HAL
open science

Robust image description with laplacian profile and radial Fourier transform

Evanthia Mavridou

To cite this version:

Evanthia Mavridou. Robust image description with laplacian profile and radial Fourier transform. Computer Vision and Pattern Recognition [cs.CV]. Université de Grenoble, 2014. English. NNT : 2014GRENM065 . tel-01555416

HAL Id: tel-01555416

https://theses.hal.science/tel-01555416v1

Submitted on 4 Jul 2017

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

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

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et Informatique**

Arrêté ministériel : 1 Octobre 2011

Présentée par

Evanthia MAVRIDOU

Thèse dirigée par **James L. CROWLEY**
et codirigée par **Augustin LUX**

préparée au sein **Laboratoire d' Informatique de Grenoble à l' INRIA Rhône-Alpes**
et de **Ecole Doctorale de Mathématiques, Sciences et Technologies de l'Information**

Robust Image Description with Laplacian Profile and Radial Fourier Transform

Thèse soutenue publiquement le **25 Novembre 2014**,
devant le jury composé de :

M. Edmond BOYER
Director of Research, INRIA Grenoble Rhône-Alpes, Président

M. Frederic JURIE
Professor, University of Caen, Rapporteur

M. Antonis A. ARGYROS
Professor, University of Crete, Rapporteur

Mme Yanxi Liu
Professor, Penn State University, Examinateur

M. James L. CROWLEY
Professor, Institut Polytechnique de Grenoble, Directeur de thèse

M. Augustin LUX
Professor, Institut Polytechnique de Grenoble, Co-Directeur de thèse

Evanthia Mavridou: *Robust Image Description with Laplacian Profile and Radial Fourier Transform,* © November 2014

Dedicated to all those people
who believed in me and supported me.
Thank you.

ABSTRACT

In this thesis we explore a new image description method composed of a multi-scale vector of Laplacians of Gaussians, the Laplacian Profile, and a Radial Fourier Transform. This method captures shape information with different proportions around a point in the image. A Gaussian pyramid of scaled images is used for the extraction of the descriptor vectors. The aim of this new method is to provide image description that can be suitable for diverse applications. Adjustability as well as low computational and memory needs are as important as robustness and discrimination power. We created a method with the ability to capture the image signal efficiently with descriptor vectors of particularly small length compared to the state of the art. Experiments show that despite its small vector length, the new descriptor shows reasonable robustness and discrimination power that are competitive to the state of the art performance.

We test our proposed image description method on three different visual tasks. The first task is keypoint matching for images that have undergone image transformations like rotation, scaling, blurring, JPEG compression, changes in viewpoint and changes in light. We show that against other methods from the state of the art, the proposed descriptor performs equivalently with a very small vector length. The second task is on pattern detection. We use the proposed descriptor to create two different Adaboost based detectors for people detection in images. Compared to a similar detector using Histograms of Oriented Gradients (HOG), the detectors with the proposed method show competitive performance using significantly smaller descriptor vectors. The last task is on reflection symmetry detection in real world images. We introduce a technique that exploits the proposed descriptor for detecting possible symmetry axes for the two reflecting parts of a mirror symmetric pattern. This technique introduces constraints and ideas of how to collect more efficiently the information that is important to identify reflection symmetry in images. With this task we show that the proposed descriptor can be generalized for more complicated applications. The set of the experiments confirms the qualities of the proposed method of being easily adjustable and requires relatively low computational and storage requirements while remaining robust and discriminative.

Keywords: multi-scale description, Gaussian pyramid, Laplacian Profile, Radial Fourier Transform, compact image description

RÉSUMÉ

L'objectif de cette thèse est l'étude d'un descripteur d'images adapté à une grande variété d'applications. Nous cherchons à obtenir un descripteur robuste et discriminant, facile à adapter et peu coûteux en calcul et en mémoire. Nous définissons un nouveau descripteur, composé de valeurs du Laplacien à différentes échelles et de valeurs d'une transformée de Fourier radiale, calculées à partir d'une pyramide Gaussienne. Ce descripteur capture une information de forme multi-échelle autour d'un point de l'image. L'expérimentation a montré que malgré une taille mémoire réduite les performances en robustesse et en pouvoir discriminant de ce descripteur sont à la heuteur de l'état de l'art.

Nous avons expérimenté ce descripteur avec trois types de tâches différentes. Le premier type de tâche est la mise en correspondance de points-clés avec des images transformées par rotation, changement d'échelle, floutage, codage JPEG, changement de point de vue, ou changement d'éclairage. Nous montrons que la performance de notre descripteur est au niveau des meilleurs descripteurs connus dans l'état de l'art. Le deuxième type de tâche est la détection de formes. Nous avons utilisé le descripteur pour la création de deux détecteurs de personnes, construits avec Adaboost. Comparé à un détecteur semblable construit avec des histogrammes de gradients (HOG) nos détecteurs sont très compétitifs tout en utilisant des descripteurs sensiblement plus compacts. Le dernier type de tâche est la détection de symétries de réflexion dans des images "du monde réel". Nous proposons une technique de détection d'axes potentiels de symétries en miroir. Avec cette tâche nous montrons que notre descripteur peut être genéralisé à des situations complexes. L'expérimentation montre que cette méthode est robuste et discriminante, tout en conservant un faible coût en calcul et en mémoire.

Mot-clés: description multi-échelle, pyramide Gaussienne, Profil Laplacien, Transformée de Fourier radial, description compacte d'image.

PUBLICATIONS

Related publications

[1] **E. Mavridou**, M.D. Hoang , J.L. Crowley, A. Lux. Scale Normalized Radial Fourier Transform as a Robust Image Descriptor. In ICPR, 2014.

[2] **E. Mavridou**, J.L. Crowley, A. Lux. Multiscale Shape Description with Laplacian Profile and Fourier Transform. In ICIAR, 2014.

The following publication is also related to the thesis subject, but it is not included as a part of the thesis:

[3] V. Jain, **E. Mavridou**, J.L. Crowley, A. Lux. Facial Expression Analysis and the Affect Space. In PRIA vol. 25, Journal "Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications." 2015.

ACKNOWLEDGMENTS

I would like to thank all the people who supported me during the three years of my PhD.

The first persons that I would like to thank are my supervisor Prof. James L. Crowley and my co-supervisor Prof. Augustin Lux for giving me the opportunity and all the means to perform my thesis in PRIMA team. Without them, this thesis would have never become true. I want to thank them for their guidance, their support, their motivation and sharing their great experience on the field. I would also like to gratefully thank Jim and Augustin for their care, encouragement and assistance in every trouble. I would also like to thank the members of my jury, DR Edmond Boyer, Prof. Frederic Jurie, Prof. Antonis A. Argyros and Prof. Yanxi Liu, for their interest in my work and accepting the task to monitor the defense of my thesis.

I would then like to thank from the bottom of my heart my teammates in PRIMA. Some people are in the team from the first moment I arrived, many passed just for a while, many were here at the beginning but left and many new came. Each one of them I will remember for their assistance to my work, welcoming manners and their friendship. I will always appreciate their effort to support me every time I asked for their help and for all their tips that made my life easier. I would like to offer my special thanks to Claudine Combe, for all her help and patience with me concerning her great work as an engineer that was a part of my research. I would also like to specially thank Lukas, Lucas, Sergi, Dominique, Amaury, Thierry, Catherine, Antoine, Julian, Varun and Dung for their assistance and collaboration. Moreover, being in a big team with lots of different subjects of expertise has a respective impact on someone's evaluation. You come across many things that you were missing from other fields and learn them first hand by people who know them. But their friendship is of no less importance to me than their help, motivation and support in my research. The ambiance in PRIMA was one of the major factors that kept me going during hard times. Some days, just the idea of seeing familiar faces with kind smiles on, is the good reason you need to drug you out of bed and get you back in front of your computer to keep fighting! Thank you all for reviewing my papers and posters, correcting my presentations, listening to my problems, giving me feedback, popping ideas, helping me with silly bugs in my code, helping me learn French, introducing me to LAN game sessions, inviting me in your homes to meet your families and friends, traveling together, having fun together and sharing your time with me. *Thank you all guys for everything! It has been quite a ride!*

During all this time in Grenoble, my people from back home were always with me. Despite being 2000 km away, my family and my friends in Thessaloniki were giving me strength and love by phone and by internet. My deepest appreciation goes to my family for making me the person that I am today,

writing this thesis and contributing to science. My parents, Stella and Panagiotis, are the reason I had everything I needed, material and spiritual, to reach all that I have reached in my life. I cannot be grateful enough for their love. Then, I would like to greatly thank my friends back in Greece for sharing my worries and my happy moments during these last three years. I would like to thank them particularly for their jokes that cheered me up when I needed it the most and for their delirious happy comments on my every little achievement. Chatting online with half a day difference in time, sharing constantly the experience of a conference in the other side of the world, is perhaps one of the most memorable things I will always have from them. It gave me a funny feeling like I was an astronaut reporting back to base on earth. *You are so far away, but I feel like you are always here!*

Finally, I would like to thank all my friends in Grenoble for keeping me company all this time. Lots of evenings, lots of trips, lots of good times to remember! And tones of photos to remind me!

I saved the last paragraph for Simon. I owe an important debt to Simon for being there for me every day, in the best and the worst. Having been a PhD himself, he gave me all the support I needed to keep up. Simon also gave me the home I missed from being so far away from Thessaloniki. With him I discovered the real France, the France of kind hospitable people with high values and great appreciation for genuine beauty and quality of life. And I want to thank Simon's family for opening their home to me and for being more than good to me. *Simon, thank you for being you and for being with me.*

Eva
Grenoble, September 30, 2014

CONTENTS

1 INTRODUCTION 5
1.1 The Technological Context of this study 5
1.2 The research problem investigated in this study 6
1.3 The experimental methods used for the investigation 7
1.4 Summary of this manuscript 8
2 IMAGE DESCRIPTION 15
2.1 What is an image descriptor? 15
2.2 Approaches to image description 16
2.2.1 Local vs global 16
2.2.2 Image intensity vs derivatives 16
2.2.3 Spectral methods 17
2.3 Objectives of image description 17
2.3.1 Invariance to Orientation 18
2.3.2 Invariance to Translation 19
2.3.3 Invariance to Scaling 19
2.3.4 Invariance to Viewpoint / Affine transformations 20
2.3.5 Invariance to Blur/ JPEG compression 21
2.3.6 Invariance to Illumination 21
2.3.7 Robustness 22
2.3.8 How to address these objectives? 22
2.4 A taxonomy of descriptors 23
2.4.1 Local descriptors 24
2.4.2 Global descriptors 35
2.4.3 Classification and taxonomy by application of existing methods. 42
2.5 A flexible new approach for a general purpose descriptor 43
3 LAPLACIAN PROFILE AND RADIAL FOURIER TRANSFORM FOR IMAGE DESCRIPTION 51
3.1 Conceiving a new method 51
3.2 Overall approach 51
3.3 Image pyramid 54
3.3.1 Gaussian pyramids vs Binomial pyramids 56
3.3.2 Analysis of the scale factor parameter 57
3.4 The Laplacian Profile vector 58
3.5 The Radial Fourier Transform 60
3.5.1 Radial Fourier on the Image pyramid or the Laplacian pyramid? 61
3.5.2 RDFT from sampling on a circle or on a disk? 63
3.5.3 Size of the RDFT sampling area 65
3.6 Conclusions: The LP-RDFT image descriptor 66
4 KEYPOINT MATCHING WITH LP-RDFT 71
4.1 Searching for the details that make the difference 71

4.2 Experiments . . . 73
4.2.1 Descriptors for comparison . . . 73
4.2.2 The LP-RDFT version for matching . . . 74
4.2.3 The procedure of comparison . . . 75
4.2.4 Textured images . . . 77
4.2.5 Textureless images . . . 77
4.2.6 How different is the LP-RDFT detector from the SIFT detector? . . . 83
4.3 Conclusions on local description . . . 84
5 PATTERN DETECTION WITH LP-RDFT 91
5.1 Detecting the visual pattern of people . . . 91
5.2 The Histogram of Oriented Gradients descriptor for people detection . . . 92
5.3 Deciding on a detector . . . 92
5.4 Experimental Method and Data . . . 93
5.4.1 Building the detection procedure . . . 93
5.4.2 Results on INRIA Person dataset . . . 94
5.4.3 Conclusions on Pattern Detection . . . 96
6 REFLECTION SYMMETRY DETECTION WITH LP-RDFT 101
6.1 Introduction to Reflection Symmetry . . . 101
6.1.1 Image description for Reflection Symmetry detection . . . 101
6.2 A new technique for Reflection Symmetry Detection . . . 103
6.2.1 Adjusting the LP-RDFT features . . . 103
6.2.2 The overall technique using LP-RDFT . . . 104
6.3 Experiments . . . 107
6.3.1 Validation framework . . . 107
6.3.2 The IEEE CVPR2013 Competition dataset . . . 109
6.3.3 Results . . . 109
6.4 Conclusions on Reflection Symmetry Detection . . . 114
7 CONCLUSIONS 119
7.1 Lessons learned . . . 120
7.2 Perspectives of this study . . . 122
7.3 Proposing future work . . . 122
7.4 Brainstorming in the aftermath . . . 123

BIBLIOGRAPHY 128

Figure 1 The tree of image description. This tree shows the distinction of descriptors that is made in this thesis in order to organize the state of the art. 23
Figure 2 The BRISK sampling pattern. The small blue circles denote the sampling locations and the red dashed circles have a radius relevant to the σ of the Gaussian kernel used for smoothing around the sampling locations. Image taken from [76]. 25
Figure 3 Illustration of the FREAK sampling pattern, which is similar to the retinal ganglion cells distribution and their corresponding receptive fields. Each circle represents a receptive field where the image is smoothed with a corresponding Gaussian kernel. Image taken from [133]. . . 25
Figure 4 Examples of SIFT descriptors on images created with the VLFeat library [135] implementation. Images taken from the INRIA Person dataset. The green square structure shows the grid of cells used for SIFT with the most important gradient orientations shown with small green arrows in the center of each cell. The yellow circle with the line in the middle of the green grid indicates the major orientation of the whole final descriptor computed from the green grid. 29
Figure 5 The NSD procedure for viewpoint and scale changes. **Left:** Viewpoint changes for long and thin foreground structures are difficult for grid descriptors due to the changes in the background. NSD selects the subset of supports that cover the foreground and have the correct scale to allow background variations. **Right:** Scale changes are problematic for non-invariant to scale grid descriptors due to changes in local support. NSD uses a subset of both large and small scale supports, ignoring intermediate scale supports that do not provide proper description. Image taken from [18]. 30
Figure 6 **Left:** Detected salient points for SURF. Hessian-based detectors find blobs. **Middle:** Haar wavelet types used for SURF. **Right:** SURF descriptors on an image at different scales and with the relevant orientations. Image taken from [9]. 34

