to imitate the AHGDK and this modified filter is called as anisotropic half exponential derivative kernel (AHEDK). AHEDK shows similar characteristics and produces similar results as that of the above explained AHGDK. We use the recursive implementation of half exponential kernel which is of order 1 and is approximately 5 times faster than the recursive implementation of anisotropic half derivative kernel. By construction, AHEDK exhibits derivative characteristics along the X direction and smoothing characteristics along the Y direction as shown in Fig.2.25(a).

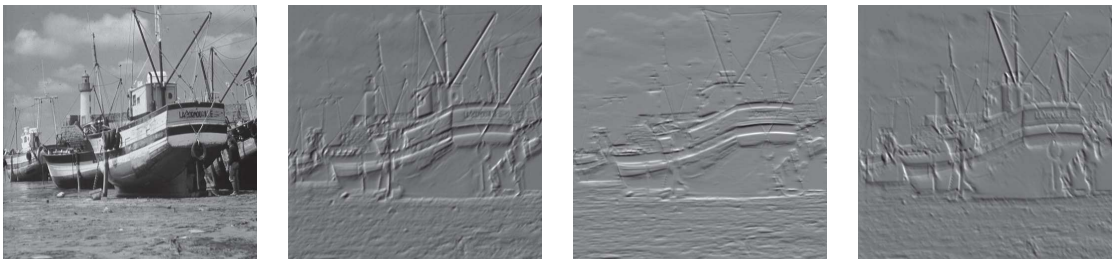
$$E_{(\mu,\lambda)}(x, y, \theta) = C_1 \cdot H(y) \cdot e^{(-\alpha_\mu \cdot y)} \cdot sign(x) \cdot e^{(-\alpha_\lambda \cdot |x|)} \tag{2.29}$$

The AHEDK is given by the Eq.2.29. The derivative information of an image is obtained by spinning the AHEDK around a pixel (x, y) as in Fig.2.25(c). In the Eq.2.29, C_1 is a normalization coefficient and $(\alpha_\mu, \alpha_\lambda)$ the height and width of the anisotropic half exponential kernel (see Fig.2.25(a)). Similar to Eq.2.28, only the causal part of this filter along the Y axis is used by cutting the kernel in the middle, in an operation that corresponds to the Heaviside function H.



(a) (b) (c) (d)

Figure 2.25: (a) AHEDK (b) discrete AHEDK (c)AHGDK with orientation angle $\Delta\theta$. (d) AHEDK in 3D.



(a) Originale (b) $\theta = 40°$ (c) $\theta = 90°$ (d) $\theta = 320°$

Figure 2.26: Image and its derivatives using AHEDK at different orientations θ, with $\alpha_\mu = 1$ et $\alpha\lambda = .2$.

2.6 Summary

To summarize this chapter, here different isotropic and anisotropic filters were presented and the applications and drawbacks of the full isotropic and anisotropic Gaussian were discussed. The advantages of splitting the full Gaussian filter in to half were highlighted and additionally, different half filters and its applications were presented. Here we have also presented different combination of half kernels like Gaussian smoothing, Gaussian derivative etc. In the rest of the chapter we refer to Anisotropic Half Gaussian Derivative Kernel as AHGDK, Anisotropic Half Exponential Derivative Kernel as AHEDK, Anisotropic Half Gaussian Smoothing Kernel as AHGSK and Difference of Half Smoothing Filters as DHSF.

Chapter 3

Image matching in videos using rotating filters

3.1 Introduction

This chapter presents a new image descriptor and matching methodology. We use the anisotropic half gaussian smoothing filter (AHGSK) explained in the previous chapter to obtain image description around a key-point. This is used as a low bit rate descriptor called Rotating Signal descriptor (RSD) and is used for image matching in videos. Finally, we also present the shortcomings of this method.

In this method, we initially use color Harris detector for finding the feature points in the image and then the Anisotropic Half Gaussian Smoothing Filter (AHGSK)is spun around the key-point to extract the color features in the form of signatures/signal. We call this signatures/signal as the Rotating Signal Descriptor (RSD). By construction, RSD is not euclidean invariant. Hence, we achieve euclidean invariance by FFT correlation between the two signatures. Moderate deformation invariance is achieved using Dynamic Time Warping (DTW) and then, by using a cascade verification scheme we improve the robustness of our matching method. Eventually, our method is illumination invariant, rotation invariant, moderately deformation invariant and partially scale invariant. Further, the dimension of RSD can be controlled by varying the angle of the rotating filter by small steps. Our descriptor with a dimension as small as 12 can give a good matching performance. The small dimension of the descriptor is the main motivation for extending the matching process to videos.

3.2 Rotating signal Descriptor (RSD)

We obtain a signature similar to [PMD12]. In [PMD12], the authors use an anisotropic half-Gaussian derivative filter/kernel (AHGDK) to obtain the signature around the key-point. In our work, we have replaced this derivative filter with a anisotropic half-Gaussian smoothing filter (AHGSK) Eq.2.26. The derivative filter has to be sampled at minute angles (rotated in many small steps) to obtain a good description. By using a smoothing filter, we can obtain good description by sparse sampling (rotating in large steps), thus reducing the size of the descriptor. The switch from derivative to smoothing filter is intended to reduce the size of the descriptor, thus increasing the speed of matching. As

in Eq.2.26 the AHGSK is given by :

$$g_{(\sigma_\xi,\sigma_\eta)}(x,y,\theta) \;=\; C \cdot S_y\left(R_\theta \begin{pmatrix} x \\ y \end{pmatrix}\right) \cdot e^{-\begin{pmatrix} x & y \end{pmatrix}.Z}$$

on a considered pixel point at $(x,\, y)$ with:

$$Z \;=\; R_\theta^{-1} \begin{pmatrix} 1/(2\sigma_\xi^2) & 0 \\ 0 & 1/(2\sigma_\eta^2) \end{pmatrix} .R_\theta \begin{pmatrix} x \\ y \end{pmatrix}$$

where σ_ξ and σ_η controls the size of the Gaussian along the two orthogonal directions, radial and axial. In our experiments, we have fixed $\sigma_\xi = 10$ and $\sigma_\eta = 1$ and S_y is a sigmoid function (along the Y axis) used to "cut" smoothly the Gaussian kernel. R_θ is a 2D rotation matrix and C a normalization coefficient. In our application for matching in videos, the length/dimension of the descriptor is fixed at 12 for each channel (i,e we rotate the filter at every 30°) and since the descriptor is obtained by rotating the filters around a key-point, its is called Rotating Signal Descriptor (RSD).

3.3 Illumination invariance

Convolution of a pixel in an image with all the kernels results in an intensity function, which depends on the direction of the kernel. Illumination invariance as proposed by diagonal illumination model [FFB95] is achieved by normalizing channel by channel:

$$(R_2, G_2, B_2)^t = M\,.\,(R_1, G_1, B_1)^t + (T_R, T_G, T_B)^t\,, \tag{3.1}$$

where: $(R_1, G_1, B_1)^t$ and $(R_2, G_2, B_2)^t$ are color inputs and outputs respectively. M is a diagonal 3x3 matrix and (T_R, T_G, T_B) represents a colour transition vector of the 3 channels. Our final descriptor is a concatenation of descriptors obtained from red, green and blue channels. As shown in the Eq 3.2:

$$\left\{g_{R_2(\sigma_\xi,\sigma_\eta)}(x,y,\theta), g_{G_2(\sigma_\xi,\sigma_\eta)}(x,y,\theta), g_{B_2(\sigma_\xi,\sigma_\eta)}(x,y,\theta)\right\} \tag{3.2}$$

3.4 Rotation invariance

The descriptor obtained from the previous step does not provide direct euclidean or deformation invariance. The common approach for achieving rotational invariance is by determining the orientation of the image region around the key-point and rotating the image region by that particular orientation. In our approach, the euclidean invariance is achieved in a simpler way by computing correlation between the descriptor curves, describing the two key-points respectively. The phase between the two curves is defined by the location of maxima of correlation.

RSD is a circular function. Clockwise rotation of RSD (in an image, where the Y axis is oriented downwards) in the image plane will shift the RSD to the right. Under these conditions, if we calculate the Euclidean distance between the two RSD's s_1 and s_2, the distance is affected by the rotation. It is therefore necessary to perform a reverse shift on one of the two RSD's, if we want to obtain a good result.

$$d(s_1,\; s_2) \;=\; \min_{\theta'} \sum_{\theta} \left(s_1(\theta) - s_2(\theta - \theta')\right)^2 \tag{3.3}$$

This distance is related to the maxima of correlation between s_1 and s_2 and is given by:

$$c(\theta) = s_1(\theta) * s_2(-\theta) \tag{3.4}$$

There are many ways to achieve correlation between 2 signals. One of the most efficient and popular way to achieve this is by using FFT as proposed by [JF08]. Through this approach, the minimum distance between s1 and s2 is achieved directly. We use the same procedure to achieve correlation between s_1 and s_2 and additionally, the FFTW3 [FJ05] has been used for faster implementation of FFT. Rotation invariance using correlation is illustrated from Fig.3.1.

3.5 Affine invariance

Since angles are not preserved under deformation or projective transforms, correlation alone is insufficient in ranking the match between a point in an image and the same point when seen in the second image under a changed viewpoint. In such a situation, curve deformation is needed to obtain affine invariant correlation scores. The simplest way to transform a curve into another curve is by the use of the dynamic time warping (DTW) algorithm. Additionally, since two signatures corresponding to the same point are relatively close (after resetting by correlation), it seems that DTW approach is apt while addressing the problem faced in our experiment. DTW is a popular similarity measure between the two temporal signals. This method has been widely used, particularly in speech recognition problems for locally deforming the time signals in order to compare them. In [Lem09], the authors have used an improved DTW for time series retrieval in pattern recognition applications.

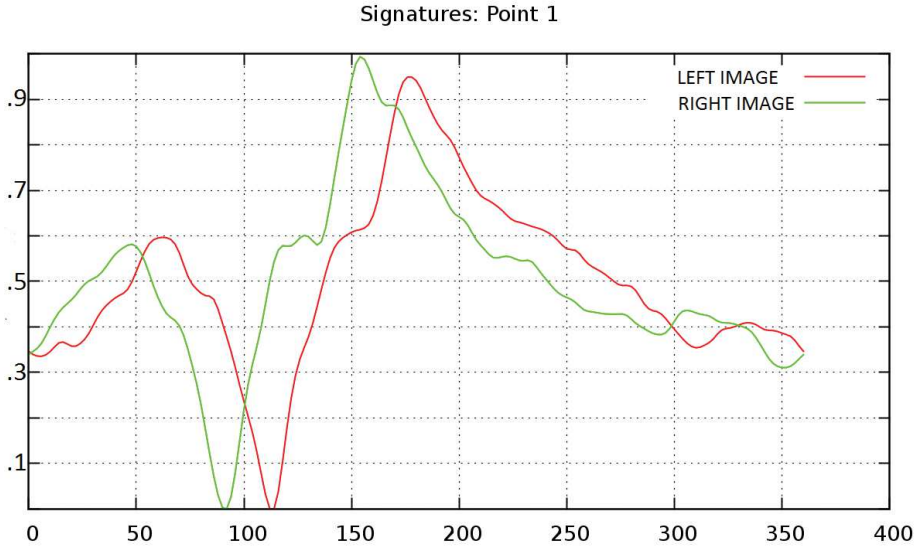
Considering the two curves that may have approximately the same shape as in Fig.3.2 and by calculating the point to point distance between the curves(As in Fig.3.2(a)), the deformation in the curves causes a false estimate of "resemblance" between the curves. But in order to find the similarity between them, it is necessary to ”warp” the time axis. If we know the transformation function between the two curves, then we can estimate the "resemblance" between them in a more realistic way as shown in Fig.3.2(b). The DTW algorithm provides this transformation function and calculates an optimal alignment between the samples of two time series that minimizes the cumulated distance.

3.5.1 Dynamic Time Warping (DTW)

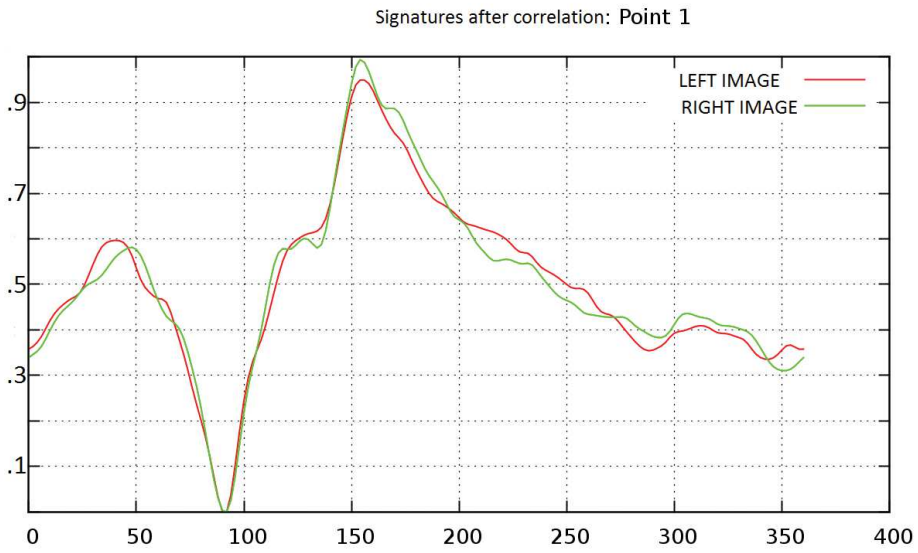
The DTW method is based on a dynamic programming algorithm which constructs a minimum cost matrix having possible offsets between the two curves. The minimum path in the matrix is then found using an inverse path. Fig.3.2(c) illustrates the search path in the matrix. Let s_1 and s_2 be the two curves that are approximately recalibrated by correlation. Consider the matrix $d[i][j]$ which is a point to point square of the difference between the two curves.

(a)

(b)

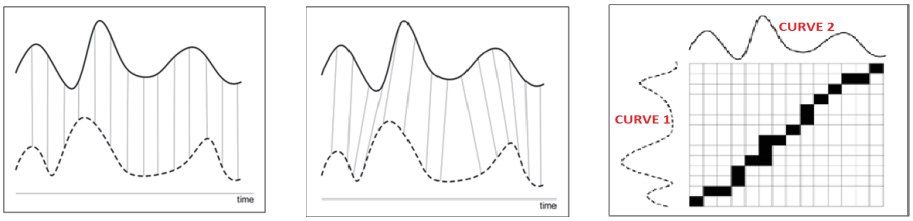


(c)



(d)

Figure 3.1: Rotation invariance using correlation at point 1. a) Left Image, b) Right Image. c) Initial RSD at point 1 in both the images ($\sigma_\xi = 10$, $\sigma_\eta = 1$, $\Delta\theta = 30°$). d) RSD after correlation.



(a) (b) (c)

Figure 3.2: Distance calculation between two curves/signature

$$d[i][j] = \big(s_1[i] - s_2[j])^2\big) = \begin{pmatrix} (s_1[1] - s_2[1])^2 & (s_1[1] - s_2[2])^2 & \cdots & (s_1[1] - s_2[n])^2 \\ (s_1[2] - s_2[1])^2 & (s_1[2] - s_2[2])^2 & \cdots & (s_1[2] - s_2[n])^2 \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ (s_1[n] - s_2[1])^2 & \cdots & \cdots & (s_1[n] - s_2[n])^2 \end{pmatrix}$$

The matrix $D[i][j]$ is a sum of squares of distances along a minimum cost path and is constructed as follows:

- $D[1][1] = d[1][1]$
- $D[1][j] = d[1][j] + D[1][j-1]$
- $D[i][1] = d[i][1] + D[i-1][1]$

and

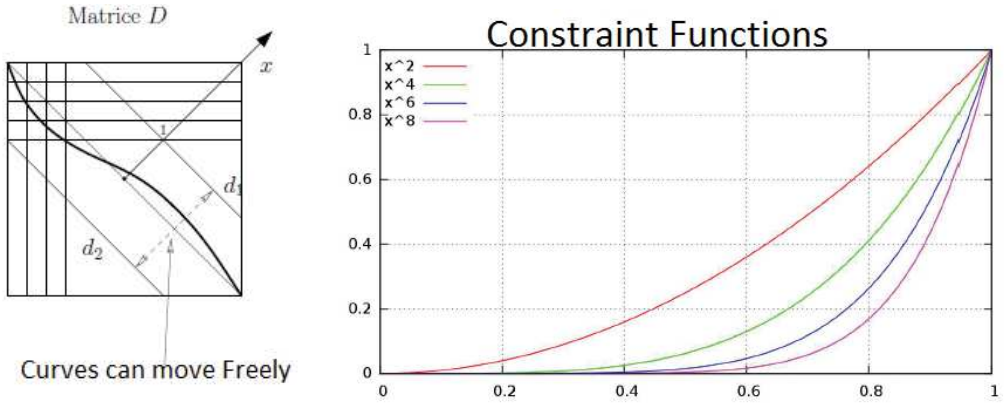
- $D[i][j] = d[i][j] + \min\{D[i-1][j-1], D[i-1][j], D[i][j-1]\}$

Finally, the transformation function of the curves s_1 and s_2 is given by the path matrix $D[i][j]$. This path is obtained by traversing the matrix from the bottom-right to top-left corner and choosing the minimum at every stage.

To avoid incompatible changes due to affine transformations, we want to constrain the DTW, so as to give a path that is close to the diagonal of the matrix $D[i][j]$. For this purpose, we add a penalty term to the matrix that is zero on the diagonal and increase as one moves away from it. The penalty term is given by a function $C(x)$ where x is the distance in the diagonal of the matrix taken in an orthogonal direction. Here, we need a function that remains close to zero in a certain band around the diagonal and polynomial of degree higher than 2 have this property. This property is illustrated in Fig.3.3. Our experiments have shown that a simple polynomial of degree 6 (Eq.3.5)gives the best results (we tested functions of the second to the eighth degree):

$$C : [0, 1] \longrightarrow \Re, \qquad C(\ x\ ) \ = \ \epsilon \cdot x^6 \tag{3.5}$$

Where,: ϵ is a normalization factor.
The value of ϵ is chosen based on the position of d_1 and d_2 with respect to the diagonal.



(a) (b)

Figure 3.3: Penalty deformation function. (a) The region around the diagonal where in the deformation path is allowed. (b) Example of penalty deformation function of degree 2, 4, 6 and 8. In our work, we use the polynomial function of degree 6.

In our experiments, the lines d_1 and d_2 are positioned on the sub-diagonal of the matrix D (on either side of the diagonal).

Fig.3.4 and Fig.3.5 illustrates the results of warping obtained on "point 1" in "Bust" sequence. As shown in Fig.3.5(a) the original signatures were recalibrated using correlation(rotation). By using constrained DTW, we are able to achieve a curve that is very close to the original curve and is illustrated in Fig.3.5(b).

3.6 Matching by Cascade Verification Scheme

We improve the robustness of the matching method by using a cascade verification scheme as shown in Fig.3.6. This scheme has 5 stages.

- In the first stage, we extract Rotating Signal Descriptors(RSD) for key-points(color Harris corners) in Image1 and for key-points in Image2. In Fig.3.6. D1,D2,D3,... and Z1,Z2,Z3,..... are the descriptors from Image1 and Image2 respectively.

- In the second stage, initially we consider the red channel part of the descriptor i,e R_2 as shown in eq.3.2. Then the FFT Correlation is performed between the descriptors D1,D2,D3,... (red channel part R_2) obtained from Image1 and the descriptors Z1,Z2,..(red channel part R_2) obtained from Image2, resulting in the correlation score S_c and angle ϕ. The same score and angle obtained is used for the green channel G_2 and blue channel B_2 part of the descriptor and the descriptors of Image 2 having an correlation score $S_c <$ threshold T, is selected. Similarly, we perform the FFT correlation between the descriptors Z1,Z2,Z3,... (red channel part R_2) of Image2 and all the descriptors of Image1.

- As in the second stage, initially we consider the red channel part of the descriptor i,e R_2 and CDTW is performed between descriptors of Image1 and all the descriptors from Image2 that has passed the first stage. The descriptors from Image2 are rotated by an angle ϕ (circularly shifted) before performing Constrained DTW (CDTW).

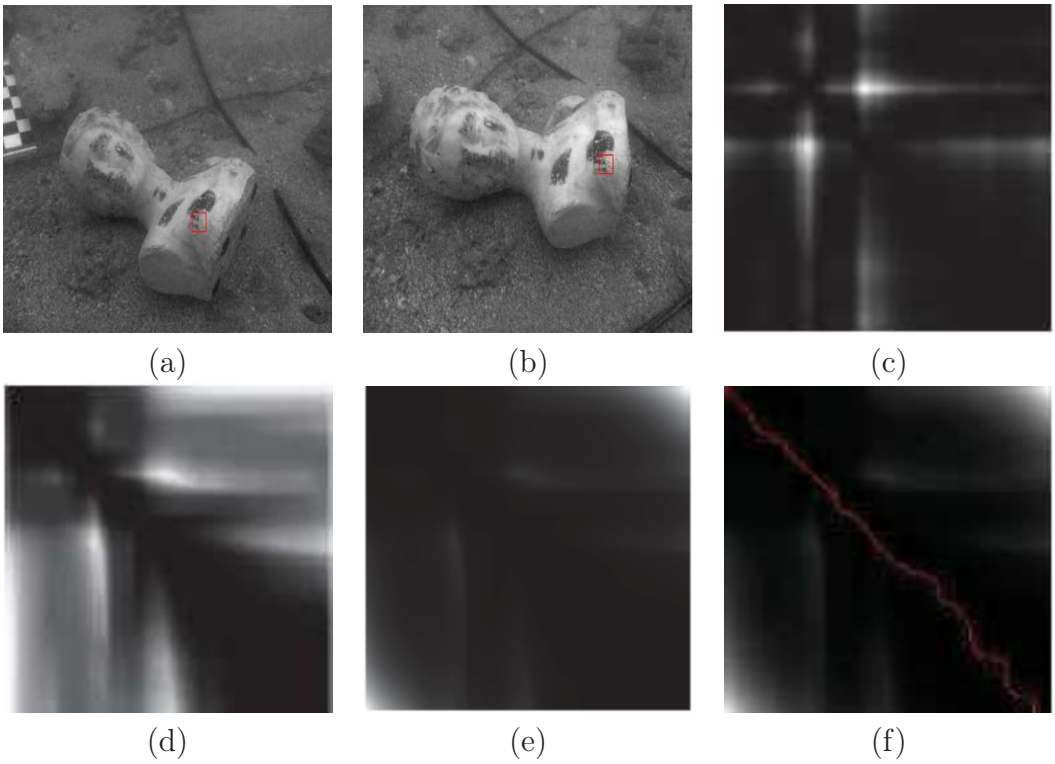
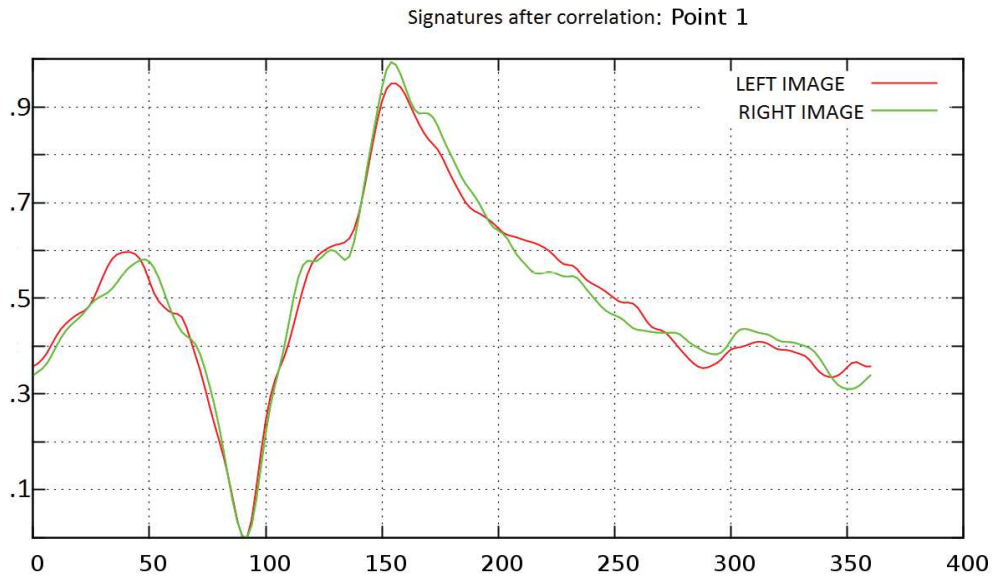
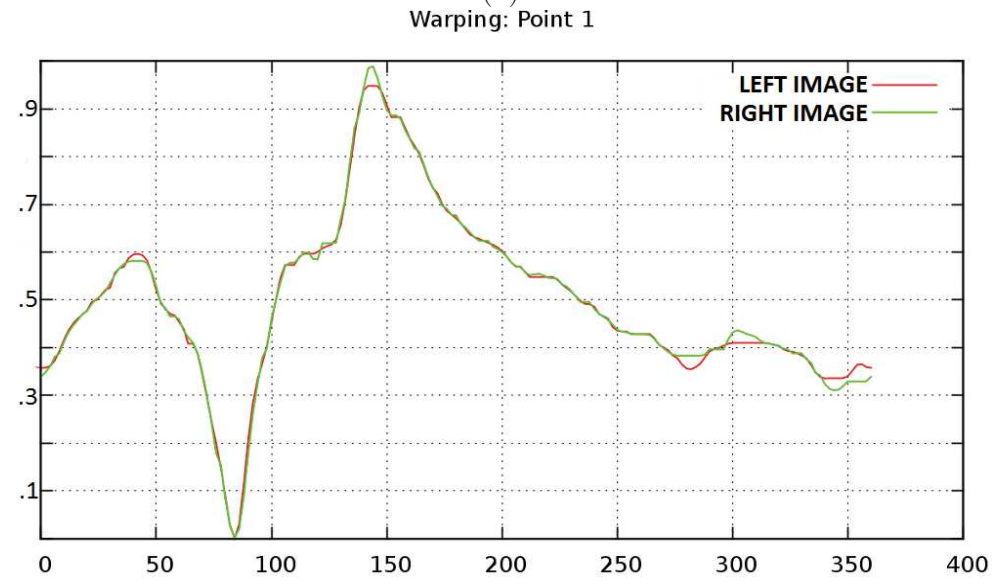


Figure 3.4: Warping results obtained on the "point 1" in the "Burst" image sequence. All the images are obtained from [PMD12]. (a) Image I1 indicating the "point 1" inside a red square . (b) Image I2 indicating the corresponding "point 1" inside a red square. (c) Matrix representing the sum of square distance between the two curves $d[i][j]$. (d) Integrated cost deformation matrix $D[i][j]$. (e) integrated + constrained distance matrix . (f) Path obtained with the constraints.



(a)



(b)

Figure 3.5: Results (a) After correlation (b) After constrained DTW

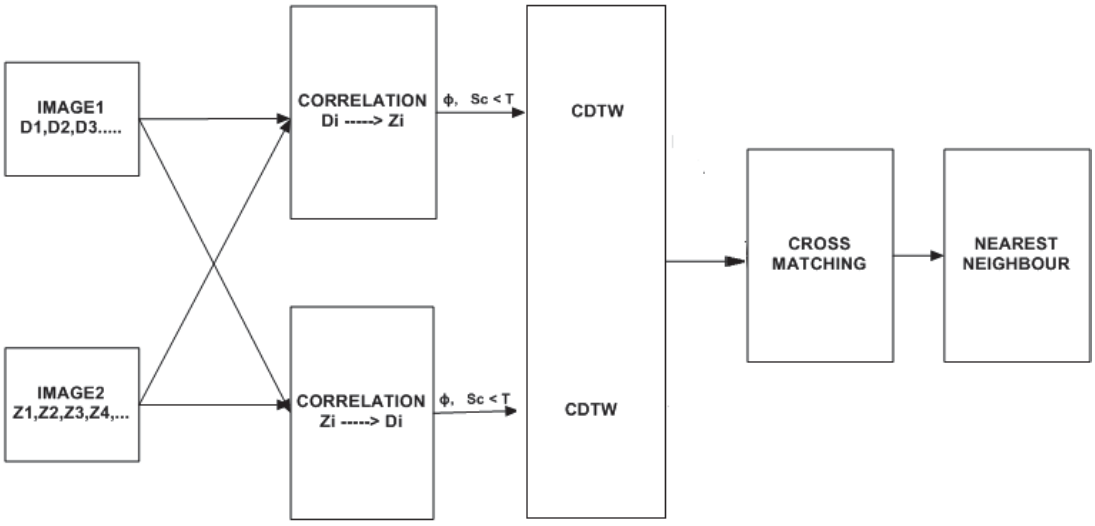


Figure 3.6: Flowchart of our cascade verification scheme

Thus, the obtained deformation score from the red channel is used for green and blue channels. Similarly, we perform CDTW between Descriptors of Image2 and all the descriptors from Image1.

- In the fourth stage, the ambiguities are removed by cross matching where the matches are computed from Image 1 towards Image 2 and from Image 2 towards Image 1 by eliminating the matches that do not correspond reciprocally. At this point we do obtain a robust set of matches but chances for false matches may still occur.

- In the Fifth stage, we consider the nearest neighbouring matches N for all the matches in Image1 . The neighbours are chosen in such a way that they are present inside a circle of radius R_{neigh} (50 pixels in our case). We do the same for all the matches in Image2. We then check whether the key-point and all its neighbours in Image1 corresponds to key-point and all its neighbours in Image2 and if they correspond, this key-point pair is chosen as the final match as seen in the example Fig.3.7. The red dots are the key-points and the green dots are the neighbours. The euclidean distance is used to compute the nearest neighbours and the radius R_{neigh} is set to 50 pixels. The number of neighbours N, is chosen from 1 to 4. In order to have a good number of matches, the value of N=2 is chosen in all our experiments. Fig.3.8 shows an image with a news anchor in a newsroom. There is a change in view point. This figure illustrates the robustness achieved by using the cascade verification scheme along with the matches obtained using different neighbouring stages.

3.7 Experiments and Discussions

The algorithm was implemented on c/c++ platform where Intel core 2 duo processor with 2.8 GHZ has been used. The complete code is our own implementation and have come

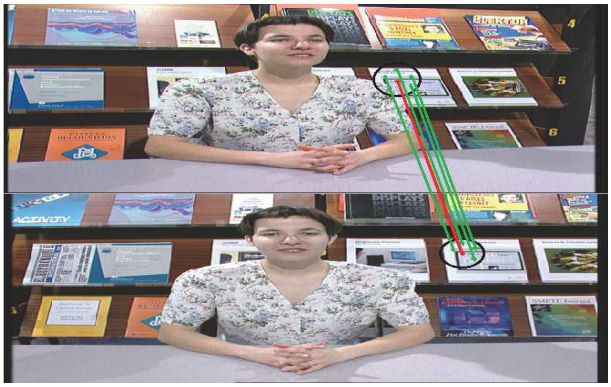


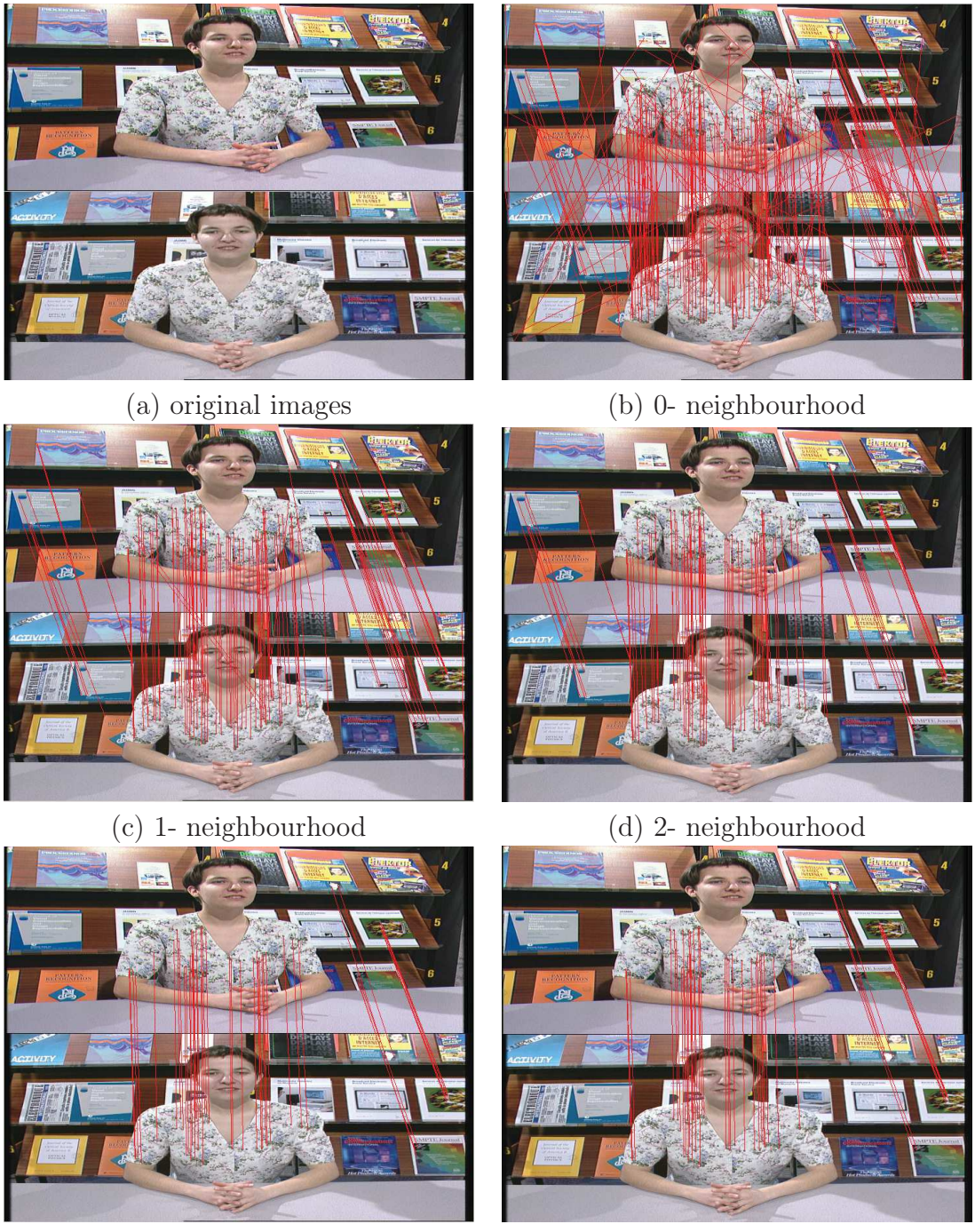
Figure 3.7: Nearest neighbour stage

up with a video framework that captures, buffers and displays the video frames in real time. Since the filter coefficients are fixed in each direction, we only need to generate the filter coefficients once and reuse them. A stack of rotating filters at different angles varying in steps of $\Delta\theta$, has also been generated once outside the video loop. This video implementation is for both gray level and color images. The image sequence used in our work are all color image sequences. Further, to improve the speed the complete code is multi-threaded.