Figure 7 Illustration of the way in which the HOG image description works for the human shape from the INRIA Person dataset. (a) An average gradient image over training examples. (b) Each square shows the maximum positive SVM weight in the block centered on the square. (c) The same for the negative SVM weights. (d) A test image. (e) The computed HOG descriptor on the test image. (f) The HOG descriptor weighted by the positive SVM weights. (g) The HOG descriptor weighted by the negative SVM weights. Image taken from [30]. 38
Figure 8 Taxonomy tree of the most substandard application per descriptor. The taxonomy first separates descriptors by the size of their support area and then by the theory they are based. It becomes clear that the support area of a descriptor depends on the size of the visual details that matter for an application and that certain theoretical approaches are preferred to address particular applications. 45
Figure 9 If two descriptor features are extracted only inside the small support areas in the yellow circles, the features will look very similar for the two salient locations. If the features use more information from the areas defined by the orange circles, the descriptor features will look different as the appearance of the two wider areas is much different. 52
Figure 10 A scale space represented by an image pyramid. 53
Figure 11 Example of a possible descriptor feature extracted with the proposed method on an image pyramid. The red dots represent the Laplacian of Gaussian values on the respective samples. The green dots represent the samples of a possible support area where the Fourier Transform can be computed. 54
Figure 12 Example of a possible descriptor vector of the proposed method projected on the lowest pyramid level that is used to extract the vector. The red dot represents the Laplacian of Gaussian value on the respective sample on this level. The green dots represent the samples of a possible support area where the Fourier Transform can be computed. The rings around the dots represent the corresponding regions on this level from the respective samples collected on higher scales. 55
Figure 13 Example of a possible descriptor feature extracted with the proposed method on an image pyramid. If we choose to extract the descriptor on different subsets of pyramid levels, we have a multi-scale descriptor that is scale invariant. 56

Figure 14 Original image for testing different Gaussian pyramid filtering versions. Its size is 400×300. 57

Figure 15 **Left:** Scaling images with an integer binomial filter that resembles a Gaussian filter of $\sigma = \sqrt{2}$. **Right:** Scaling images with a Gaussian filter of $\sigma = \sqrt{2}$. It is obvious that both filters work efficiently for the scaling of the image. **Above left and right:** Digital display of the first level of a Half-Octave Gaussian pyramid of image 14. We can see the borders of filtering around each level image. **Below left and right:** Under each display of the first level of the pyramid, there is a figure with keypoints found on it. Keypoints are found as Laplacian extrema and away from the borders (unfiltered area). These are the most stable keypoints kept after Non-Maximum Suppression. 58

Figure 16 **Left:** Scaling images with a Gaussian filter of $\sigma = \sqrt{2}$. **Right:** Scaling images with a Gaussian filter of $\sigma = 2$. The filter with the larger σ offers better smoothing but less keypoints. **Above left and right:** Digital display of the first level of a Half-Octave Gaussian pyramid of image 14. We can see the borders of filtering around each level image. **Below left and right:** Under each display of the first level of the pyramid, there is a figure with keypoints found on it. Keypoints are found as Laplacian extrema and away from the borders (unfiltered area). These are the most stable keypoints kept after Non-Maximum Suppression. 59

Figure 17 An example of a Laplacian Profile extracted from a continuous scale space. Laplacian values are collected in a vector at every scale σ. The higher the scale is, the larger the corresponding neighborhood described by the Laplacian value is on the original image. 60

Figure 18 Rotation experiment for the first image from each case folder at Affine Covariant Features dataset [91]. The RMSE of the one-level descriptors of sampling area at radius = 1 for the RDFT, using intensity values (blue line) and Laplacian values (red line). In both cases, we keep only the magnitude values from the RDFT coefficients to make sure we have rotation invariant features. It is shown that according rotation changes, the RDFT is better to be computed on intensity values. 62

Figure 19 The two tested RDFT sampling areas of the descriptor on one level. The central red dot in each pattern represents an LP element (as the location on a region of a pyramid level where this Laplacian of Gaussian was computed). The green surrounding dots represent the sampling areas for the RDFT around an LP element. The radius of the circle where they are sampled is the same for both sampling patterns, the circle and the disk. In this example, we collect eight samples on the periphery of the circle. 63

Figure 20 Rotation experiment for the first image from each case folder at Affine Covariant Features dataset [91]. The RMSE of the one-level descriptors with sampling area at radius = 5 for the RDFT, performing sampling on the periphery of the circle or in the entire disk area. In both cases, we kept only the magnitude values from the RDFT coefficients to make sure we have rotation invariant elements. It is shown that according rotation changes, the RDFT from samples on a disk can handle better the situation but it requires a lot more vector elements. 64

Figure 21 Two possible RDFT sampling areas of the descriptor on one level. The central red dot in each pattern represents an LP element (as the location on a region of a pyramid level where this Laplacian of Gaussian was computed). The green surrounding dots represent the sampling areas for the RDFT around the LP elements. **Left:** 4 neighbors are taken in linearly around an LP element for the 1D RDFT, representing the circle with the smallest possible radius. **Right:** A circle with wider radius is taken around an LP element for the 1D RDFT. 65

Figure 22 A simplified representation of the visual system for keypoint detection. The green node is the step of the system that the proposed descriptor is involved in. 71

Figure 23 Image transformation can make salient locations disappear or new to appear. **Above:** Rotation of the image with rotation center the center of the image (blue target in the center). The rotation of the image causes new salient locations to appear and other to disappear. **Below:** Scaling of the image. Scaling usually causes salient locations to disappear as high frequencies are successively removed or new to appear as large regions become small points. 72

Figure 24 Are all the above location matches valid? The size of the above yellow circles is relevant to the scale where the salient location was detected. Image transformation can cause salient location in different images to be found shift in the x and y directions and on different scales. The salient locations that seem the same to a human can be considered as different to a program. The decision if a match is good or not depends on the constraints of an application. 73
Figure 25 Affine Covariant Features dataset - Blur ("bikes"). Keypoint matching of image 1 to the rest. 78
Figure 26 Affine Covariant Features dataset - Blur ("trees"). Keypoint matching of image 1 to the rest. 78
Figure 27 Affine Covariant Features dataset - Viewpoint ("graf"). Keypoint matching of image 1 to the rest. 79
Figure 28 Affine Covariant Features dataset - Viewpoint ("wall"). Keypoint matching of image 1 to the rest. 79
Figure 29 Affine Covariant Features dataset - Zoom + rotation ("bark"). Keypoint matching of image 1 to the rest. 80
Figure 30 Affine Covariant Features dataset - Zoom + rotation ("boat"). Keypoint matching of image 1 to the rest. 80
Figure 31 Affine Covariant Features dataset - Light ("leuven"). Keypoint matching of image 1 to the rest. 81
Figure 32 Affine Covariant Features dataset - JPEG compression ("ubc"). Keypoint matching of image 1 to the rest. 81
Figure 33 Images from MIRFLICKR Retrieval Evaluation dataset with low texture information. 82
Figure 34 Rotation tests for the collected textureless images from MIRFLICKR dataset. 83
Figure 35 Scaling tests for the collected textureless images from MIRFLICKR dataset. 83
Figure 36 LP-RDFT keypoints. The keypoints are represented as yellow circles with their diameter relevant to the scale where they are detected. 85
Figure 37 SIFT keypoints. The keypoints are represented as yellow circles with their diameter relevant to the scale where they are detected. 85
Figure 38 A closer look to LP-RDFT keypoints. This is a part from the upper left side of figure 36. The keypoints detected on the Gaussian pyramid of LP-RDFT capture important intensity perturbations. Therefore, they are more often closer or on edges, corners or blobs. The keypoints are represented as yellow circles with their diameter relevant to the scale where they are detected. 86

Figure 39 A closer look to SIFT keypoints. This is a part from the upper left side of figure 37. The keypoints detected on the Gaussian pyramid of SIFT can capture either important or very small intensity perturbations. Therefore, keypoints can be found either closer or on edges, corners and blobs but also on areas that for humans look rather homogeneous. The keypoints are represented as yellow circles with their diameter relevant to the scale where they are detected. 86
Figure 40 This image is an example of the SIFT implementation from the VLFeat library. The green circles are the keypoint with their diameter being relevant to the scale where they were found and the green line in each circle indicates the orientation of the keypoint. SIFT can find and match keypoints that are hard for a human to identify, for example at the region of the image where there is the sky. Image taken from [135] 87
Figure 41 A visual system for pattern detection. The green nodes are the steps of the system where LP-RDFt is involved. . 91
Figure 42 Some of the test images from INRIA Person dataset. As it can be seen by the example images, a person's pattern, meaning the shape of a person, can be very diverse and also very "mixed"with the background. These facts make this detection problem hard to solve. 92
Figure 43 ROC for the techniques. Detection rate refers to true positive detections and False positive rate refers to false positive. 95
Figure 44 Groundtruth examples of reflection symmetry from the IEEE CVPR2013 Competition [82]. The blue line segments are the reflection symmetry axes that separate the symmetric patterns into the two parts, the one part being the reflection of the other. 101
Figure 45 A simplified representation of the visual system for reflection symmetry detection. The green nodes are the steps of the system that the proposed technique is involved in. 103
Figure 46 The creation of an LP-RDFT feature vector. The final descriptor vector consists of the LP, magnitude information from the RDFT around each LP element and one phase element. The vector is normalized with the L_2-norm to improve robustness to illumination except the phase element at the end of the vector. 104

Figure 47 Precision-recall curves for images with a single symmetry axis. For the highest possible recall that the new technique reaches, it outperforms the baseline in both recall and precision. For images containing one symmetry axis, the proposed technique reaches higher recall values with better precision that the baseline algorithm. 110
Figure 48 Detected axes in images containing a single symmetry axis. **Above:** the new technique. **Below:** the baseline algorithm. We keep at most three detections per image with the highest confidence score. For the baseline algorithm, the symmetric matches that indicate a axis are given with dots of the same color. We use blue rectangles to indicate the support area (symmetric matches) for each axis. 110
Figure 49 Precision-recall curves for images with multiple symmetry axes. For the highest possible recall that the new technique reaches, it outperforms the baseline in both recall and precision. For images containing multiple symmetry axes, the baseline algorithm reaches better recall values but with very low precision. 111
Figure 50 Detected axes in images containing multiple reflection symmetry axes. **Above:** the new technique. **Below:** the baseline algorithm. We keep at most 5 detections per image with the highest confidence score. For the baseline algorithm, the symmetric matches that indicate a axis are given with dots of the same color. We use blue rectangles to indicate the support area (symmetric matches) for each axis. 111
Figure 51 Examples of matches for working on images with a single symmetry axis. **Left:** groundtruth axes (blue line segments). **Right:** symmetric keypoint matches found and the most voted axes (magenta line segments). Each match is shown as a couple of yellow circles representing the keypoints matched, with the radius indicating the scale where the keypoint was detected and a green line segment that connects the two keypoints. In the third case, we can see that the proposed algorithm considers symmetry differently from a human. 112

Figure 52 Examples of matches working on images with multiple symmetry axes. **Left:** groundtruth axes (blue line segments). **Right:** symmetric keypoint matches found and the most voted axes (magenta line segments). Each match is shown as a couple of yellow circles representing the keypoints matched, with the radius indicating the scale where the keypoint was detected and a green line segment that connects the two keypoints. In the first, second and fourth case, we can see that the proposed algorithm considers symmetry differently from a human. 113

LIST OF TABLES

Table 1 Classification table for existing image descriptors. The classification is made according to the theory they are based and the qualities of the descriptors. The descriptors are presented in alphabetical order. Each one can be found on the previous sections by consulting the first column "SUPPORT AREA" (Local or Global) and then the group of columns "APPROACH" (the type of theoretical approach that the descriptor is based on.) 44

LIST OF ALGORITHMS

1 The generalized procedure of extracting a local LP-RDFT descriptor vector on a selected location in an image. 67

2 The followed matching procedure between two images. The procedure is the same for LP-RDFT and all the competing descriptors. 75

3 The followed procedure for detecting a pattern in an image. The procedure is the same for both LP-RDFT and HOG. 97

4 The proposed procedure for reflection symmetry axes detection in an image with LP-RDFT. The procedure for the baseline algorithm requires the flipping of the image and the recomputation of descriptors in the new image. The matching of symmetric pairs of keypoints for the baseline happens between the two images. LP-RDFT does not require flipping of the image because it allows the descriptor vectors to be independent of orientation. A separate phase element kept for each descriptor vector can later be used in the RSM formula to measure symmetry between matched keypoints. 108

LP Laplacian Profile

RDFT Radial Discrete Fourier Transform

LP-RDFT Laplacian Profile and Radial Discrete Fourier Transform descriptor

A-KAZE Accelerated KAZE

ASIFT Affine-SIFT

BOLD Bunch Of Lines Descriptor

BRIEF Binary Robust Independent Elementary Features descriptor

BRISK Binary Robust Invariant Scalable Keypoints descriptor

CARD Compact And Real-time Descriptors

CBP Centralized Binary Patterns

CENTRIST CENsus TRansform hISTogram descriptor

CHOG Compressed Histogram of Oriented Gradients

CMD Covariance Matrix Descriptors

CSIFT Color-SIFT

CS-LBP Center-Symmetric LBP

CS-LTP Center-Symmetric LTP

DSIFT Dense SIFT

FAsT-Match Fast Affine Template Matching descriptor

FP False positive

FREAK Fast Retina Keypoint descriptor

FTS Fuzzy Texture Spectrum descriptor

GaFour Gabor / Fourier descriptor

GEF Gabor Energy filters

GF-HOG Gradient Field HOG

GIH Geodesic-Intensity Histogram descriptor

GLAC Gradient Local Auto-Correlations descriptor

GLOH Gradient Location and Orientation Histogram descriptor

G-RIF Generalized-Robust Invariant Features descriptor

KAZE KAZE descriptor (the word means *wind* in Japanese)

LARK Locally Adaptive Regression descriptor

LBP Local Binary Patterns

LESH Local Energy based Shape Histogram descriptor

LHS Local Higher-order Statistics descriptor

LINE LINEarizing the Memory for Parallelization descriptor

LIOP Local Intensity Order Patterns

LTP Local Ternary Patterns

MAR Mobile Augmented Reality

MROGH Multisupport Region Order-Based Gradient Histogram descriptor

MRRID Multisupport Region Rotation and Intensity Monotonic Invariant Descriptor

MSLD Mean-Standard deviation Line Descriptor

NSD Nested Shape Descriptors

ORB Oriented (keypoints) and Rotated BRIEF descriptor

OSID Ordinal Spatial Intensity Distribution descriptor

PCA Principal Component Analysis

PD Patch Duplets descriptor

PHOG Pyramid HOG descriptor

PHOW Pyramidal Histogram of Visual Words descriptor

PVEP Peak Valley Edge Patterns

RMSE Root Mean Square Error

RSM Reflection Symmetry Magnitude

RIFF Rotation-Invariant Fast Features descriptor

RIFT Rotation Invariant Feature Transform descriptor

SID Scale Invariant Descriptor

SIFT Scale-Invariant Feature Transform descriptor

SIFT-GC SIFT using Global Context

SILT Scale-Invariant Line Transform descriptor

SURF Speeded Up Robust Features descriptor

SYMD SYMmetry Descriptor

SYM-FISH SYMmetric-aware Flip Invariant Sketch Histogram descriptor

TP True positive

WLD Weber Local Descriptor

INTRODUCTION

Nous présentons un nouveau descripteur d'images qui est fondé sur des fonctions mathématiques invariantes aux transformations affines de l'image, et qui fournit une description des signaux efficace et très compacte. Ce descripteur combine deux éléments : premièrement le Profil Laplacien; deuxièmement, nous utilisons la Transformée de Fourier qui est connue et largement utilisée dans différents domaines scientifiques pour ses propriétés de capture de l'information de signal.

Nous effectuons des expériences sur la reconnaissance des points d'intérêt dans les images qui ont subi des changements. Dans une seconde série d'expériences, nous testons la méthode proposée sur la reconnaissance de formes par la recherche de piétons dans des images. Les expériences sont ensuite étendues à une tâche plus complexe qui est la détection de symétrie de réflexion en images (réflexion d'un modèle perpendiculairement à un axe). Les résultats dans tous les cas montrent que la méthode proposée présente des performances compétitives à l'état de l'art.

Avec notre méthode nous poursuivons deux objectifs principaux. D'abord nous voulons encoder les données de signal de manière efficace afin de fournir les informations nécessaires. Puis, nous voulons fournir une description aussi compacte que possible, afin d'être utilisée pour des tâches coûteuses ou des machines à faible puissance de calcul. Nous voulons répondre à ces objectifs en utilisant une base théorique unique qui est suffisamment flexible pour être adaptée à différentes applications, mais sans changer le type de données de signaux capturées. De cette façon, l'information calculée peut être partagée entre les différentes applications.

1 INTRODUCTION

1.1 THE TECHNOLOGICAL CONTEXT OF THIS STUDY

The search for a perfect image descriptor has been the "Holy Grail" since earliest days of Computer Vision. Many methods and variations have been proposed. While many of these have proved effective for particular tasks, no image description has been found to provide the most effective description for all tasks.

The three classic types of feature structures used for image description are edges, blobs and corners. Edges are defined as boundaries where changes in the pixel intensity take place [20, 11, 67, 38, 144, 127]. They consist of a set of pixels that form a line or a curve. The more abrupt the change in the pixel intensity, the more well defined the edge. Although shapes maybe detected as assemblies of edges, it is difficult to detect edges when the intensity changes are smooth. Blobs, on the other hand, overcome this particular limitation of edges.

A blob is a group of pixels that form a simple shape with smooth surface in the image. The classic method for blob detection is to first threshold the pixel intensities. Pixels on one side of the threshold are considered as the background and the groups of pixels on the otherside are considered as the blobs. Blob detection has been further extended with the use of more discriminative features to overcome the limitations of simple thresholding of binary values [90, 93, 88]. This method works fine when the background and the objects in the scene have homogeneous surfaces with small intensity variations. But for images with many intensity variations, blobs can be less effective than edges.

In comparison to edges and blobs, corners are very small locations in the image where the intensity has important variations compared to their surrounding pixel neighborhood. Corners can be defined with a position as small as a pixel. Corners are widely used for the detection of salient locations for matching images with similar content [48, 121, 6]. While all these approaches can be efficient for image description, their usability is limited and their advantages are complementary.

Color is a very important image cue and has played an important role in vision research. Different color spaces can be found in the literature, each one having advantages for particular applications [125]. Some of the most commonly used color spaces in image description are the RGB (red-green-blue) and the HSV (hue-saturation-value) color spaces. Color values can be used for either probabilistic image description [44, 130, 57, 74] or for deterministic description techniques [38, 67, 16, 137, 132, 15]. The main limitations for the use of color cues for image description are that color is sensitive to changes in illumination and that similar things do not necessarily have the same color.

Therefore, color based image description can be highly efficient for applications where color details remain stable, but in general this approach can have important limitations.

A variety of local features have been used to define image descriptors. Gaussian derivatives [39, 94] are such an approach that has significant advantages as a description method. They can be also used to provide the basis for more complicated description methods [84, 30, 9]. Haar wavelets [107] can be used to create techniques that require modest computational time and provide efficient description. Gabor filters have also been shown to capture texture in different orientations and resolutions [55].

The Fourier Transform has been widely used for image description in several ways, either as a descriptor itself or as a part of the theory of a descriptor [70, 87]. The main advantage of the Fourier Transform is that it can provide fine frequency information in a very compact way.

The common property of these methods is that they all have the capacity to represent the image signal in a sufficiently distinctive manner to achieve efficient description. The problem is that each one of them has been proven successful in particular applications and according to particular criteria (for example, amount of computational time), but none has been found to be generally the best. The large diversity of methods often leads to the need for different calculations on the same signal to extract different representations of the same information for different stages of processing.

The objective of this work is to introduce a new description method based on invariant operations that can be appropriate for a large variety of visual tasks. This method meet two criteria. Therefore this method can be adapted is to provide a relatively compact descriptor for tasks on machines with limited memory. Although, it can be used to encode a larger descriptor for tasks that require a more discriminative descriptor. These criteria are met using a single theoretical approach that is flexible enough to be customized to different applications. Without changing the type of captured signal details, computed information can then be shared between different visual tasks.

1.2 THE RESEARCH PROBLEM INVESTIGATED IN THIS STUDY

In this study we investigate a general purpose image descriptor based on the combination of a Laplacian Profile with a Radial Fourier Transform. The Laplacian Profile is a vector of Laplacian of Gaussian values collected over a range of scales. Gaussian derivatives are widely used for their ability to capture shapes in images [20, 78, 84, 116]. The Laplacian of Gaussian has the advantage of being a rotation invariant mathematical operation. Therefore, the Laplacian Profile can provide multi-scale rotation invariant description of a signal.

The Fourier Transform is widely used as a descriptor for signals. Its coefficients can be expressed as frequency magnitude and phase. For a Radial Fourier Transform taken on a disk or a circle, the phase of frequencies is an indication of the the orientation of the image intensity in a pixel neighborhood [147]. On the other hand, the magnitude of frequencies is invariant to

orientation. The magnitude and phase of frequencies can be used respectively to create an orientation invariant or a very discriminative image description method.

The idea behind the proposed descriptor is to capture and express the appearance of an image neighborhood with a small feature vector that is robust to image changes. The trick is that the small size of a descriptor vector useful for fast computations but this must not affect the discrimination power of the descriptor significantly compared to the state of the art.

Though discriminative power has always been an important criterion for image description, computational and memory requirements can be equally important for applications. Our contribution is an easily calculated, robust and very compact descriptor based on the Laplacian of Gaussian and the Fourier Transform computed on a logarithmic scale space. The proposed method captures shape in two different ways, 1) as changes in intensity due to the Laplacian of Gaussians and 2) as low frequencies around them due to the Fourier Transform.

1.3 THE EXPERIMENTAL METHODS USED FOR THE INVESTIGATION

In order to evaluate the proposed description method we investigate its performance for a diverse set of applications. We experimentally compared variations of the new descriptor with established image descriptors on three visual tasks.

The first task evaluates the capability of the proposed method to provide good image description for small image neighborhoods. The appropriate task to do this is keypoint matching between images. Keypoint matching measures the capacity of a descriptor to identify the same locations in the image scene after changes in the image plane. The testing images for our experimentations are taken from the Affine Covariant Features benchmark dataset and the MIR-FLICKR Retrieval Evaluation dataset. We use a set of descriptors of different theoretical approaches to compare their performance with the proposed descriptor. These are SIFT [84], SURF [9], ORB [112], BRISK [76], FREAK [133], BRIEF [19] and NSD [18]. This experimental frame work is a standard manner in the literature to simply compare different descriptors.

The results on keypoint matching reveal that the proposed descriptor has the capacity to identify correctly the same locations in different images. Compared to the performance of the other descriptors used for these experiments, the proposed descriptor works very well with scaling changes in images and can overcome low resolution problems. Though, in other cases its performance is less competitive than the state of art. As seen by the performance of the other descriptors, in each case there is a different descriptor that works the best or the worst. Nonetheless, an important observation concerning our descriptor is that compared to the state of the art it has a statistically significant small vector size. The conclusion from keypoint matching is that the proposed descriptor is effective though not always the most efficient.

The next task is pedestrian detection. This task is selected in order to prove that the descriptor can be used to identify patterns and structures in images. The dataset for this task is the well known INRIA Person dataset and the competing descriptor is the also well known HOG descriptor [30] that was developed on this dataset. The capacity of HOG to describe structures and shapes have established it as the baseline for this type of visual task.

The results on pedestrian detection showed that our proposed descriptor can provide similar detection rates with HOG. Although it has more false detections than HOG, the proposed descriptor vectors are about 8 times smaller than the HOG descriptor vectors. The results attest to the fact that the proposed descriptor is enough discriminative to be used for describing large image neighborhoods containing patterns with acceptable performance but in a significantly more compact manner.

The last task for the evaluation is reflection symmetry detection in images. This is a hard visual task that requires particular information from the image signal. The major problem in this task is that the definition of symmetry in an image can be very different between a human and a computer. Computer algorithms may consider symmetry in different orientations than humans or find symmetrical patterns that humans do not consider significant. On the contrary, as humans have the capacity to be more abstract in the way they conceive things, they can identify symmetry where algorithms cannot. In addition, the existence of symmetry in an image can be also disputable between humans. Therefore, the evaluation of symmetry in images cannot be as perfectly objective as for example compared to tasks such as the detection of humans in an image.

The dataset and experimental framework for this task are adapted from the Symmetry Detection from Real World Images Competition 2013 in IEEE CVPR2013. The results on reflection symmetry detection are modest compared to the baseline algorithm provided by the contest. The results from this task show that the proposed descriptor has the capacity to carry the required information and integrate it within an appropriate technique that can provide legitimate results on this highly demanding task. While having less good performance than the baseline algorithm, the proposed technique can offer a simpler solution with smaller descriptor vectors.

1.4 SUMMARY OF THIS MANUSCRIPT

Chapter 2 is a review of the existing methods of image description. We give an exact explanation of the meaning of image description and we exhibit its objectives. The objectives of image description can be organized in eight main types: invariance to translation, invariance to scaling, invariance to rotation, invariance to viewpoint / affine transformations, invariance to blurring / image compression (especially JPEG), invariance to illumination changes, resistance to clutter / partial occlusion and robustness. We distinguish between local and global description based on the size of the area that a method describes compared to the size of the image. We name local those descriptors that are

capable to describe efficiently a small region of the image independently from the rest of the image. Global descriptors are those descriptors that are computed using a large window or the whole image in order to give sufficient information to perform a visual task. We further classify existing methods according to their theoretical approach. The descriptors are separated into four groups according to the theory on which they are based: intensity based descriptors, gradient / first order partial derivative based descriptors, Laplacian / second order partial derivative based descriptors, spectrum based descriptors. For each descriptor found in the literature, there is a short summary of how it works.

The purpose of this chapter is to investigate all the extent of the state of the art, sort the different approaches, study their characteristics and search for gaps in the state of the art where there is still room for research. At the end, we provide a classification table and a taxonomy tree with the existing methods. In these two schemes, the methods are attributed by their particular properties and the applications they are suitable for. This regularization of the state of the art helps us to set goals and leads the way to the proposition of the new image description method.

Chapter 3 introduces the new method. In this chapter we show how the Laplacian of Gaussian can be collected on a logarithmic scale space and be organized into a vector in order to provide invariant image description and how its discrimination power can be enhanced by the Fourier theory.

The Gaussian pyramid can be computed with the Half-Octave Gaussian pyramid algorithm [29]. The Laplacian of Gaussian values can be easily collected as differences of adjacent levels on the created pyramid. Collecting a set of Laplacians of Gaussians on corresponding coordinates in several scales provides us with a multi-scale vector that captures shape information on increasingly larger proportions when projected on the original image. This vector is named the Laplacian Profile and it is the spine of the proposed description method.

The Fourier Transform can be calculated radially around each of the Laplacian of Gaussians collected for a Laplacian Profile. There are different possibilities for designing this descriptor concerning the sampling area for the Fourier Transform and the formula used for its computation. The sampling area can be a circle of samples or a disk of samples around a selected Laplacian of Gaussian. On a circle of samples, we can compute an 1D Radial Fourier Transform, considering the periphery of the circle as a linear sequence of samples. On a disk area, we can compute a 2D Radial Fourier Transform after transferring the disk from the Cartesian to polar coordinates. In general, sampling a disk of samples provides more discriminative descriptors but sampling on a circle offers shorter computational time. The parameters of the possible designs are discussed in order to see the advantages and the disadvantages for each manner. Eventually, a number of variable parameters is given for the design of the proposed method in order to be tested in the experimental part of this thesis.

Chapter 4 describes experiments of keypoint matching. The comparison framework focuses on testing several descriptors using textured and textureless images. The proposed method performs similar to the state of the art,

showing comparable results in some cases and better in others. It is shown that scaling changes in the images are better handled by the proposed descriptor than any other type of image change.

The memory needs of a description method is an important matter that is also addressed. We find two ways in the literature that are usually followed to deal with memory consumption. It is either attempted to create very small descriptor vectors or convert a descriptor to binary [76, 112, 19, 18]. Our idea is that any descriptor can be binarized afterwards by exploiting a suitable binarization formula. Consequently, we work on developing a new description method that is able to encode visual information correctly while creating small descriptor vectors.

We test our proposed method on the Affine Covariant Features benchmark dataset and a set of textureless images that we collect from the MIRFLICKR Retrieval Evaluation dataset. A small portion of the large set of image descriptors is chosen for comparing and evaluating our proposed method. The literature indicates the descriptors SIFT [84], SURF [9], ORB [112], BRISK [76], FREAK [133], BRIEF [19] and NSD [18]. These descriptors are chosen because they are either well established and widely used or very new. The results show that our proposed method can compete with the rest of the descriptors with vector length almost 5 times smaller than that of SIFT or even with a very tiny vector of only 7 elements long for textureless images.