This method has been tested on 2 video sequences and both are color image (RGB) sequence. The first video which is the fish sequence from [JY09] is a small sequence with 132 frames and having a resolution of 320x240. The sequence is about a fish moving in an aquarium. In the middle of the sequence, where the fish overlaps another big yellow fish, the appearance is made indistinguishable due to lack of texture and low video quality due to heavy compression. The second sequence, which is the bottle sequence, is our own sequence generated by using a web-cam. In this sequence, we can see the bottle being rotated, turned around and tilted sideways. The resolution of this sequence is 640x480 with 1892 frames. We compare our results with that of SIFT descriptor [Low04]. Since, there is no algorithmic implementation of SIFT for matching in videos, we have extracted significant frames from the video and applied SIFT on those individual frames. The SIFT code is provided by [Low04]. For all the video sequences that we have used, the rotating filter angle $\theta = 30°$. This results in a descriptor with 12 dimension .

We first tested our method on the ***'fish'*** video sequence. From the Fig.3.9, we can clearly see that our method performs similar to or better than SIFT. This is same for almost all the frames in sequence. Results of this video in Fig.3.9 demonstrates that our method can deal with low quality videos and images. In this case, the frame rate achieved was around 10 frames per second on an average. The matching results in video can be found in the link [1]. The second video sequence is that of a ***'bottle'***. From the Fig.3.10 we can clearly see that our method and SIFT gives almost similar results. Results of this video demonstrates that this method can deal with rotation, deformation and small changes in scale. In this case, the frame rate achieved was around 6 frames per second on average. This video can be found in the link [2].

[1] `https://www.youtube.com/watch?v=49MkqVc0OeM`
[2] `https://www.youtube.com/watch?v=N9GzV7bUzmU`

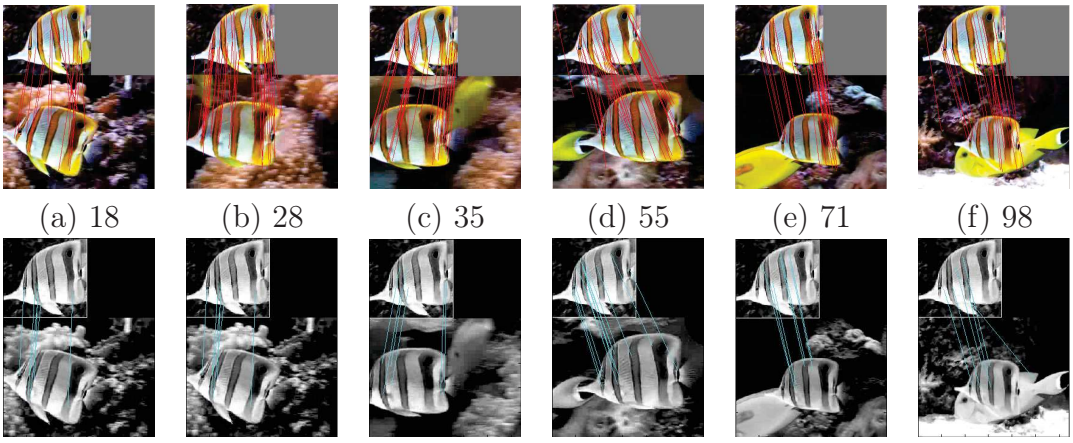


(a) original images (b) 0- neighbourhood

(c) 1- neighbourhood (d) 2- neighbourhood

(e) 3- neighbourhood (f) 4- neighbourhood

Figure 3.8: Robustness in matching with varying number of matches in the neighbourhood



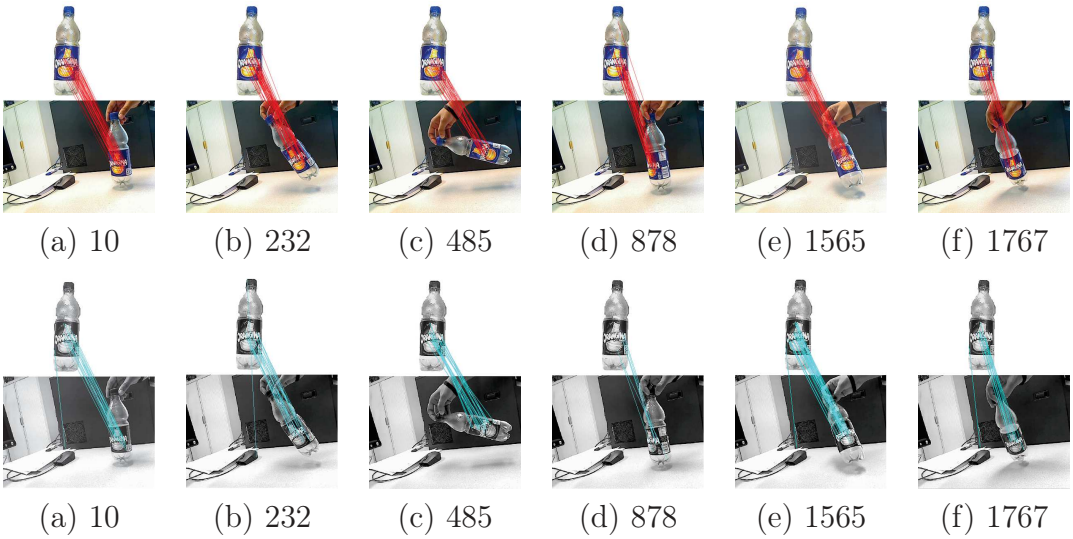
(a) 18 (b) 28 (c) 35 (d) 55 (e) 71 (f) 98

(a) 18 (b) 28 (c) 35 (d) 55 (e) 71 (f) 98

Figure 3.9: first six color images are output of matching using our method. last six gray images are output of matching using SIFT. Numbers shown are the frame numbers used in both SIFT and our method

We further experimented our RSD on the Oxford dataset [3]. Details of this dataset is given in the chapter.4.5.1. Here, the RSD is extracted from gray-scale image patches. In this experiment, we exclude the matching process (correlation, DTW and nearest neighbour) as explained above. The rotation and affine normalized gray-scale images patches are extracted as in [MS04] and is explained in detail in the next methodology. The results of RSD (dimensions 12, 36 and 72) is compared with that of SIFT and PCA SIFT for variations in blur, scale, rotation, compression, viewpoint and brightness. The

[3] `http://www.robots.ox.ac.uk/~vgg/research/affine/)`



(a) 10 (b) 232 (c) 485 (d) 878 (e) 1565 (f) 1767

(a) 10 (b) 232 (c) 485 (d) 878 (e) 1565 (f) 1767

Figure 3.10: first six color images are output of matching using our method. last six gray images are output of matching using SIFT. Frame numbers for both SIFT and our method is indicated below the images.

comparisons can be seen in Fig.3.11, Fig.3.12, Fig.3.13, Fig.3.14, Fig.3.15.

(a)BIKE 1 (b)BIKE 4 (c)BIKE 1-4

Figure 3.11: Comparison of RSD with SIFT and PCA-SIFT for variations in blur.

(a)BOAT 1 (b)BOAT 4 (c)BOAT 1-4

Figure 3.12: Comparison of RSD with SIFT and PCA-SIFT for variations in scale and rotation.

(a)COMPRESSION 1 (b)COMPRESSION 4 (c)COMPRESSION 1-4

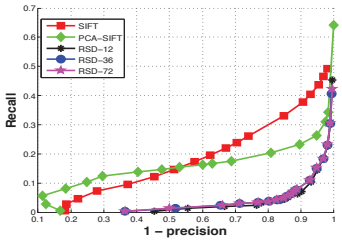
Figure 3.13: Comparison of RSD with SIFT and PCA-SIFT for variations in compression.

3.8 Summary

In this chapter a new method for image matching has been explained. Initially, the image features (RSD) are extracted by spinning the AHGSK around a key-point and then the cascade matching scheme is used to achieve rotation and affine invariance. Additionally, we also achieved robust matches using the nearest neighbouring scheme. This methodology forms our first contribution and resulted in the publication [VMD15]. Though the above explained method was different and new, it had the following drawbacks:

(a)GRAFF 1

(b)GRAFF 4

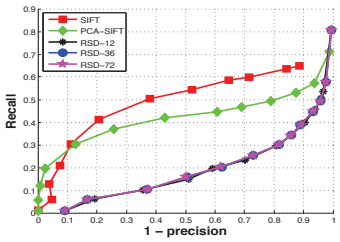


(c)GRAFF 1-4

Figure 3.14: Comparison of RSD with SIFT and PCA-SIFT for variations in viewpoint.

(a)LUVEN 1

(b)LUVEN 4



(c)LUVEN 1-4

Figure 3.15: Comparison of RSD with SIFT and PCA-SIFT for variations in brightness.

- Our assumption of using the same correlation and DTW score for the other two channels (Green and Blue) may result in loosing some of the good matches. To overcome this, we can apply correlation and DTW independently to the signatures of three channels to the corresponding images. But this would result in different "warpings" of the signatures for each key-point pair. Additionally, we should also design a way to fuse the individual distances. One way to deal with this issue is to use Multi Dimension Dynamic Time Warping (MDDTW) [tHRH07]. But, MDDTW is very complicated and time consuming.
- Dynamic time warping modifies the curves. If the constraint function is not chosen carefully, then a good match may end up as a bad match and vice-versa. To find a global constraint function to overcome this issue, we need to introduce a learning stage.
- The proposed method using DTW provides moderate deformation/affine invariance.
- The proposed method is not scale invariant.
- Since, RSD is obtained by spinning the AHGSK around the key-point(color Harris corner), the information extracted by the RSD is very weak. RSD alone fails to capture the geometry of the region around the key-point. As a result, when this descriptor is evaluated for the standard dataset (Here only grey images were used in the evaluation), it gives weak results when compared to that of the SIFT descriptor. The results can be seen in the Fig.3.11, Fig.3.12, Fig.3.13, Fig.3.14, Fig.3.15. These drawbacks can be negated by using RSD-HOG descriptor introduced in the next section.

Chapter 4

RSD-HoG

In the previous chapter we had proposed a new low bit-rate image descriptor called RSD (Rotating Signal descriptor) and a new cascade image matching method based on correlation and dynamic time warping. RSD was a weak descriptor and it failed to capture the geometry of the region around the key-point. The cascade matching stage had many drawbacks too. In this chapter, we tried to over come the drawbacks of the previous image descriptor by proposing a new robust image patch descriptor called RSD-HoG.

4.1 Introduction

In the literature review, we spoke about image descriptors based on gradients, curvature and filter responses. Most of the descriptors based on filter response were weak and as a result failed to compete with the gradient based descriptors. Since these filter based descriptors were obtained at individual key-points, they failed to capture the complete geometry of the region around the key-point. Our descriptor can be considered as a combination of the descriptors based on gradients, curvature and filter responses.

In our descriptor, we have embedded the response (RSD) of the half Gaussian filter in the Histogram of oriented Gradient framework and hence, the name RSD-HoG. In this descriptor:

- The response of a rotating Anisotropic Half Gaussian Derivative Kernel (AHGDK) or Anisotropic Half Gaussian Exponential Kernel(AHEDK) around pixels is used to construct signatures (RSD).
- RSD is constructed for each pixel in the region around the key-point i.e we follow a dense approach.
- The orientation of the edges and the anisotropic gradient directions are extracted from the RSD, thus capturing the geometry/curvature of the region around the key-point.
- These orientations are binned separately in different ways as in HoG, to get different variants of the descriptor.

4.2 Filtering Stage

When compared to isotropic filters, anisotropic filters have an advantage in detecting large linear structures. For anisotropic filters, the gradient magnitude at the corners decreases as the edge information under the scope of the filter decreases (Fig.4.1(a)). Consequently, robustness to noise when dealing with tiny geometric structures weakens. This drawback can be nullified by using Half filters such as AHGDK or AHEDK. In addition to this, with the use of elongated and oriented half kernels in our experiments, we are able to estimate the two principal edge directions as in Fig.4.1(b).

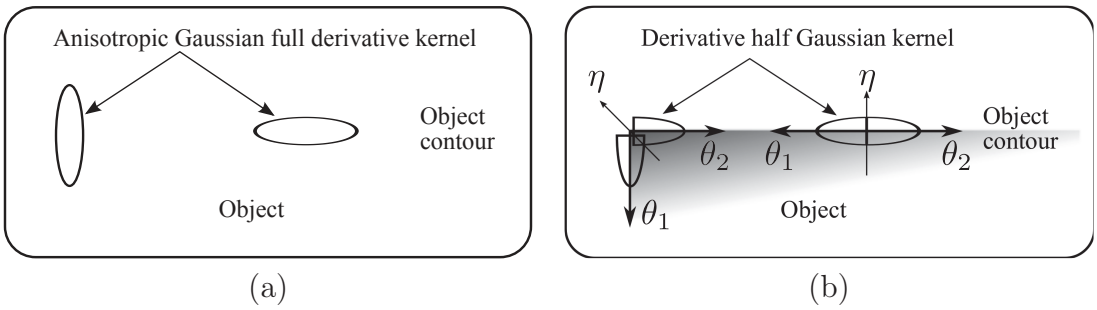


Figure 4.1: (a) Anisotropic full Gaussian at a corner. (b) Edge directions and gradient direction extraction using Anisotropic half Gaussian

In this work, we use Anisotropic Half Gaussian Derivative Kernel (AHGDK) for descriptor generation and we also use Anisotropic Half Gaussian Exponential Kernel(AHEDK) to obtain a faster variant of our descriptor. We then compare the two descriptors with that of the descriptor obtained from Anisotropic Full Gaussian derivative kernel and show the advantage of using Half kernel over Full kernels. These kernels has been explained in detail in chapter.2.

At each pixel (x, y) in the region around the key-point, we spin the AHGDK at different orientations θ to obtain the derivative information or signature (RSD) $\mathcal{Q}(x, y, \theta)$ as in Eq.4.2. Replacing the AHGDK with the AHEDK, we get the RSD $\mathcal{E}(x, y, \theta)$ as in Eq.4.4.

$$\begin{aligned} \mathcal{Q}(x,y,\theta) &= I_{R_\theta} * \mathcal{G}_{(\mu,\lambda)}(x,y) && (4.1)\\ &= I_{R_\theta} * C_1 \cdot H(-y) \cdot x \cdot e^{-\left(\frac{x^2}{2\lambda^2} + \frac{y^2}{2\mu^2}\right)} && (4.2)\\ \mathcal{E}(x,y,\theta) &= I_{R_\theta} * E_{(\mu,\lambda)}(x,y) && (4.3)\\ &= I_{R_\theta} * C_1 \cdot H(y) \cdot e^{(-\alpha_\mu \cdot y)} \cdot sign(x) \cdot e^{(-\alpha_\lambda \cdot |x|)} && (4.4) \end{aligned}$$

Examples of $\mathcal{Q}(x, y, \theta)$ for a few selected points is shown in Fig.4.2. In the direction of the edges, the filter responds with positive or negative peaks in the function $\mathcal{Q}(x, y, \theta)$. Point 1 belongs to a smooth/homogeneous region and as a result, the response of the filter is almost 0 with no peaks. Since point 2 belongs to a textured region, the signal is relatively stochastic with low amplitude. Points 3 and 4 belongs to horizontal and oblique edges respectively. The two peaks in the signal $\mathcal{Q}(x, y, \theta)$ indicates the orientation at which the edges are present. It is the same for the points 5 and 6 at the corners where the peaks indicate the direction of the edges.

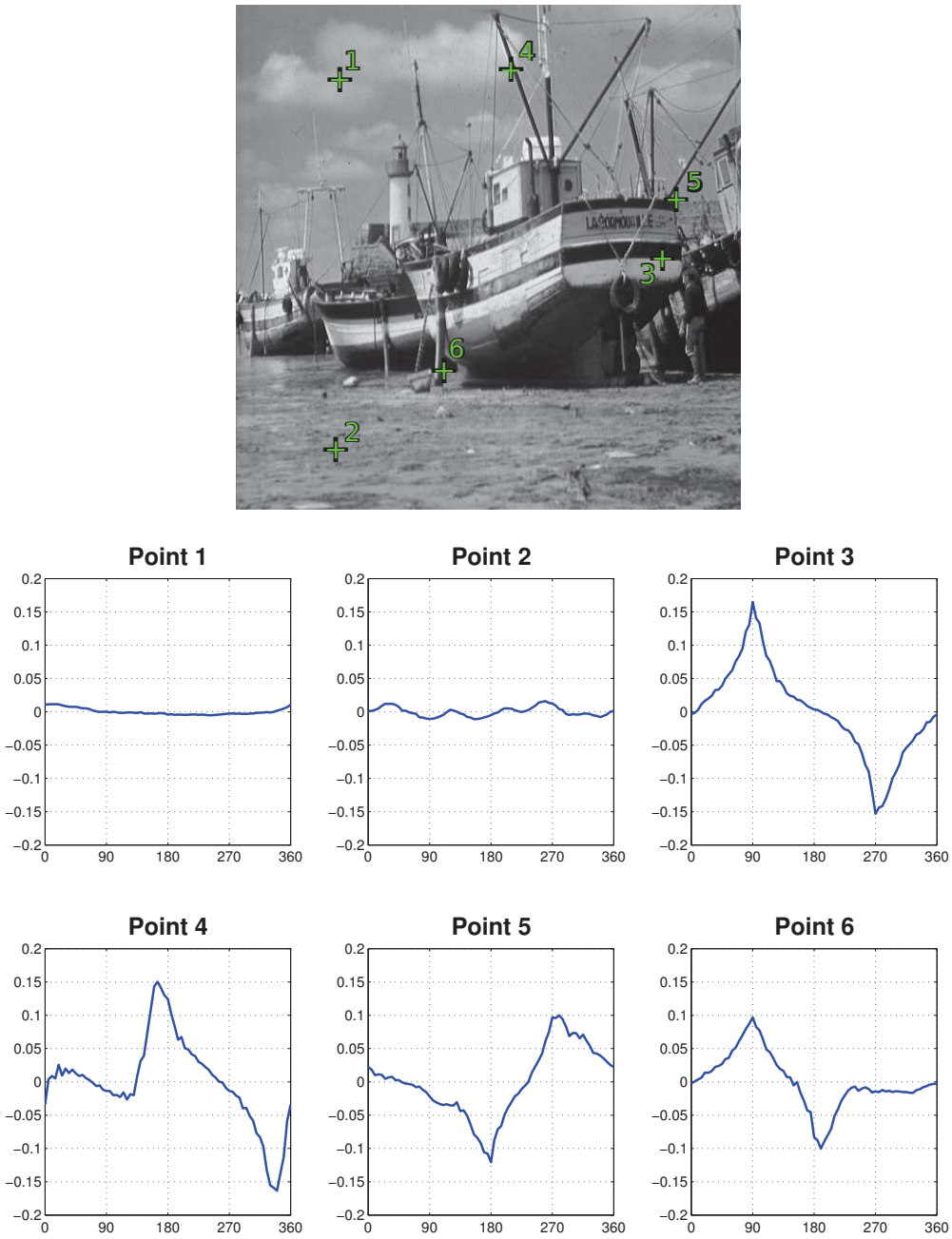
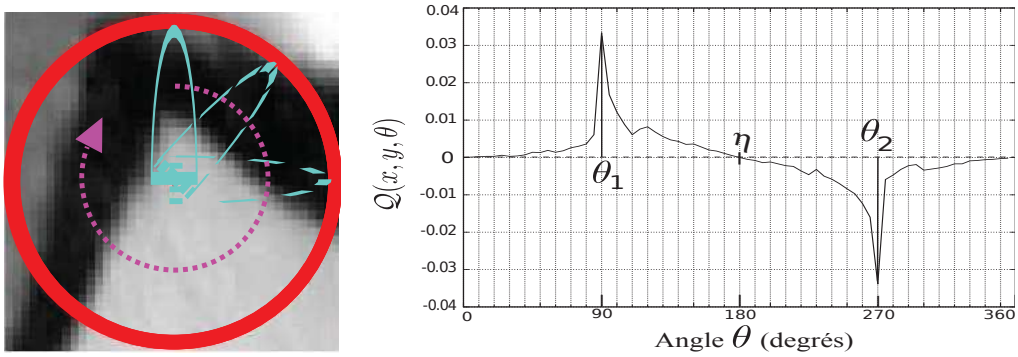


Figure 4.2: Point selection in the top image. Functions $\mathcal{Q}(x, y, \theta)$ for each point with $\Delta\theta = 5°$, $\mu = 5$ et $\lambda = 1$. X axis represents the orientation of the filter, whereas y axis represents the response of the filter.

4.3 Anisotropic gradient magnitude and direction estimation



(a) (b)

Figure 4.3: (a) AHGDK applied to a key-point x_p, y_p in the image patch to obtain $\mathcal{Q}(x, y, \theta)$. (b) An example of $\mathcal{Q}(x, y, \theta)$.

We construct the signature $\mathcal{Q}(x_p, y_p, \theta)$ by considering the response of the above described rotating filter at a pixel location (x_p, y_p) as in Fig.4.3(a). Fig.4.3(b) shows a sample signature obtained by applying the AHGDK at the the pixel location (x_p, y_p) in steps of 2°. Anisotropic gradient magnitude $\|\nabla I\|_a$ and its associated direction η for the key-point at location (x_p, y_p) are obtained by considering the global extrema G_{max} and G_{min} of the function $\mathcal{Q}(x_p, y_p, \theta)$ along with θ_1 and θ_2. The two angles θ_1 and θ_2 defines a curve crossing the pixel (an incoming and outgoing direction) and thus, representing the two edge directions. Both these global extrema are combined to maximize the gradient $\|\nabla I\|_a$, i.e:

$$\begin{cases} G_{max} = \max\limits_{\theta\in[0,360°[} \mathcal{Q}(x_p, y_p, \theta) & \text{and} \quad \theta_1 = \arg\max\limits_{\theta\in[0,360°[}(\mathcal{Q}(x_p, y_p, \theta)) \\ G_{min} = \min\limits_{\theta\in[0,360°[} \mathcal{Q}(x_p, y_p, \theta) & \text{and} \quad \theta_2 = \arg\min\limits_{\theta\in[0,360°[}(\mathcal{Q}(x_p, y_p, \theta)) \\ \|\nabla I\|_a = G_{max} - G_{min} \end{cases} \tag{4.5}$$

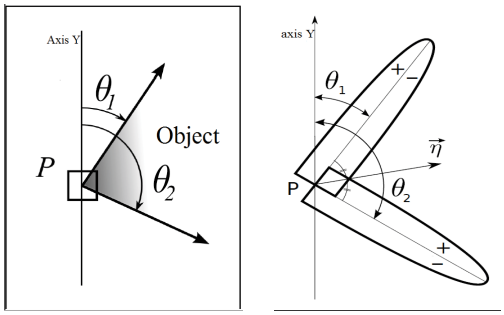
We calculate η (Fig.4.4) by taking the average of the two angles θ_1 and θ_2 :

$$\eta = \frac{\theta_1 + \theta 2}{2} \tag{4.6}$$

The Fig.4.5, illustrates the calculation of different gradients $\|\nabla I\|$ and angles $(\eta, \theta_1, \theta_2)$ having different parameters of λ and μ.

4.4 Methodology

In this section, we describe the various steps involved in the construction of the descriptor, as shown in the Fig.4.6. Initially, we use the Harris affine feature detector to find the affine regions in the image. The detected regions are of different elliptical sizes, which



(a) (b)

Figure 4.4: Calculation of $\eta(x, y)$ from θ_1 and θ_2. (a) θ_1 and θ_2 corresponds to the two directions of an object edge at a pixel P. (b)Angle η is the bisector of the two angles θ_1 and θ_2.

depends on the detection scale. Once we have this detected region, the next step is to affine normalize the region.

4.4.1 Normalization

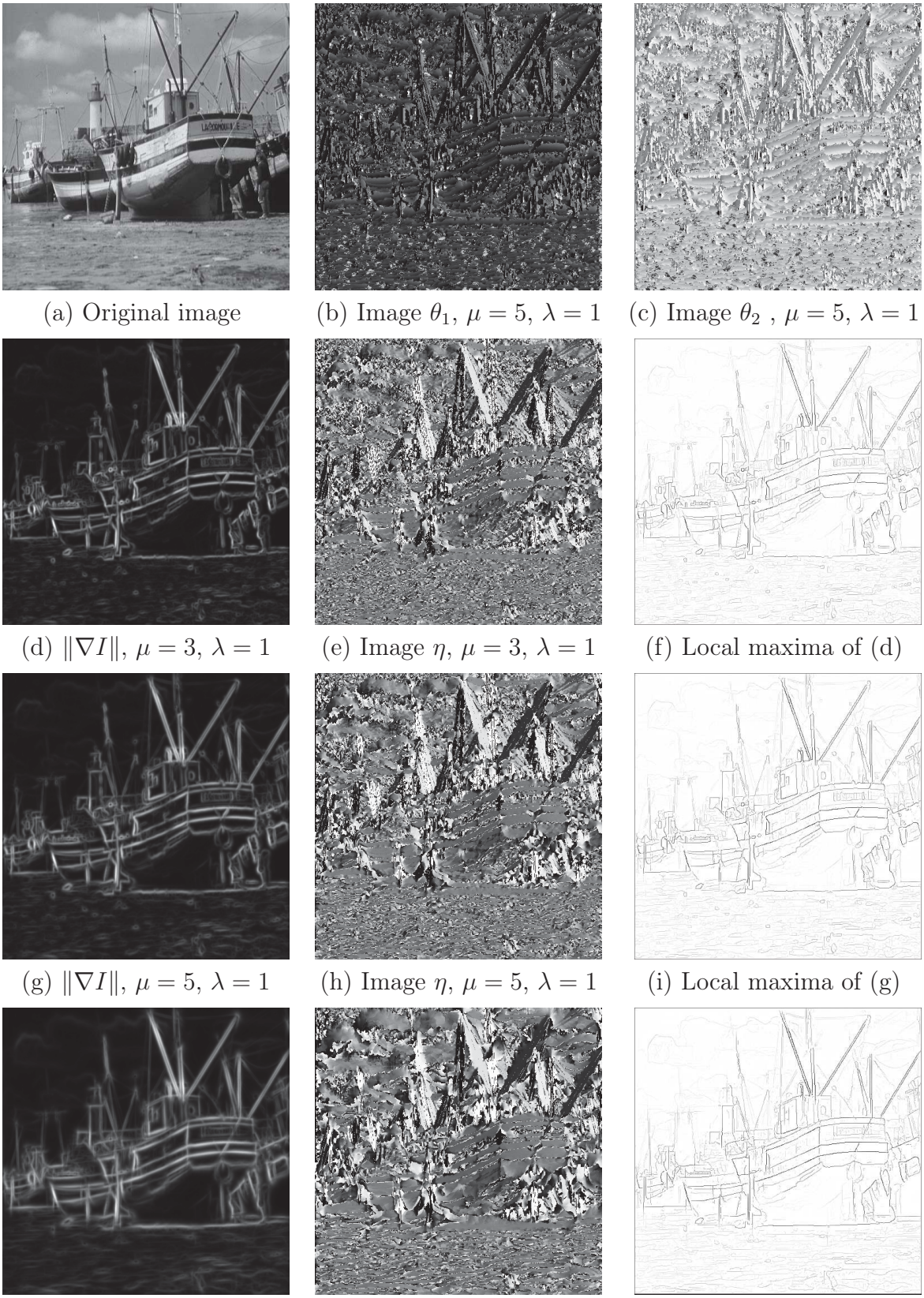
We achieve normalization of the region as proposed by [MS03]. As there is a possibility to change the size or shape of the detected region by scale or affine covariant construction, we can modify the set of pixels that contribute to the descriptor computation [MS03]. Typically, larger regions contain more signal variations. Hessian-Affine detector mainly detects blob-like structures for which the signal variations lies on the blob boundaries. We make sure that these signal changes are included into the descriptor by considering a measurement region that is three times larger than the detected region. Next, we map all the detected regions to a circular region of a constant radius to obtain scale and affine invariance. In order to represent the local structure at a sufficient resolution, the size of the normalized region is chosen such that it is not too small. In our case, this size is arbitrarily set to 41×41 pixels. Authors in [KS04, MS03] have used a similar patch size. If the detected region is larger than the normalized region, then the image of the detected region can be smoothed by a Gaussian kernel before region normalization. The standard derivation of Gaussian used for smoothing is given by the ratio of detected region to the normalized region [MS03].

As in [FWH12], consider a detected region denoted by a symmetrical matrix $M \in \Re^{2\times 2}$. For any point X in the region, it satisfies the condition:

$$X^T M X \leq 1 \tag{4.7}$$

If $M = \frac{1}{c^2}E$, where E is the identity matrix, then the region is a circular region and c is its radius. Otherwise, it is an elliptical region. The aim of normalization step is to warp the detected elliptical/circular region into a canonical circular region as shown in Fig.4.7. The same point X' which belongs to a normalized region satisfies Eq.4.8, where r is the radius of the normalized region,

$$X'^T X' \leq r^2 \tag{4.8}$$



(a) Original image (b) Image θ_1, $\mu = 5$, $\lambda = 1$ (c) Image θ_2 , $\mu = 5$, $\lambda = 1$

(d) $\|\nabla I\|$, $\mu = 3$, $\lambda = 1$ (e) Image η, $\mu = 3$, $\lambda = 1$ (f) Local maxima of (d)

(g) $\|\nabla I\|$, $\mu = 5$, $\lambda = 1$ (h) Image η, $\mu = 5$, $\lambda = 1$ (i) Local maxima of (g)

(j) $\|\nabla I\|$, $\mu = 10$, $\lambda = 1$ (k) Image η, $\mu = 10$, $\lambda = 1$ (l) Local maxima of (j)

Figure 4.5: Estimation of gradient $\|\nabla I\|$ and the angles θ_1, θ_2 and η with different parameters μ and λ. $\Delta\theta = 5°$.

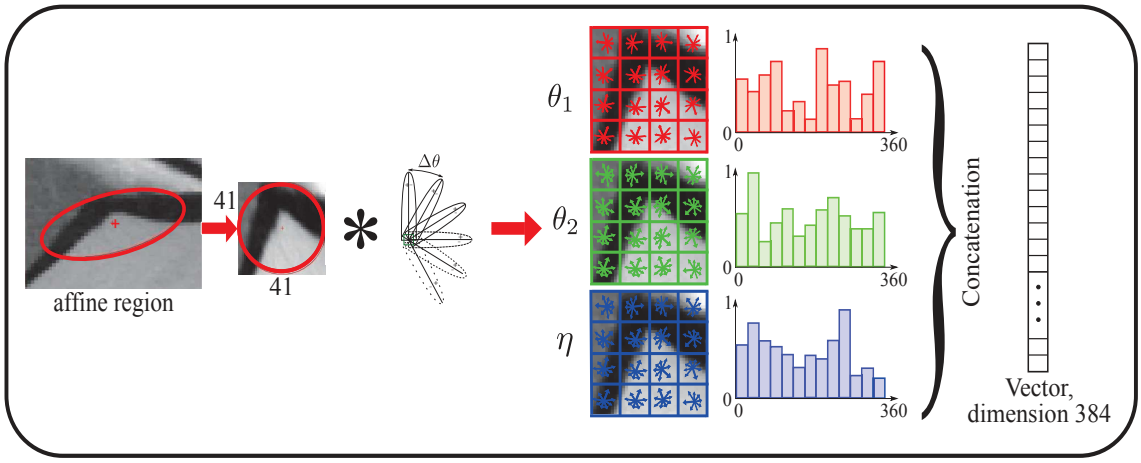


Figure 4.6: Descriptor construction methodology

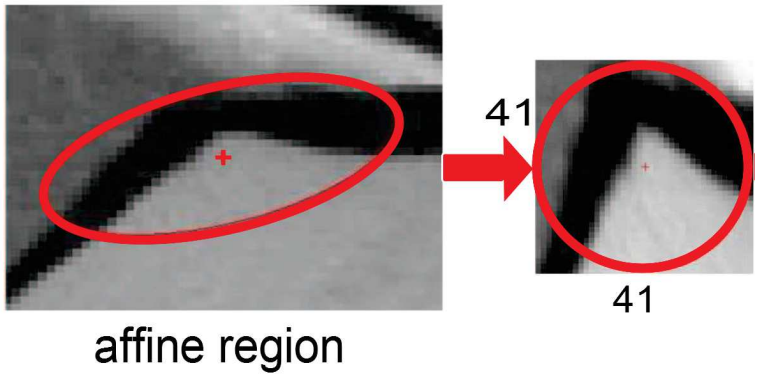


Figure 4.7: Region normalization

When both the above equations are combined, we have:

$$X = \frac{1}{r} M^{-.5} X = T^{-1} X' \tag{4.9}$$

Therefore, for each sample point X' in the normalized region, we calculate its corresponding point X in the detected region and take the intensity of X as the intensity of X' in the normalized region, i,e $I(X') = I(X)$. Since X is not exactly located at a grid point, we use bilinear interpolation to obtain $I(X)$.

Once we have achieved the affine invariance, the next step will be to achieve the rotation invariance. This is done by rotating the normalized region by the dominant gradient orientation, that is computed on a small neighbourhood of the region center. We estimate the dominant gradient orientation as in [Low04] by building a histogram of gradient angles weighted by the gradient magnitude and selecting the orientation that corresponds to the largest histogram bin.

4.4.2 Descriptor construction

Once we have the rotation and affine normalized image patch, the next step is to convolve the image patch with AHGDK or AHEDK at different orientations and extract the RSD for each pixel. As explained in the previous section, the RSD for each pixel is extracted, where:

- Angle at the maxima, θ_1 and the response G_{max} at θ_1.
- Angle at the minima, θ_2 and the response $\|G_{min}\|$ at θ_2.
- Anisotropic gradient angle η and its magnitude $\|\nabla I\|_a$.

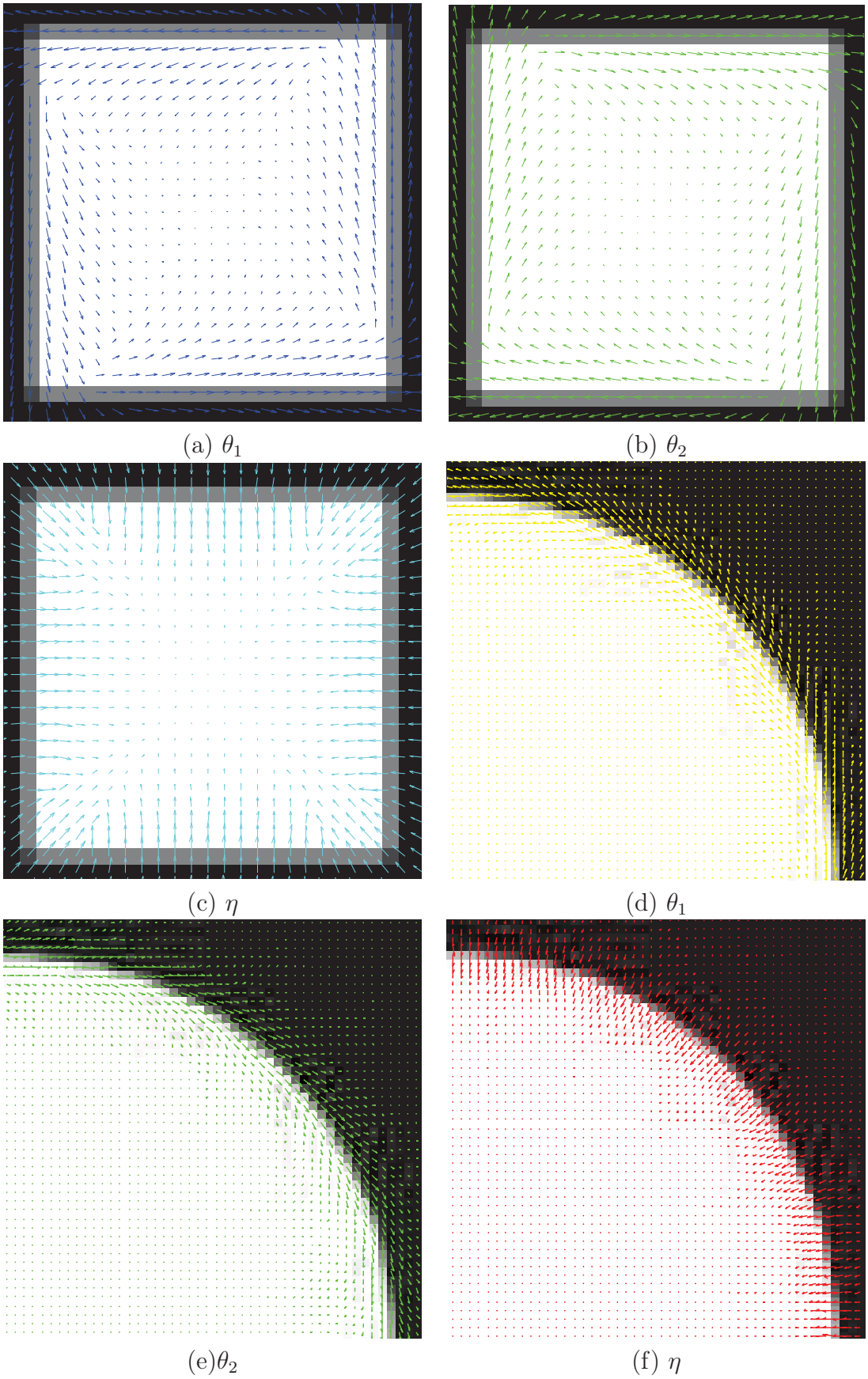
By construction, this method detects curvature information with θ_1 and θ_2, by using only 1st order derivatives. Examples of the curvature captured by our method can be seen in Fig.4.8. The angles $(\theta_1, \theta_2 \& \eta)$ for the synthetic square were obtained using the AHGDK. Whereas, the angles $(\theta_1, \theta_2 \& \eta)$ for the synthetic quarter circle are captured using the AHEDK. Once we have these informations, the next step is to form three intermediate descriptors by constructing the Histogram of oriented gradient (HoG) having all the 3 angles separately. For the HoG construction, we follow the same approach as in [Low04], and divide the patch into 4×4 blocks of equal size (blocks on the extreme right and bottom contain 11×11 pixels).

- Angle at the maxima, θ_1 is binned by weighing with its response G_{max} to form: $HoG_{\theta_1} = \{\theta_{1_{bin1}}, \theta_{1_{bin2}}, \theta_{1_{bin3}}, \theta_{1_{bin4}}.....\theta_{1_{bin128}}\}$
- Angle at the minima, θ_2 is binned by weighing with its response G_{min} to form: $HoG_{\theta_2} = \{\theta_{2_{bin1}}, \theta_{2_{bin2}}, \theta_{2_{bin3}}, \theta_{2_{bin4}}.....\theta_{2_{bin128}}\}$
- The anisotropic gradient, η is binned by weighing with its response $\|\nabla I\|_a$ to form: $HoG_{\eta} = \{\eta_{bin1}, \eta_{bin2}, \eta_{bin3}, \eta_{bin4}.....\eta_{bin128}\}$

Finally, we combine the three intermediate descriptors in 4 different ways:

- **DESCT1-theta1-eta** : A 256 dimension(length) descriptor obtained by concatenating HoG_{θ_1} and HoG_{η}.
- **DESCT2-theta2-eta** : A 256 dimension descriptor obtained by concatenating HoG_{θ_2} and HoG_{η}.
- **DESCT3-theta1-theta2** : A 256 dimension descriptor obtained by concatenating HoG_{θ_1} and $HoG_{\theta 2}$.
- **DESCT4-theta1-theta2-eta** : A 384 dimension descriptor obtained by concatenating HoG_{θ_1}, HoG_{θ_2} and HoG_{η}.

We follow the same procedure to construct the descriptors using AHEDK and call them **EXP-DESCT1-theta1-eta**, **EXP-DESCT2-theta2-eta**, **EXP-DESCT3-theta1-theta2** and **EXP-DESCT4-theta1-theta2-eta** respectively.



(a) θ_1 (b) θ_2

(c) η (d) θ_1

(e)θ_2 (f) η

Figure 4.8: Curvature captured by our method illustrated on synthetic square and quarter circle.

Dataset	**Image Change**	# **Images**	# **Key-points**	**Resolution**	**Harris threshold**
BIKE	blur	5	878,665,624, 482,384	1000x700	1000
BOAT	rotation+zoom	5	3023,2935,2379, 1423,1199	800x640	1000
GRAFF	Viewpoint	5	1758,1973,2172, 1976,2153	800x640	1000
COMPRESSION	Compression	5	1402,1425,1421 1400,1540	800x640	1000
LEUVEN	illumination	5	902,723,615 500,399	921x614	1000

Table 4.1: Sequences and images used in the image matching experiments: 1st column indicates the dataset name. 2nd column indicates the variation in the image. 3rd column indicates the number of images in the sequence used in our experiments. 4th column indicates the number of key-points in each of the 5 images starting from the 1st image to the 5th image. 5th column indicates the resolution of the images in the sequence. 6th Column indicates the threshold used in the Harris affine region detector

4.5 Experiments and results

In this section, we initially present the dataset that is used in our experiments and then the details of the evaluation protocol used for image matching experiments. Later, we give the implementation details and finally conclude this section with the results and discussion.

4.5.1 Dataset

The descriptors used in our experiments are tested with the dataset provided by Mikolajczyk et al.[1]. where the standard dataset includes several image sequences. Each image sequence generally contains 6 images. The image sequence have different geometric and photometric transformations such as image blur, lighting, viewpoint, scale changes, zoom, rotation and JPEG compression. Additionally, they have provided the ground truth homographies for every image transformation with respect to the first image of every sequence. Some of the images in different sequences are shown in the Fig.4.9. The properties of the image sequence in the dataset is tabulated in the Table.4.1.

4.5.2 Evaluation criteria

We use the *recall vs* $1 - precision$ plot as proposed by Mikolajczyk et al. in [MS03]. The criterion is based on the number of correct matches and the number of false matches

[1] `http://www.robots.ox.ac.uk/~vgg/research/affine/`

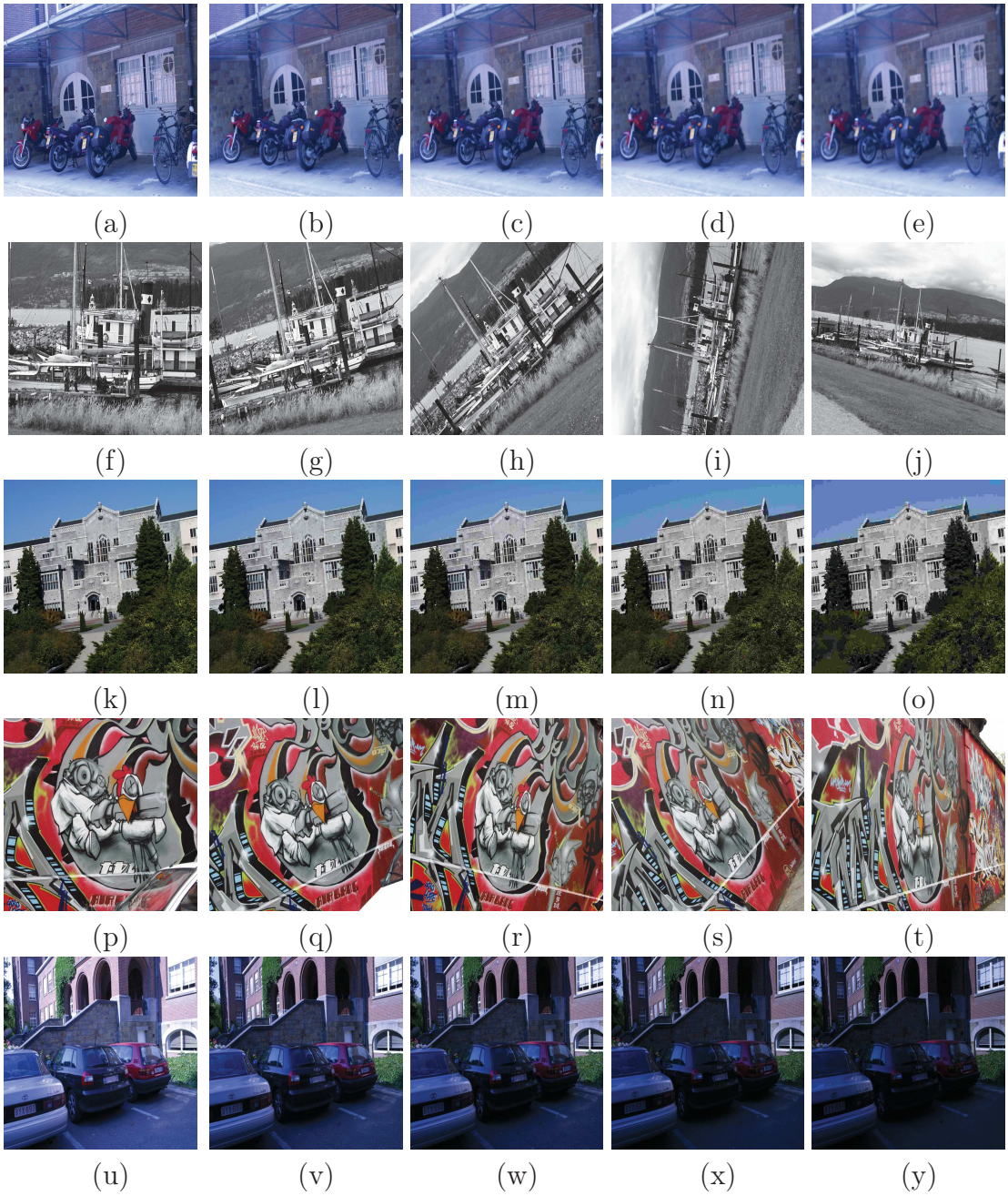


Figure 4.9: Oxford dataset for image matching. The images in the first row are named as BIKE dataset. It has variations in blur. The images in the 2nd row are called as the BOAT dataset having rotation and zoom variations. The images in the 3rd row have variations in compression and are named as COMPRESSION dataset. The images in the 4th row have variations in viewpoint and are named as GRAFF dataset. The images in the final row belongs to the LEUVEN dataset and have variations in brightness.

obtained for an image pair:

$$\text{recall} = \frac{\#\text{ correct matches}}{\#\text{ correspondences}} \tag{4.10}$$

$$\text{1-precision} = \frac{\#\text{ false matches}}{\#\text{ correct matches} + \#\text{ false matches}} \tag{4.11}$$

The number of correct matches and the correspondences is determined by the overlap error. The two regions A and B are said to correspond to each other if the overlap error is ε_0, which is defined as an error in the image area covered by sufficiently small region, as shown in Eq.4.12. For the evaluation, we have used the plugin/code (without any modification) provided by [2].

$$\varepsilon_0 = 1 - \frac{A \cap H^T \cdot B \cdot H}{A \cup H^T \cdot B \cdot H} \tag{4.12}$$

4.5.3 Parameters

Our descriptor and its variants depends on 4 different parameters (Table.4.2): $\Delta\theta$, $No-of-bins$, height and width (μ, λ for AHGDK and α_μ, α_λ for AHEDK). The rotation step $\Delta\theta$ is fixed to 10°. An increase in the rotation step results in loss of information. For constructing the histogram, the image patch is divided into 16 blocks. All blocks are of size 10x10 (Since we are using a patch of size 41×41 , the blocks in the extreme right and bottom have 11x11 size). The number of bins ($No-of-bins$) is fixed to 8 per block, resulting in a $8 * 16 = 128$ bins for 16 blocks. Increasing the number of bins gives almost the same performance but it increases the dimensionality of the descriptor. AHGDK height μ and width λ is fixed to 6 and 1 respectively. AHEDK height α_μ and width α_λ is fixed to 1 and 0.2 respectively. Width and height parameters are chosen empirically, so as to have a ratio sharpness length suitable for robust edge detection, which generally gives good results in most cases. This ratio is compatible with the angle filtering step. Euclidean distance is used as the comparison metric. We have used the recursive implementation of Gaussian filter as in [Der93] and similarly, the recursive implementation for the exponential filter has been used. When implemented recursively, the exponential filter is upto 4 times faster than the recursive implementation of Gaussian filter, providing similar performance.

Table 4.2: Parameters

filter Height (μ, α_μ)	filter Width (λ, α_λ)	Rotation step ($\Delta\theta$)	No of BINS
6 , 1	1 , 0.2	5°	8

4.5.4 Descriptor Performance

The performance of descriptors obtained using both AHGDK and AHEDK are compared against SIFT, GLOH and DAISY. For SIFT and GLOH, the descriptors are extracted from the binaries provided by Oxford group [3]. DAISY descriptor for patches is extracted

[2] http://www.robots.ox.ac.uk/~vgg/research/affine/
[3] http://www.robots.ox.ac.uk/~vgg/research/affine/

from the code provided by [TLF10].

Here, we test the descriptors using similarity threshold based matching, as this technique is better suited for representing the distribution of the descriptor in its feature space [MS03]. These descriptors has been compared on all the images in the dataset. Initially, we show the quantitative results using the *Recall vs* $1 - Precision$ plot that can be found in Fig.4.10, Fig.4.11, Fig.4.12, Fig.4.14 and Fig.4.13. Each figure has 12 graphs. In the first 4 graphs of every figure, we compare the descriptors based on AHGDK and its variants with SIFT, SURF and DAISY. In the next 4 graphs, we compare the descriptors based on AHEDK and its variants with SIFT, SURF and DAISY and the final 4 graphs of these figure gives the comparison between descriptors based on AHGDK , AHEDK and Anisotropic Full Gaussian Derivative Kernel(AFGDK). Finally, we show the qualitative results using nearest neighbour matching strategy.

Variations in blur. (BIKE)

The quantitative results for the BIKE sequence can be seen in Fig.4.10. The qualitative results can be found in Fig.4.15.

- From the first four graphs it is seen that the descriptors based on AHGDK outperforms SIFT, DAISY and GLOH.
- The same can be said about descriptors based on AHEDK (the next four graphs).
- In the final four graphs, when the descriptors based on half filters is compared to that of full filter, we see that the descriptors based on half filters perform better than that of the AFGDK. Amongst all the descriptors compared in the Fig.4.10, the descriptor **DESCT4-theta1-theta2-eta** gives good performance. But, it has the highest dimension.

Variations in rotation and zoom. (BOAT)

The quantitative results for the BOAT sequence can be seen in Fig.4.11. The qualitative results can be found in Fig.4.16.

- From the first four graphs it can be seen that, the descriptors based on AHGDK outperforms SIFT, DAISY and GLOH.
- The same can be said about descriptors based on AHEDK (the next four graphs).
- When compared with AFGDK, we see that the descriptors based on half filters perform better than that of the AFGDK. Amongst all the descriptors compared in the Fig.4.10, the descriptor **DESCT4-theta1-theta2-eta** gives good performance, whereas, the descriptor based on AFGDK performs badly.

Variations in Compression. (COMPRESSION)

The quantitative results for the COMPRESSION sequence can be seen in Fig.4.11. The qualitative results can be found in Fig.4.17

- Filters based on AHGDK outperforms SIFT, DAISY and GLOH.

- The same can be said about descriptors based on AHEDK (the next four graphs).
- Here all the descriptor based on AHGDK, AHEDK and AFGDK shows good performance.

Variations in brightness. (LUVEN)

The quantitative results for the LEUVEN sequence can be seen in Fig.4.13. The qualitative results can be found in Fig.4.18.

- Filters based on AHGDK outperform SIFT, DAISY and GLOH.
- The same can be said about descriptors based on AHEDK (the next four graphs).
- Here, the descriptor based on AHGDK and AHEDK gives better performance than AFGDK.

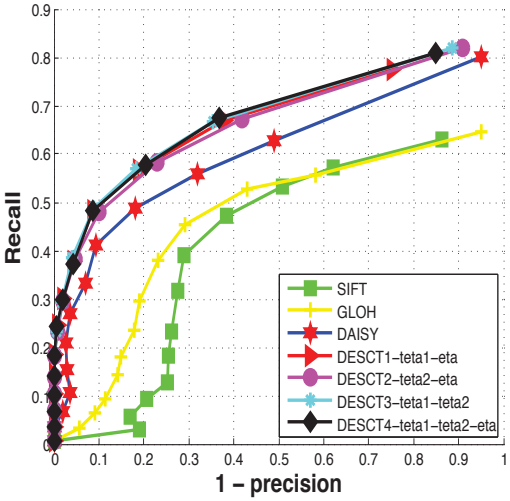
Variations in viewpoint. (GRAFF)

The quantitative results for the GRAFF sequence can be seen in Fig.4.14. It is a challenging sequence and the the qualitative results can be found in Fig.4.19.

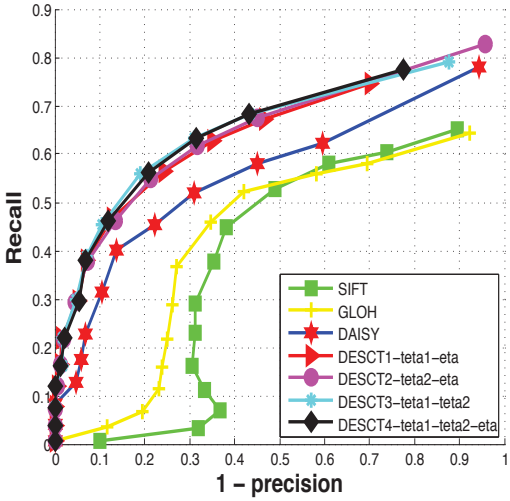
- In the first two graphs, filters based on AHGDK outperforms SIFT, DAISY and GLOH. In the third graph the DESCT-teta1-teta2-eta has an edge over the rest. In the 4th graph it is seen that all the descriptors fail.
- Here, the descriptor based on AHEDK performs well in first two graphs but fails in the next two graphs.
- Here, the descriptor based on AHGDK and AHEDK gives better performance than AFGDK.

4.6 Summary

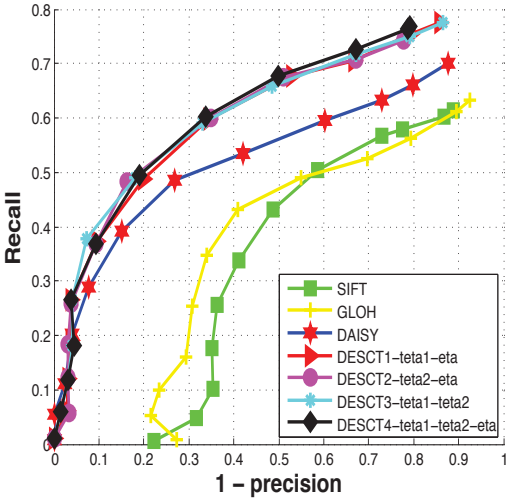
In this chapter, we propose a new image descriptor called RSD-HoG whose principle and results have been published in [VMDM15c, VMDM15a]. The originality of this method is that, it captures the geometry of the image patch by embedding the response of the AHGDK or AHEDK in a HoG framework. Our method incorporates edge direction as well as anisotropic gradient direction for generating the descriptor and its variations. On the standard dataset provided by the Oxford group, our descriptor and its variants outperform SIFT, GLOH and DAISY. By this method, we are able to overcome the drawbacks of the descriptor that was proposed in chapter.3.



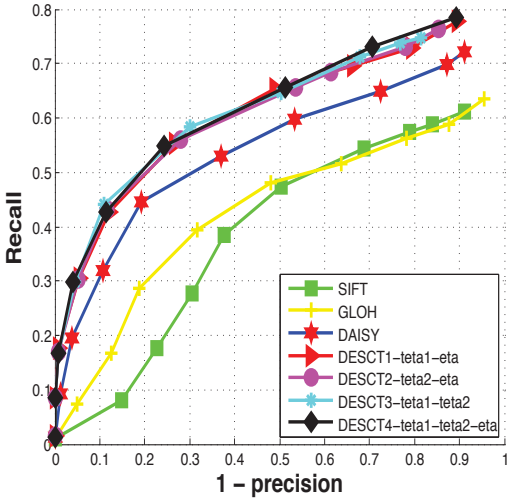
(a) BIKE 1-2 (AHGDK)



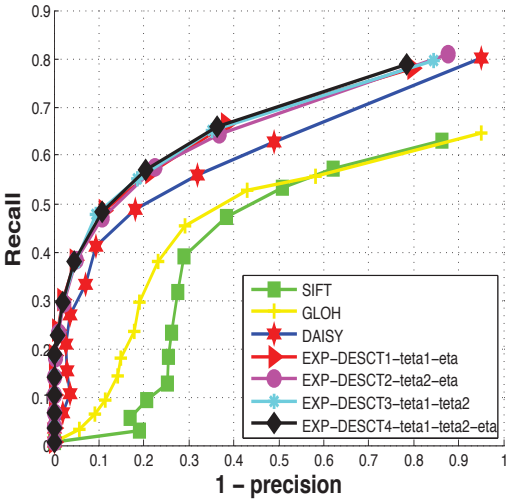
(b) BIKE 1-3 (AHGDK)



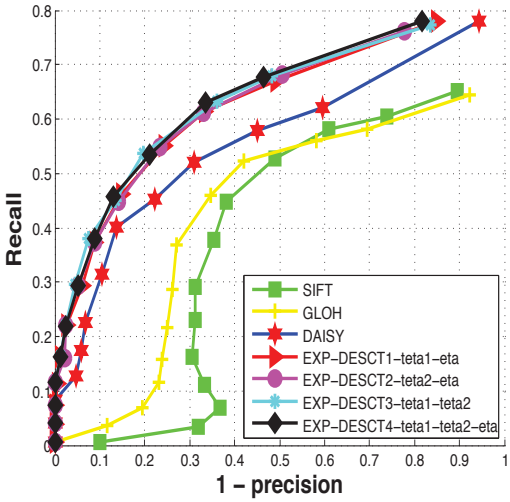
(c) BIKE 1-4 (AHGDK)



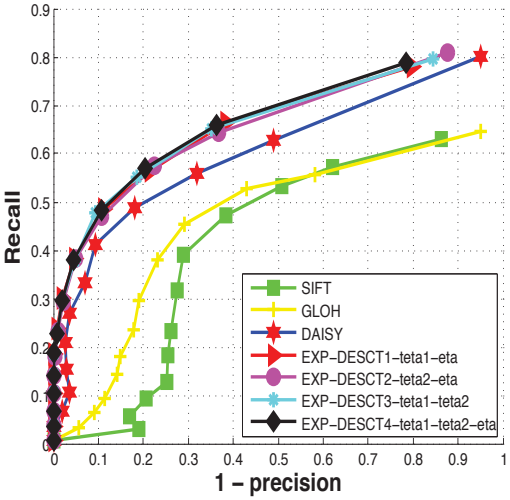
(d) BIKE 1-5 (AHGDK)



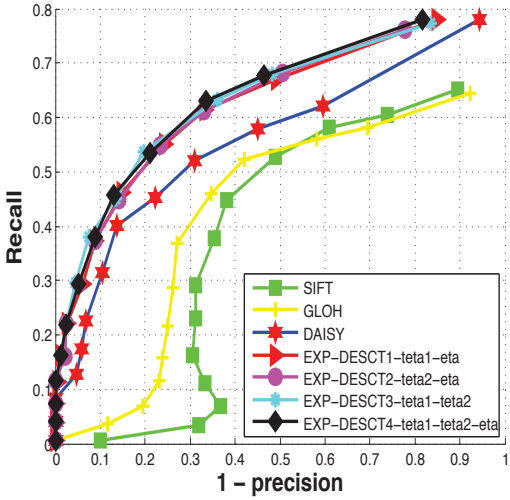
(e) BIKE 1-2 (AHEDK)



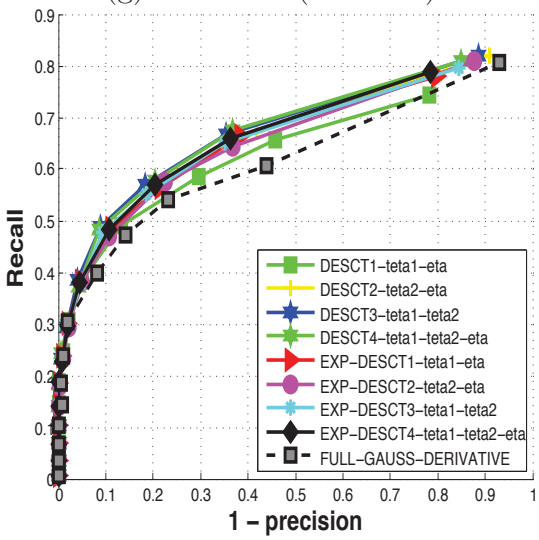
(f) BIKE 1-3 (AHEDK)



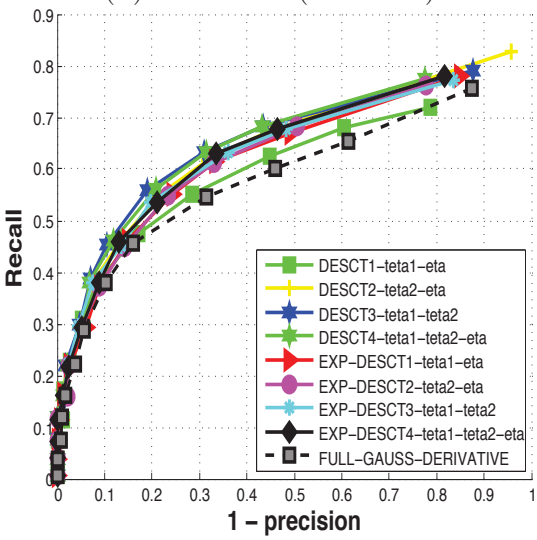
(g) BIKE 1-4 (AHEDK)



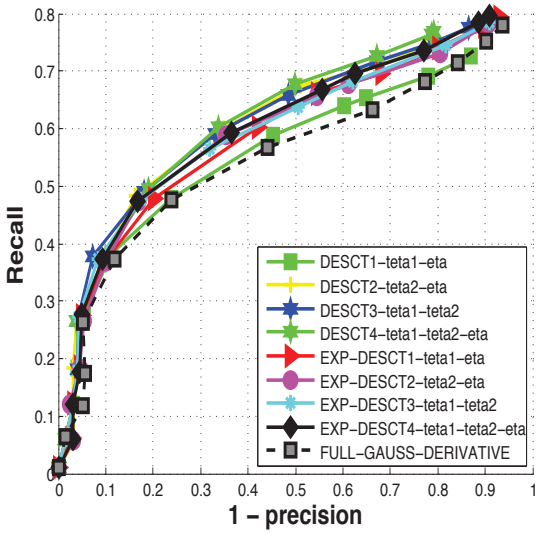
(h) BIKE 1-5 (AHEDK)