In chapter 5, we employ a shorter version of the proposed descriptor with an Adaboost classifier in order to produce a detector for standing and walking people (pedestrians). We use the well known INRIA Person dataset containing images with pedestrians for our experiments and compare to a detector made with the HOG descriptor [30]. The results show that our proposed descriptor can perform similarly to HOG and be more than 8 times smaller in vector length.

We test two versions of our descriptor, both introduced in chapter 3, and discuss on the results. The first version of the proposed descriptor uses samples from the Gaussian pyramid (intensity values) for the computation of the Radial Fourier Transform. The second version uses Laplacian values on samples from the Gaussian pyramid for the computation of the Radial Fourier Transform. The second version aims to increase the discrimination power of the descriptor by using Laplacian values, as explained in chapter 3. Both versions have the same detection rate as HOG but find more false positives. Considering that the vector length of both versions of the proposed descriptor is significantly smaller than the vector length of HOG, they are very suitable for applications for which the detection rate is more important than the accuracy while the computational and memory costs are limited.

In chapter 6, we use the proposed description method to detect reflection symmetry in images. Symmetry is mostly defined by the shapes existing in an image. The shapes that involve reflection symmetry have specific characteristics that distinguish them from repeating patterns. Reflecting shapes have to be sufficiently close and in the same time not overlapping while being the one the mirror image of the other. A technique for reflection symmetry detection has to be capable to distinguish between two patters/ shapes that are

really symmetrical or simply similar. As the proposed method is able to capture shape and independently the orientation, it is suitable for experimenting for this visual task.

We use the proposed descriptor to introduce a new technique for reflection symmetry detection. We provide formulas and constraints that fit to the characteristics of the proposed descriptor and test it on the dataset provided for reflection symmetry by the Symmetry Detection from Real World Images Competition 2013 in IEEE CVPR2013. The performance comparison within this framework showed interesting results.

Chapter 7 concludes with the best choices of parameters and discusses for future development and perspectives of this method. Advices on the best parameters come from the results and the observations of the experimentations in chapters 4, 5 and 6. Furthermore, based on general observations and the capabilities of the Laplacian of Gaussian and the Fourier theory, we give guidelines for further research on the proposed descriptor. The chapter ends with a personal point of view towards research on computer vision.

The proposed descriptor provides acceptable results in all the experiments in this thesis. Although its performance is not the best among the sate of the art, the proposed descriptor works for all tasks. The ability to perform reasonably in all the experimental tasks is an attestation that a general purpose descriptor can exist, though as expected its performance is modest.

DESCRIPTION DE L'IMAGE

Dans ce chapitre, nous passons en revue, en la structurant, la grande variété de descripteurs d'images qui peuvent être trouvés dans l'état de l'art. Nous discutons les approches principales pour la description d'images et leurs objectifs. En fonction de ces approches et objectifs, nous produisons un tableau de classification et une taxinomie sous la forme d'un arbre pour l'ensemble des descripteurs analysés. Nous utilisons la table et l'arbre pour tirer des conclusions sur les forces et les faiblesses des différentes méthodes. En fonction de cette conclusion nous enonçons les objectifs de notre travail.

2 IMAGE DESCRIPTION

2.1 WHAT IS AN IMAGE DESCRIPTOR?

Image description, as a part of computer vision, offers the methods and tools to describe image signals, extracting information according to a purpose. Research in the field started in the 60's, at a time when available computational power was measured in kilobytes and Kilocycles per second. Early image descriptors were limited to thresholding or simple additions and subtractions of pixel values due to the low computational power of computers at that time.

Image description methods may work with 2D images or may be generalized to more dimensions to represent such things as depth, color or time. In this thesis, we will concentrate on methods that work on 2D images.

Descriptors

Image descriptors describe properties of images, image regions or individual image locations. These properties are typically called "features". Most image descriptors are composed of vectors of real valued features, although features may be binary, categorical, ordinal, integer-valued or real-valued. Image descriptors can describe a specific position in an image, a region, or a population of positions. Image descriptors that describe populations of positions are said to be statistical or probabilistic. Feature values can correspond to values of individual pixels, properties of regions such as size or orientation or statistical properties such as the frequency of occurrence of colors in a region. Probabilistic methods may use Principal Component Analysis, Gaussian Mixture Models, covariance matrices or other statistics [130, 106, 119] to describe a collection of pixels or images. The method presented in this thesis extracts individual numerical vectors with real valued features on specified locations in an image.

Methods can be developed for general use for specific applications, such as object recognition, or they can be developed for particular tasks, such as character recognition. In order to use a method to a different kind of applications, the method usually requires important modifications.

In this chapter we review the large variety of image descriptors that can be found in the state of the art. Section 2.2 discusses the different major approaches to image description. Section 2.3 introduces common objectives for image description. Section 2.4 reviews existing approaches, following a generalized classification scheme. This leads to a table containing the reviewed methods and a taxonomy in the format of a tree. From this we draw conclusions on what can be achieved and what is missing in the state of the art.

In section 2.5 at the end of this chapter, we propose a new method that combines two well known theories, the Laplacian of the image signal and the Fourier Transform. This new method is experimentally compared to the state of the art for three visual tasks in the following chapters.

2.2 APPROACHES TO IMAGE DESCRIPTION

2.2.1 Local vs global

Local description methods describe a specified neighborhood of an image or on a small salient region in the image to create a feature vector. Global methods use most or all of the pixels of an image to produce a feature vector. In some cases local features that are weekly discriminant can be collected over an entire image to create a strongly discriminant global feature. Global descriptors are usually used as parts of a system that involves learning and detection of patterns.

Local methods work on defined local neighborhoods around specific locations in the image, the size of which varies according to the method. The specified locations can be selected according to a saliency measurement. In this case, the locations are known as *keypoints* or *points of interest* or *salient points*, if they are only pairs of coordinates for single pixels, or *salient regions*, when pixel neighborhoods are indicated. The specified locations though may be alternatively determined using a-priori information.

Keypoints or salient regions are determined by detectors. It is useful to combine descriptors with detectors that share a similar feature space. For example, a detector that indicates edges using image gradients can be combined with a descriptor based on gradients. The detector and the descriptor can share the same Gaussian pyramid (explained in subsection 2.3.3) and the computed gradients, resulting in less computational cost and time [84]. However, for many applications the most effective detector does not necessarily rely on the same feature space as the most effective descriptor.

Difference of Gaussians (DoG) is a popular method that can be easily computed on image pyramids and widely used for multi-scale descriptors [26]. The FAST (Features from Accelerated Segment Test) detector is a more recent but also well know corner detector [111]. Other detectors, such as Harris detector [48] or Canny detector [20], search for corners or edges. Salient regions can be detected using the Harris-affine, the Hessian and the Hessian-affine region detectors [90], the Harris-Laplace and Hessian-Laplace affine region detectors [93] or the MSER (Maximally Stable Extremal Regions) detector [88]. Several other methods have been proposed [129, 62, 114] and the research in this area is very active. Detection of keypoints and salient regions in images is an important problem of computer vision on its own, so we will not explore further.

2.2.2 Image intensity vs derivatives

Raw pixel intensities descriptors can be used to construct fast and inexpensive image descriptors and often used for real time applications. The main representative of this class in the Local Binary Patterns (LBP) descriptor [101]. LBP has extended the approach of using intensity comparison tests into a simple scheme of image description using a binary comparison test for pixel

intensities. Several descriptors have derived after this approach using intensity comparison tests for either local or global image description [51, 123, 139]. Intensity based descriptors tend to be more insensitive to illumination changes in images and less expensive compared to derivative based descriptors.

The gradient of the image is also very commonly used for local and global image description. Edges and corners are region boundaries and well captured with gradients. Also, the orientation of the gradient can provide a powerful feature for discrimination. A few intensity based descriptors use gradients to introduce orientation in their vectors [76, 133]. The Laplacian is less commonly used despite its capacity to capture changes in the image signal in part because of its sensitivity to noise. The first and second order partial derivatives in each of the two dimensions x and y are very rarely used. The two major representatives for this class are the Scale-Invariant Feature Transform descriptor (SIFT) [84] and the Histograms of Oriented Gradients descriptor (HOG) [30]. It should be mentioned that the majority of local descriptors in the literature use gradients. Derivative based descriptors tend to be invariant and more discriminant though more expensive than intensity based descriptors.

2.2.3 Spectral methods

Spectral methods describe the Fourier frequencies of the signal. This approach is commonly used for global descriptors as it makes it possible to collect important image content information and ignore details (high frequencies). Global descriptors that use spectra have very low computational cost and are highly efficient for fast classification of very large datasets.

2.3 OBJECTIVES OF IMAGE DESCRIPTION

We want image description methods that can decrypt visual signals but work fast and be computationally inexpensive. The information that can be provided to a visual system by an image descriptor is constrained by physical and computational limitations. Even if the scene is constant, images may contain minor variations due to rounding, digitization and photons noise. To be reliable, an image descriptor must accommodate such noise.

Discrimination power

Discrimination is the most common use of descriptors. Discrimination refers to the ability to chose. Some methods sacrifice efficiency for effective discrimination. Unfortunately, the most effective methods often sacrifice invariance for discrimination. There are appropriate learning techniques that can be combined with low discriminative methods in order to boost discrimination such as Adaboost [41], cascades of linear classifiers [138], Bag of Words (BoW). Nonthe-less, discriminality is the major objective for image description methods.

Invariance

We can distinguish two qualities that descriptors should have, invariance and robustness. Invariance for a description method is the property to remain constant when the image undergoes some transformation. A descriptor may be mathematically invariant to a parameter, or it may be made invariant by normalization. In the first case, the descriptor is based on a function that is in-

sensitive to changes. In the second case, orientation is estimated as a characteristic value and the descriptor vector is normalized with that value. Invariance can be distinguished from equivariance. Equivariance, or covariance, is the ability of a method to follow a change in an image and provide a description that retains the same structure but "shifted" in some parameter along with the change. Robustness is the ability to tolerate change. Invariance and equivariance are typically the result of mathematical properties of the descriptor while robustness can often be achieved by algorithmic methods. In the rest of this section, invariance towards usual image changes, resistance to partial occlusion and robustness are discussed in order to be able to annotate methods in the next section.

Equivariance

Robustness

2.3.1 Invariance to Orientation

Invariance to orientation provides an example of the possible approaches to invariance. A descriptor can be mathematically invariant to orientation or it may be made invariant by normalization to an estimated orientation. For example, the Laplacian of Gaussian (LoG) at location (x, y) on an image, $\nabla^2 G_\sigma(x, y)$, is a function that does not involve orientation:

$$\nabla^2 G_\sigma(x, y) = \frac{\partial^2 G_\sigma(x, y)}{\partial x^2} + \frac{\partial^2 G_\sigma(x, y)}{\partial y^2} = \frac{x^2 + y^2 - \sigma^2}{\sigma^4} \exp \frac{-(x^2 + y^2)}{2\sigma^2}. \quad (1)$$

Therefore, LoG is mathematically invariant to orientation.

An alternative is to normalize the descriptor to a local estimate of orientation. The signal orientation in a pixel neighborhood is usually considered as the direction from pixels with low intensities towards pixels with high intensities. The image gradient at location (x,y) on an image, $\nabla(x, y)$, is defined as:

$$\vec{\nabla}(x, y) = \left[\frac{\partial(x,y)}{\partial x}, \frac{\partial(x,y)}{\partial y} \right]. \quad (2)$$

The orientation of the gradient at (x, y) can be estimated as:

$$\theta = \tan^{-1}(\Delta y \backslash \Delta x). \quad (3)$$

The estimated orientation can be used to normalize the gradient to the dominant orientation

$$\nabla(x, y) = \Delta x \cos(\theta) + \Delta y \sin(\theta). \quad (4)$$

The image gradient needs to be normalized with θ in order to provide description that is invariant to orientation.

There are several ways to estimate orientation and use this estimate to "rotate" the descriptor. An alternative to equation 4 is to translate the image signal to polar or log-polar coordinates. [70, 77, 6]. Another way followed by some binary descriptors is shifting the vector elements or counting the vector element transitions between possible values [101].

Rotation between two images can be expressed with a transformation matrix R:

$$R_{AtoB} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where α is the angle of rotation, A is the source image and B is the destination (rotated) image. Changes by rotation can significantly affect the quality of the image signal. Because images are represented by a discrete grid, rotation changes cause the resampling of the pixels. The resampling can cause the alteration of the image pixel values. Rotations on exactly one, two or three quarters of the circle are usually the easiest case to handle due to direct translation of the image pixels' position on the discrete grid of the image. Interpolation is required for the other rotation angles. To overcome rotation changes, a description method must capture the appearance of a rotated pixel neighborhood without being affected by the rotated coordinates of the pixels or their interpolated values.

2.3.2 Invariance to Translation

Translation is the change of position of the image pixels. This image transformation can be expressed with a transformation matrix T:

$$T_{AtoB} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where dx and dy are the change in the two image directions respectively, A is the source image and B is the destination (translated) image.

A descriptor may be mathematically invariant to translation. For example, the histogram of image intensities is not affected by translation as long as it is taken within the limits of the image. For other descriptors the position on the image is important. For example, local descriptors cannot use the same salient locations after the image signal is translated. The salient locations need to be normalized by the translation parameters dx and dy.

Translation changes are the most easily handled type of image changes with current techniques. Given a small image neighborhood, translation can be seen as simply changing the image position of this neighborhood without any influence to the pixel values. Descriptors can overcome the change in position of points in the scene by working only on salient image positions and try to match the same salient positions in different images, or by using a window scanning method on every position of the image. Translation invariance is coming assumed for most techniques.

2.3.3 Invariance to Scaling

Scale changes result from the change of the distance between the camera and the scene or the objects in the scene. Unlike rotation and translation, there is

no obvious image descriptor that is mathematically invariant to scale. Scale change can expressed with a transformation matrix S:

$$S_{AtoB} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where s_x and s_y is the scaling factors in the two directions of the image respectively, A is the source image and B is the destination (scaled) image.

The scaling of an image to larger sizes stretches the image as well as adds more details (high frequency information). Scaling to smaller sizes shrinks the image and removes details. The challenge of handling scale changes is to recognize the same region in the scene when it is presented with a different amount of details in two images. A common approach to accommodate scale change in information is to remove small details from the image signal and retain more salient structures. By targeting salient structures in different scales, the problem is reduced to matching coarse structure.

Scale invariant descriptors often use pyramidal image structures [28] to represent a scale space and extract their features. Scaling is directly relevant to smoothing/ blurring as high frequencies are lost in both cases, therefore scale invariant descriptors usually perform against smoothing/ blurring. As small artifacts can be created in the image from bad scaling, Gaussian filtering is commonly prefered for scaling as its smoothness avoids artifacts and allows efficient interpolation.