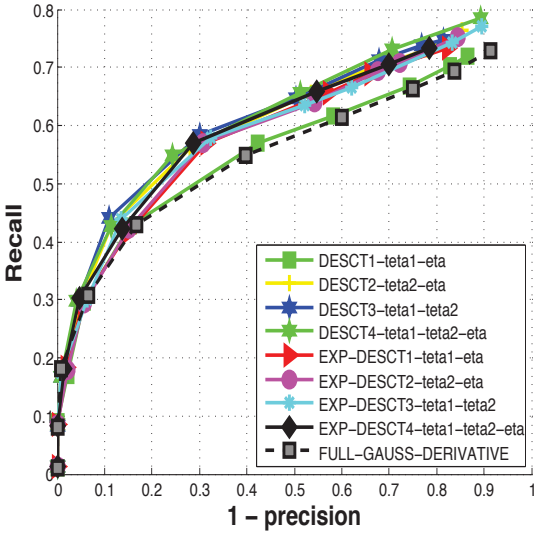
(i) BIKE 1-2 (AHGDK vs AHEDK)



(j) BIKE 1-3 (AHGDK vs AHEDK)

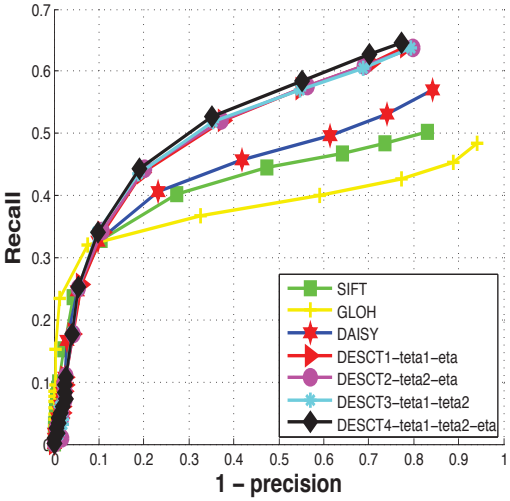


(k) BIKE 1-4 (AHGDK vs AHEDK)

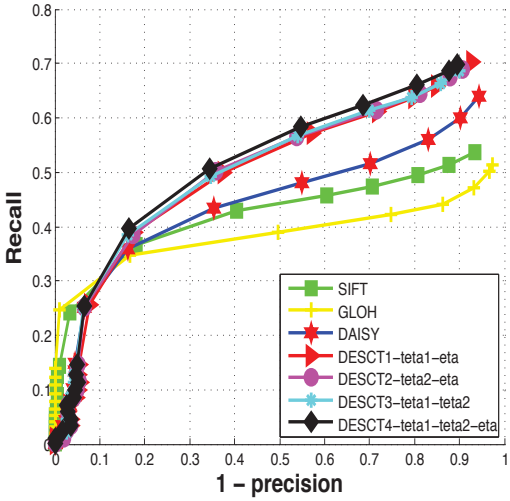


(l) BIKE 1-5 (AHGDK vs AHEDK)

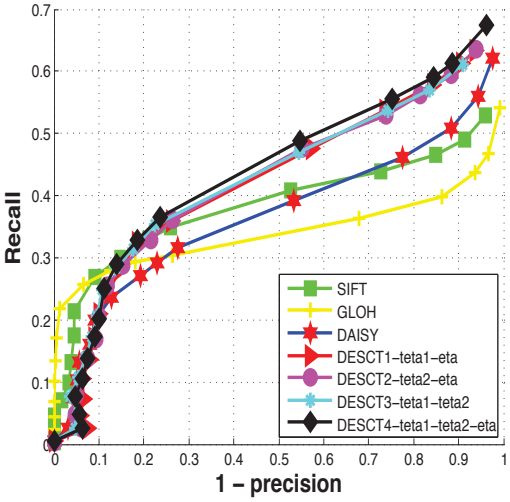
Figure 4.10: BIKE DATASET with variations in BLUR



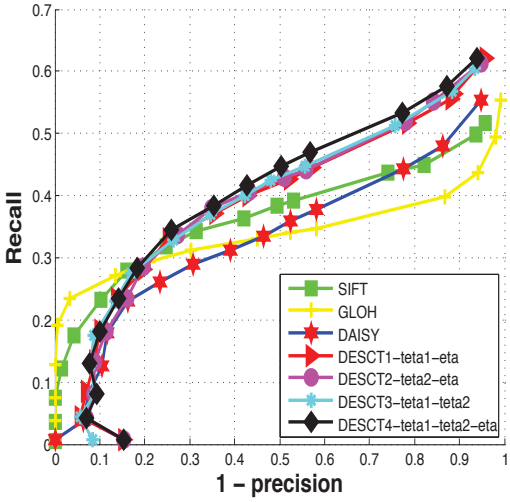
(a) BOAT 1-2 (AHGDK)



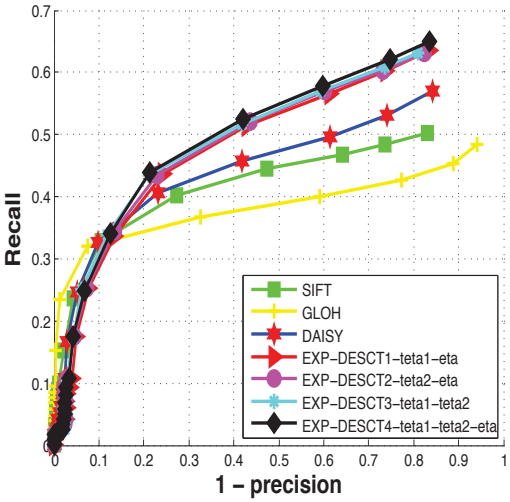
(b) BOAT 1-3 (AHGDK)



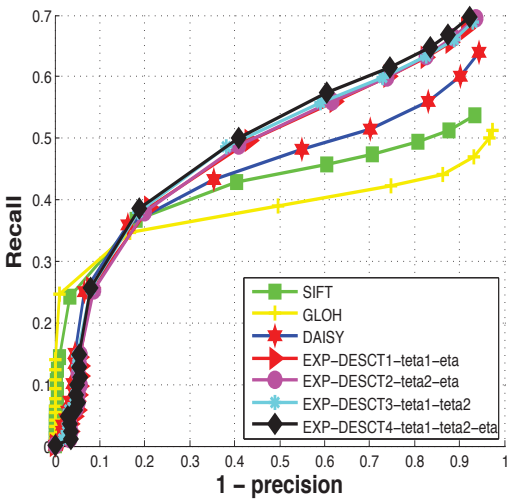
(c) BOAT 1-4 (AHGDK)



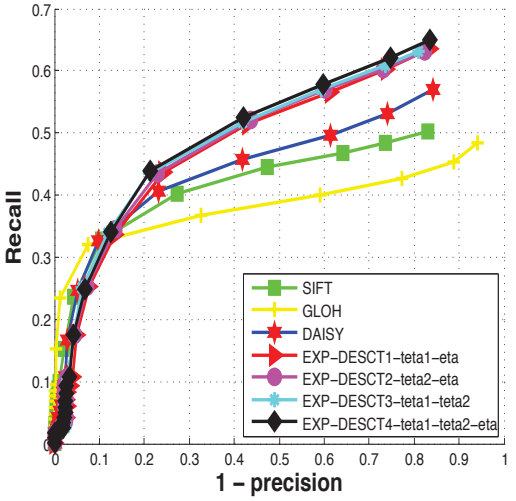
(d) BOAT 1-5 (AHGDK)



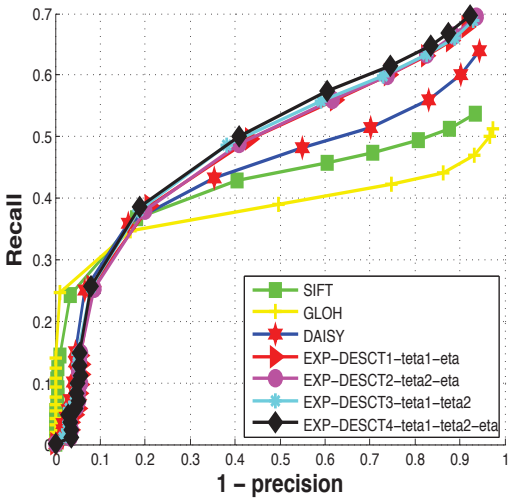
(e) BOAT 1-2 (AHEDK)



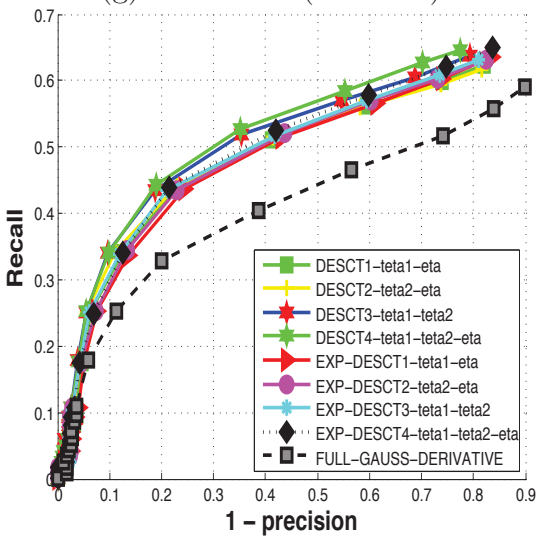
(f) BOAT 1-3 (AHEDK)



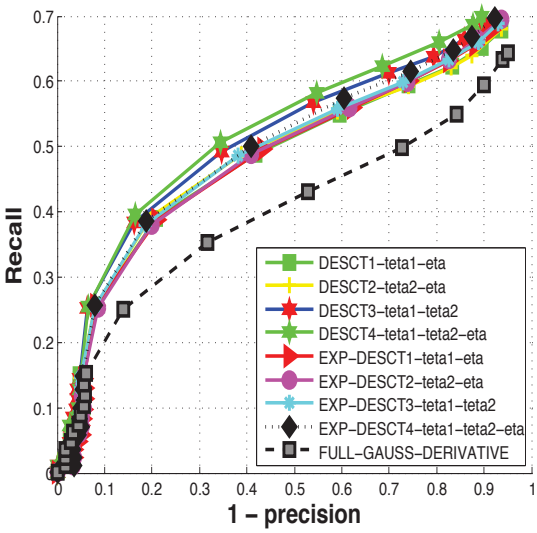
(g) BOAT 1-4 (AHEDK)



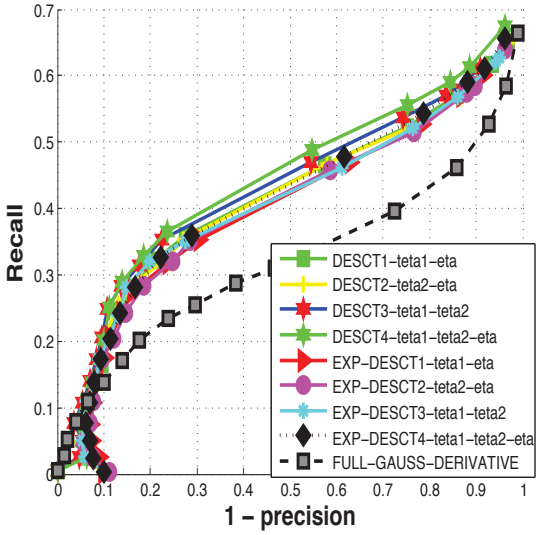
(h) BOAT 1-5 (AHEDK)



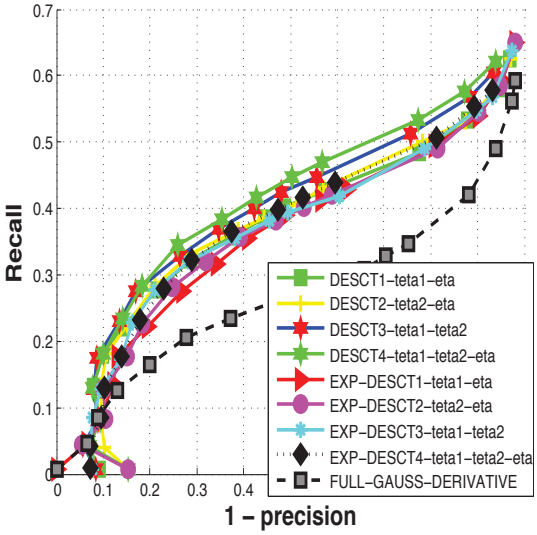
(i) BOAT 1-2 (AHGDK vs AHEDK)



(j) BOAT 1-3 (AHGDK vs AHEDK)

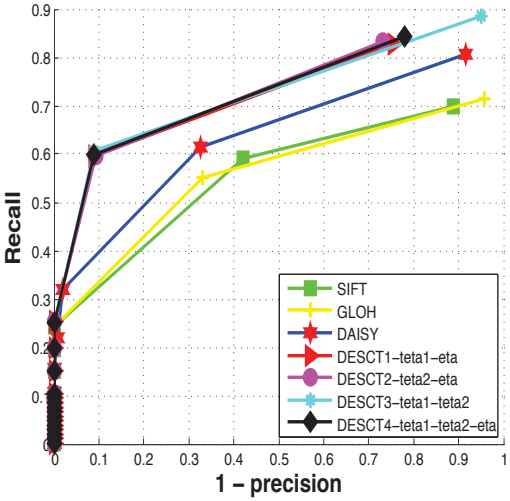


(k) BOAT 1-4 (AHGDK vs AHEDK)

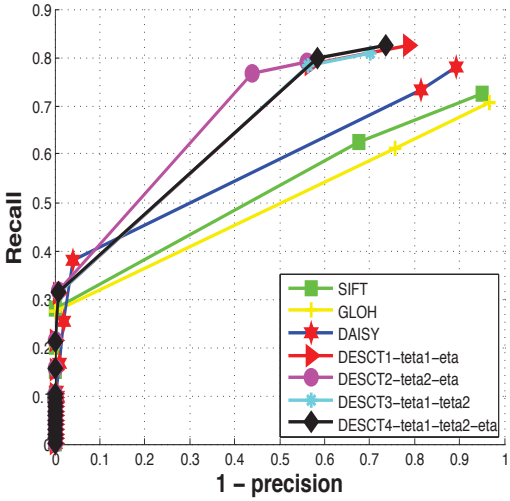


(l) BOAT 1-5 (AHGDK vs AHEDK)

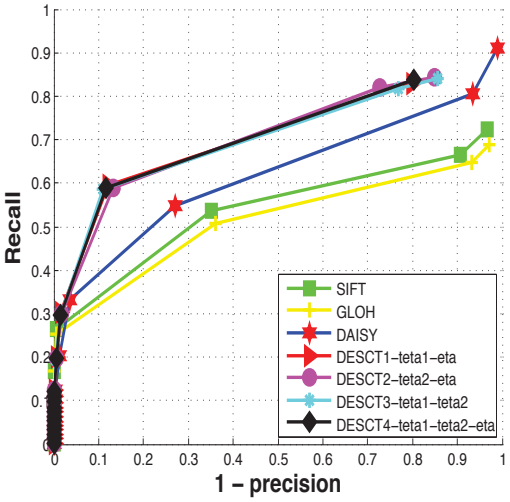
Figure 4.11: BOAT DATASET with changes in rotation and zoom



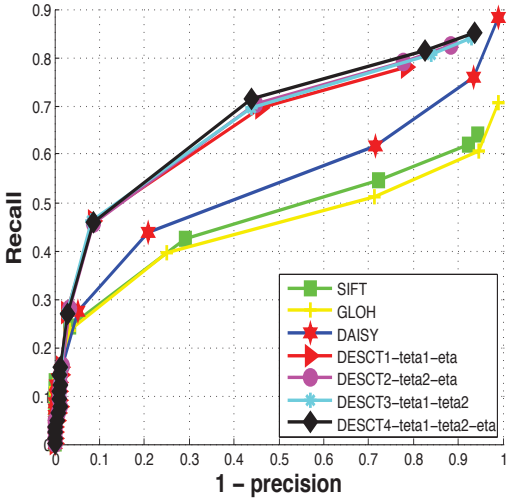
(a) COMP 1-2 (AHGDK)



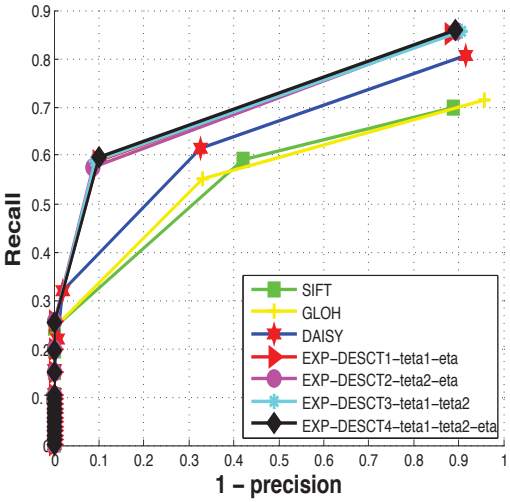
(b) COMP 1-3 (AHGDK)



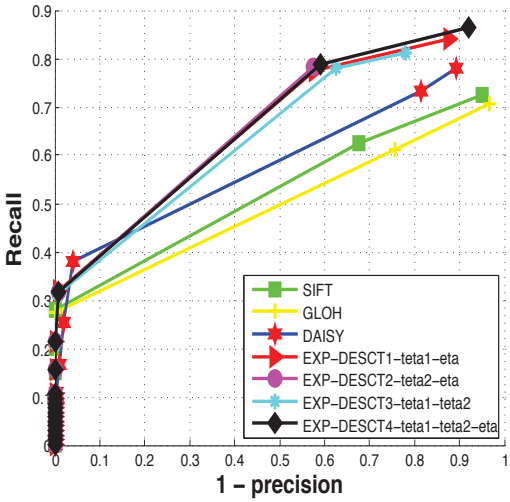
(c) COMP 1-4 (AHGDK)



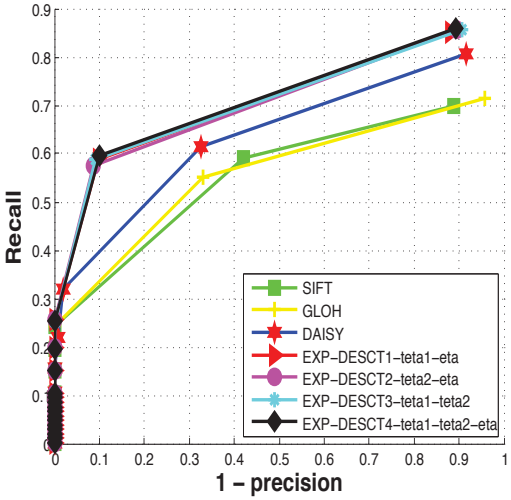
(d) COMP 1-5 (AHGDK)



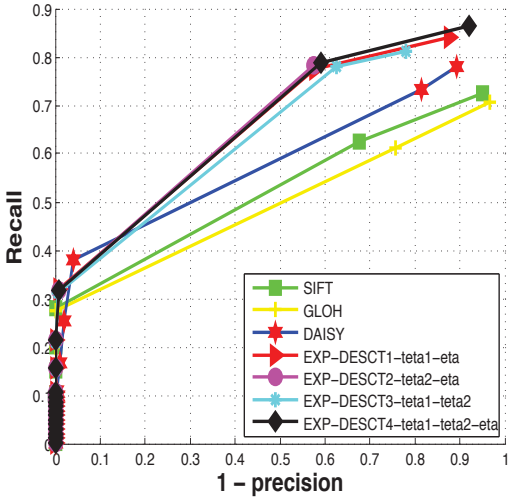
(e) COMP 1-2 (AHEDK)



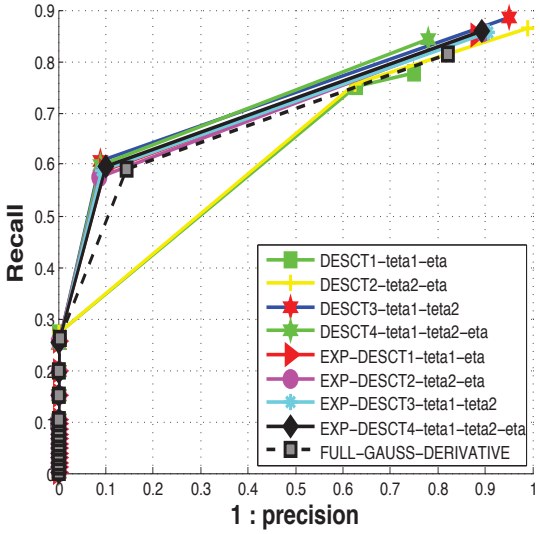
(f) COMP 1-3 (AHEDK)



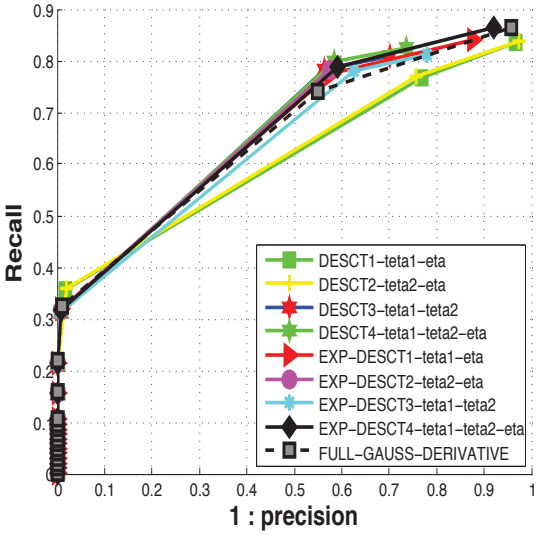
(g) COMP 1-4 (AHEDK)



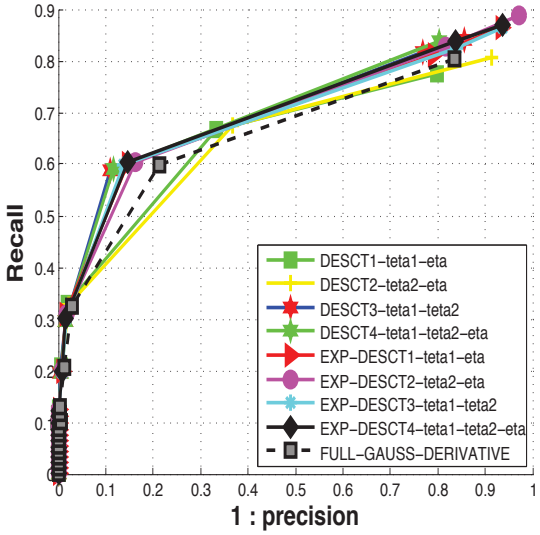
(h) COMP 1-5 (AHEDK)



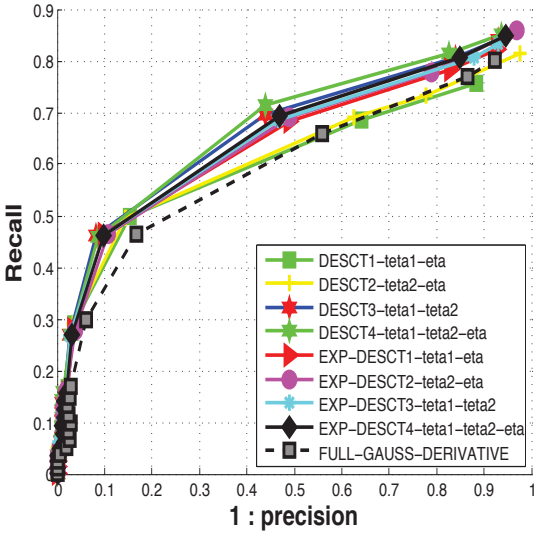
(i) COMP 1-2 (AHGDK vs AHEDK)



(j) COMP 1-3 (AHGDK vs AHEDK)

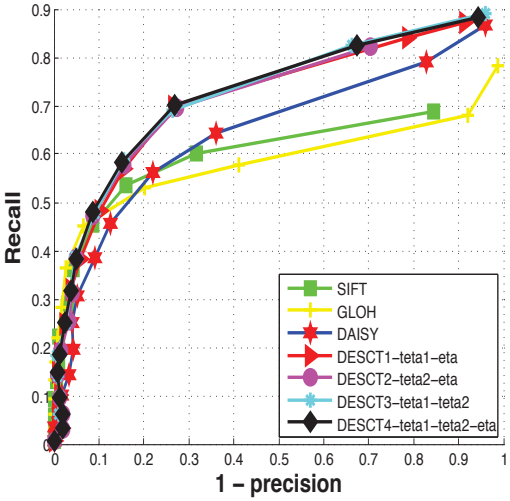


(k) COMP 1-4 (AHGDK vs AHEDK)

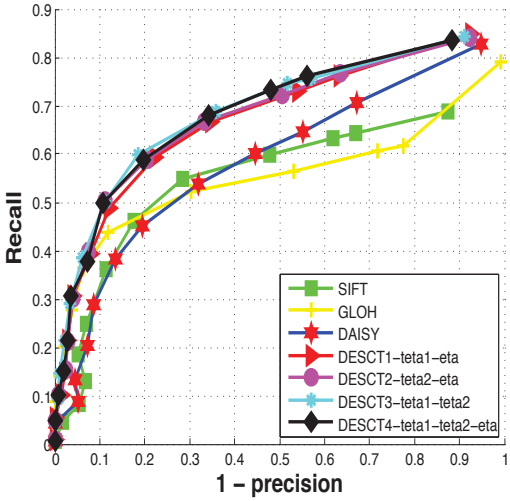


(l) COMP 1-5 (AHGDK vs AHEDK)

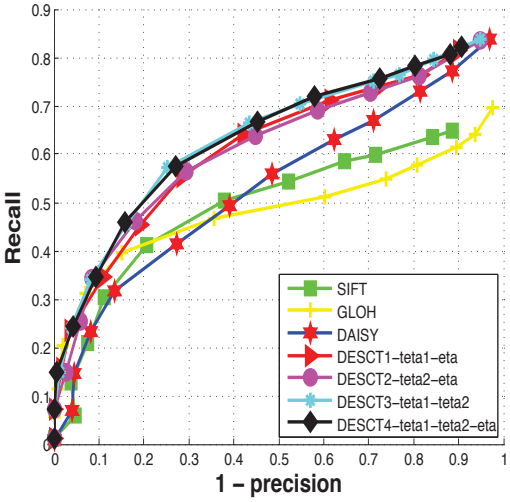
Figure 4.12: COMPRESSION DATASET with variations in compression



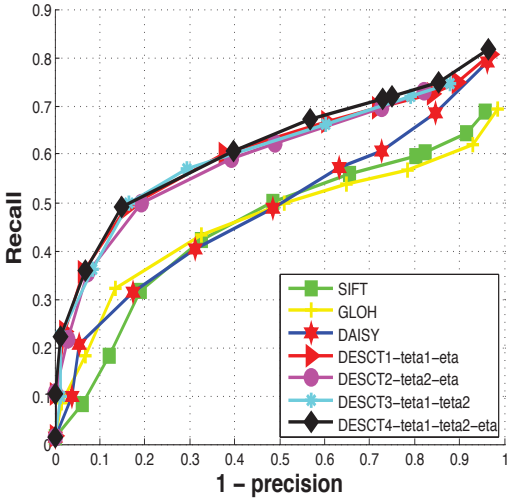
(a) LUVEN 1-2 (AHGDK)



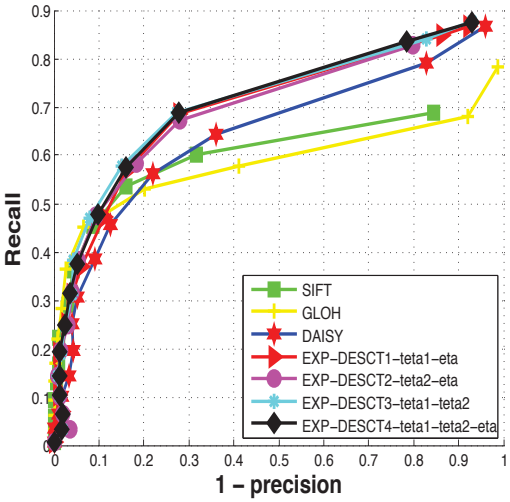
(b) LUVEN 1-3 (AHGDK)



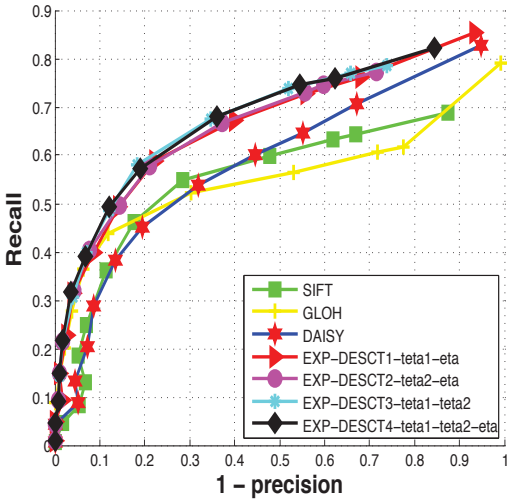
(c) LUVEN 1-4 (AHGDK)



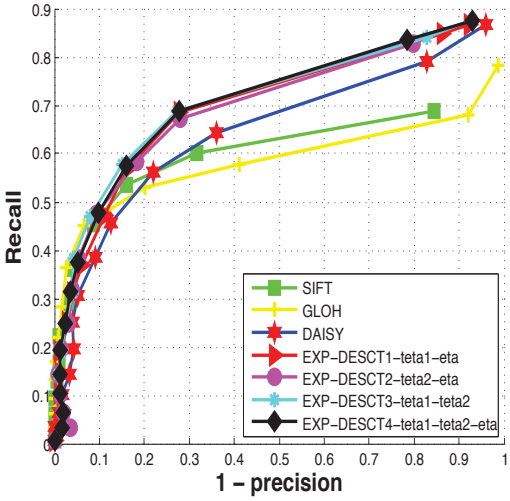
(d) LUVEN 1-5 (AHGDK)



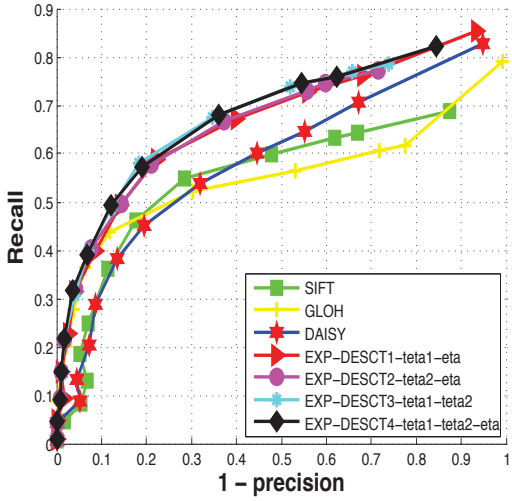
(e) LUVEN 1-2 (AHEDK)



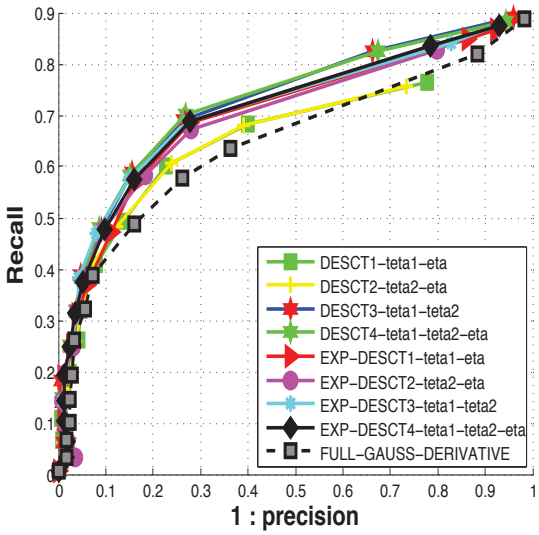
(f) LUVEN 1-3 (AHEDK)



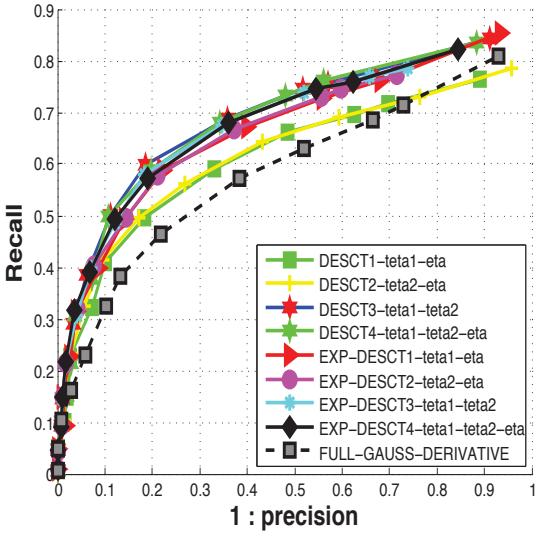
(g) LUVEN 1-4 (AHEDK)