Pyramids are an effective structure when the descriptors use derivatives, because pyramids allow fast computation of derivatives as simple differences of samples on adjacent levels. The preferred scale space is the logarithmic because the ratio of scaling between two successive images remains constant. The loss of a constant amount of information offers a more stable description of the same point in a scene in different scales. The property of collecting and allowing the matching of features in different image scales is named scale invariance.

2.3.4 Invariance to Viewpoint / Affine transformations

Affine transformation can combine scale, rotation and translation transformations. The affine transformation between two images can be expressed with a transformation matrix A:

$$A_{AtoB} = \begin{bmatrix} s_x \times \cos(\alpha) & \sin(\alpha) & dx \\ -\sin(\alpha) & s_y \times \cos(\alpha) & dy \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where α is the angle of rotation, dx and dy are the change in the two image directions respectively, s is a scaling factor, A is the source image and B is the destination (rotated) image.

This is a hard case of image changes as the set of pixel values of a certain transformed neighborhood can differ significantly from the original. The problem remains hard even after keeping only the most salient structures in an image as they also have changed appearance and look different. The descriptors have to overcome both the difference in pixel values and the change in the geometry of shapes in the image.

There is no feature that known to be mathematically invariant to affine transformations. An effective way to create affine invariant descriptors is to iterate through possible templates (possible transformated images in regard to a source image) and choose the template that fits the best to the transformed image [96, 73]. The drawback with affine customized descriptors is that they have to take into account several image views/ templates and are therefore computationally more expensive. A simple translation, scale and rotation invariant descriptor with low computational cost can be an efficent low-cost alternative instead of a fully affine invariant descriptor.

2.3.5 Invariance to Blur/ JPEG compression

Invariance to low resolution from blurring or image compression is relevant to image scaling to smaller sizes. The similarity is that, in all three cases, high frequencies are lost. The difference is that the images do not shrink after blurring or JPEG compression, which makes the problem a sense a little easier than scaling. As JPEG images are extremely usual and one of the most widely used test datasets include this test case [91], it is common in the literature to refer to it. The major problem of JPEG compression is the artifacts that are created in the image due to the algorithm and cause false visual information to appear. The similarity of blurring and image compression to scaling allows scale invariant descriptors to be also well invariant for these two changes.

2.3.6 Invariance to Illumination

Illumination refers to photometric changes between images. This type of image change causes the pixel intensity value to vary between darker and lighter tones or even change color. For example, dark grey can look like light grey with higher illumination, but in another case higher illumination could make brown appear as yellow. While color is a strong discriminative cue, illumination changes can cause important misinterpretations. The two ways to deal with illumination changes is to use features that are invariant to illumination changes or to normalize the descriptor values to a selected space of possible values in order to remove large variations.

Intensity based descriptors with ordinal or circular binning tend to be by default invariant to illumination as they measure differences of intensity than actual values [124]. Intensity based descriptors work most of the time better against illumination changes than those based on derivatives or spectra.

Normalization for illumination changes is usually done at the final descriptor vector with the L_1 or L_2-norm of the vector [30, 70]. Other ways to in-

duce illumination invariance are to prepare the image before description with gamma correction to enhance the local dynamic range of the image, high-pass filtering to remove the influence of overall intensity gradients and contrast equalization [123, 30].

2.3.7 Robustness

The manner robustness is induced in descriptors differs from that of invariance. It is the capacity to handle successfully small amounts of the above image changes as well as added noise. Robustness can be expressed in different ways depending on the method's usability. For a descriptor that describes small pixel neighborhoods in order to find the same in another image, robustness means to tolerate small viewpoint changes or noise [18]. For a method that creates object templates, robustness means to be able to recognize the objects even if they are partially occluded or their overlay or shape have changed [102]. Robustness cannot replace invariance but it surely rises the performance. The advantage of robustness against invariance is that it allows smaller descriptor vectors. While invariance searches for precise characteristics and needs long descriptor vectors to well contain them, robustness aims to the most overall appearance of the signal and to aims to encode it less strictly. Inducing robustness is a better choice than invariance when the size of descriptor vectors is an important matter.

Robustness is very important for partial occlusions often found in cluttered scenes. Occlusions appear when shapes/ objects in the image scene overlap. Also, an object can be partially occluded if it is not fully inside the image scene. In order to overcome this problem, description methods have to be able to describe shapes with as less information as possible while keeping information for their structure. On way to deal with occlusions is creating descriptors that are based on template matching schemes or encode in some other way geometrical relationships among the edges/ curves/ corners of shapes [144, 140, 73, 127]. The other way is to create robust descriptors that are not dramatically affected by small occlusions [11, 30].

2.3.8 How to address these objectives?

A useful lesson taken from this section is that we can assure invariance to certain image transformation mathematically and avoid additional normalization for them. For an efficient and inexpensive descriptor especially, this approach is very meaningful. Though, not all types of invariance can be addressed simultaneously by one theoretical approach. It is wise to select a theoretical approach or to combine theoretical approaches that address one type of invariance. For example, we can create a descriptor that is mathematically invariant to rotation by using LoG. Afterwards, a descriptor can be further normalized to become robust to other image transformations. For example, the descriptor with LoG can be made invariant to illumination by normalizing the vector with the L_2-norm. A good manner to create an efficient descriptor is to address

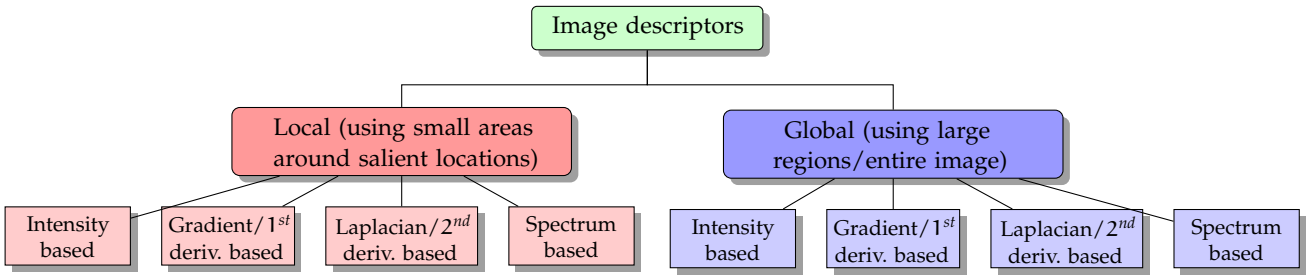

Figure 1: The tree of image description. This tree shows the distinction of descriptors that is made in this thesis in order to organize the state of the art.

mathematically objectives that are costly to achieve otherwise and use normalizing for those objectives that are less expensive or can be only achieved this way.

2.4 A TAXONOMY OF DESCRIPTORS

In this section we review existing descriptors by classifying them in subcategories according to the extent of the area they exploit, the type of the mathematical basis they use and their capabilities as well as their applications, as illustrated in figure 1. We use the term feature vector in order to refer to the final descriptor vector that a method produces. The feature vector can have thousands of elements or just a few elements depending on the used method. First, we briefly visit the methods to learn how they work. At the beginning of each subsection that contain many descriptors, we present first those descriptors that are important for the following chapters of this thesis where the experiments are presented. Next, we set up a classification table on descriptor characteristics and a taxonomy tree with the usability of theoretical approaches towards substandard vision applications and the relevant derived descriptors. The revision of existing methods lead us to the proposition of a new approach.

The descriptors presented in this chapter include well established descriptors that still compete in the state of the art as well as recently proposed descriptors. Other description methods exist, but it is unnecessary to compare every one of them, as many provide performance characteristics that are inferior to the state of the art. The following classification aims to show the state of the art that is valuable for future research in the field.

The first step for reviewing the descriptors separates descriptors in local and global. The separation is made regarding their support area, as defined by the size of the neighborhood that these methods exploit in order to create meaningful descriptor vectors.

Classification scheme for the existing descriptors.

The second step for the classification of the reviewed methods concerns their theoretical bases. As we show, the same mathematical bases can be used for both local and global description methods but within different schemes. We set four different descriptor classes at this point: a) intensity based methods that use sums, differences or statistics of pixel values, b) gradient or 1st order (partial) derivative based methods that exploit the changes in the image signal, c) Laplacian or 2nd order (partial) derivative based methods that exploit the

extrema in the image signal and d) spectrum based methods that rely on the frequencies of an image. The approach used in a method directly depends on the type of information that an application requests from the method.

For the large variety of descriptor presented in this chapter, we select a representative sample of them and use them in chapters 4, 5 and 6 for the experimental comparison with the proposed descriptor. These selected descriptors are indicated in the next subsections by the side-comments that refer to the chapter in which they are used for the experiments.

The outcome of the descriptors' classification is the creation of families of descriptors sharing similar qualities and scopes. The grouping of description methods can be made in many ways. For the aims of this work, the grouping of descriptors have been made with the classification table for descriptor characteristics and the taxonomy tree by substandard applications.

2.4.1 Local descriptors

2.4.1.1 *Intensity based local descriptors*

BRISK is used in the experiments of chapter 4.

BRISK. Binary Robust Invariant Scalable Keypoints descriptor (BRISK) [76] uses keypoints on a scale space and creates binary feature vectors by concatenating the results of simple brightness comparison tests. The scale-space pyramid layers of BRISK consist of n octaves c_i and n intra-octaves d_i, for $i = 0, 1, ..., n-1$ and typically $n = 4$. The octaves are formed by progressively half-sampling the original image. Samples for the tests are taken on N locations equally spaced on concentric circles around a keypoint. Gaussian smoothing with progressively larger σ, relevant to the distance between the points on the respective circle, is applied on each concentric circle. The sampling pattern is applied rotated according to the estimated orientation of the keypoint. Figure 2 shows the resampling pattern of BRISK. Though BRISK is intensity based, it uses gradients to compute orientation.

FREAK is used in the experiments of chapter 4.

FREAK. Fast Retina Keypoint descriptor (FREAK) [133] computes a cascade of binary strings with the use of local binary tests on image intensities at keypoints that are extracted on multiple scales by a multiscale detector. Taken a pair P of two compared samples, the binary test $T(P)$ is defined as:

$$T(P) = \begin{cases} 1, & \text{if } (I(P_1) - I(P_2) > 0; \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $I(P_x)$ is the intensity of a sample and P_1 and P_2 are the two samples of the pair P. A subset of 512 binary tests are kept from all possible local binary tests in order to be used in the final descriptor vector. The sampling around keypoints is performed in a circular pattern that mimics the topology of the human retina, i.e. concentric rings smoothened with Gaussian filters with σ relevant to their distance from the center. Orientation is computed on selected pairs of samples. Figure 3 shows the resampling pattern of BRISK. FREAK is intensity based binary descriptor but uses gradients to compute orientation.

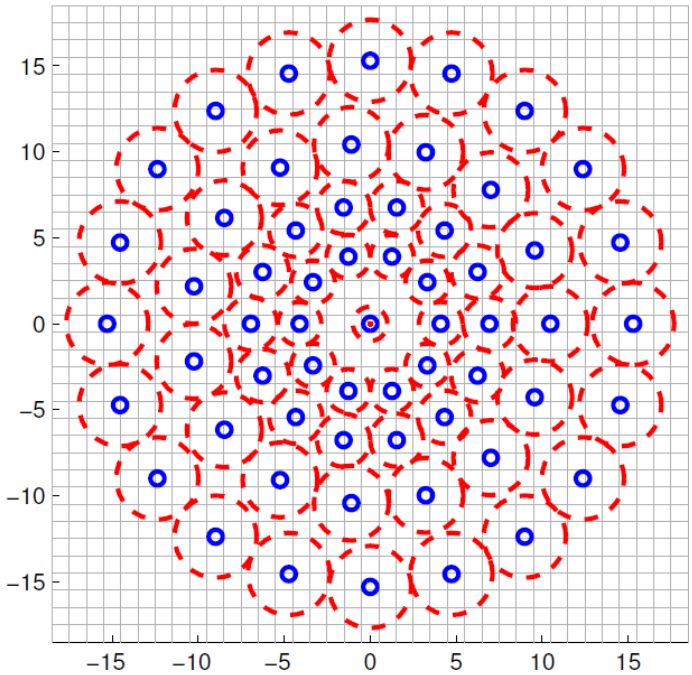

Figure 2: The BRISK sampling pattern. The small blue circles denote the sampling locations and the red dashed circles have a radius relevant to the σ of the Gaussian kernel used for smoothing around the sampling locations. Image taken from [76].

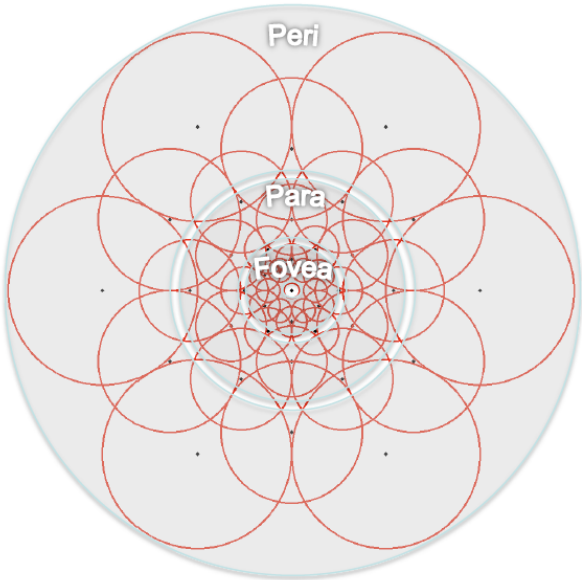

Figure 3: Illustration of the FREAK sampling pattern, which is similar to the retinal ganglion cells distribution and their corresponding receptive fields. Each circle represents a receptive field where the image is smoothed with a corresponding Gaussian kernel. Image taken from [133].

BRIEF is used in the experiments of chapter 4.

BRIEF. Binary Robust Independent Elementary Features descriptor (BRIEF) [19] is a fast binary descriptor that works on the basis of a relatively small number of pairwise intensity comparisons. A patch p is firstly defined and smoothed. Then, pairwise tests are performed on pixel intensities at specified locations. A binary test $T(p)$ for BRIEF on patch p of size $S \times S$ is defined as:

$$T(p; x, y) = \begin{cases} 1, & \text{if } (p(x) < p(y); \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $p(x)$ is the pixel intensity in a smoothed version of p at $x = (u, v)^T$. Choosing a set of n_d (x, y)-location pairs of pixels uniquely defines a set of binary tests. The resulting BRIEF feature vector consists of either 128, 256 or 512 bits that can be stored using 16, 32 or 64 bytes, each bit encoding the result of one of the pairwise tests. This descriptor is an example of compact binary image description.

ORB is used in the experiments of chapter 4.

ORB. The rotation invariant version of BRIEF is the Oriented (keypoints) and Rotated BRIEF descriptor (ORB) [112]. ORB uses a measure named intensity centroid which assumes that the intensity of a keypoint is an offset from its center, and this vector may be used to compute an orientation θ. A patch is defined around a keypoint and pairwise tests are performed on pixel intensities at specified locations, in the same way as for BRIEF. Then, the BRIEF features are normalized using the estimated orientation of the keypoint.This is done by defining for a set of n binary tests at location (x_i, y_i), a $2 \times n$ matrix S. Using the orientation θ, S is steered and becomes S_θ. So the BRIEF operations are turned into steered BRIEF operations for the ORB as:

$$g_n(p, \theta) := f_n(p) | (x_i, y_i) \in S_\theta \tag{11}$$

ORB uses a variation of FAST [111] keypoints that respect orientation and are calculated in multiple scales. The dimensionality of ORB is the same as for BRIEF.

CS-LBP and CS-LTP. Center-Symmetric LBP (CS-LBP) [51] are a version of LBP (Local Binary Patterns, presented in section 2.4.2.1) for keypoint description. This method actually combines the desirable properties of the SIFT and LBP into one description method. In detail, the LBP is modified such that around a central pixel, neighboring pixels that are opposite to each other are compared in order to generate a binary vector (center-symmetric pairs comparison). A patch is defined around a detected keypoint and it is divided into cells. For each cell, a CS-LBP histogram is built, using the CS-LBP features collected on the pixels of the cell. The final descriptor vector is the combination of the cell histograms. Similarly to CS-LBP, LTP (section 2.4.2.1) can be integrated to Center-Symmetric LTP (CS-LTP) and further to Histogram of Relative Intensities CS-LTP (HRI-CSLTP) [46] for keypoint matching.

Ferns. Ferns are sets [106] of binary tests on pixel intensities defined on neighborhoods around specified pixel locations. The locations of the tests in the neighborhood are arbitrary so the feature vector represents the structure of the neighborhood it describes.

GIH. Geodesic-Intensity Histogram descriptor (GIH) [79] is a deformation invariant binary descriptor. GIH uses the geodesic distance measure to collect samples around a keypoint. Then the intensity values of the samples is used to create a histogram which remains unaffected by important affine changes of the image signal.

LIOP. Local Intensity Order Patterns (LIOP) [139] are another version of LBP (section 2.4.2.1) for local description. Affine salient regions are collected with an affine covariant region detector and these are normalized to become disk regions. Disks are divided into subregions (ordinal bins) based on intensity orders. For each pixel in a subregion, a LIOP feature is constructed by measuring intensities of pixels sampled on a circle. To obtain rotation invariance, the surrounding pixels are sampled on the circle starting from the direction defined from the center of the region towards the used central pixel. Each LIOP is weighted to achieve robustness to noise. The collected intensities around the central pixel are sorted into a small vector. This small sorted vector is given an index according to a pre-defined index table. The index number corresponds to a binary string. All LIOP features of a subregion are first accumulated together, using the index table, into a subregion histogram. The histograms of the subregions are concatenated to the final region descriptor.

MRRID. Multisupport Region Rotation and Intensity Monotonic Invariant Descriptor (MRRID) [32] divides a given normalized support region into several rings and computes local intensity features at sample points in each ring. The samples are collected on a local xy coordinate system defined by a given keypoint and one of the samples. The same method can be used with gradient values instead of intensities and it is then known as Multisupport Region Order-Based Gradient Histogram (MROGH), presented in section 2.4.1.2. The configuration of MRRID feature vectors allows them to be binary by default.

OSID. Ordinal Spatial Intensity Distribution descriptor (OSID) [124] constructs a 2D histogram where the pixel intensities are binned in the ordinal space as well as in the spatial space. OSID is fully invariant to illumination changes.

Self-similarity descriptor. The Self-similarity descriptor [120] creates a correlation surface around a keypoint by correlating a small pixel neighborhood with a larger one, both centered on the keypoint. The correlation surface is then transformed into a binned log-polar representation to give a vector of 182 elements. These features are then attributed with geometrical relationships in order to create global ensembles of local descriptors. Every ensemble is an object template.

SILT. Scale-Invariant Line Transform descriptor (SILT) [65] detects line segments on a scale space.Each line segment is encoded as differences of boxes using integral images [138] over a surrounding local neighborhood. Principal Component Analysis (PCA) is further used for dimensionality reduction. The line matching is performed by the Manhattan distance.

Spin Images. A spin image is a descriptor feature vector for surface matching in 3D images [59]. It creates 3D oriented points on a polygonal mesh with vertices, using the positions and surface normals of those vertices. The position and the normal of each vertex define a local cylindrical coordinate system. This

local coordinate system is used to express the surrounding surface neighborhood into a 2D binned map using a defined projection function. Spin images were introduced for 3D object recognition by 3D keypoint matching (vertices positions). The parameters of this descriptor are adjustable to the needs of different applications. The matching of spin images is performed by computing the correlation coefficient between two spin images.

SYMD. This descriptor is developed for extracting local features from images of architectural scenes. Its name is given by the exploitation of local symmetries in the image which are considered as salient for matching, hence SYMmetry Descriptor (SYMD) [49]. A measure of local symmetry is used, based on analyzing image differences across symmetry axes. This measure is densely computed on different scales on an image and image patches are scored according the quantity of horizontal, vertical, and rotational symmetry computed. The patches with the highest scores, which contain patterns with some kind of symmetry, are collected as good candidates for matching. The measure for scoring the patches can be computed using either pixel intensities or gradients. For each collected patch, symmetry is measured based on pixel intensities on a grid of 20 angular cells and 4 radial cells resulting in a 240-dimensional descriptor vector.

2.4.1.2 *Gradient or first order partial derivative based local descriptors*

SIFT is used in the experiments of chapters 4 and 6.

SIFT. The Scale-Invariant Feature Transform descriptor (SIFT) [84] uses local histograms of the orientation of image derivatives over a grid of small windows to provide image description on a logarithmic scale space. The logarithmic scale space of SIFT is created by a Gaussian pyramid algorithm. In the original paper of SIFT, this pyramid is created by Gaussian filtering using $\sigma = \sqrt{2}$ and a resampling step of 1.5 samples in each direction. Salient points are collected as extrema using the DOG method on the pyramid. After locating a keypoint on a particular scale (a level on the pyramid), the local gradients are extracted. One or more orientations are assigned to the keypoint based on these gradients, which will be used to make the extracted descriptor invariant to rotation. All future operations are made relative to the assigned keypoint orientation(s). A Gaussian smoothening takes place in order to give priority to the gradients that are closer to the keypoint. Then, the gradient orientations are computed. A local histogram of the orientation of the local neighborhood can be built by the weighted orientations using 36 bins. The 36 bins cover the 360 degree range of rotations. The main orientation(s) is the peak in this histogram of local orientations. Then, with the same precomputed gradients and their orientations, a grid of local orientation histograms can be created around the keypoint. This grid of histograms is the descriptor vector. The histograms of orientations in the grid are "rotated" to match the main orientation found before. This is done by simply subtracting the main orientation. For a keypoint with more than one main orientations, we can have one different descriptor for each of the keypoint's main orientations. This first algorithm of SIFT uses two adjacent scales that are different by one octave to compute feature vectors of

160 elements. In the first scale it computes orientation histograms on a 4×4 grid and in the next scale on a a 2×2 grid. Considering 8 bin orientation histograms, the final descriptor vector has $(8 \times 4 \times 4) + (8 \times 2 \times 2) = 160$ elements.

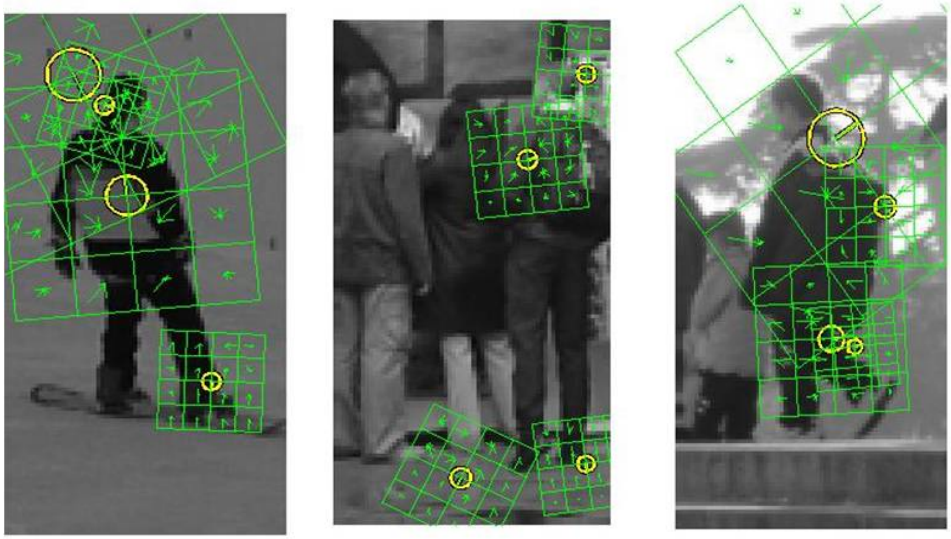

Figure 4: Examples of SIFT descriptors on images created with the VLFeat library [135] implementation. Images taken from the INRIA Person dataset. The green square structure shows the grid of cells used for SIFT with the most important gradient orientations shown with small green arrows in the center of each cell. The yellow circle with the line in the middle of the green grid indicates the major orientation of the whole final descriptor computed from the green grid.

In the more recent version of SIFT [83], the Gaussian pyramid is created using octaves of scales, each octave ending with the doubling of the scale factor σ, but with a number of more intermediate scales in an octave. The number of interval scales in an octave is variable. The image is rescaled only after a full octave. The created pyramid is denser and allows more scales to be examined. Again, the main orientation is the peak in a histogram of local gradient orientations. This SIFT algorithm computes a descriptor on only one scale and works efficiently with 4×4 grids of cells, with each cell having 8 orientation bins. Therefore, it gives $4 \times 4 \times 8 = 128$ elements per descriptor vector. This particular SIFT algorithm is very efficient and has been widely used. Figure 4 shows examples of detected SIFT keypoints in an image surrounded by the grid of bins.

SIFT, though old, is still competing in the state of the art today. The theory of SIFT with its numerous variations has dominated the field due to its high performance. In [91] is shown that SIFT and SIFT-like descriptors performed the best for both recall and repeatability.

NSD. Nested Shape Descriptors (NSD) [18] are constructed by pooling oriented gradients over a geometric structure of successively larger nested circles. This nested correlation structure resemble overlapping Hawaiian earrings

NSD is used in the experiments of chapter 4.

around a keypoint and enables a robust local distance function called the nesting distance. The construction of NSD can be easily made with an image pyramid. NSD using the nesting distance metric can select the best of the nested circles (supports), so the best support area. Figure 5 is a demonstration of the manner NSD features are computed around keypoints in images. NSD can be

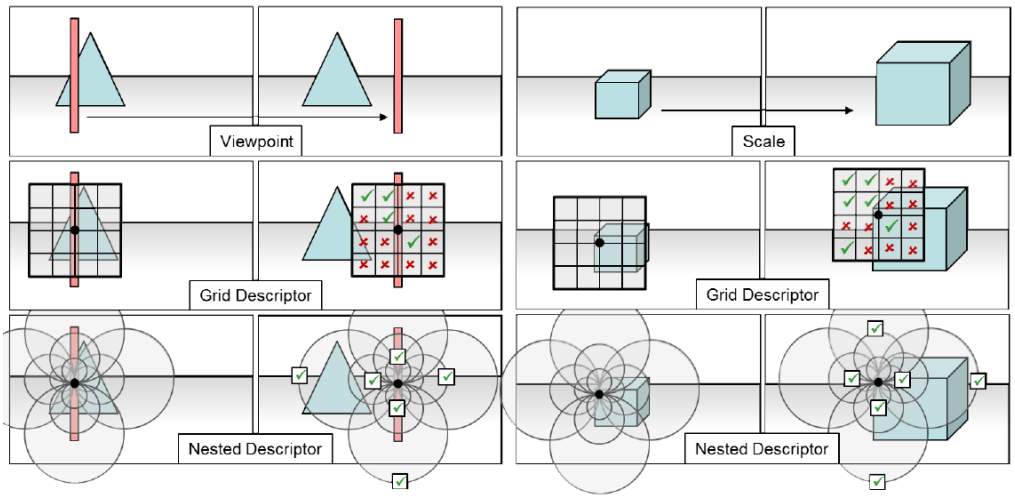

Figure 5: The NSD procedure for viewpoint and scale changes. **Left:** Viewpoint changes for long and thin foreground structures are difficult for grid descriptors due to the changes in the background. NSD selects the subset of supports that cover the foreground and have the correct scale to allow background variations. **Right:** Scale changes are problematic for non-invariant to scale grid descriptors due to changes in local support. NSD uses a subset of both large and small scale supports, ignoring intermediate scale supports that do not provide proper description. Image taken from [18].

binarized by computing the sign of the oriented gradients computed in the defined support area. The function D for the binarization of NSD is:

$$D(i,j,k) = \begin{cases} 1, & \text{if} d(i,j,k) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $d(i,j,k)$ is the descriptor for orientation i, lobe of Hawaiian earring j and lobe scale k. As the original nested structure is too dense, NSD can be created considering a substructure named the seed of life (SOL). SOL is generally formed using Hawaiian earrings with n-fold rotational symmetry. The advantage of NSD against usual grid descriptors is that it can have a large support region in the image best fitted to each described keypoint.

ASIFT or Affine-SIFT. Affine-SIFT (ASIFT) [96] is a fully affine invariant SIFT-like descriptor. SIFT is by default invariant to scaling, rotation and translation but not to affine transformations. ASIFT takes into consideration the angles defining the camera axis orientation. It applies a dense set of image translations, rotations, and camera zooms to two compared images. Then, it uses SIFT to match each artificial image from the first image to each created one from the second image. ASIFT is able to identify features that have undergone very large affine distortions but has twice the complexity of SIFT.

BOLD. The Bunch Of Lines Descriptor (BOLD) [127] is an edge based recognition method for textureless objects developed for indoor robotic systems. Many objects to be observed by an indoor robot have no important texture information on their surfaces, such as refrigerators and microwave ovens, but have strong geometrical consistency within the same object class. BOLD uses pairwise line segments with geometric relationships between them. Line segments are found as edges detected within a logarithmic scale space. BOLD performs edge matching between an input image and a set of training images and retrieves the best matched training images. Then, the Generalized Hough Transform is used to find the most consistent subset of edge correspondences and the object pose is computed through a Least-Square Estimation of the required transformation (for example similarity or homography).