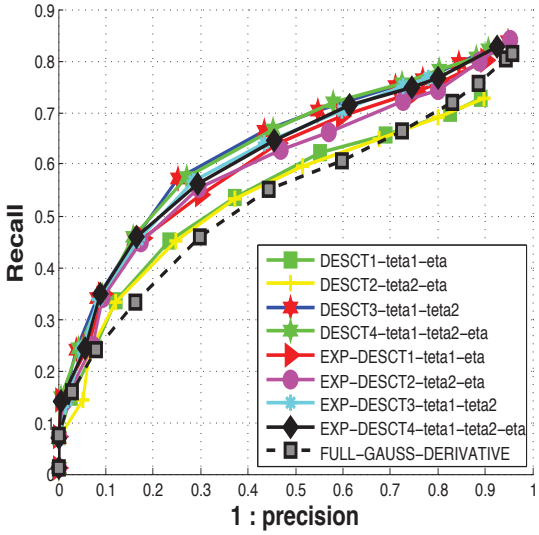
(h) LUVEN 1-5 (AHEDK)



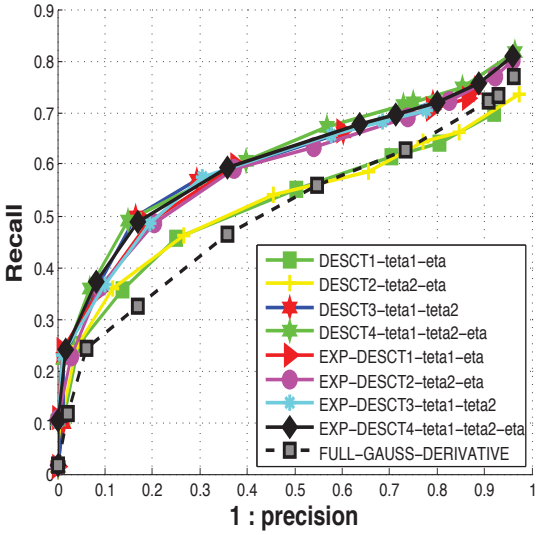
(i) LUVEN 1-2 (AHGDK vs AHEDK)



(j) LUVEN 1-3 (AHGDK vs AHEDK)

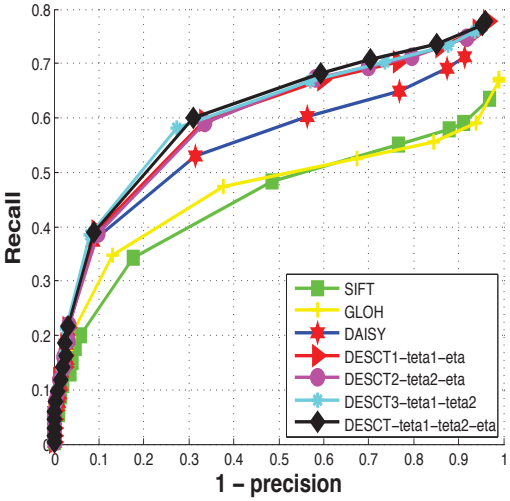


(k) LUVEN 1-4 (AHGDK vs AHEDK)

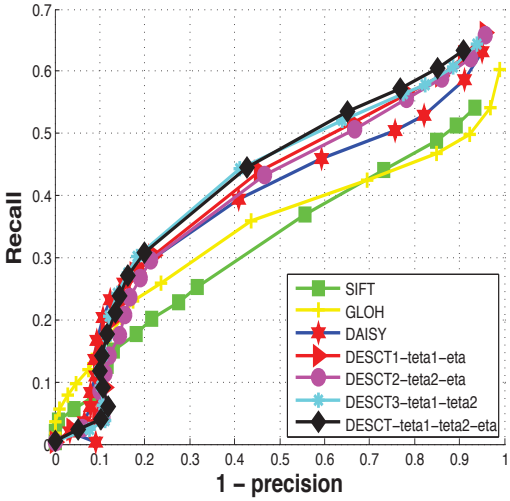


(l) LUVEN 1-5 (AHGDK vs AHEDK)

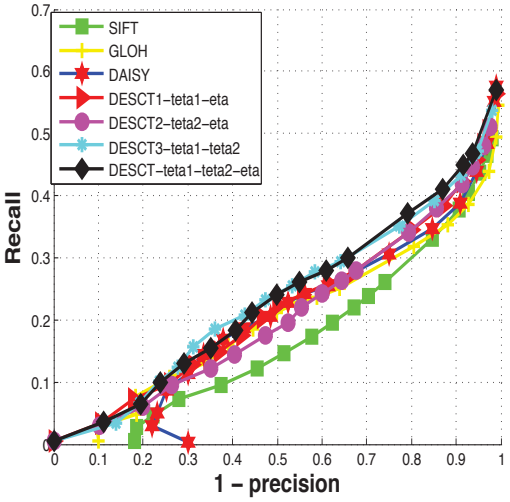
Figure 4.13: LEUVEN DATASET with variations in Birghtness



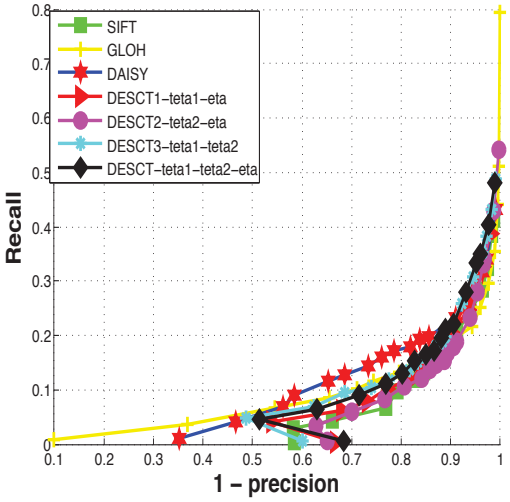
(a) GRAFF 1-2 (AHGDK)



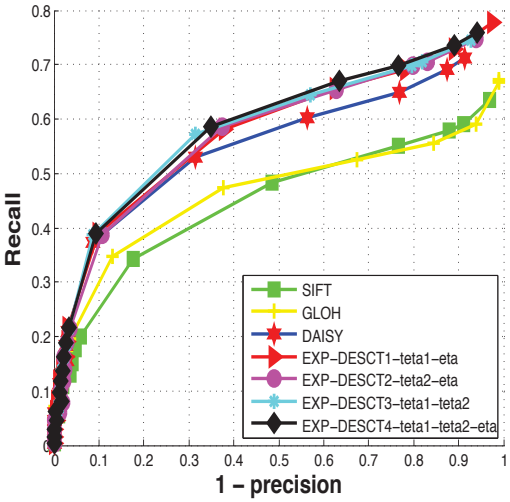
(b) GRAFF 1-3 (AHGDK)



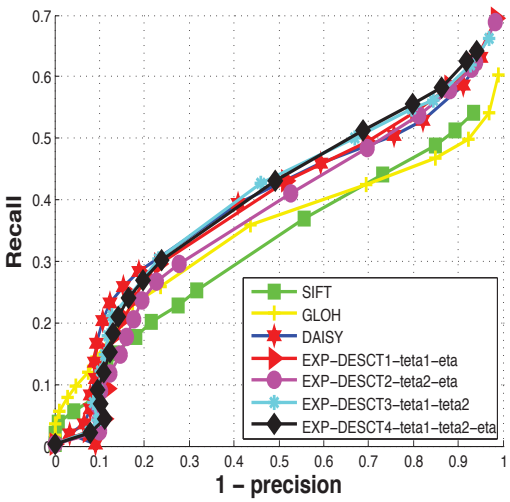
(c) GRAFF 1-4 (AHGDK)



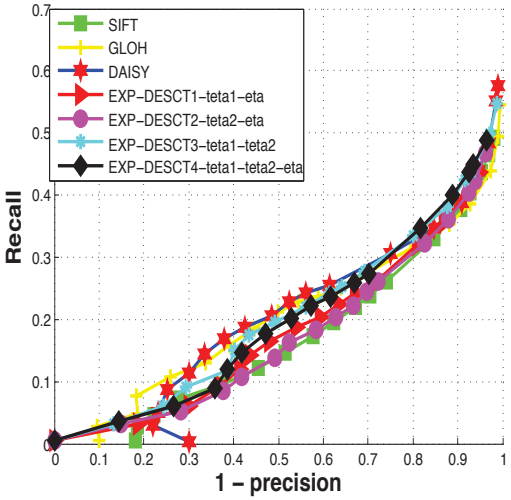
(d) GRAFF 1-5 (AHGDK)



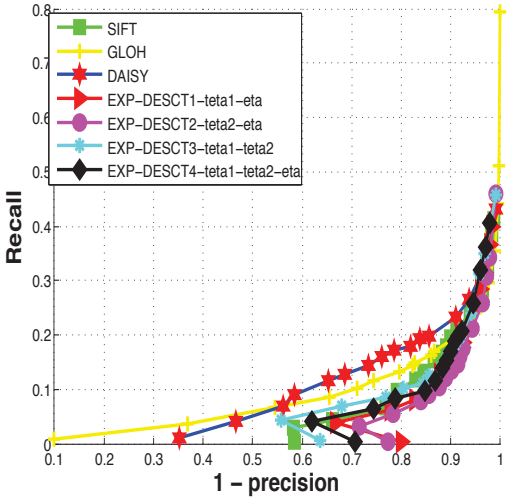
(e) GRAFF 1-2 (AHEDK)



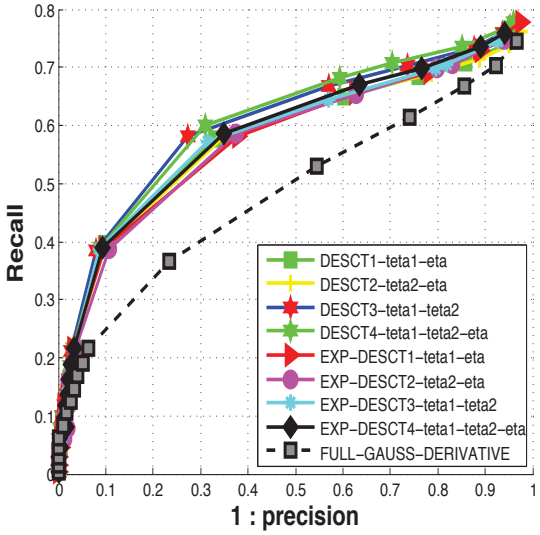
(f) GRAFF 1-3 (AHEDK)



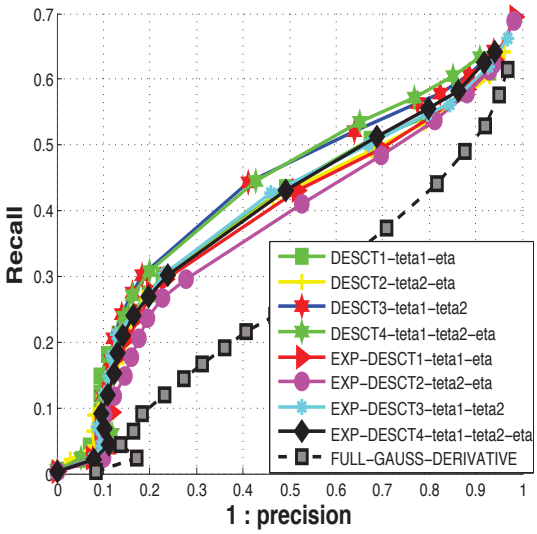
(g) GRAFF 1-4 (AHEDK)



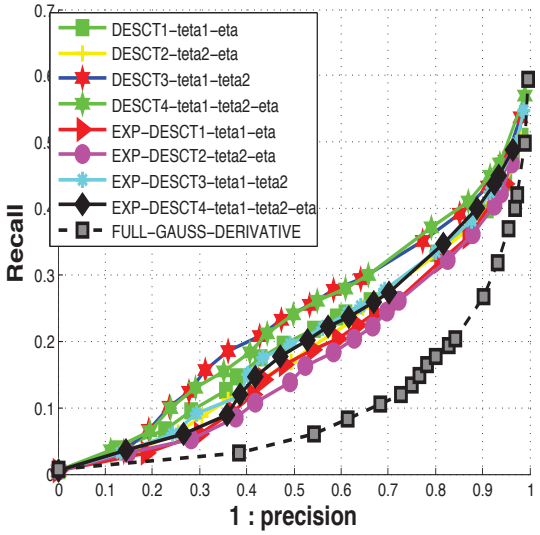
(h) GRAFF 1-5 (AHEDK)



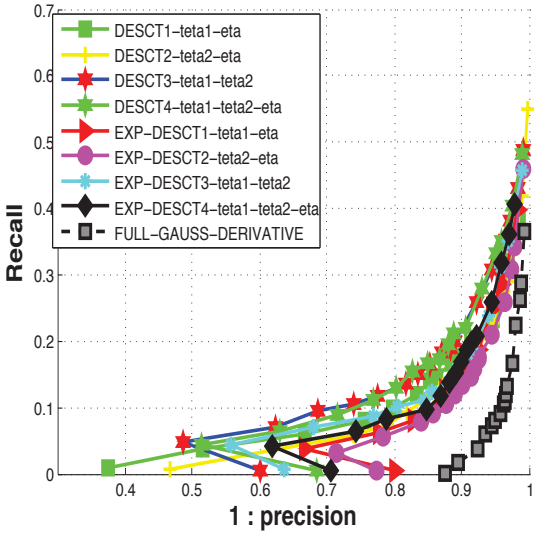
(i) GRAFF 1-2 (AHGDK vs AHEDK)



(j) GRAFF 1-3 (AHGDK vs AHEDK)

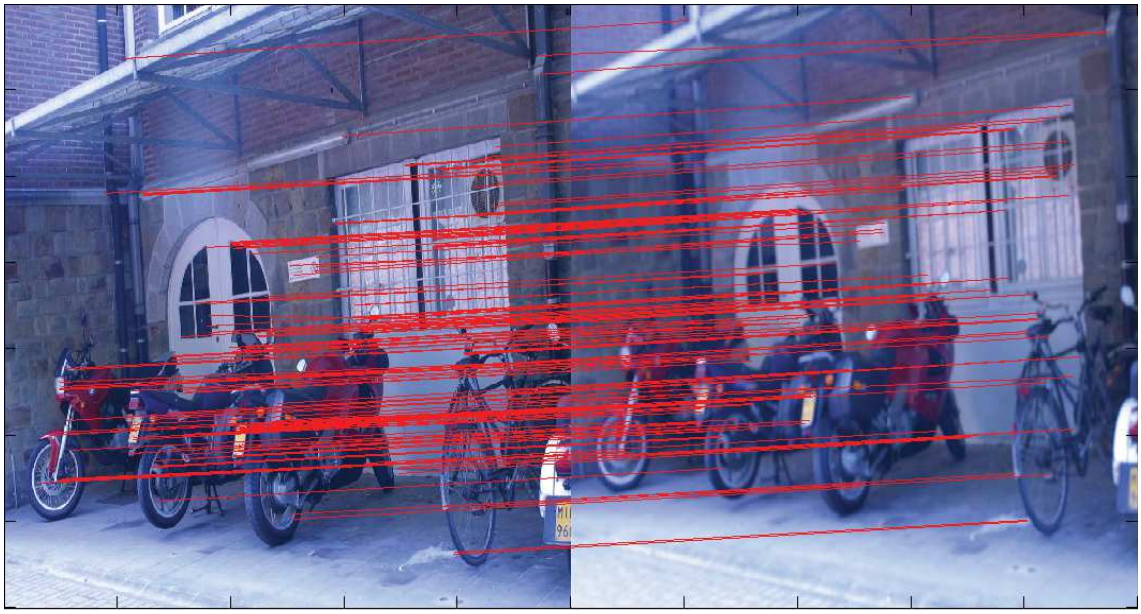


(k) GRAFF 1-4 (AHGDK vs AHEDK)

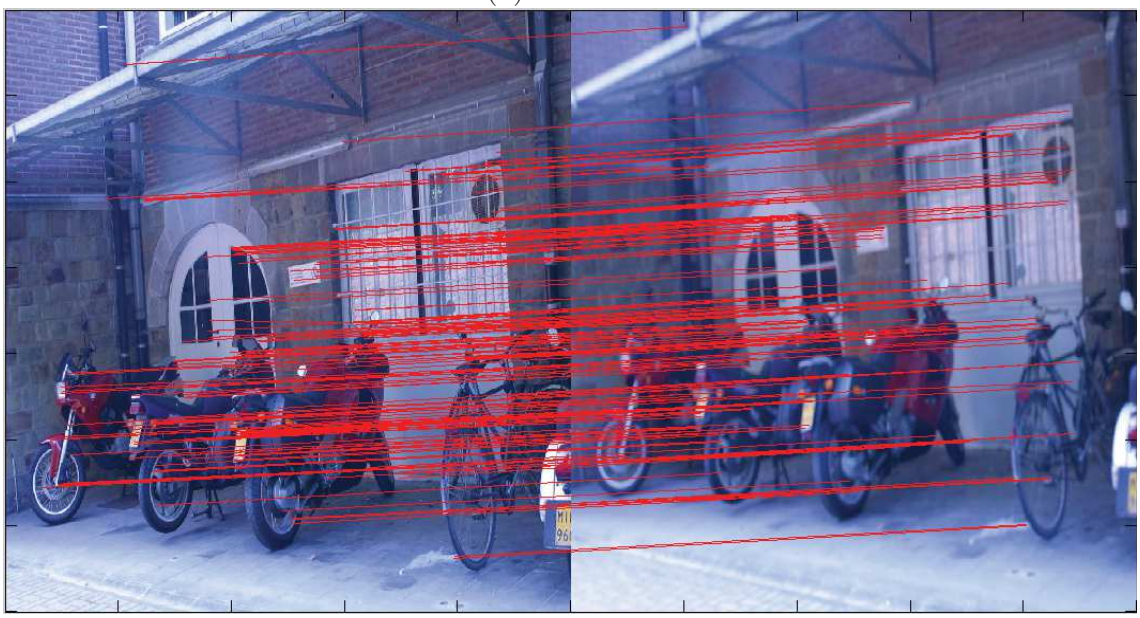


(l) GRAFF 1-5 (AHGDK vs AHEDK)

Figure 4.14: GRAFF DATASET with variations in Birghtness

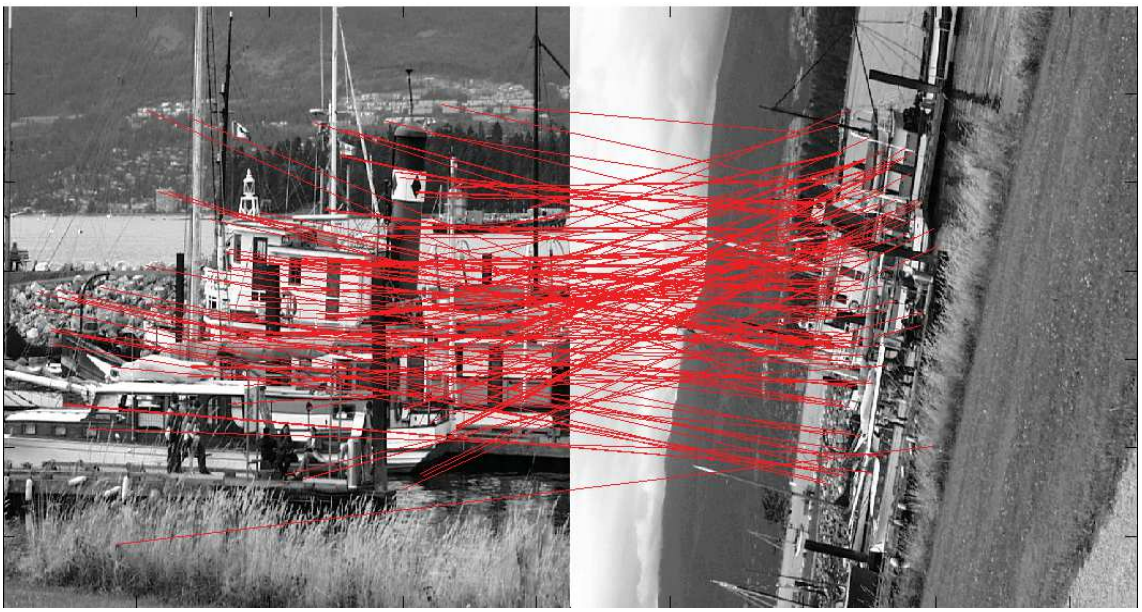


(a) AHGDK



(b) AHEDK

Figure 4.15: Qualitative results using nearest neighbour matching approach. (a) Matching between BIKE1 and BIKE4 using AHGDK (b) Matching between BIKE1 and BIKE4 using AHEDK.



(a) AHGDK

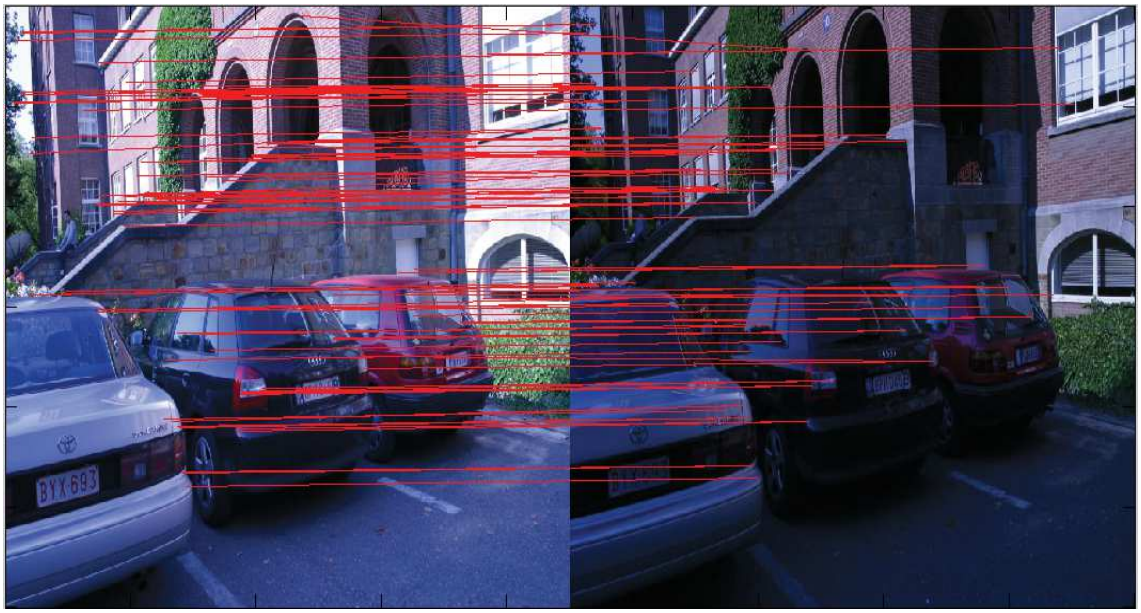
(b) AHEDK

Figure 4.16: Qualitative results using nearest neighbour matching approach. (a) Matching between BOAT1 and BOAT4 using AHGDK (b) Matching between BOAT1 and BOAT4 using AHEDK.

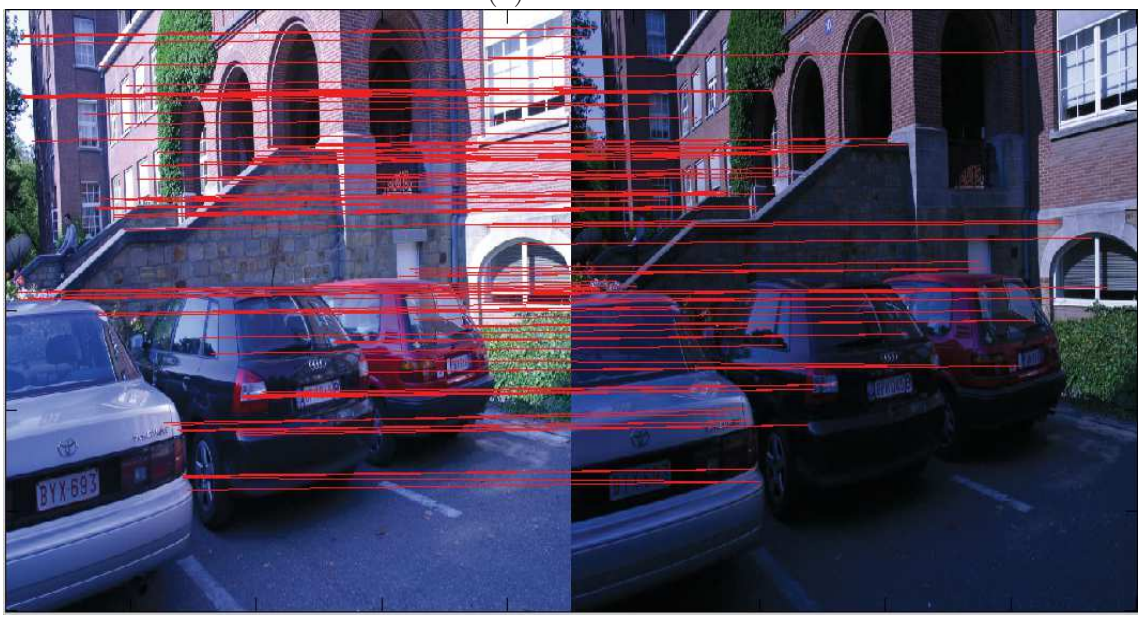
(a) AHGDK

(b) AHEDK

Figure 4.17: Qualitative results using nearest neighbour matching approach. (a) Matching between compression1 and compression4 using AHGDK. (b) Matching between compression1 and compression4 using AHEDK.



(a) AHGDK



(b) AHEDK

Figure 4.18: Qualitative results using nearest neighbour matching approach. (a) Matching between luven1 and luven4 using AHGDK. (b) Matching between luven1 and luven4 using AHEDK.

(a) AHGDK

(b) AHEDK

Figure 4.19: Qualitative results using nearest neighbour matching approach. (a) Matching between GRAFF1 and GRAFF4 using AHGDK. (b) Matching between GRAFF1 and GRAFF4 using AHEDK.

Chapter 5

RSD-DoG

In the previous chapter we proposed a descriptor called RSD-HoG. In RSD-HoG, we bin the orientations at which the edges occur and the angles of the anisotropic gradient to form an image descriptor. Orientation of the edges and anisotropic gradient orientation are obtained using first order gradient information. Computer vision literature shows that second order image statistics provide valuable information about the image. In this chapter, we propose another new descriptor based on second order image derivatives called RSD-DOG. Here, we treat the image patch as a 3D surface, with intensity being the 3rd dimension. The considered 3D surface has a rich set of second order features/statistics such as ridges, valleys, cliffs and so on. These second order statistics can be easily captured using the difference of rotating semi Gaussian filters. The originality of our method lies in successfully combining the response of the directional filters with that of the Difference of Gaussian (DOG) approach.

5.1 Introduction

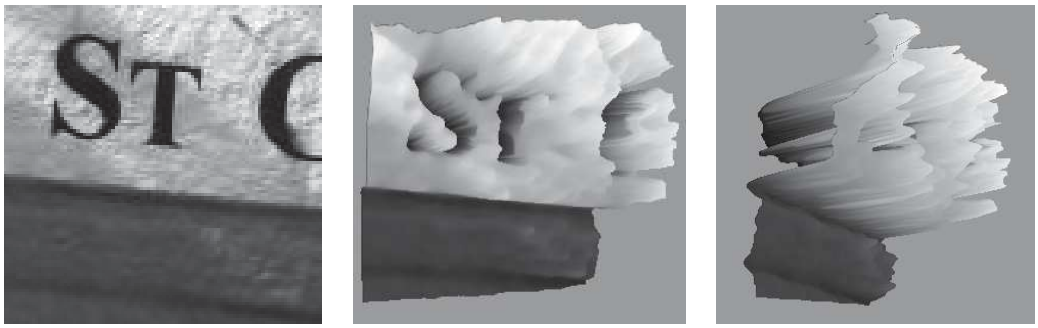
In the computer vision literature, features related to first order image statistics such as segments, edges, image gradients and corners have been used in abundance for image matching and object detection. In one dimension, first order gradient extracted at a point gives the slope of the curve at that point. In case of an image, first order gradient at a pixel measures the slope of the luminance profile at that pixel, while, second order derivatives are abundantly present in many key-point detectors based on Hessian and laplacian. But, only first order gradient information is included in the state of the art local image descriptors such as SIFT [Low04], GLOH [MS03], DAISY [TLF10], and LBP [OPM02b]. Whereas, features related to second order statistics such as cliff, ridges, summits, valleys and so on have been sparsely used for the image matching and object recognition purpose. In one dimension, second order derivative at a point measures the local curvature at that point i.e. how much the curve bends at a given point. This part of the thesis mainly concentrates on the use of second order statistics for the task of image matching.

Digital images are represented in 2 dimensions as $I = D(x, y)$, where I is the intensity at the pixel location (x, y) in the digital image D. Alternatively, an image can also be a 3D surface where every (x, y) pair has a third coordinate in the z plane. Here, the intensity values are used to represent z axis. This representation has a surface that has three coordinates for every point. This type of mapping from an open set $\Re^2$ into $\Re^3$, is

known as a Monge patch and can be written in the parametric form as:

$$s(x, y) = (x, y, D(x, y)) \tag{5.1}$$

where, $D(x, y)$ is the original Digital image and $s(x, y)$ is a 2-D surface in $\Re^3$. Parametrizing an image as a 3D surface allows for the creation of a new feature space using differential geometry. Example of a digital image represented as a 3D image surface is shown in Fig.5.1. Such a surface in 3D, consists of features such as ridges, valleys, summits or basins and so on. The geometric properties of these features can be accurately characterized by local curvatures of differential geometry through second order statistics. The motivation behind this work is to extract these 2nd order image statistics and represent them as a compact image descriptor for image matching.



(a) (b) (c)

Figure 5.1: (a) Image representation in 2D. (b) and (c) The same image being represented in 3D, with intensity being the third dimension. Both (b) and (c) are viewed from different angles. We can see that the 3D image is made of many second order statistics such as ridges, valleys, summits, edges etc.

Here, for the generation of the RSD-DoG:

- Our idea is to consider the 2D image patch as 3D surface made of second order statistics such as ridges, valleys, summits and so on. Then, we extract these second order statistics by using a local directional maximization or minimization of the response of difference of two rotating half smoothing filters.

- These directions correspond to the orientation of ridges, valleys or a junction of ridges/valleys. The orientations at which these second order statistics occur are binned to form a local image descriptor RSD-DOG of dimension/length 256. By construction, the dimension of our descriptor is almost 3 to 4 times less when compared to other descriptors [ZK13, OT01] based on second order statistics.

- This descriptor is evaluated for invariance to blur, rotation, compression, scale and viewpoint changes. By construction, our descriptor shows enormous robustness to variations in illumination. To highlight this property, we rigorously evaluate our descriptor on dataset consisting of images with linear and non-linear illumination changes.

5.2 Directional Filter

Initially, we use the Anisotropic half Gaussian smoothing Filter (AHGSK) as discussed in the chapter.2. Here, we cut the full smoothing filter into half by using the heaviside function instead of the sigmoid function. To recap, the AHGSK is given by:

$$G_{(\mu,\lambda)}(x,y,\theta) = C.H\left(R_\theta \begin{pmatrix} x \\ y \end{pmatrix}\right) e^{-\begin{pmatrix} x & y \end{pmatrix} R_\theta^{-1} \begin{pmatrix} \frac{1}{2\,\mu^2} & 0 \\ 0 & \frac{1}{2\lambda^2} \end{pmatrix} R_\theta \begin{pmatrix} x \\ y \end{pmatrix}} \tag{5.2}$$

where C is a normalization coefficient, R_θ a rotation matrix of angle θ, x and y are pixel coordinates and μ and λ the standard-deviations of the Gaussian filter. By convolving the image patch with AHGSK at different orientations, we obtain a stack of directional smoothed image patches $I_\theta = I * G_{(\mu,\lambda)}(\theta)$. To reduce the computational complexity, in the first step, we rotate the image at some discrete orientations from 0 to 360 degrees (of $\Delta\theta = 1, 2, 5$, or 10 degrees, depending on the angular precision required and the smoothing parameters) before applying non rotated smoothing filter. As the image is rotated instead of the filters, the filtering implementation can use efficient recursive approximation of the Gaussian filter. As presented in [MM10], the implementation of the method is clear and direct. In the second step, we apply an inverse rotation of the smoothed image and obtain a bank of $360/\Delta\theta$ images.

At every pixel in the image patch, we are required to estimate a smoothed second order derivative of the image along a curve crossing these pixels. In one dimension, the second order derivative of a signal can be easily estimated using a Difference Of Gaussian (DOG) operator. In our method, we directly apply two half Gaussian filters with two different λ and the same μ to obtain the directional derivatives (as in Fig.5.2). Later, we compute the difference of the response of these two filters to obtain the desired smoothed second order derivative information in the ridge/valley directions. We refer to this half Gaussian filter combination as the difference of half smoothing filters(DHSF). More details of DHSF can be found in Chapter2. To recap the DHSF is given by:

$$D(x,y,\theta) = G_{(\mu,\lambda_1)}(x,y,\theta) - G_{(\mu,\lambda_2)}(x,y,\theta) \tag{5.3}$$

μ, λ_1 and λ_2 correspond to the standard-deviations of the Gaussians. At each pixel in the image patch, we are interested in the response of the DHSF at θ_{M_1}, θ_{M_2}, θ_{m_1} and θ_{m_2}. Where, θ_{M_1} and θ_{M_2} are the directions at which the local maxima of the function D occurs. $D(x,y,\theta_{M_1})$ and $D(x,y,\theta_{M_2})$ are the response of DHSF at θ_{M_1} and θ_{M_2}. θ_{m_1} and θ_{m_2} are the directions at which the local minima of the function D occurs. $D(x,y,\theta_{m_1})$ and $D(x,y,\theta_{m_2})$ are the response of DHSF (Fig.5.6) at θ_{m_1} and θ_{m_2}.

Some examples of the signal $D(x,y,\theta)$ obtained by spinning the DHSF around the selected key-points extracted from the synthetic image are shown in Fig.5.3. On a typical valley (point 1 in Fig. 5.3), the pixel signal at the minimum of a valley consists of at least two negative sharp peaks. For ridges (point 7 in Fig. 5.3), the pixel signal at the maximum of a ridge contains at least two positive peaks. These sharp peaks correspond to the two directions of the curve (an entering and leaving path). In case of a junction, the

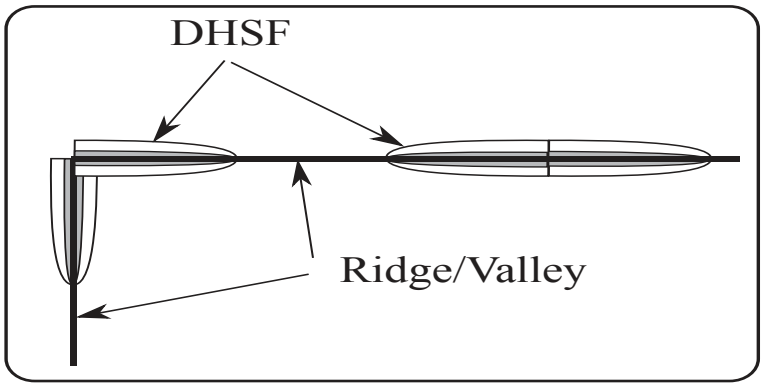


Figure 5.2: DHSF in the ridge/valley directions.

number of peaks corresponds to the number of crest lines (ridges/valleys) in the junction (point 4 in Fig. 5.3). We obtain the same information for the bent lines (illustrated in point 2 on Fig. 5.3). Finally, due to the strong smoothing (parameter μ), D is close to 0 in the presence of noise without any crest line nor edge (illustrated in point 10 in Fig. 5.3). This illustrates the robustness of this method in the presence of noise.

5.3 Methodology

The methodology is shown in Fig.5.4. We use the Harris affine region detector for detecting the interest regions. We follow the same procedure as explained in the Chapter.4 for affine and rotation normalization of the key-region to obtain 41×41 patch.

We consider this image patch as a 3D surface, with intensity being the 3rd dimension. As in Fig.5.4, for each pixel in the image patch, we spin the DHSF and obtain a stack of DOG patches. From this stack of DOG patches, for each pixel we extract the signal $D(x, y, \theta)$ (for simplicity and proper viewing, in Fig.5.4 signal is not shown and a stack of image patch is shown). From each signal we extract the four angles θ_{M_1}, θ_{M_2}, θ_{m_1}, θ_{m_2} and their corresponding responses $||D(x, y, \theta_{M_1})||$, $||D(x, y, \theta_{M_2})||$, $||D(x, y, \theta_{m_1})||$ and $||D(x, y, \theta_{m_2})||$. Once these informations are obtained, for each pixel P, we estimate the average angles η_1 and η_2 and their respective average magnitudes δ_1 and δ_2 by:

$$\begin{cases} \eta_1(x, y) = (\theta_{M_1} + \theta_{M_2})/2 \\ \eta_2(x, y) = (\theta_{m_1} + \theta_{m_2})/2 \\ \delta_1 = (||D(x, y, \theta_{M_1})|| + ||D(x, y, \theta_{M_2})||)/2 \\ \delta_2 = (||D(x, y, \theta_{m_1})|| + ||D(x, y, \theta_{m_2})||)/2 \end{cases}$$

The angle η_1 is weighed by δ_1 and η_2 by δ_2 and binned as in Eq. 5.4. Later, H_{η_1} and H_{η_2} are concatenated to form the final 256 length/dimension RSD-DOG descriptor.

$$\begin{cases} H_{\eta_1} = \{\eta_{1_{bin1}}, \eta_{1_{bin2}}, \eta_{1_{bin3}}, \eta_{1_{bin4}} \ldots\ldots\ldots \eta_{1_{bin128}}\} \\ H_{\eta_2} = \{\eta_{2_{bin1}}, \eta_{2_{bin2}}, \eta_{2_{bin3}}, \eta_{2_{bin4}} \ldots\ldots\ldots \eta_{2_{bin128}}\} \end{cases} \quad (5.4)$$

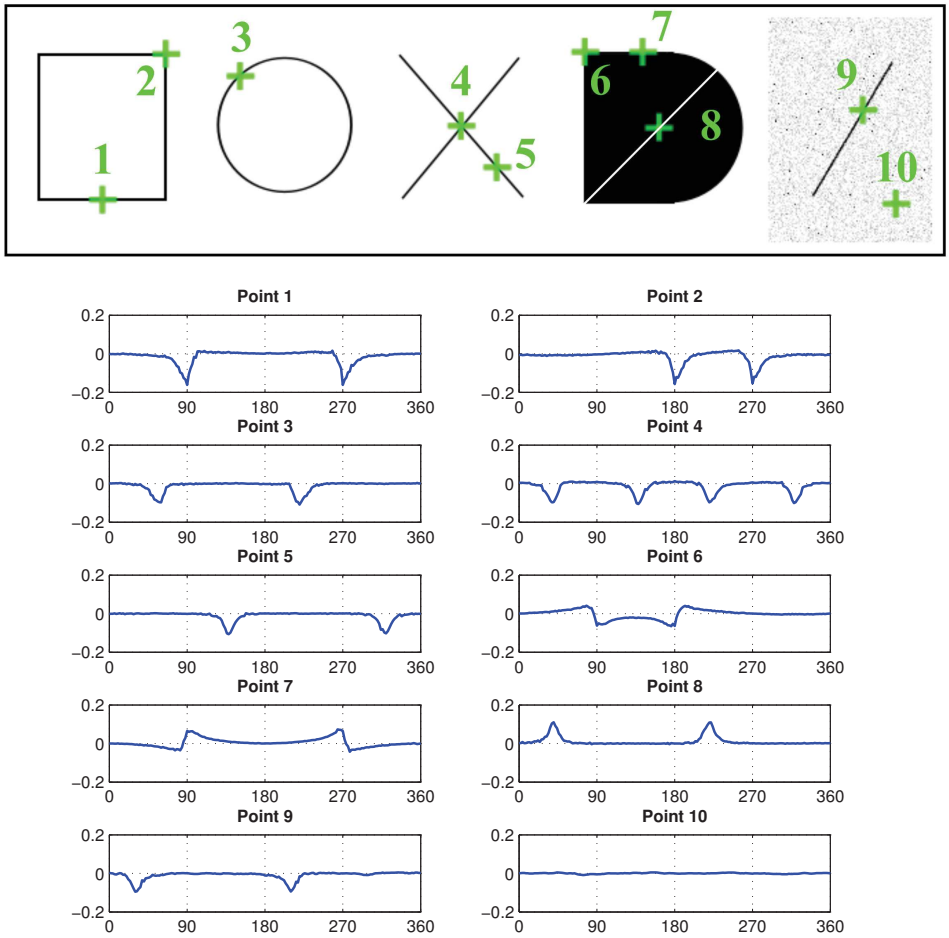


Figure 5.3: Points selection on a synthetic image. Examples of functions $D(x, y, \theta)$ on the points selected on synthetic image using $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$. *The x-axis corresponds to the value of* θ *(in degrees) and the y-axis to* $D(x, y, \theta)$.

5.4 Experiments and results

Matlab platform is used for the experiments. Harris affine key points [MS02] were used for image patch extraction as well as the key points obtained from other detectors can also be used for extracting these image patches. The descriptors used in our experiments are tested with the dataset provided by Mikolajczyk et al.[1]. It is the same dataset used for evaluating our descriptor RSD-HoG. More details about the dataset can be found in Chapter.4.

In order to study in detail the performance of our descriptor for changes in illumination, we also evaluated our descriptor on four image pairs, with complex illumination changes; the data set for the same is publicly available[2]. The complex illumination dataset has 4 set of images, namely 'desktop', 'corridor', 'square' and 'square root'. The first two sets, 'desktop' and 'corridor' have drastic illumination changes, whereas 'square' and 'square root' datasets are obtained by a square and square root operation on the second image

[1] `http://www.robots.ox.ac.uk/~vgg/research/affine/`
[2] http://zhwang.me/publication/liop/

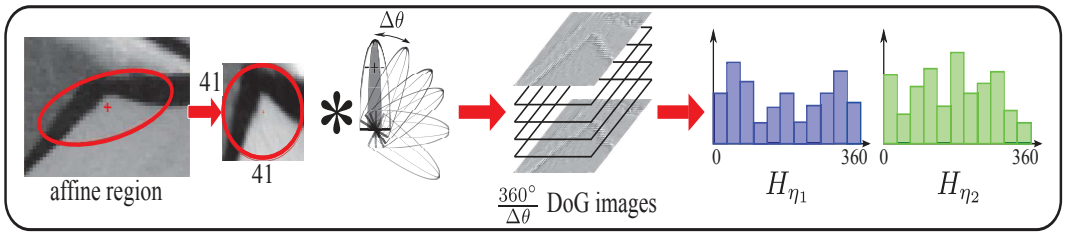
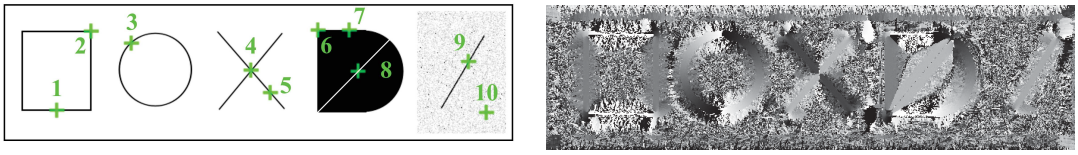
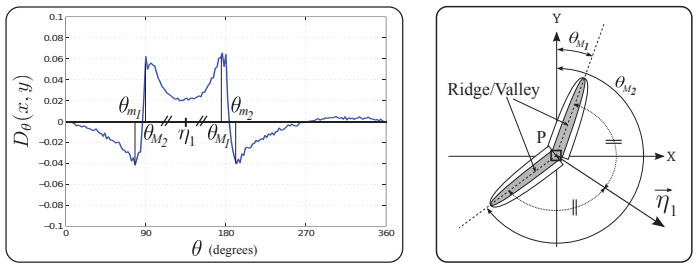


Figure 5.4: Methodology involved in the construction of RSD-DOG descriptor.



(a) Original Image (b) η_1 Image

Figure 5.5: Example of an η_1 Image. The parameters used are $\Delta\theta = 2^\circ$, $\mu = 6, \lambda_1 = 1, \lambda_2 = 1.5$.



(a) (b)

Figure 5.6: (a) η_1 computation from θ_{M_1} and θ_{M_2}. (b) η_1 corresponds to the direction perpendicular to the ridge/valley at the level of a pixel P.

of the 'desktop' set [WFW11]. The images of this illumination dataset can be found in the Fig.5.7. For evaluating the RSD-DoG descriptor, we use the same protocol using the Recall vs 1-Precision curves. The protocol is explained in Chapter.4.

Our descriptor depends on 5 different parameters: $\Delta\theta$, $No-of-bins$, μ, λ_1 and λ_2. The rotation step $\Delta\theta$ is fixed to 10°. Increasing the rotation step results in loss of information. As in [Low04], for histogram construction, the image patch is divided into 16 blocks. All blocks are of the size 10x10 (Since we are using a patch of size 41x41, the blocks in the extreme right and bottom have 11x11 size). As in [Low04], the number of bins ($No-of-bins$) is fixed to 8 per block, resulting in a $8 * 16 = 128$ bins for 16 blocks. Increasing the number of bins results in same performance as in previous case, but it increases the dimensionality of the descriptor. Filter height μ is fixed to 6. As in [Low04], for DHSF the ratio between successive scales is fixed to $\sqrt{2}$. So, filter widths λ_1 and λ_2 are fixed to 2 and $2\sqrt{2}$ respectively. In our experiments, we obtain state of art results by using just two scales. Height ($\mu = 6$) and Width ($\lambda_1 = 2$, $\lambda_2 = 2\sqrt{2}$) parameters are chosen empirically so as to have a ratio sharpness length that is suitable for robust second order feature detection , which generally gives good results in most cases. This ratio is

compatible with the angle filtering step. The parameters are tabulated in Table.5.1.

Table 5.1: Parameters

filter Height (μ)	filter Width (λ_1, λ_2)	Rotation step ($\Delta\theta$)	No of BINS
6	$2, 2\sqrt{2}$	10°	8

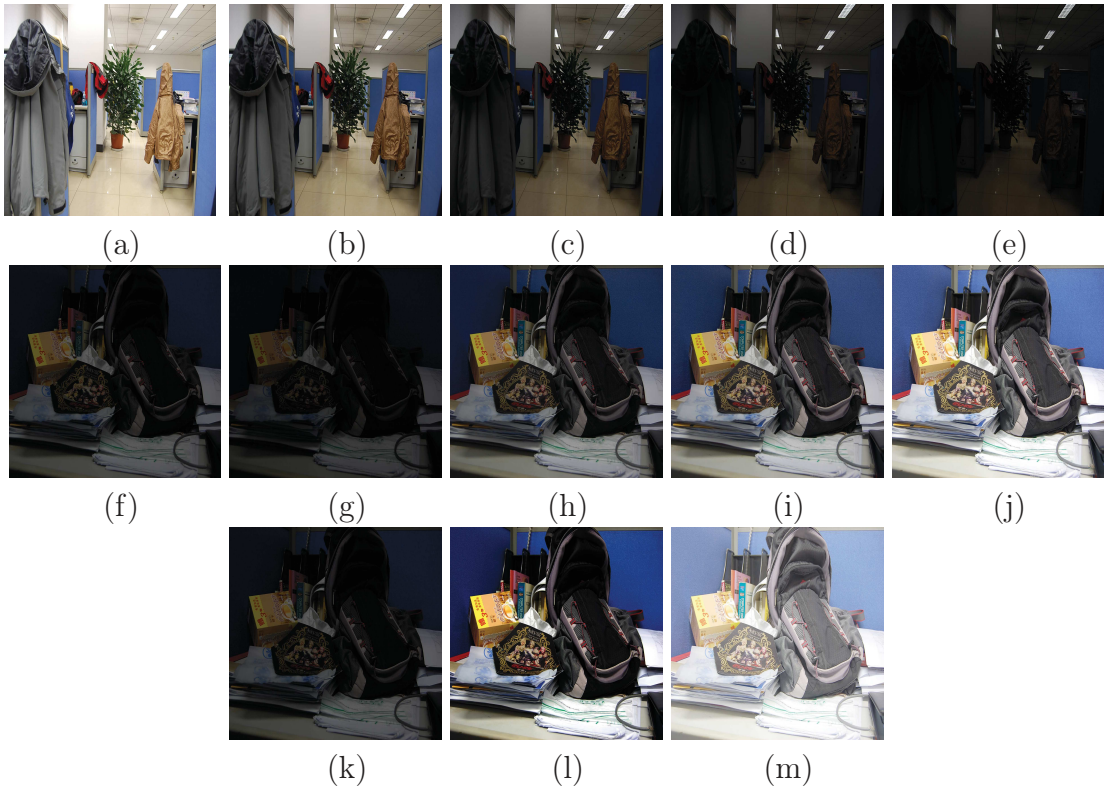


Figure 5.7: Illumination dataset for image matching. The images in the first row is named as '*corridor*' sequence. The images in the 2nd row is called as the '*desktop*'. Both the sequence has drastic variations in illumination. The images in the 3rd row is called '*square*' and '*square root*'.

5.4.1 Descriptor Performance

In our experiments, we have used two variations of our descriptor (1) RSD-DOG (2-SCALES) with height $\mu = 6$ and width $\lambda = 2, 2\sqrt{2}$ respectively. This has a dimension of 256. (2) RSD-DOG (3-SCALES) with height $\mu = 6$ and width $\lambda = 2, 2\sqrt{2}, 4$ respectively. Here, in step one, we smooth the image patch with $\mu = 6$, $\lambda_1 = 2$ and $\lambda_2 = 2\sqrt{2}$ to obtain a 256 length descriptor. In the second step, we smooth the image patch with $\mu = 6$, $\lambda_1 = 2\sqrt{2}$ and $\lambda_2 = 4$ and obtain another 256 length descriptor. Lastly, we concatenate the two parts to form a 512 size RSD-DOG(3-SCALES) descriptor. All our experiments are based on similarity matching and euclidean distance is used as the distance measure.

The performance of these two variants of RSD-DOG is compared with the performance of SIFT, GLOH, DAISY, GIST and LIDRIC descriptors. GIST and LIDRIC descriptors

are based on Gabor filters. Zambanini et al. [ZK13] propose LIDRIC descriptor, based on multi-scale and multi-oriented even Gabor filters. The descriptor is constructed in such a way that typical effects of illumination variations like changes of edge polarity or spatially varying brightness changes at each pixel are taken into account for illumination insensitivity. LIDRIC has a dimension of 768. Oliva et al. [OT01] employ Gabor filters to the grey-scale input image at four different angles and at four spatial scales to obtain the GIST descriptor. The descriptor has a dimension of 512 and is more global. For SIFT and GLOH, the descriptors are extracted from the binaries provided by Oxford group. For DAISY descriptor, the patches are extracted from the code provided by [3]. The matlab code for GIST and LIDRIC descriptors were obtained from [4] and [5] respectively.

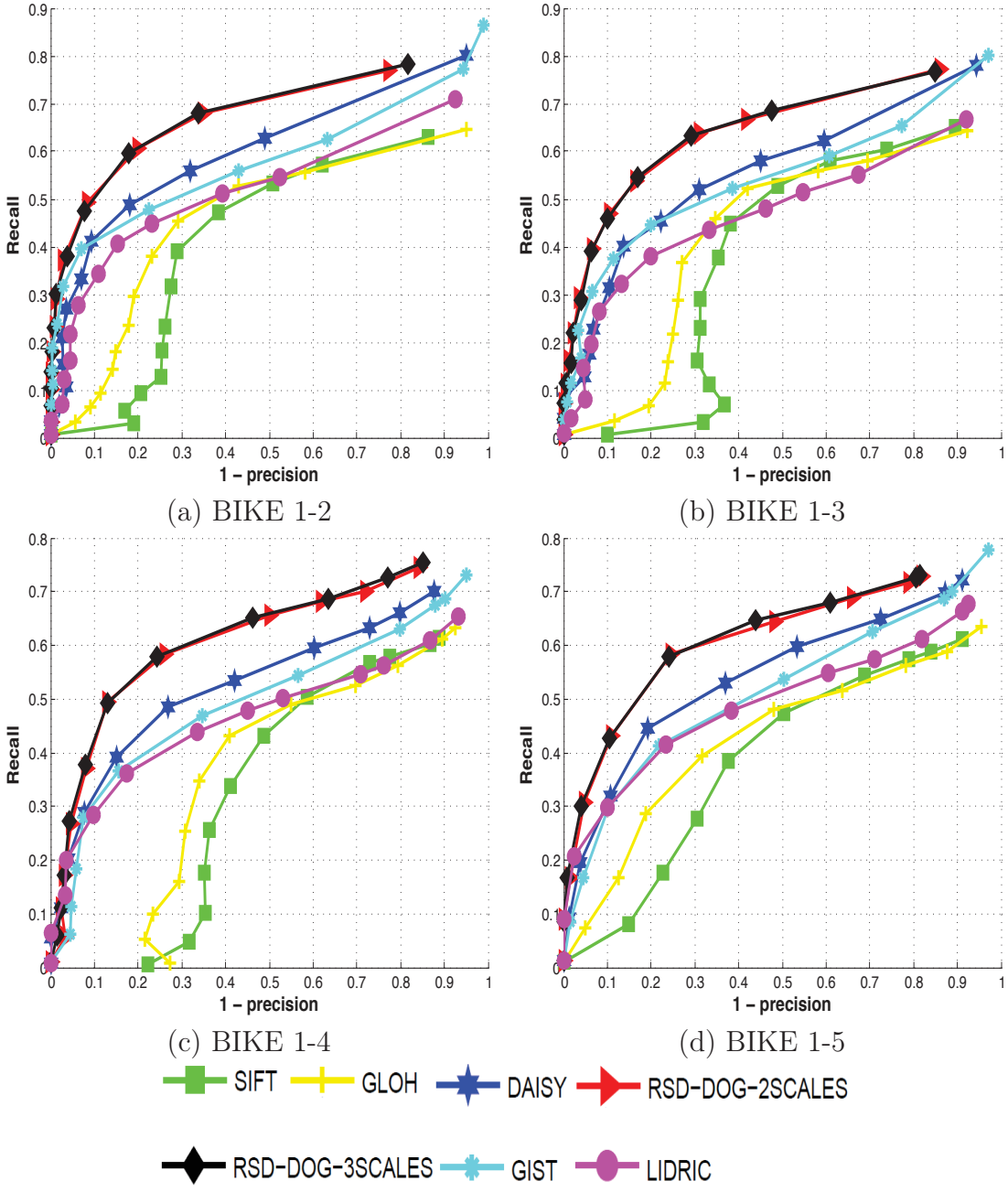
For changes in rotation, viewpoint, blur and compression both variants of the RSD-DOG shows better performance than the other 5 descriptors. The recall vs 1-precision plots in the Fig.5.8, Fig.5.9 and Fig.5.10 illustrates the superiority of our descriptor. In Fig.5.11, image pair graf(1-5) is a complex image pair. As a result, performance of all the descriptors deteriorates. It should be noted that, in most of the cases, RSD-DOG (3-SCALES) performs similar to or slightly better than that of RSD-DOG (2-SCALES). Our hypothesis is that since the region around the key-point is normalized, the effect of scale is reduced. So, increasing the number of scales increases the complexity and descriptor dimension with very little gain in performance.

For variations in illumination, in all cases (Fig.5.12, Fig.5.13 and Fig.5.14) both the variants of RSD-DOG performs consistently better than all the other descriptors. When it comes to 'square' and 'square root' images (Fig.5.15) SIFT, DAISY, LIDRIC and GIST descriptors exhibit poor performance and GLOH descriptor fails miserably.

5.5 Summary

The paper proposes a novel image patch descriptor based on second order statistics such as ridges, valleys, basins and so on. The originality of our method lies in combining the response of directional filter with that of the Difference of Gaussian (DOG) approach. One of the advantage of the proposed descriptor is the dimension/length. Our descriptor has a dimension of 256, which is almost 2 to 4 times less than other descriptors based on second order statistics. Our dataset shows good variations to complex illumination changes. On the standard dataset provided by the Oxford group our descriptor outperforms SIFT, GLOH, DAISY, GIST and LIDRIC. This methodology forms our final contribution and resulted in the publication [VMDM15b].

[3] http://cvlab.epfl.ch/software/daisy
[4] http://people.csail.mit.edu/torralba/code/spatialenvelope/
[5] http://www.caa.tuwien.ac.at/cvl/project/ilac/



(a) BIKE 1-2 (b) BIKE 1-3

(c) BIKE 1-4 (d) BIKE 1-5

Figure 5.8: Recall vs 1-Precision curves for BIKE sequence.

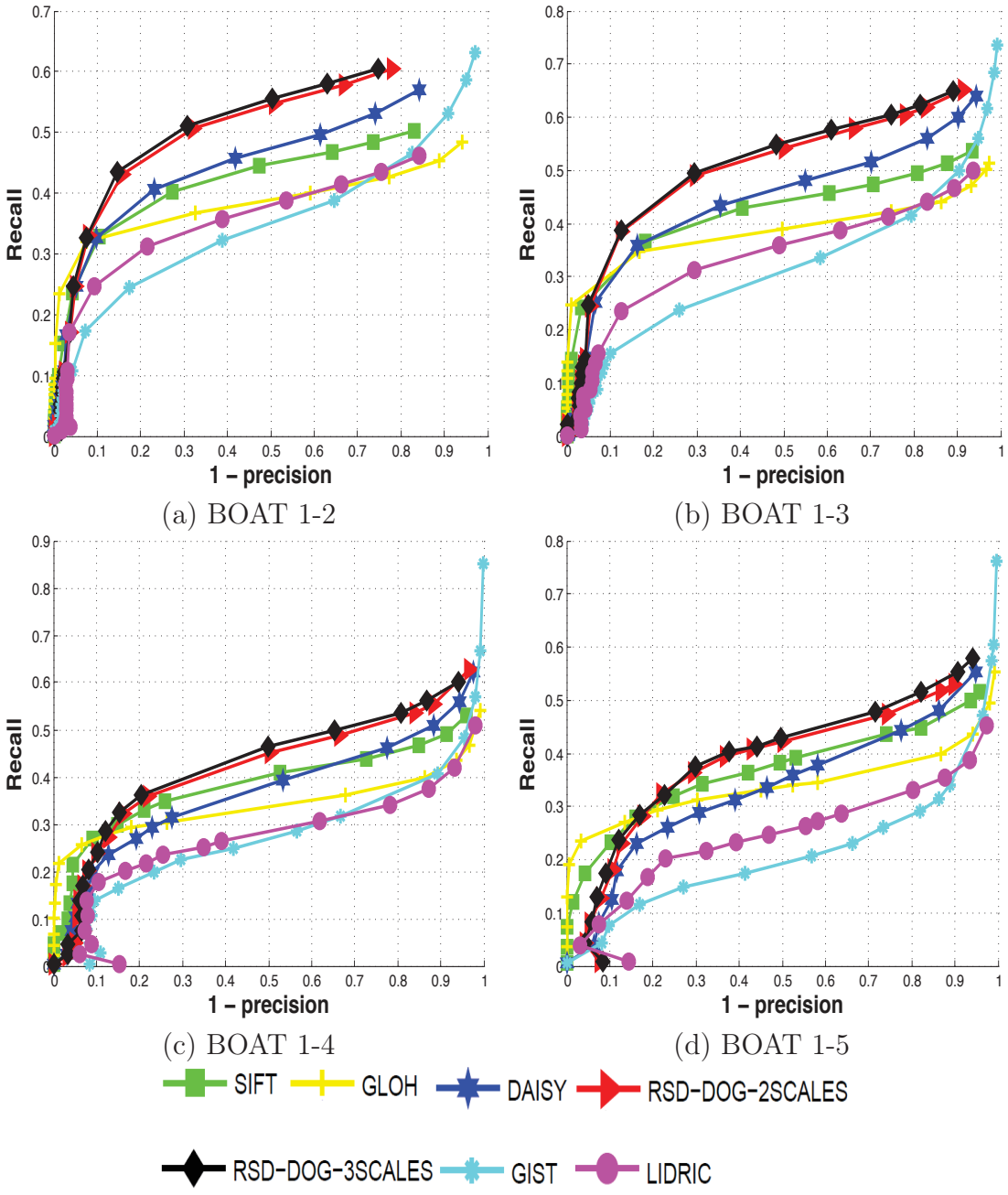
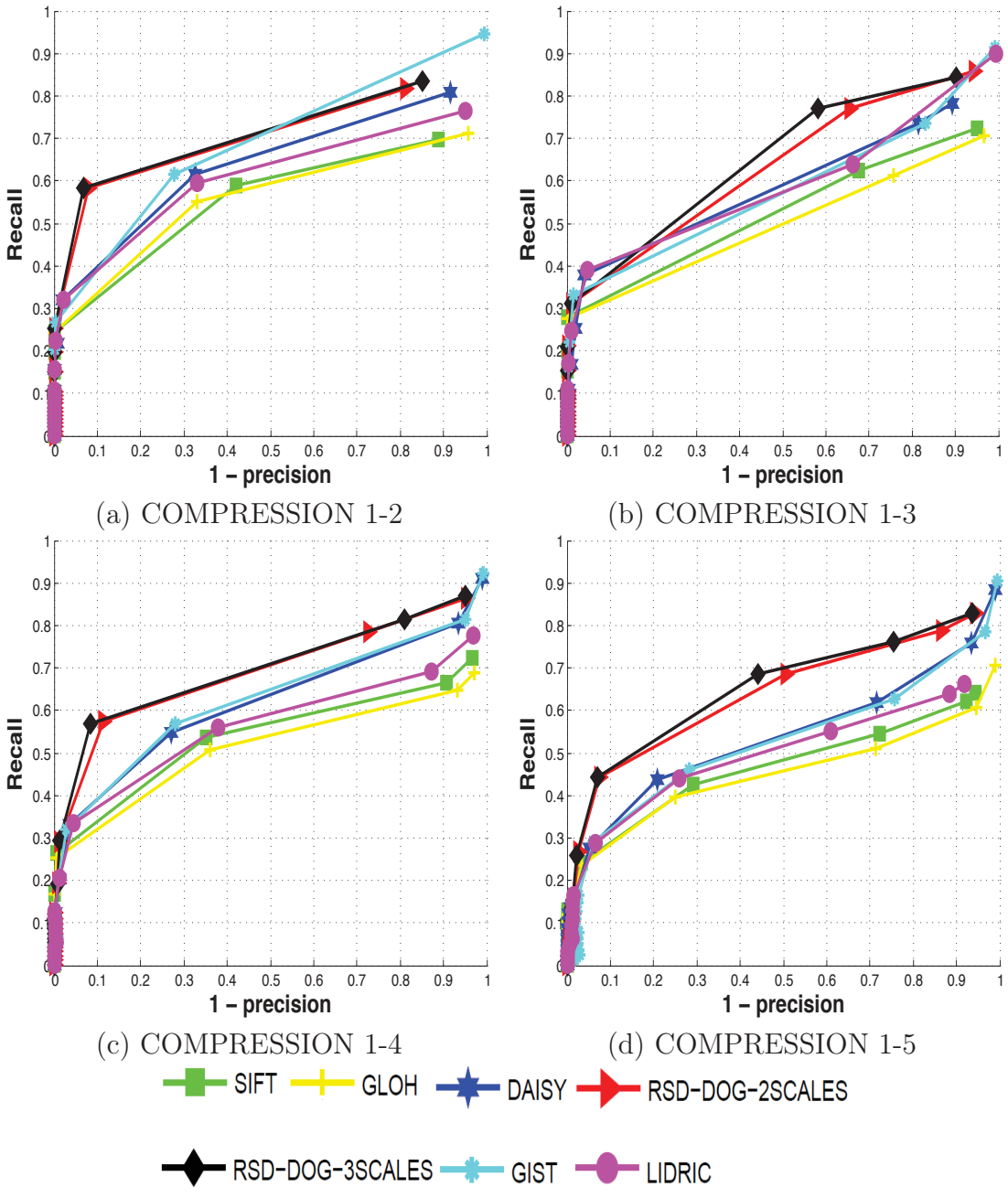


Figure 5.9: Recall vs 1-Precision curves for BOAT sequence.