CARD. Compact And Real-time Descriptors (CARD) method [6] is based on lookup tables for extracting histograms of oriented gradients. Keypoints are extracted on an image pyramid by a corner detector. CARD applies a log-polar binning pattern rotated by the dominant orientation. The elements of the obtained vectors are then quantized according to a lookup table. Finally, the vectors are converted to short binary codes by a technique named learning-based sparse hashing.

Colour Contour Frames. The Colour Contour Frames descriptor [38] creates similarity invariant texture patches defined around contour segments. The method detects color edges and their orientations and use them to create line and ellipse segments. The color edges are computed on the three image channels with the Canny detector [20] and their orientation with the double angle representation method [45]. For each line or ellipse segment, a similarity invariant frame is defined and a gradient patch is extracted using the invariant frame.

CHOG. Compressed Histogram of Oriented Gradients (CHOG) [23] uses histograms of gradients very compactly but still efficiently for matching. First, a histogram-of-gradients based descriptor is computed on a window around a keypoint. Orientation is set to the direction of the most dominant gradient. Vector Quantization (VQ) of the gradient distribution is performed on a grid that is not necessarily made with rows and columns of cells. The best grid has a cross-like shape. This descriptor is named Uncompressed HOG (UHOG). Then, trees are used in order to reduce a UHOG feature vector to a binary feature vector with respect to the distribution of the UHOG. The final compressed descriptor is the CHOG.

CSIFT or Color-SIFT. Color based SIFT [16] uses three color components taken from a linear transformation of the three RGB channels of an image. One component represents intensity and the other two are chromatic components. Color-SIFT (CSIFT) performs well under illumination changes due to shadow, shading and highlights. It generally more discriminative than the original gray-level SIFT but it is less invariant.

There are many more variants of SIFT that use different color information. These variants show similar performance to CSIFT. An example is PI-SIFT [108] that creates features on a scale space that are partially invariant to photometric

variations. MSIFT [15] uses the RGB color channels and near-infrared (NIR). In [132], there is a list of more SIFT color variants analyzed and compared.

DAISY. DAISY is a SIFT-like multi-scale local descriptor that computes Gaussian gradients on concentric rings of different radius around a keypoint [126]. The sigma of the Gaussian filter used at each ring is relevant to the radius of the ring, giving a scaled version of the image neighborhood on the respective ring. The name derives from the look of the multi-scale pattern of the exploited image neighborhood projected on the original image.

GLOH. Gradient Location and Orientation Histogram descriptor (GLOH) [91] extends the theory of SIFT by using more spatial regions for the gradient histograms on a log-polar location grid. The log-polar grid has three bins in radial direction (the radius set to 6, 11, and 15) and 8 in angular direction. SIFT is computed in each bin. The higher dimensionality of the descriptor is reduced from 272 to 64 dimensions through PCA.

HALCON template matching tool. The shape-based matching tool of the HALCON library [144] is an edge based template matching description method. First, a salient region is determined to create a model. Then the matching and locating of an object is achieved by using an affine transform. This method can find objects even with a single template image and localize objects with good accuracy in real-time.

KAZE and A-KAZE. KAZE descriptor (the word means *wind* in Japanese) (KAZE) [4] is another SIFT inspired descriptor. It creates a logarithmic scale space using Additive Operator Splitting (AOS) techniques for nonlinear diffusion filtering that keeps edges unaffected. KAZE computes first order derivatives over a 24×24 grid and finds the dominant orientation. Then all samples in the grid are rotated and again first derivatives are calculated according to the dominant orientation. A faster version of KAZE is Accelerated KAZE (A-KAZE) [5]. For A-KAZE, a new mathematical framework called Fast Explicit Diffusion (FED) is introduced, which speeds up dramatically feature detection in nonlinear scale spaces.

MROGH. Multisupport Region Order-Based Gradient Histogram descriptor (MROGH) [32] divides a given normalized support region into several rings and computes local gradient features at sample points in each ring. The samples are collected on a local xy coordinate system defined by the keypoint and one of the samples. The same method can be used with intensities instead of gradients and it is then known as Multisupport Region Rotation and Intensity Monotonic Invariant Descriptor (MRRID), presented in section 2.4.1.1. In comparison with MRRID, an MROGH feature vector is not binary by default.

MSLD. The Mean-Standard deviation Line Descriptor (MSLD) [140] is a robust multi-scale line matching algorithm. First, edges are extracted on the image in order to compose line segments. MSLD uses a support area around a line segment where it extracts gradients. The support area is defined by subregions and a grid over each subregion. The gradients are collected separately for each subregion. Then, the gradient orientations are aligned relative to the overall orientation of the line. This is a similar procedure to SIFT but extended for line segments. The gradients taken at each subregion are stacked in a matrix. As the size of this matrix is variable depending on the line length, MSLD uses

a combination of the mean and standard deviation to reduce the descriptor to a particular size. Hence, the mean vector and the standard deviation vector of matrix column vectors are computed. Then the two vectors are respectively normalized with the unit norm and concatenated into the final descriptor vector. The subregion size and the subregion grid size are variable and must be set at the beginning. The concept of MSLD can be extended for curves by the Mean-Standard deviation Curve Descriptor (MSCD).

PCA-SIFT. This is a variant of SIFT with reduced dimensions [64]. PCA-SIFT uses gradients over a 39×39 image region. The extracted vector has 3042 elements. Therefore PCA is used to reduce the elements to 36. The small dimensionality of PCA-SIFT requires less storage memory and results to a faster matching, though its performance is slightly less good than SIFT.

PD. Patch Duplets descriptor (PD) [58] are pairwise descriptor combining keypoints into pairs. The descriptor feature is a pair of patches, each patch centered around on a keypoint. The keypoints are extracted on a local orientation image. This image is computed with the double angle representation method [45]. The patch orientation is determined by the relative location of the two keypoints and the patch size is relevant to the distance between the two keypoints. The pairs are determined by geometrical and perceptual constraints. For each patch, 16 samples are collected at a 4×4 grid on the orientation image, giving 32 complex valued samples for a duplet. The values are reduced to 16 per duplet by PCA.

RIFF. Mobile Augmented Reality (MAR) systems rely on real time tracking and recognition. Rotation-Invariant Fast Features descriptor (RIFF) [122] is a fast local descriptor that exploits the Radial Gradient Transform (RGT) to collect rotation invariant gradients over a grid of Voronoi cells at radius 20 pixels around a keypoint. RIFF does not outperform the state of the art but works efficiently enough with lower computational cost.

RIFT. Rotation Invariant Feature Transform descriptor (RIFT) [75] is a more efficiently rotation invariant version of SIFT. First, the keypoints used for this descriptor are attributed with an orientation determined by their unit normal vector. The position and orientation of the samples collected around each keypoint is measured relative to a neighborhood determined coordinate system, rather than the true or world coordinate system. RIFT collects samples on a circular neighborhood around a keypoint divided into concentric rings of equal width. A gradient orientation histogram is computed from each ring. Orientation is measured relative to the direction pointing outward from the center in order to induce rotation invariance.

Shape Context. The idea behind this descriptor is to match one point on an shape with the best matching point on a second shape. Shape Context [11] describes shape around a keypoint by computing edges on an images and then collecting edge values at coordinates on a log-polar grid of bins around the keypoint. A feature vector of shape context expresses the configuration of the entire shape relative to a reference point (keypoint).

SIFT-GC. SIFT using Global Context (SIFT-GC) is a version of SIFT that considers the global context of the image [97] for more efficient keypoint matching. The SIFT feature vector is combined with a 60 element long vector of Global

Context (GC). This second vector is a histogram collected on a log-polar grid that extends over a large portion of the image.

SYM-FISH. SYMmetric-aware Flip Invariant Sketch Histogram descriptor (SYM-FISH) [21] is a reflection invariant version of the Shape Context descriptor that involves symmetry relationships between features. First, Shape Context features are extracted and they are rotated according to a dominant orientation depending on the two denser bins of the feature. This are the FISH features. Then, a symmetry table is used to keep symmetry relationships among visual words encoded with FISH and that characterize an object (sketch). Symmetry is measured densely using the kurtosis coefficient [7] for every 10° of rotation of the sketched image. The resulting 36-dimension vector of scores indicates the possible symmetry axes existing in the sketched image and symmetry relationships between visual words are attributed based on this vector.

2.4.1.3 *Laplacian based local descriptors*

SURF is used in the experiments of chapter 4.

SURF. Speeded Up Robust Features descriptor (SURF) [9] is a SIFT inspired local descriptor. SURF utilizes sums of 2D Haar wavelet responses (section 2.4.2.1) in order to approximate first and second order partial Gaussian derivatives. First, orientation is extracted from a circular region around each keypoint. Then, a square region aligned to this orientation is taken around the keypoint, and the approximated second order Gaussian derivatives are computed. The size of the square region depends on the scale of the keypoint. SURF is a lot faster and more robust than SIFT though slightly less invariant to rotation and illumination changes. SURF uses an integer approximation of the Hessian determinant for keypoint detection on different scales. Figure 6 gives an intuition about the image description made with SURF.

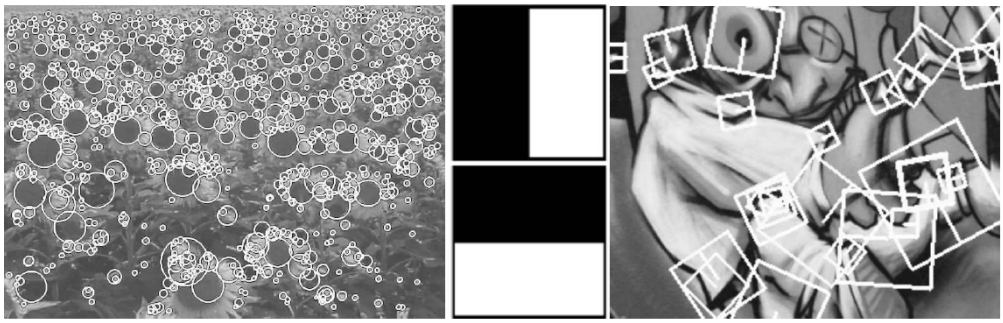

Figure 6: **Left:** Detected salient points for SURF. Hessian-based detectors find blobs. **Middle:** Haar wavelet types used for SURF. **Right:** SURF descriptors on an image at different scales and with the relevant orientations. Image taken from [9].

2.4.1.4 *Spectrum based local descriptors*

SID. The Scale Invariant Descriptor (SID) [70] starts with log-polar sampling on a set of image scales around keypoints and then obtains the local amplitude,

orientation and phase measurements of a 2D signal by the use the Monogenic signal [35] and the Fourier Transform Modulus [22]. The L_1-norm is used for further normalization. SID shows invariance to scale, rotation and illumination.

2.4.2 Global descriptors

2.4.2.1 *Intensity based global descriptors*

CENTRIST. This scene recognition descriptor models the distribution of local structures in an image while suppressing textural information [142]. CENsus TRansform hISTogram descriptor (CENTRIST) is easily implemented and works very fast. The key element is the Census transform that compares the intensity value of a pixel with its 8 neighboring pixels. The resulting descriptor is a holistic representation that resembles a sketched copy of the original image.

CMD. Covariance Matrix Descriptors (CMD) [130] are features based on the covariance matrices in image regions. The covariance matrix is basically made on pixel intensities but it can be combined with color, gradients, Laplacians or other filter responses to increase discriminality. The computation of the covariance matrix is easy due to the use of integral images. This is a texture descriptor but can also work robustly for object detection.

Eigenfaces. Eigenvectors can be used to encode important signal information. Eigenfaces are a description method that uses PCA to encode facial characteristics for detection and recognition [128]. Each computed eigenvector, when displayed in 2D, looks like a sketch of a face, thus called Eigenface. The Eigenfaces are considered as features that characterize the variation between face images, such as that a face can be represented as a linear combination of the Eigenfaces.

FAsT-Match. Fast Affine Template Matching descriptor (FAsT-Match) algorithm [73] performs fast template matching under 2D affine transformations by minimizing the Sum-of-Absolute-Differences (SAD) error measure. FAst-Match considers an image as the template and search for this template in another image. The idea is that the difference in the affine transformations of two images with similar content can be approximated by inspecting only a small fraction of pixels from the two images. Then, the SAD error can be used to evaluate this estimation. The template is a set of intensities that are mapped on the target image.

Fisherfaces. A relative approach to Eigenfaces for face detection and recognition is Fisherfaces [10]. Instead of PCA, Fisherfaces are extracted with Fisher's Linear Discriminant (FLD) [36]. Fisherfaces perform well for large variations in lighting and take into account facial expressions. The basis vectors (eigenvectors) of FLD, when displayed in 2D, look like faces curved on a surface, hence Fisherfaces. Every face image can be represented as a linear combination of Fisherfaces.

Grid shape descriptors. These type of descriptors project shapes onto a grid of fixed size and interpret the contour of the shape into a binary sequence [86]. This sequence can be easily stored in a small row of of bits. Though charac-

terized by low complexity, these descriptors suffer from low discrimination power.

Haar wavelets. Haar features [107] are widely used for detecting people, faces and other visual classes. These features resemble differences of boxes and can be easily computed at very low computational cost using integral images. Because Haar features are based on sums of pixels over rectangular regions, they can be unstable when used to detect forms that are not aligned with the rows and columns of the image. Nonetheless, they are widely used for detecting visual classes in real time applications due to their fast computation.

Moments. Some theoretical approaches for shape representation propose algebraic functions that encode shape attributes in mathematical values [53, 66, 37]. Moments are weighted averages of pixel intensities. These description approaches search for shapes by examining the amount of particular types of meaningful geometrical relationships in the image signal. Several theoretical approaches are exploited to introduce such functions, such as the Fourier Theory, derivatives and orthogonal polynomials. The resulting function are used to measure image properties as the centroid (center of gravity), the major and minor axes, the eccentricity, etc. Moments are very basic descriptors but work very fast and well for not complicated shapes in images.

LBP. Local Binary Patterns (LBP) [101] and their many variants have proven to be a very powerful descriptor. Their predecessor, the Texture Unit [50], is defined on a 3×3 neighborhood. In this neighborhood, a function is comparing the central pixel intensity with the other eight pixels and the output is three possible values per compared pair: lower, equal, higher (0, 1, 2). A frequency function of all the Texture Units of an image gives the image Texture Spectrum. LBP reduces the theory of Texture Unit to binary: the output of the comparison function can have only two possible values, lower or higher (0, 1) [100]. In a more recent version, the comparison is made with a number of pixels on a circle around a central point with variable radius [101] and has possible values lower or higher-equal (0, 1). Rotation invariance can be simply imposed by rotating the binary vector of comparison values until it starts with 0. Another way to induce rotation invariance is to count the transitions between 0 and 1 (uniform patterns). LBP with invariance to contrast can be made by using the local pixel variance. LBP can be averaged over larger regions in order to achieve a generalized image description. This approach separates an image into a grid of cells. In each cell, every pixel is used as the central pixel around which the comparison function works on its respective neighbors. After using all the pixels of the cell as central pixels, a histogram of this cell is created. The histogram in a particular cell shows which neighbors tend to be higher in intensity than their central pixel and which lower.