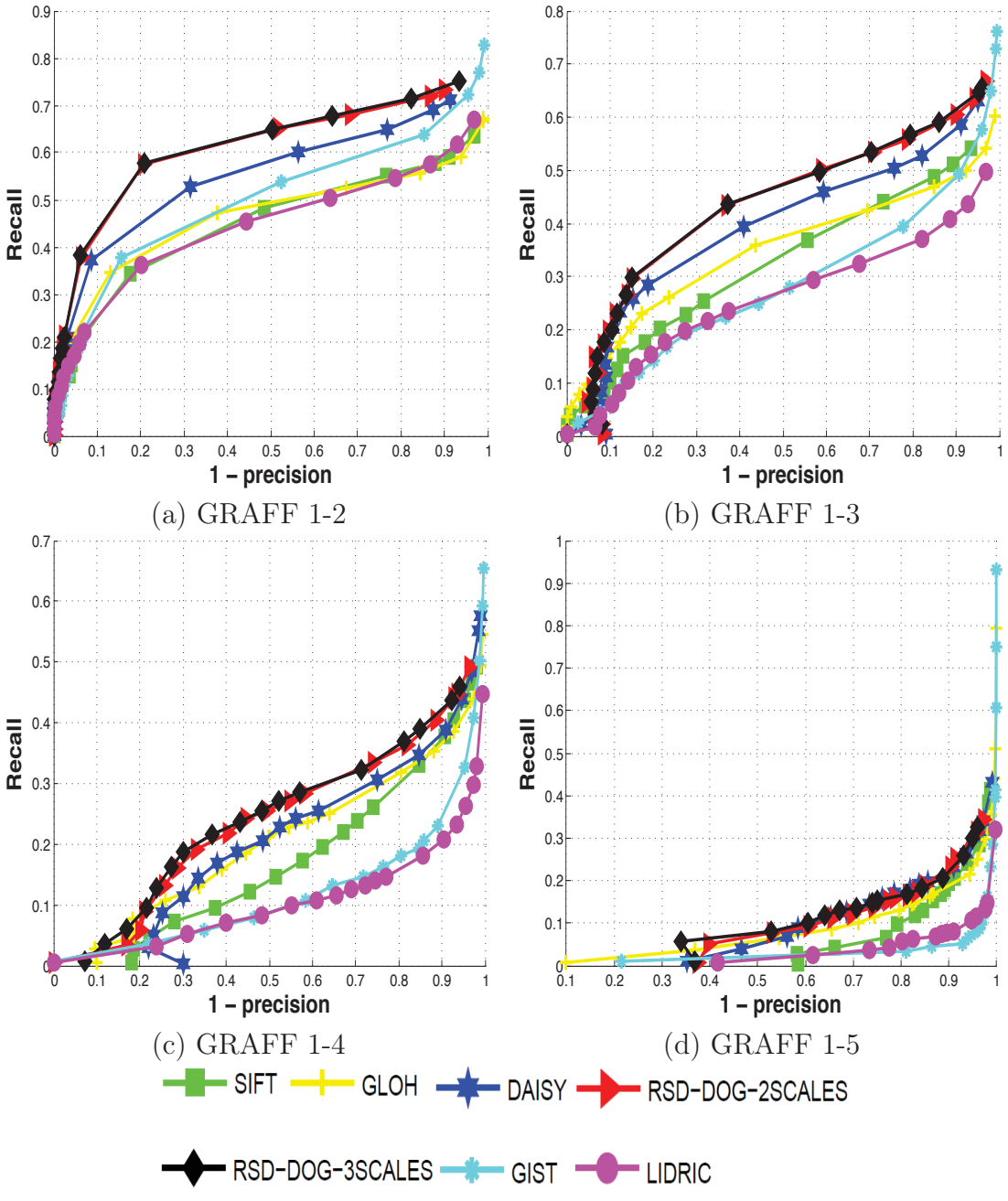
(a) COMPRESSION 1-2

(b) COMPRESSION 1-3

(c) COMPRESSION 1-4

(d) COMPRESSION 1-5

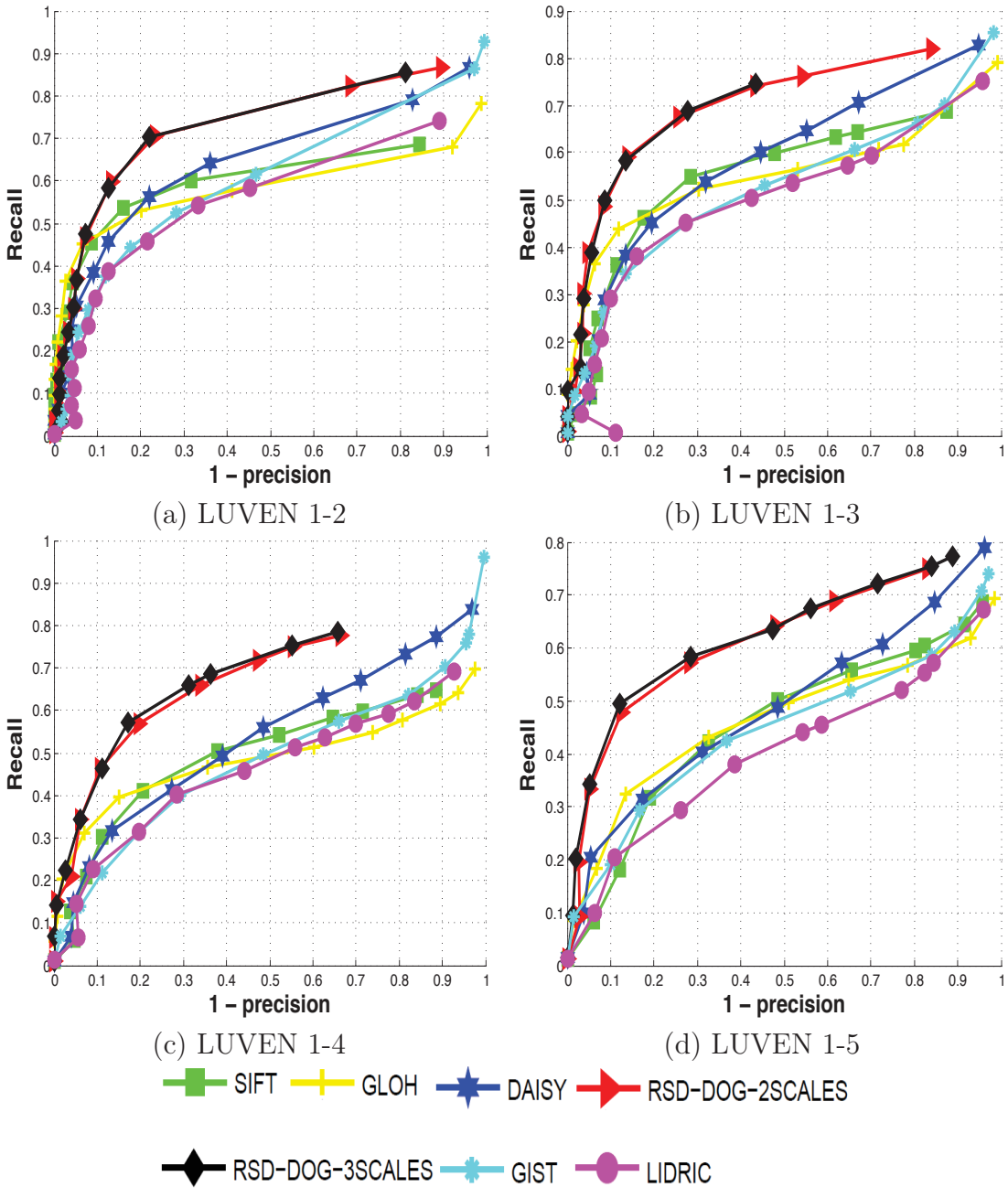
Figure 5.10: Recall vs 1-Precision curves for COMPRESSION sequence.



(a) GRAFF 1-2 (b) GRAFF 1-3

(c) GRAFF 1-4 (d) GRAFF 1-5

Figure 5.11: Recall vs 1-Precision curves for GRAFF sequence.



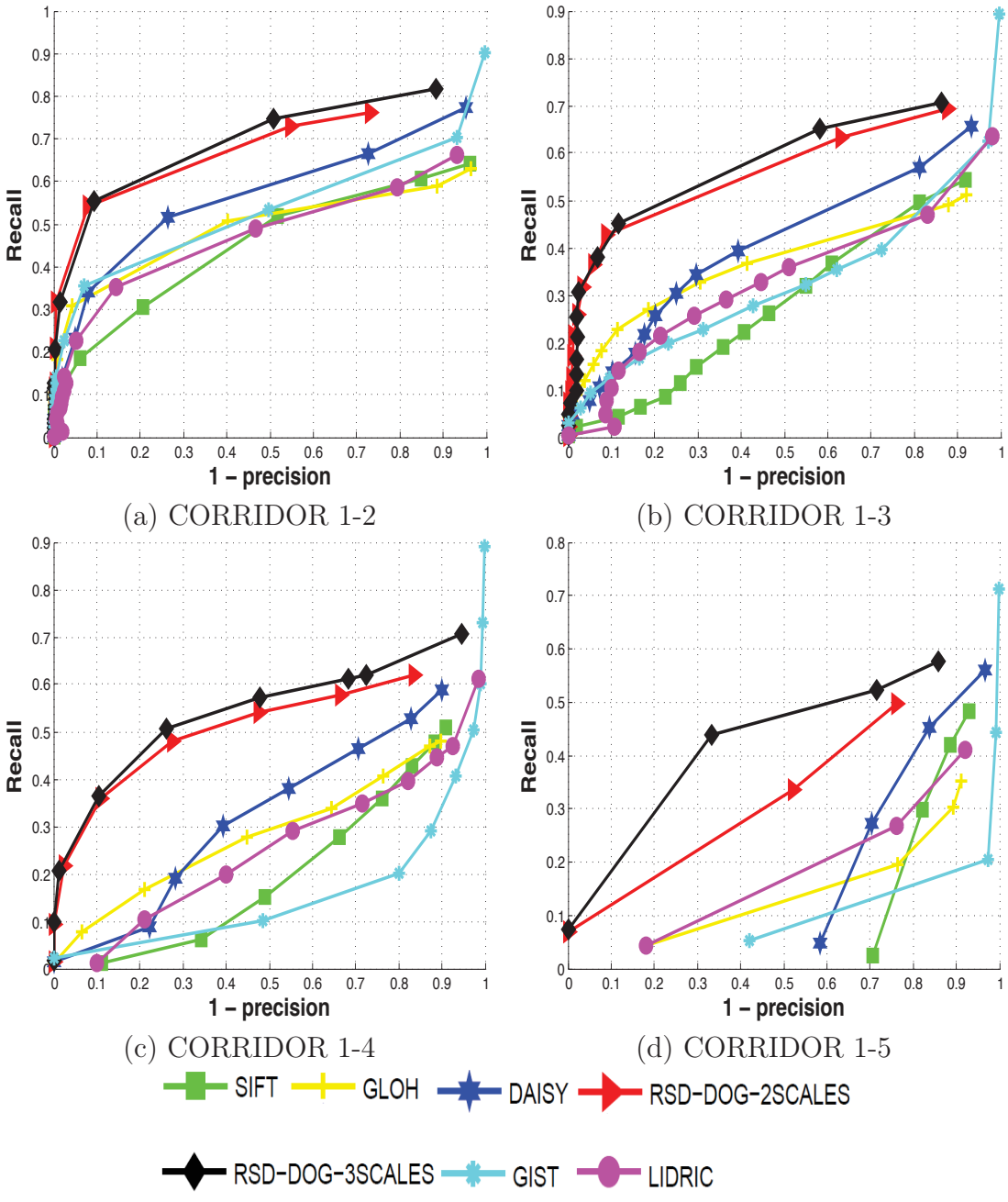
(a) LUVEN 1-2

(b) LUVEN 1-3

(c) LUVEN 1-4

(d) LUVEN 1-5

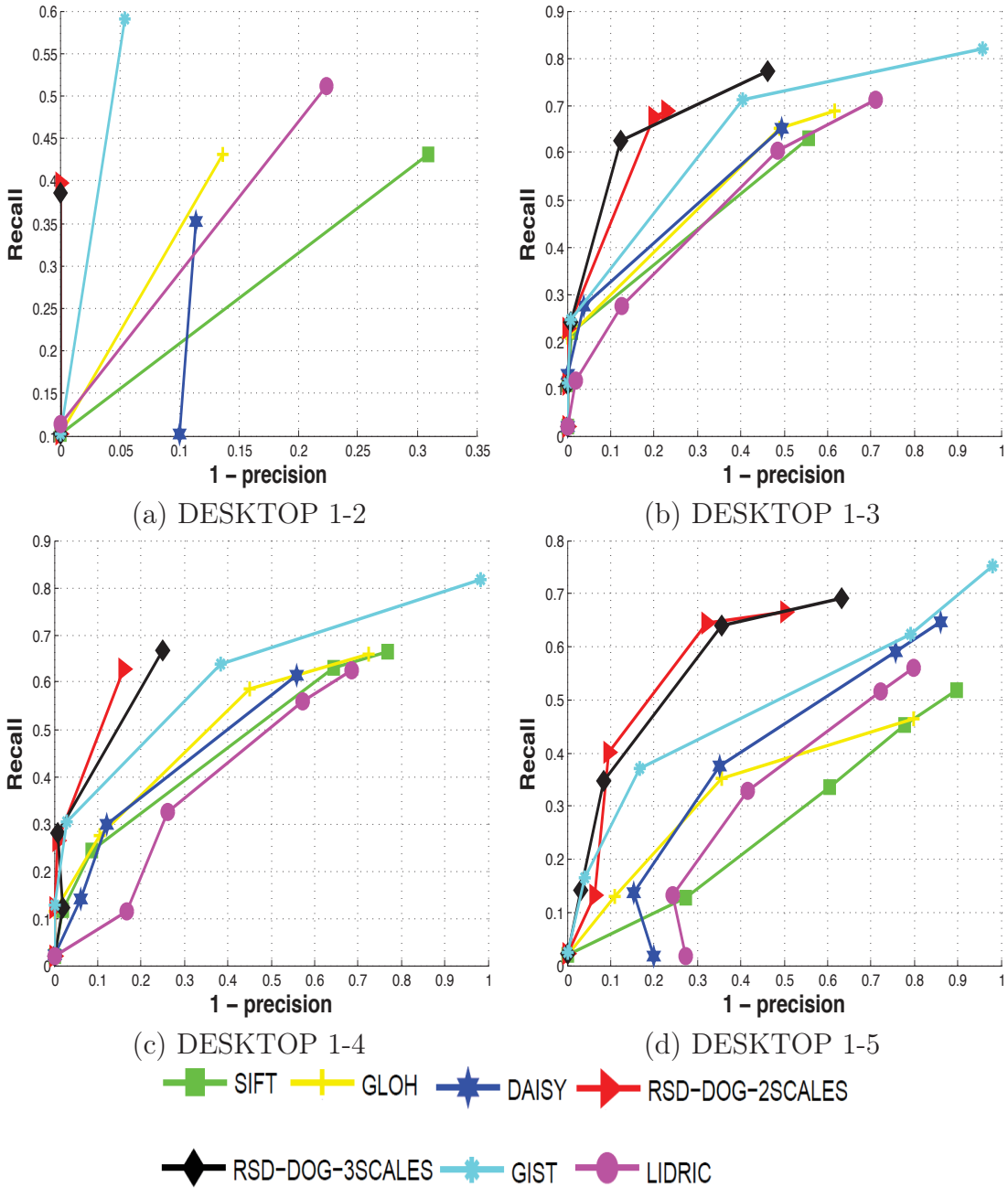
Figure 5.12: Recall vs 1-Precision curves for LUEVEN sequence.



(a) CORRIDOR 1-2 (b) CORRIDOR 1-3

(c) CORRIDOR 1-4 (d) CORRIDOR 1-5

Figure 5.13: Recall vs 1-Precision curves for CORRIDOR sequence.



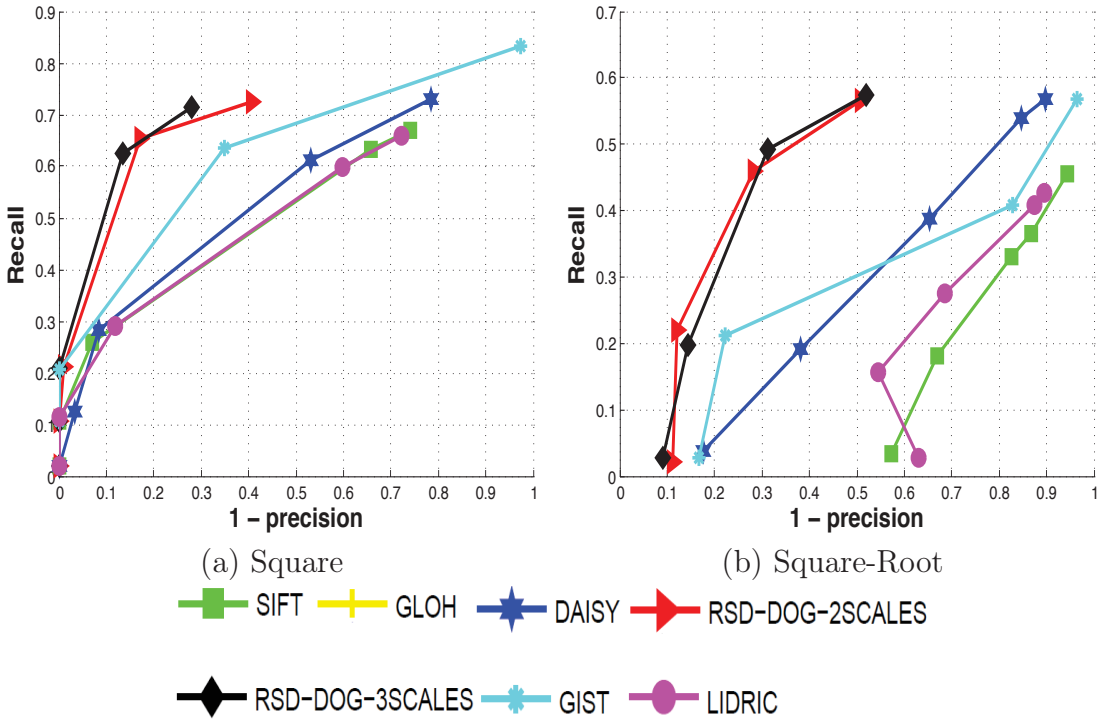
(a) DESKTOP 1-2

(b) DESKTOP 1-3

(c) DESKTOP 1-4

(d) DESKTOP 1-5

Figure 5.14: Recall vs 1-Precision curves for DESKTOP sequence.



(a) Square (b) Square-Root

Figure 5.15: Recall vs 1-Precision curves for SQUARE and SQUARE-ROOT sequence.

Chapter 6

Conclusion & Future work

To the best of our knowledge, image descriptors based on anisotropic filters or any half filters were not fully explored in the computer vision field. In this thesis, we have proposed new image descriptors based on a family of anisotropic half filters which outperforms many state of the art descriptors that are based on first order image statistics and filter responses. This unique work can be used in many different computer vision applications.

6.1 Conclusion

Initially, we conducted a brief literature review on the image matching pipeline and we discussed about different interest point detectors emphasizing on the popular interest point and region detectors. This was followed by a brief overview on different categories of descriptors where we identified their advantages and disadvantages. Next was the matching stage which is usually based on minimization of the inter-distance between the descriptors. This stage allows to determine the pairs of points with the best likeness.

The literature review was followed by discussion on different isotropic and anisotropic filters. In the first part of the chapter.2 different isotropic and anisotropic filters based on Gaussian and exponential were discussed. The latter, considers the advantage of half filters over the full filters and explores different anisotropic smoothing and derivative half filters.

In chapter.3 a new low bit-rate descriptor called RSD was discussed. By construction, the descriptor doesn't exhibit Euclidean invariance. FFT correlation was used to obtain rotation invariance and constrained dynamic time warping was used to achieve partial affine/deformation invariance. A cascade matching scheme was developed to improve the robustness of the final matches. Drawbacks of the proposed method were discussed and since the descriptor was of low dimension, it was extended to the video platform. This was our first contribution to the thesis.

The next important contribution was, the new image patch descriptor RSD-HoG presented in Chapter.4. The descriptor was constructed in a novel way by embedding the response of the rotating half filter in the histogram of orientation gradient (HoG) framework. The proposed descriptor captured the angles at which the edges occurred and the direction of the anisotropic gradient. This combination of features exhibited enormous robustness to geometrical and photometric variations. Image matching results on standard dataset using the standard matching protocol proves the superiority of the RSD-HoG

descriptor against the state of the art descriptors.

Final important contribution, the RSD-DoG descriptor, was presented in Chapter.5. The presented method follows a different approach by considering the image patch as a 3D surface. Thus, considered image patch is made of many second order image statistics such as ridges, valley etc. We then extract the angles at which these second order statistics are present by combining the response of the half filter with that of the Difference of Gaussian (DoG) method. Matching results using the recall vs 1-precision curves proves the strength of the RSD-DoG descriptor.

6.2 Future work

In conclusion, this thesis verified that using half filters we can extract the first order and second order image statistics effectively to form the local image patch descriptors for image matching . In the future we would like to include the following improvements:

- Our methods uses many parameters for descriptor construction. In the future, we would like to introduce a learning step from which we can learn the parameters to maximise the performance of the descriptor.
- In the future, we would like to test the performance of our descriptor on textured image and High resolution image dataset.
- Both RSD-HoG and RSD-DoG uses the first order and second order image statistics such as anisotropic gradient, edge direction, ridges, valleys etc. But it doesn't use the already existing response of the half filters. Like in SURF, we can use the response of the half filters to further enhance the RSD-HoG and RSD-DoG descriptors.
- Our descriptors and in particular the approach used for extracting the local features can be used effectively in medical image analysis especially in retina image analysis. In the future, we would also like to enhance the popularity of our descriptors by using it in the medical imaging field and biometrics field(Finger print recognition).
- In the future, we would like to use the proposed descriptors in applications related to content based image and video retrieval.
- One of the limiting aspect of the proposed RSD-HoG and RSD-DOG descriptors is the speed: the local features are slow to compute. This can be addressed by using GPU or parallel programming. But, this performance is unacceptable in consumer oriented mobile platforms applications and hence, we would like to address this issue in the near future.
- One of the distant scope of the thesis is to use computer vision techniques to help the old and disabled people in *smart homes*. The proposed descriptors were not used in any applications such as (object detection and gesture recognition) that could help the old and disabled in *smart homes*. Due to lack of time we couldn't propose any application. In the near future, we would like to use our work in applications related to *smart homes*.

Publication in the context of this thesis

- **Object matching in videos using rotational signal descriptor**. Darshan Venkatrayappa, Philippe Montesinos, and Daniel Diep. In Three-Dimensional Image Processing, Measurement (3DIPM), and Applications, San francisco, California, United States, volume 9393. SPIE, 2015.

- **RSD-HOG: A new image descriptor**. Darshan Venkatrayappa, Philippe Montesinos, Daniel Diep, and Baptiste Magnier. In Image Analysis - 19th Scandinavian Conference, SCIA 2015, Copenhagen, Denmark, June 15- 17, 2015. Proceedings, volume 9127 of Lecture Notes in Computer Science, pages 400–409. Springer, 2015.

- **A novel image descriptor based on anisotropic filtering**. Darshan Venkatrayappa, Philippe Montesinos, Daniel Diep, and Baptiste Magnier. In Computer Analysis of Images and Patterns - 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015 Proceedings, Part I, volume 9256 of Lecture Notes in Computer Science, pages 161–173. Springer, 2015. (**Best Paper award**)

- **RSD-DOG : A new image descriptor based on second order derivatives**. Darshan Venkatrayappa, Philippe Montesinos, Daniel Diep, and Baptiste Magnier. In Advances Concepts for Intelligent Vision Systems , ACIVS 2015, Catania,Italy, October 26-29, 2011. (To Appear), Lecture Notes in Computer Science. Springer, 2015.

Bibliography

[AB13] Aniruddha K. Acharya and Venkatesh R. Babu. A real time implementation of sift using gpu. In *The 4th National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, 18-21 Dec. 2013, Jodhpur, India*, pages 1 – 4. IEEE Computer Society, 2013.

[ABD12] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J. Davison. KAZE features. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI*, volume 7577 of *Lecture Notes in Computer Science*, pages 214–227. Springer, 2012.

[ABD13] Pablo F. Alcantarilla, Luis Miguel Bergasa, and Andrew J. Davison. Gauge-surf descriptors. *Image Vision Comput.*, 31(1):103–116, 2013.

[AF06] Alaa E. Abdel-Hakim and Aly A. Farag. CSIFT: A SIFT descriptor with color invariant characteristics. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1978–1983. IEEE Computer Society, 2006.

[AI08] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

[AK09] Andrey Andreev and Nikolay Kirov. Word image matching based on hausdorff distances. In *10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009*, pages 396–400. IEEE Computer Society, 2009.

[AKB08] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2008.

[AMHP09] Timo Ahonen, Jiri Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Image Analysis, 16th Scandinavian Conference, SCIA 2009, Oslo,*

Norway, June 15-18, 2009. Proceedings, volume 5575 of *Lecture Notes in Computer Science*, pages 61–70. Springer, 2009.

[AMN+98] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998.

[AOV12] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 510–517. IEEE Computer Society, 2012.

[AP09] Sanun Srisuk Amnart Petpon. Face recognition with local line binary pattern. In *International Conference on Image and Graphics, ICIG 2009*, pages 533–539. IEEE Computer Society, 2009.

[ARG10] Faraj Alhwarin, Danijela Ristic-Durrant, and Axel Gräser. VF-SIFT: very fast SIFT feature matching. In *Pattern Recognition - 32nd DAGM Symposium, Darmstadt, Germany, September 22-24, 2010. Proceedings*, volume 6376 of *Lecture Notes in Computer Science*, pages 222–231. Springer, 2010.

[AWRG08] Faraj Alhwarin, Chao Wang, Danijela Ristic-Durrant, and Axel Gräser. Improved sift-features matching for object recognition. In *Visions of Computer Science - BCS International Academic Conference, Imperial College, London, UK, 22-24 September 2008*, pages 178–190. British Computer Society, 2008.

[Bau00] Adam Baumberg. Reliable feature matching across widely separated views. In *2000 Conference on Computer Vision and Pattern Recognition (CVPR 2000), 13-15 June 2000, Hilton Head, SC, USA*, pages 1774–1781. IEEE Computer Society, 2000.

[Bea78] P. Beaudet. Rotationally invariant image operators. In *4th International Joint Conference on Pattern Recognition*, page 579–583, 1978.

[BEA83] Peter J. Burt, Edward, and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications.*, 4:532–540, 1983.

[Bel11] Fabio Bellavia. Matching image features. *PhD thesis*, 2011.

[BHW11] Matthew Brown, Gang Hua, and Simon A. J. Winder. Discriminative learning of local image descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):43–57, 2011.

[BL97] Jeffrey S. Beis and David G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), June 17-19, 1997, San Juan, Puerto Rico*, pages 1000–1006. IEEE Computer Society, 1997.

[BL02] Matthew Brown and David G. Lowe. Invariant features from interest point groups. In *Proceedings of the British Machine Vision Conference 2002, BMVC 2002, Cardiff, UK, 2-5 September 2002*, pages 1–10. British Machine Vision Association, 2002.

[BL03] Matthew Brown and David G. Lowe. Recognising panoramas. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 1218–1227. IEEE Computer Society, 2003.

[BMP02] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[BN07] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern Recognition and Machine Learning. J. Electronic Imaging*, 16(4):049901, 2007.

[BOZB14] Abdelkader Bellarbi, Samir Otmane, Nadia Zenati, and Samir Benbelkacem. MOBIL: A moments based local binary descriptor. In *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2014, Munich, Germany, September 10-12, 2014*, pages 251–252. IEEE Computer Society, 2014.

[BRF10] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 244–252. Curran Associates, Inc., 2010.

[BS11] Matthew Brown and Sabine Süsstrunk. Multi-spectral SIFT for scene category recognition. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 177–184. IEEE Computer Society, 2011.

[BTG06] Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool. SURF: speeded up robust features. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006.

[BZlCl15] Du Bo, Zhangguan-liang, and Cuixiao-long. An algorithm of image matching based on mahalanobis distance and weighted knn graph. In *Information Science and Control Engineering (ICISCE)*, pages 116 – 121. IEEE, 2015.

[BZM07] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007, Amsterdam, The Netherlands, July 9-11, 2007*, pages 401–408. ACM, 2007.

[BZM08] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(4):712–727, 2008.

[Can86] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[CF06] David Connah and Graham D. Finlayson. Using local binary pattern operators for colour constant image indexing. In *3rd European Conference on Colour in Graphics, Imaging, and Vision, CGIV 2006, Leeds, UK, June 19-22, 2006*, pages 60–64. IS&T - The Society for Imaging Science and Technology, 2006.

[CH07] Warren Cheung and Ghassan Hamarneh. N-sift: N-dimensional scale invariant feature transform for matching medical images. In *Proceedings of the 2007 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Washington, DC, USA, April 12-16, 2007*, pages 720–723. IEEE, 2007.

[CHYC15] Li-Chih Chen, Jun-Wei Hsieh, Yilin Yan, and Duan-Yu Chen. Vehicle make and model recognition using sparse representation and symmetrical surfs. *Pattern Recognition*, 48(6):1979–1998, 2015.

[CLB10] Gail Carmichael, Robert Laganière, and Prosenjit Bose. Global context descriptors for SURF and MSER feature descriptors. In *Canadian Conference on Computer and Robot Vision, CRV 2010, Ottawa, Ontario, Canada, May 31 - June 2, 2010*, pages 309–316. IEEE Computer Society, 2010.

[CLSF10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: binary robust independent elementary features. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792. Springer, 2010.

[CP84] James L. Crowley and Alice C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(2):156–170, 1984.

[CSS09] Samuel Cheng, Vladimir Stankovic, and Lina Stankovic. Improved sift-based image registration using belief propagation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009, 19-24 April 2009, Taipei, Taiwan*, pages 2909–2912. IEEE, 2009.

[CT11] Ricardo Chincha and Yingli Tian. Finding objects for blind people based on SURF features. In *2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW), Atlanta, GA, USA, November 12-15, 2011*, pages 526–527. IEEE, 2011.

[CTC+09] Vijay Chandrasekhar, Gabriel Takacs, David Chen, Sam S. Tsai, Jatinder Singh, and Bernd Girod. Transform coding of image feature descriptors. In *In Proc. of visual communications and image processing conference (VCIP)*, 2009.

[CTC+12] Vijay Chandrasekhar, Gabriel Takacs, David M. Chen, Sam S. Tsai, Yuriy A. Reznik, Radek Grzeszczuk, and Bernd Girod. Compressed histogram of gradients: A low-bitrate descriptor. *International Journal of Computer Vision*, 96(3):384–399, 2012.

[Der87] Rachid Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.

[Der93] Rachid Deriche. Recursively implementating the Gaussian and its derivatives. Research Report RR-1893, 1993.

[DJ94] Marie-Pierre Dubuisson and Anil K. Jain. A modified hausdorff distance for object matching, 1994.

[DN81] Leonie Dreschler and Hans-Hellmut Nagel. Volumetric model and 3d-trajectory of a moving car derived from monocular tv-frame sequence of a street scene. In Patrick J. Hayes, editor, *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), Vancouver, BC, Canada, August 1981*, pages 692–697. William Kaufmann, 1981.

[DSC09] Geng Du, Fei Su, and Anni Cai. Face recognition using surf features. In *Proceedings of the SPIE,2009*, 2009.

[DSHN09] Philippe Dreuw, Pascal Steingrube, Harald Hanselmann, and Hermann Ney. Surf-face: Face recognition under viewpoint consistency constraints. In *British Machine Vision Conference, BMVC 2009, London, UK, September 7-10, 2009. Proceedings*, pages 1–11. British Machine Vision Association, 2009.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893. IEEE Computer Society, 2005.

[FA91] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(9):891–906, 1991.

[FB14] Philipp Fischer and Thomas Brox. Image descriptors based on curvature histograms. In *Pattern Recognition - 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings*, volume 8753 of *Lecture Notes in Computer Science*, pages 239–249. Springer, 2014.

[FBA+06] Quanfu Fan, Kobus Barnard, Arnon Amir, Alon Efrat, and Ming Lin. Matching slides to presentation videos using SIFT and scene background matching. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR 2006, October 26-27, 2006, Santa Barbara, California, USA*, pages 239–248. ACM, 2006.

[FFB95] Graham D. Finlayson, Brian V. Funt, and Kobus Barnard. Color constancy under varying illumination. In *ICCV*, pages 720–725, 1995.

[FJ05] Matteo Frigo and Steven G. Johnson. The design and implementation of fftw3. In *Proceedings of the IEEE - Special issue on Program Generation, Optimization, and Platform Adaptation*, pages 216–231, 2005.

[FN75] Keinosuke Fukunaga and Patrenahalli M. Narendra. A branch and bound algorithms for computing k-nearest neighbors. *IEEE Trans. Computers*, 24(7):750–753, 1975.

[FWH12] Bin Fan, Fuchao Wu, and Zhanyi Hu. Rotationally invariant descriptors using intensity order pooling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10):2031–2045, 2012.

[FZA+11] Johannes Feulner, Shaohua Kevin Zhou, Elli Angelopoulou, Sascha Seifert, Alexander Cavallaro, Joachim Hornegger, and Dorin Comaniciu. Comparing axial CT slices in quantized n-dimensional SURF descriptor space to estimate the visible body region. *Comp. Med. Imag. and Graph.*, 35(3):227–236, 2011.

[GC10] Xiaojie Guo and Xiaochun Cao. FIND: A neat flip invariant descriptor. In *20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*, pages 515–518. IEEE Computer Society, 2010.

[GC12] Xiaojie Guo and Xiaochun Cao. MIFT: A framework for feature descriptors to be mirror reflection invariant. *Image Vision Comput.*, 30(8):546–556, 2012.

[GC14] Matheus A. Gadelha and Bruno M. Carvalho. DRINK: discrete robust invariant keypoints. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 821–826. IEEE, 2014.

[GD04] Kristen Grauman and Trevor Darrell. Fast contour matching using approximate earth mover's distance. In *CVPR (1)*, pages 220–227, 2004.

[GL06] Iryna Gordon and David G. Lowe. What and where: 3d object recognition with accurate pose. In *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*, pages 67–82. Springer, 2006.

[GL11] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 817–824. IEEE Computer Society, 2011.

[GSSF13] Zeinab Ghassabi, Amin Sedaghat, Jamshid Shanbehzadeh, and Emad Fatemizadeh. An efficient approach for robust multimodal retinal image registration based on UR-SIFT features and PIIFD descriptors. *EURASIP J. Image and Video Processing*, 2013:25, 2013.

[GSvdW02] Jan-Mark Geusebroek, Arnold W. M. Smeulders, and Joost van de Weijer. Fast anisotropic gauss filtering. In *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part I*, volume 2350 of *Lecture Notes in Computer Science*, pages 99–112. Springer, 2002.

[GvdBSG01] Jan-Mark Geusebroek, Rein van den Boomgaard, Arnold W. M. Smeulders, and Hugo Geerts. Color invariance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(12):1338–1350, 2001.

[GXXS13] Ting Gao, Yu Xu, Ting-xin Xu, and Li Shuai. Multi-scale image registration algorithm based on improved SIFT. *Journal of Multimedia*, 8(6):755–761, 2013.

[GZZ10] Zhenhua Guo, Lei Zhang, and David Zhang. Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recognition*, 43(3):706–719, 2010.

[GZZZ10] Zhenhua Guo, Lei Zhang, David Zhang, and Su Zhang. Rotation invariant texture classification using adaptive LBP with directional statistical features. In *Proceedings of the International Conference on Image Processing, ICIP 2010, September 26-29, Hong Kong, China*, pages 285–288. IEEE, 2010.

[HAP08] Chu He, Timo Ahonen, and Matti Pietikäinen. A bayesian local binary pattern texture descriptor. In *19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA*, pages 1–4. IEEE Computer Society, 2008.

[HBW07] Gang Hua, Matthew Brown, and Simon A. J. Winder. Discriminant embedding for local image descriptors. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, pages 1–8. IEEE, 2007.

[HDF12] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II*, volume 7573 of *Lecture Notes in Computer Science*, pages 759–773. Springer, 2012.

[HP01] Joon Hee Han and Timothy Poston. Chord-to-point distance accumulation and planar curvature: a new approach to discrete curvature. *Pattern Recognition Letters*, 22(10):1133–1144, 2001.

[HPS06] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with center-symmetric local binary patterns. In *Computer Vision, Graphics and Image Processing, 5th Indian Conference, ICVGIP 2006, Madurai, India, December 13-16, 2006, Proceedings*, volume 4338 of *Lecture Notes in Computer Science*, pages 58–69. Springer, 2006.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference, AVC 1988, Manchester, UK, September, 1988*, pages 1–6. Alvey Vision Club, 1988.

[HSZ07] Adel Hafiane, Guna Seetharaman, and Bertrand Zavidovique. Median binary pattern for textures classification. In *Image Analysis and Recognition, 4th International Conference, ICIAR 2007, Montreal, Canada, August 22-24, 2007, Proceedings*, volume 4633 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 2007.

[HZWC14] Di Huang, Chao Zhu, Yunhong Wang, and Liming Chen. HSOG: A novel local image descriptor based on histograms of the second-order gradients. *IEEE Transactions on Image Processing*, 23(11):4680–4695, 2014.

[IKM08] Dimitrios K. Iakovidis, Eystratios G. Keramidas, and Dimitris Maroulis. Fuzzy local binary patterns for ultrasound texture characterization. In *Image Analysis and Recognition, 5th International Conference, ICIAR 2008, Póvoa de Varzim, Portugal, June 25-27, 2008. Proceedings*, volume 5112 of *Lecture Notes in Computer Science*, pages 750–759. Springer, 2008.

[JDS11] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011.

[JF08] Steven G. Johnson and Matteo Frigo. Implementing ffts in practice. In C. Sidney Burrus, editeur, Fast Fourier Transforms, chapitre 11. Connexions, Rice University, Houston TX, USA, September 2008.

[JLLT04] Hongliang Jin, Qingshan Liu, Hanqing Lu, and Xiaofeng Tong. Face detection using improved LBP under bayesian framework. In *Third International Conference on Image and Graphics, ICIG 2004, Hong Kong, China, December 18-20, 2004*, pages 306–309. IEEE Computer Society, 2004.

[JPG12] Umarani Jayaraman, Surya Prakash, and Phalguni Gupta. An efficient color and texture based iris image retrieval technique. *Expert Syst. Appl.*, 39(5):4915–4926, 2012.

[JSFJ11] Xinghao Jiang, Tanfeng Sun, Bing Feng, and Chengming Jiang. A space-time SURF descriptor and its application to action recognition with video words. In *Eighth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2011, 26-28 July 2011, Shanghai, China*, pages 1911–1915. IEEE, 2011.

[JU04] Mathews Jacob and Michael Unser. Design of steerable filters for feature detection using canny-like criteria. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(8):1007–1019, 2004.

[JY09] Hao Jiang and Stella X. Yu. Linear solution to scale and rotation invariant object matching. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 2474–2481. IEEE Computer Society, 2009.

[JZ97] Anil K. Jain and Douglas E. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(12):1386–1391, 1997.

[KB01] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[KCL15] Tae-Koo Kang, In-Hwan Choi, and Myo-Taeg Lim. MDGHM-SURF: A robust local image descriptor based on modified discrete gaussian-hermite moment. *Pattern Recognition*, 48(3):670–684, 2015.

[KD09] Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 1042–1050. Curran Associates, Inc., 2009.

[KMS08] Alexander Kläser, Marcin Marszalek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proceedings of the British Machine Vision Conference 2008, Leeds, September 2008*, pages 1–10. British Machine Vision Association, 2008.

[KR82] Les Kitchen and Azriel Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, 1(2):95–102, 1982.

[KS04] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR (2)*, pages 506–513, 2004.

[KvD87] Jan J. Koenderink and Andrea J. van Doorn. Representation of local geometry in the visual system. *Biological cybernetics.*, 55:367–375, 1987.

[KvdWHR06] David Knossow, Joost van de Weijer, Radu Horaud, and Rémi Ronfard. Articulated-body tracking through anisotropic edge detection. In *Dynamical Vision, ICCV 2005 and ECCV 2006 Workshops, WDV 2005 and WDV 2006, Beijing, China, October 21, 2005, Graz, Austria, May 13, 2006. Revised Papers*, volume 4358 of *Lecture Notes in Computer Science*, pages 86–99. Springer, 2006.

[KZB04] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*, volume 3021 of *Lecture Notes in Computer Science*, pages 228–241, 2004.

[LB03] Tony Lindeberg and Lars Bretzner. Real-time scale selection in hybrid multi-scale representations. In *Scale Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 148–163. Springer, 2003.

[LBS14] Henning Lategahn, Johannes Beck, and Christoph Stiller. DIRD is an illumination robust descriptor. In *2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, June 8-11, 2014*, pages 756–761. IEEE, 2014.

[LC07] Shu Liao and Albert C. S. Chung. Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude. In *Computer Vision - ACCV 2007, 8th Asian Conference on Computer Vision, Tokyo, Japan, November 18-22, 2007, Proceedings, Part II*, volume 4844 of *Lecture Notes in Computer Science*, pages 672–679. Springer, 2007.

[LCS11] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2548–2555. IEEE, 2011.

[LDDP12] Anders Boesen Lindbo Larsen, Sune Darkner, Anders Lindbjerg Dahl, and Kim Steenstrup Pedersen. Jet-based local image descriptors. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III*, volume 7574 of *Lecture Notes in Computer Science*, pages 638–650. Springer, 2012.

[Lem09] Daniel Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition*, 42(9):2169–2180, 2009.

[LHZW10] Xi Li, Weiming Hu, Zhongfei Zhang, and Hanzi Wang. Heat kernel based local binary pattern for face representation. *IEEE Signal Process. Lett.*, 17(3):308–311, 2010.

[Lin98] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[LJLC04] Thomas Lewiner, João D. Gomes Jr., Hélio Lopes, and Marcos Craizer. Arc-length based curvature estimator. In *XVII Brazilian Symposium on Computer Graphics and Image Processing, (SIBGRAPI 2004) 17-20 October 2004, Curitiba, PR, Brazil*, pages 250–257. IEEE Computer Society, 2004.

[LM01a] Thomas K. Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[LM01b] Thomas K. Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[LMB+05] D.A. Lisin, M.A. Mattar, M.B. Blaschko, E.G. Learned-Miller, and M.C. Benfield. Combining local and global image features for object class recognition. *Workshop on Learning in Computer Vision and Pattern Recognition.*, 2005.

[LMGY04] Ting Liu, Andrew W. Moore, Alexander G. Gray, and Ke Yang. An investigation of practical approximate nearest neighbor algorithms. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 825–832, 2004.

[LMSR08] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008.

[Low99] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LSP05a] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1265–1278, 2005.

[LSP05b] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1265–1278, 2005.

[LSP06] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 2169–2178. IEEE Computer Society, 2006.

[LWZ11] Jianguo Li, Tao Wang, and Yimin Zhang. Face detection using SURF cascade. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*, pages 2183–2190. IEEE, 2011.

[LYH11] Congxin Liu, Jie Yang, and Hai Huang. P-SURF: A robust local image descriptor. *J. Inf. Sci. Eng.*, 27(6):2001–2015, 2011.

[MCS10] Rui Ma, Jian Chen, and Zhong Su. MI-SIFT: mirror and inversion invariant generalization for SIFT descriptor. In *Proceedings of the 9th ACM International Conference on Image and Video Retrieval, CIVR 2010, Xi'an, China, July 5-7, 2010*, pages 228–235. ACM, 2010.

[MCUP02] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference 2002, BMVC 2002, Cardiff, UK, 2-5 September 2002*, pages 1–10. British Machine Vision Association, 2002.

[MDS05] Eric N. Mortensen, Hongli Deng, and Linda G. Shapiro. A SIFT descriptor with global context. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 184–190. IEEE Computer Society, 2005.

[MEO11] Antonio Monroy, Angela Eigenstetter, and Björn Ommer. Beyond straight lines - object detection using curvature. In *18th IEEE International Conference on Image Processing, ICIP 2011, Brussels, Belgium, September 11-14, 2011*, pages 3561–3564. IEEE, 2011.

[MFM04] David R. Martin, Charless Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004.

[MGD98] Philippe Montesinos, Valérie Gouet, and Rachid Deriche. Differential invariants for color images. In *Fourteenth International Conference on Pattern Recognition, ICPR 1998, Brisbane, Australia, 16-20 August, 1998*, pages 838–840. IEEE Computer Society, 1998.

[MH80] D. Marr and E. Hildreth. Theory of edge detection. In *Proceedings of the Royal Society of London. Series B, Biological Sciences*, page 1167), 1980.

[MHB+10] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerd Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II*, volume 6312 of *Lecture Notes in Computer Science*, pages 183–196. Springer, 2010.

[MM86] Farzin Mokhtarian and Alan K. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):34–43, 1986.

[MM96] B. S. Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[MM10] Philippe Montesinos and Baptiste Magnier. A new perceptual edge detector in color images. In *Advanced Concepts for Intelligent Vision Systems - 12th International Conference, ACIVS 2010, Sydney, Australia, December 13-16, 2010, Proceedings, Part I*, volume 6474 of *Lecture Notes in Computer Science*, pages 209–220. Springer, 2010.

[MM12] Ondrej Miksik and Krystian Mikolajczyk. Evaluation of local detectors and descriptors for fast feature matching. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, pages 2681–2684. IEEE Computer Society, 2012.

[MM14] Baptiste Magnier and Philippe Montesinos. Oriented half gaussian kernels and anisotropic diffusion. In *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications, Volume 1, Lisbon, Portugal, 5-8 January, 2014*, pages 73–81. SciTePress, 2014.

[MMD11a] Baptiste Magnier, Philippe Montesinos, and Daniel Diep. Texture removal by pixel classification using a rotating filter. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pages 1097–1100. IEEE, 2011.

[MMD11b] Baptiste Magnier, Philippe Montesinos, and Daniel Diep. Texture removal in color images by anisotropic diffusion. In *VISAPP 2011 - Proceedings of the Sixth International Conference on Computer Vision Theory and Applications, Vilamoura, Algarve, Portugal, 5-7 March, 2011*, pages 40–50. SciTePress, 2011.

[MMD11c] Philippe Montesinos, Baptiste Magnier, and Daniel Diep. Perceptual crest line extraction. In *IEEE IVMSP, Image, Video, and Multidimensional Signal Processing, Ithaca, New York, June 16-17 2011*, 2011.

[MMD13] Baptiste Magnier, Philippe Montesinos, and Daniel Diep. A tool for brain magnetic resonance image segmentation. In *VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications, Volume 2, Barcelona, Spain, 21-24 February, 2013.*, pages 75–79. SciTePress, 2013.

[MMP10] Philippe Montesinos, Baptiste Magnier, and Jean-Louis Palomares. A new perceptual edge detector in color images. In *In Proceedings of the third International Workshop on Image Analysis : IWIA 2010*, page 185–192, 2010.

[MN95] Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[MP04a] Topi Mäenpää and Matti Pietikäinen. Classification with color and texture: jointly or separately? *Pattern Recognition*, 37(8):1629–1640, 2004.

[MP04b] Pierre Moreels and Pietro Perona. Common-frame model for object recognition. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 953–960, 2004.

[MREM04] Greg Mori, Xiaofeng Ren, Alexei A. Efros, and Jitendra Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR (2)*, pages 326–333, 2004.

[MS01] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *ICCV*, pages 525–531, 2001.

[MS02] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *ECCV , European Conference on Computer Vision, Copenhagen, Denmark*, volume 2350 of *Lecture Notes in Computer Science*, pages 128–142. Springer, 2002.

[MS03] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pages 257–263. IEEE Computer Society, 2003.

[MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[MS05] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[MSM13] Hunny Mehrotra, Pankaj Kumar Sa, and Banshidhar Majhi. Fast segmentation and adaptive SURF descriptor for iris recognition. *Mathematical and Computer Modelling*, 58(1-2):132–146, 2013.

[MTEF06] Kevin P. Murphy, Antonio Torralba, Daniel Eaton, and William T. Freeman. Object detection and localization using local and global features. In *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*, pages 382–400. Springer, 2006.

[MWW12] Xianglin Meng, Zhengzhi Wang, and Lizhen Wu. Building global image features for scene recognition. *Pattern Recognition*, 45(1):373–380, 2012.

[MXM13] Baptiste Magnier, Huanyu Xu, and Philippe Montesinos. Half gaussian kernels based shock filter for image deblurring and regularization. In *VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications, Volume 1, Barcelona, Spain, 21-24 February, 2013.*, pages 51–60. SciTePress, 2013.

[MY09] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Imaging Sciences*, 2(2):438–469, 2009.

[NLB10] Loris Nanni, Alessandra Lumini, and Sheryl Brahnam. Local binary patterns variants as texture descriptors for medical image analysis. *Artificial Intelligence in Medicine*, 49(2):117–125, 2010.

[NPH+13] Thao Nguyen, Eun-Ae Park, Jiho Han, Dong-Chul Park, and Soo-Young Min. Object detection using scale invariant feature transform. In *Genetic and Evolutionary Computing - Proceedings of the Seventh International Conference on Genetic and Evolutionary Computing, ICGEC 2013, August 25-27, 2013, Prague, Czech Republic*, volume 238 of *Advances in Intelligent Systems and Computing*, pages 65–72. Springer, 2013.

[OAZ] Robert M. Haralick Oscar A. Zuniga. Corner detection using the facet model. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 30–37, 1983.*

[OJL07] Margarita Osadchy, David W. Jacobs, and Michael Lindenbaum. Surface dependent representations for illumination insensitive image comparison. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):98–111, 2007.

[OP99] Timo Ojala and Matti Pietikäinen. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, 32(3):477–486, 1999.

[OPH96] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[OPM01] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In *Advances in Pattern Recognition - ICAPR 2001, Second International Conference Rio de Janeiro, Brazil, March 11-14, 2001, Proceedings*, volume 2013 of *Lecture Notes in Computer Science*, pages 397–406. Springer, 2001.

[OPM02a] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[OPM02b] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[OT01] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[OT06] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Visual Perception, Progress in Brain Research.*, 155, 2006.

[Per92] Pietro Perona. Steerable-scalable kernels for edge detection and junction analysis. In *Computer Vision - ECCV'92, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 19-22, 1992, Proceedings*, volume 588 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 1992.

[Per95] Pietro Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(5):488–499, 1995.

[PHZA11] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. Computer vision using binary patterns. 2011.

[PLYP12] Yanwei Pang, Wei Li, Yuan Yuan, and Jing Pan. Fully affine invariant SURF for image matching. *Neurocomputing*, 85:6–10, 2012.

[PMD12] Jean-Louis Palomares, Philippe Montesinos, and Daniel Diep. A new affine invariant method for image matching. In *2012 3DIP Image Processing and Applications*, pages 756–761, 2012.

[PPC12] Paolo Piccinini, Andrea Prati, and Rita Cucchiara. Real-time object detection and localization with sift-based clustering. *Image Vision Comput.*, 30(8):573–587, 2012.

[PWF09] Florin Alexandru Pavel, Zhiyong Wang, and David Dagan Feng. Reliable object recognition using SIFT features. In *2009 IEEE International Workshop on Multimedia Signal Processing, MMSP '09, Rio de Janeiro, Brazil, October 5-7, 2009*, pages 1–6. IEEE, 2009.

[RBS09] Keerthi Ram, Yogesh Babu, and Jayanthi Sivaswamy. Curvature orientation histogram s for detection and matching of vascular landmarks in retinal images. In *In Proceedings of SPIE-Medical Imaging*, 2009.

[RL09] Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 1509–1517. Curran Associates, Inc., 2009.

[RLD07] Javier Ruiz-del-Solar, Patricio Loncomilla, and Christ Devia. A new approach for fingerprint verification based on wide baseline matching using local interest points and descriptors. In *Advances in Image and Video Technology, Second Pacific Rim Symposium, PSIVT 2007, Santiago, Chile, December 17-19, 2007, Proceedings*, volume 4872 of *Lecture Notes in Computer Science*, pages 586–599. Springer, 2007.

[RM03] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 10–17. IEEE Computer Society, 2003.

[RO09] M. G. Ross and A. Oliva. Estimating perception of scene layout properties from global image features. *Journal of vision.*, 10, 2009.

[RPD10] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):105–119, 2010.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. ORB: an efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2564–2571. IEEE, 2011.

[SAS07a] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th International Conference on Multimedia 2007, Augsburg, Germany, September 24-29, 2007*, pages 357–360. ACM, 2007.

[SAS07b] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th International Conference on Multimedia 2007, Augsburg, Germany, September 24-29, 2007*, pages 357–360. ACM, 2007.

[SB97] Stephen M. Smith and J. Michael Brady. SUSAN - A new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[SBBF12] Christoph Strecha, Alexander M. Bronstein, Michael M. Bronstein, and Pascal Fua. Ldahash: Improved matching with smaller descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(1):66–78, 2012.

[SC92] Jun Shen and Serge Castan. An optimal linear operator for step edge detection. *CVGIP: Graphical Model and Image Processing*, 54(2):112–133, 1992.

[Sch01] Cordelia Schmid. Constructing models for content-based image retrieval. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 39–45. IEEE Computer Society, 2001.

[SF96] Eero P. Simoncelli and Hany Farid. Steerable wedge filters for local orientation analysis. *IEEE Transactions on Image Processing*, 5(9):1377–1382, 1996.

[SH08] Chanop Silpa-Anan and Richard Hartley. Optimised kd-trees for fast image descriptor matching. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008.

[SHK12] Michael Schaeferling, Ulrich Hornung, and Gundolf Kiefer. Object recognition and pose estimation on embedded hardware: Surf-based system designs accelerated by FPGA logic. *Int. J. Reconfig. Comp.*, 2012:368351:1–368351:16, 2012.

[sj14] scott jrig. Computer vision metrics: Survey, taxonomy, and analysis. Apress Berkely, CA, USA, 2014.

[SK87] Lawrence Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 4(3):519–524, 1987.

[SKFP09] Jan Svab, Tomas Krajnık, Jan Faigl, and Libor Preucil. Fpga based speeded up robust features. In *IEEE International Conference on Technologies for Practical Robot Applications, 2009, Boston, USA*, 2009.

[SLMS05] Edgar Seemann, Bastian Leibe, Krystian Mikolajczyk, and Bernt Schiele. An evaluation of local shape-based features for pedestrian detection. In *Proceedings of the British Machine Vision Conference 2005, Oxford, UK, September 2005*. British Machine Vision Association, 2005.

[SM97a] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):530–535, 1997.

[SM97b] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):530–535, 1997.

[SMB00] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[SZ02] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do I organize my holiday snaps?". In *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part I*, volume 2350 of *Lecture Notes in Computer Science*, pages 414–431. Springer, 2002.

[TBL10] Bart Thomee, Erwin M. Bakker, and Michael S. Lew. TOP-SURF: a visual words toolkit. In *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*, pages 1473–1476. ACM, 2010.

[TCFL13] Tomasz Trzcinski, C. Mario Christoudias, Pascal Fua, and Vincent Lepetit. Boosting binary keypoint descriptors. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 2874–2881. IEEE, 2013.

[TCGP09] Duy-Nguyen Ta, Wei-Chao Chen, Natasha Gelfand, and Kari Pulli. Surftrac: Efficient tracking and continuous object recognition using local feature descriptors. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 2937–2944. IEEE Computer Society, 2009.

[TFW08] Antonio Torralba, Robert Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008.

[TG00] Tinne Tuytelaars and Luc J. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of the British Machine Vision Conference 2000, BMVC 2000, Bristol, UK, 11-14 September 2000*, pages 1–14. British Machine Vision Association, 2000.

[TG04] Tinne Tuytelaars and Luc J. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[THL09] Nutao Tan, Lei Huang, and Changping Liu. Face detection using improved LBP under bayesian framework. In *16th IEEE International Conference on Image Processing ICIP, 2009*, pages 1237 – 1240. IEEE Computer Society, 2009.

[tHRH07] G.A. ten Holt, M.J.T. Reinders, and E.A. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In *In Thirteenth annual conference of the advanced school for computing and imaging*, 2007.

[TL12] Tomasz Trzcinski and Vincent Lepetit. Efficient discriminative projections for compact binary descriptors. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I*, volume 7572 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 2012.

[TLCT09] Feng Tang, Suk Hwan Lim, Nelson L. Chang, and Hai Tao. A novel feature descriptor invariant to complex brightness changes. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 2631–2638. IEEE Computer Society, 2009.

[TLF10] Engin Tola, Vincent Lepetit, and Pascal Fua. DAISY: an efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):815–830, 2010.

[TM07] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.

[TP91] Matthew Turk and Alex Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition*, pages 586–591. IEEE, 1991.

[TT07] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. volume 4778 of *Lecture Notes in Computer Science*, pages 168–182. Springer, 2007.

[TWY08] Yun-Ta Tsai, Quan Wang, and Suya You. CDIKP: A highly-compact local feature descriptor. In *19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA*, pages 1–4. IEEE Computer Society, 2008.

[vdSGS08] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008.

[vdWGB06] Joost van de Weijer, Theo Gevers, and Andrew D. Bagdanov. Boosting color saliency in image feature detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(1):150–156, 2006.

[VL07] Christoffer Valgren and Achim J. Lilienthal. Sift, SURF and seasons: Long-term outdoor localization using local features. In *Proceedings of the 3rd European Conference on Mobile Robots, EMCR 2007, September 19-21, 2007, Freiburg, Germany*, 2007.

[VMD15] Darshan Venkatrayappa, Philippe Montesinos, and Daniel Diep. Object matching in videos using rotational signal descriptor. In *Three-Dimensional Image Processing, Measurement (3DIPM), and Applications, San francisco,California, United States*, volume 9393. SPIE, 2015.

[VMDM15a] Darshan Venkatrayappa, Philippe Montesinos, Daniel Diep, and Baptiste Magnier. A novel image descriptor based on anisotropic filtering. In *Computer Analysis of Images and Patterns - 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015 Proceedings, Part I*, volume 9256 of *Lecture Notes in Computer Science*, pages 161–173. Springer, 2015.

[VMDM15b] Darshan Venkatrayappa, Philippe Montesinos, Daniel Diep, and Baptiste Magnier. RSD-DOG: A new image descriptor based on second order derivatives. In *Advanced Concepts for Intelligent Vision Systems - 16th International Conference, ACIVS 2015, Catania, Italy, October 26-29, 2015, Proceedings*, volume 9386, pages 23–34. Springer, 2015.

[VMDM15c] Darshan Venkatrayappa, Philippe Montesinos, Daniel Diep, and Baptiste Magnier. RSD-HoG: A new image descriptor. In *Image Analysis - 19th Scandinavian Conference, SCIA 2015, Copenhagen, Denmark, June 15-17, 2015. Proceedings*, volume 9127 of *Lecture Notes in Computer Science*, pages 400–409. Springer, 2015.

[VSMM14] Darshan Venkatrayappa, Désiré Sidibé, Fabrice Meriaudeau, and Philippe Montesinos. Adaptive feature selection for object tracking with particle filter. In *Image Analysis and Recognition - 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part II*, volume 8815 of *Lecture Notes in Computer Science*, pages 395–402. Springer, 2014.

[VvHR05] Manuela M. Veloso, Felix von Hundelshausen, and Paul E. Rybski. Learning visual object definitions by observing human activities. In *5th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2005, Tsukuba, Japan, December 5-7, 2005.*, pages 148–153. IEEE, 2005.

[WB07] Simon A. J. Winder and Matthew Brown. Learning local image descriptors. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society, 2007.

[WFW11] Zhenhua Wang, Bin Fan, and Fuchao Wu. Local intensity order pattern for feature description. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 603–610. IEEE, 2011.

[WHT08] Lior Wolf, Tal Hassner, and Yaniv Taigman. Y.: Descriptor based methods in the wild. In *In: Faces in Real-Life Images Workshop in ECCV. (2008) (b) Similarity Scores based on Background Samples*, 2008.

[WHY09] Xiaoyu Wang, Tony X. Han, and Shuicheng Yan. An HOG-LBP human detector with partial occlusion handling. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 32–39. IEEE, 2009.

[Wit83] Andrew P. Witkin. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence. Karlsruhe, FRG, August 1983*, pages 1019–1022. William Kaufmann, 1983.

[WL08] Shou-Der Wei and Shang-Hong Lai. Fast template matching based on normalized cross correlation with adaptive multilevel winner update. *IEEE Transactions on Image Processing*, 17(11):2227–2235, 2008.

[XTFZ14] Xianwei Xu, Lu Tian, Jianjiang Feng, and Jie Zhou. OSRI: A rotationally invariant binary descriptor. *IEEE Transactions on Image Processing*, 23(7):2983–2995, 2014.

[XTZ14] Lingxi Xie, Qi Tian, and Bo Zhang. Max-sift: Flipping invariant descriptors for web logo search. In *2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014*, pages 5716–5720. IEEE, 2014.

[YAR08] Chuohao Yeo, Parvez Ahammad, and Kannan Ramchandran. Rate-efficient visual correspondences using random projections. In *Proceedings of the International Conference on Image Processing, ICIP 2008, October 12-15, 2008, San Diego, California, USA*, pages 217–220. IEEE, 2008.

[YC03] Cheng-Hao Yao and Shu-Yuan Chen. Retrieval of translated, rotated and scaled color textures. *Pattern Recognition*, 36(4):913–929, 2003.

[YC07] Jian Yao and Wai-kuen Cham. Robust multi-view feature matching from multiple unordered views. *Pattern Recognition*, 40(11):3081–3099, 2007.

[YC12] Xin Yang and Kwang-Ting Cheng. LDB: an ultra-fast feature for scalable augmented reality on mobile devices. In *11th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2012, Atlanta, GA, USA, November 5-8, 2012*, pages 49–57. IEEE Computer Society, 2012.

[YCWQ14] Harry Yang, Shengnan Caih, Jingdong Wang, and Long Quan. Low-rank SIFT: an affine invariant feature for place recognition. In *2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014*, pages 5731–5735. IEEE, 2014.

[YDS01] Weichuan Yu, Konstantinos Daniilidis, and Gerald Sommer. Approximate orientation steerability based on angular gaussians. *IEEE Transactions on Image Processing*, 10(2):193–205, 2001.

[Yia93] Peter N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas.*, pages 311–321. ACM/SIAM, 1993.

[YKHP14] Juha Ylioinas, Juho Kannala, Abdenour Hadid, and Matti Pietikäinen. Learning local image descriptors using binary decision trees. In *IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, March 24-26, 2014*, pages 347–354. IEEE, 2014.

[YYQW11] Xiaoli Yuan, Jing Yu, Zengchang Qin, and Tao Wan. A sift-lbp image retrieval model based on bag-of-features. In *2011 IEEE International Conference on Image Processing*, pages 1061–1064. IEEE, 2011.

[ZBC10] Chao Zhu, Charles-Edmond Bichot, and Liming Chen. Multi-scale color local binary patterns for visual object classes recognition. In *20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*, pages 3065–3068. IEEE Computer Society, 2010.

[ZCCY10] Gangqiang Zhao, Ling Chen, Gencai Chen, and Junsong Yuan. KPB-SIFT: a compact local feature descriptor. In *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*, pages 1175–1178. ACM, 2010.

[ZCSS13] Yingxuan Zhu, Samuel Cheng, Vladimir Stankovic, and Lina Stankovic. Image registration using BP-SIFT. *J. Visual Communication and Image Representation*, 24(4):448–457, 2013.

[ZCX+07] Lun Zhang, Rufeng Chu, Shiming Xiang, ShengCai Liao, and Stan Z. Li. Face detection based on multi-block LBP representation. In *Advances in Biometrics, International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007, Proceedings*, volume 4642 of *Lecture Notes in Computer Science*, pages 11–18. Springer, 2007.

[ZGZL10] Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, and Jianzhuang Liu. Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor. *IEEE Transactions on Image Processing*, 19(2):533–544, 2010.

[Zit10a] C. Lawrence Zitnick. Binary coherent edge descriptors. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II*, volume 6312 of *Lecture Notes in Computer Science*, pages 170–182. Springer, 2010.

[Zit10b] C. Lawrence Zitnick. Binary coherent edge descriptors. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II*, volume 6312 of *Lecture Notes in Computer Science*, pages 170–182. Springer, 2010.

[ZK13] Sebastian Zambanini and Martin Kampel. A local image descriptor robust to illumination changes. In *Image Analysis, 18th Scandinavian Conference, SCIA 2013, Espoo, Finland, June 17-20, 2013. Proceedings*, volume 7944 of *Lecture Notes in Computer Science*, pages 11–21. Springer, 2013.

[ZN13] Wanlei Zhao and Chong-Wah Ngo. Flip-invariant SIFT for copy and object detection. *IEEE Transactions on Image Processing*, 22(3):980–991, 2013.

[ZP07] Guoying Zhao and Matti Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):915–928, 2007.

[ZR14] Jun Zhu and Mingwu Ren. Image mosaic method based on SIFT features of line segment. *Comp. Math. Methods in Medicine*, 2014:926312:1–926312:11, 2014.

[ZSG+05] Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang. Local gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, pages 786–791. IEEE Computer Society, 2005.

[ZTH+14] Shiliang Zhang, Qi Tian, Qingming Huang, Wen Gao, and Yong Rui. USB: ultrashort binary descriptor for fast visual matching and retrieval. *IEEE Transactions on Image Processing*, 23(8):3671–3683, 2014.

[ZTL+13] Shiliang Zhang, Qi Tian, Ke Lu, Qingming Huang, and Wen Gao. Edge-sift: Discriminative binary descriptor for scalable partial-duplicate mobile search. *IEEE Transactions on Image Processing*, 22(7):2889–2902, 2013.

[ZWY+13] Sheng Zhong, Jianhui Wang, Luxin Yan, Lie Kang, and Zhiguo Cao. A real-time embedded architecture for SIFT. *Journal of Systems Architecture - Embedded Systems Design*, 59(1):16–29, 2013.