Soft histograms for LBP (SLBP) [3] increase the robustness of LBP using fuzzy comparison function, with soft margins (fuzzy membership functions) instead of a decision threshold, for improved texture recognition. Adaptive Soft Histogram Local Binary Patterns (ASLBP) [145] is a variation of SLBP for face recognition based on adaptively learning the soft margin of decision boundaries.

CBP. Centralized Binary Patterns (CBP) [42] for facial expression recognition is another variant better adjusted for noisy images. CBP extend the comparison function to consider both the central pixel's intensity with respect to the relevant surrounding sampled pixels as well intensity differences between these surrounding sampled pixels that exist symmetrically to each other considering the center.

LTP. An illumination invariant extension of LBP for face recognition is the Local Ternary Patterns (LTP) [123]. The support region of LTP is the eight closest neighbors of the central pixel. Before feature extraction, the face image undergoes gamma correction, high-pass filtering, masking to remove unnecessary image regions and contrast equalization. The comparison function of LTP can take three possible values, lower, equal and higher (-1, 0, 1). Therefore, the preliminary LTP feature is a vector composed by elements with three possible values. The LTP is split into two LBP features, named the negative and positive LBP features. Each one encodes either the negative and equal values (-1, 0) with 1 and 0 or the positive and equal values (0, 1) with 0 and 1 respectively (uniform patterns). Separate histograms and similarity metrics are computed for the positive and negative LBP features and then they are combined The face image is divided into cells and a histogram of LBP features are computed for each cell.

FTS. Fuzzy Texture Spectrum descriptor (FTS) [8] is used for texture analysis and characterization by expressing texture as a spectrum of intensity relations. This descriptor is also based on the predecessor of LBP, the Texture Unit. FTS takes a vector from a 3×3 window of pixels' intensities centered on a pixel. Then three values are assigned to every element of the vector, each showing the degree to which the gray-levels of surrounding pixels are lighter, similar or darker than the central pixel. This procedure repeats for all image pixels. After dimensionality reduction, a spectrum of the surrounding intensity relative to the central pixel is produced.

2.4.2.2 *Gradient or first order partial derivative based global descriptors*

HOG. Local histograms of image derivatives [116, 117] provide an effective image description for indexing and recognizing visual classes. The SIFT descriptor [84] adopted this approach, using histograms of gradient orientations computed over a grid of small windows (or cells). The power and generality of local histograms of oriented gradients has been demonstrated and made popular by Dalal and Triggs [30] under the name HOG. HOG descriptors are computed as a histograms of gradient orientations from a grid of small cells, and thus requires setting three parameters: the number of gradient orientations used in the histograms, the size of the cells and the size of the grid of cells. A fourth parameter concern the degree of smoothing used in computing the image derivatives. In experiments, Dalal and Triggs found that the most effective descriptor for detecting humans was obtained using histograms of 9 gradient orientations computed within a 3×3 grid of 6×6 cells. This provides a vector of 2916 features that was then used with a Support Vector Machine (SVM) classifier to find human forms in a test data base of images.

HOG is used in the experiments of chapter 5.

Figure 7 is a demonstration of the manner HOG is computed on an image containing a human shape. Though initially proposed for human detection, HOG is used for object recognition in general.

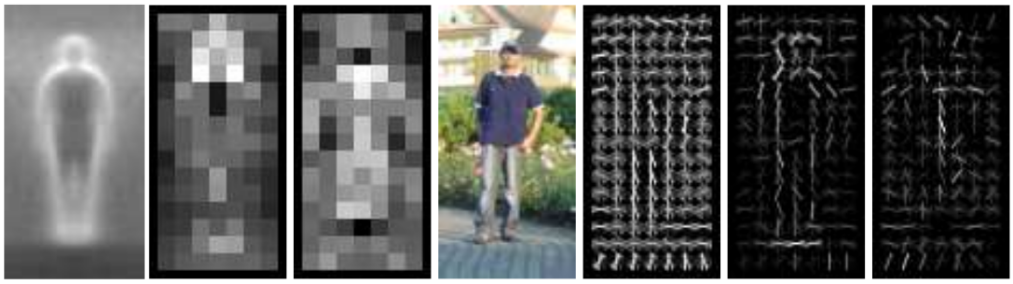

Figure 7: Illustration of the way in which the HOG image description works for the human shape from the INRIA Person dataset. (a) An average gradient image over training examples. (b) Each square shows the maximum positive SVM weight in the block centered on the square. (c) The same for the negative SVM weights. (d) A test image. (e) The computed HOG descriptor on the test image. (f) The HOG descriptor weighted by the positive SVM weights. (g) The HOG descriptor weighted by the negative SVM weights. Image taken from [30].

DSIFT or Dense SIFT. SIFT features can be used densely on an image on pre-designed locations at a fixed scale considering zero orientation in Dense SIFT (DSIFT) [134]. Then, local averaging on every 4×4 block of descriptors is performed, giving one averaged descriptor for each block. Local contrast normalization is used on the blocks for partially inducing invariance to illumination changes.

Gaussian derivatives (first order partial and gradient). Gaussian derivatives have long been popular because they can provide scale and rotation invariant description [39, 94]. A Gaussian Pyramid [28] provides a fast algorithm for creating Gaussian derivatives at multiple scales. The resulting descriptors are steerable [39] and can be used to construct affine invariant descriptions [94]. Gaussian derivatives are widely used to detect edges [20] and salient points [78, 84] as well as for multidimensional histograms of appearance [116]. They have been used with Log Polar Histograms for face detection [47]. First order Gaussian derivatives (partial and gradient) capture visual information for image structures such as bars, blobs and corners. Higher order Gaussian derivatives can be useful for describing more complicated structures but tend to be sensitive to image noise [95, 69].

GIST in color. GIST can be also computed using gradients instead of Fourier magnitudes. A color-based GIST variant [57] computes gradients on a grid over the RGB and HSV color spaces and then uses PCA for dimensionality reduction.

GLAC. Gradient Local Auto-Correlations descriptor (GLAC) [68] utilizes 2nd order statistics, spatial and orientational auto-correlations, of local image gradients. This method is an extension of Higher-order Local Auto-Correlation (HLAC) [104] method that uses pixel intensities instead of gradients. More ex-

tensions of this theory are the Steerable Filter Local Auto-Correlation (SLAC), the Normal Local Auto-Correlation (NLAC) for 3D images and the Cubic HLAC for motion images as well as the Color HLAC for multichannel images from [105].

Color Invariant HOG. An improvement of HOG for object detection in scenes under cast shadows uses color invariant gradients instead of gray-level gradients [137]. The color invariant gradients are extracted using three different color models: the RGB, the $c_1c_2c_3$ and the o_1o_2 color models.

GF-HOG. Gradient Field HOG (GF-HOG) [54] represents an image as a structure of a dense gradient field interpolated from a sparse set of edge pixels. This method is an alternative to Bag of Words (BoW) with local descriptors scheme for object localization and recognition.

Fast HOG. An lighter form of HOG is used with cascade classifiers [146]. Feature vectors are computed onto smaller blocks of different sizes and let the cascade training procedure select the most significant. This approach is less discriminant than the original HOG but it is faster to compute and can detect a variety of human body parts. Larger blocks capture information about larger portions of the human form while small blocks cover parts such as legs or arms, providing an improved robustness.

LARK. A local steering kernel (LSK) is the main element of Locally Adaptive Regression descriptor (LARK) [118]. An LSK is taken over an image patch. It exploits image gradients to determine the size and shape of a radially symmetric kernel function. This kernel function encodes local geometric structures in the image signal. The size of the patch, where an LSK is computed, depends on the query image (an image that contain an object of a requested class). A set of LSK is computed densely on an image and PCA is used for dimensionality reduction. The LSK features are compared by the cosine similarity measure.

LHS. Local Higher-order Statistics descriptor (LHS) [119] are a generalization of LBP (and LTP) that employs higher-order statistics of local non-binarized pixel patterns. LHS works on 3×3 neighborhood measuring the intensity differences between the central pixel and the surrounding eight neighbors, just like LBP. But instead of a binary quantization, LHS uses a parametric Gaussian Mixture Model (GMM) to derive a probabilistic representation of the differential space. To achieve this they use a scoring method that characterize an observed feature vector by its gradient with respect to the parameters of the GMM.

LINE. LINE [52] is an gradient based template matching recognition method for 3D textureless objects. It uses image gradients and gradient orientations and represents a 3D object with a limited set of templates called response maps. The name LINE derives from the term LINEarizing the Memory for Parallelization descriptor (LINE), which is the way this algorithm stores descriptor vectors in the memory of the computer to speed up computations. LINE has been integrated into three forms: (LINE-2D) that used gradients on 2D images, (LINE-3D) that uses surface normals from 3D images and LINE-MOD (multimodal) that uses both gradients and surface normals on RGB-D images.

PHOG. HOG can become a multi-scale feature when extracted on an image pyramid, named Pyramid HOG descriptor (PHOG) [12]. Edges are used to make a fuzzy histogram of gradient directions and increased discrimination power.

PHOW. Pyramidal Histogram of Visual Words descriptor (PHOW) [13] is a variant of DSIFT but extracted at multiple scales. PHOW-color [134] extracts the same type of descriptors but on the three HSV image channels.

Polar-HOG. HOG can be rotation invariant if the gradients are collected on the polar coordinate system [77]. Instead of a rectangular patch, Polar-HOG collects a disk patch and maps this disk on polar coordinates. The object direction is estimated by double-scale direction estimation and rotation correction is applied on the polar coordinates. If more than one candidate directions are estimated, a HOG feature vector is computed for each of these directions.

PVEP. Peak Valley Edge Patterns (PVEP) [98] are a modification of LBP that uses first order derivatives on 3×3 neighborhoods for given directions. Directional edges on $0°$, $45°$, $90°$ and $135°$ are obtained by intensity differences between a pixel and its two closest neighbors on this direction. The comparison function of PVEP has a ternary function with values 0, 1 and 2. The PVEP feature is then split into two binary patterns, Peak Edge Pattern (PEP) and Valley Edge Pattern (VEP). This approach is similar to an LTP feature being split in a positive LBP feature and a negative LBP feature. LBP extracts relationships between a central pixel and its relevant neighbors, whereas PVEP extracts relationships between each pixel in the 3×3 neighborhood with its two closest neighbors along a given direction.

Rotation-Invariant HOG. Another method to build rotation-invariant HOG feature vectors [81] uses Fourier analysis in polar coordinates. The histogram of gradients is considered as a continuous angular signal which can be well represented by the Fourier basis. This descriptor can be adjusted for 3D by using spherical coordinates instead of polar.

Scene-Tensor Duplets. A tensor is a matrix structure that can combine local lower-level features into more complex descriptors and so integrate different type of information over a neighborhood. A fourth order tensor, defined on a projective space, uses descriptors over a large region to integrate them into a regional descriptor [99]. This tensor representing line segments by their orientation, center of gravity and covariance relative to a local coordinate system, is also known as a scene tensor. In order to construct the Scene-Tensor duplets [121], scene tensors are created over an image so that each one indicates two line segments on a region, defined in terms of position, extension (how long they are) and orientation. Each tensor is computed on a corner detected by also using tensor representations over the image. The final duplet is a pair of scene tensors connected with a set of geometrical relationships defined by the line segments they contain.

WLD. Weber Local Descriptor (WLD) [24] is a robust method inspired by Weber's Law (concerning images, this law states that the ratio of the noticeable difference in intensity for discrimination to the background intensity is a constant). WLD first computes the ratio of the relative intensity differences of a central pixel against its neighbors (first-order derivatives) in a 3×3 neighbor-

hood to the intensity of this pixel. Then the arc tangent of the ratio is taken. If the final value is is positive, the surrounding pixels are lighter than the central one and visa versa. An orientation computed over the 3×3 neighborhood is also attributed to the WLD feature. WLD features are computed over a grid of cells following a delta function pattern and are accumulated into a 2D histogram for the final image descriptor.

2.4.2.3 *Laplacian or second order partial derivative based global descriptors*

Gaussian derivatives (second order partial and Laplacian). Similarly to the first order Gaussian derivatives, second order Gaussian derivatives (partial or Laplacian) capture visual information for image structures such as bars, blobs and corners. Their advantage is that they capture shapes independently from orientation.

Snakes. Active contour models, also known as snakes, are energy minimizing models that detect shapes in images [63]. Staring from a selected image point, a contour is continuously spreading through the image by minimizing an energy function using the pixel neighbors on the periphery of the contour. The energy function has three terms that measure the strength and orientation of pixel intensities, edges and corners using intensity values and properties of second order derivatives. The contour spreads towards the direction that each time the function is minimized. This type of contours are referred to as active because their shape is formed functionally and dynamically. The nickname snakes comes from the way they spread on the image like they were alive.

2.4.2.4 *Spectrum based global descriptors*

Gabor wavelets, GEF & GaFour. The Gabor wavelet transform is widely used in image processing and computer vision due to each ability to capture texture in different orientations and resolutions. Gabor wavelets [55] are used in a discrete and finite form as band-pass filters for capturing information from 1D or 2D signals, for example speech recordings and images. A Gabor filter may be defined as the product of a Gaussian kernel with a complex sinusoid, therefore it is a filter with real and imaginary part. Their use include texture characterization, face detection, face pose estimation and facial expression recognition. An alternative way that Gabor wavelets can be used is by obtaining a magnitude on a location from the addition of several Gabor filters. This magnitude corresponds to the energy of the existing frequencies at the location. In this case, the filters are named Gabor Energy filters (GEF) [103] and they are sensitive to edges giving more precise responses. GEF is used for expression/ emotion recognition. Another descriptor based on Gabor wavelets is the GaFour features descriptor [87] for face recognition and face pose estimation. The Gabor / Fourier descriptor (GaFour) features are a combination of 1-D Gabor wavelets and the Fourier transform. The image is sliced in rows and for each there is an application of 1-D Gabor filters of five different frequencies. Then, for each created magnitude signal GaFour uses the Fourier transform and take all the vectors of the real and imaginary parts in the final asymmetry feature vectors.

GIST. GIST [102] captures a holistic representation of the image context the gist of the image for easy classification. The idea is that the specific information about object shapes or identities is not essential for scene recognition and that modeling a holistic representation of the scene gives enough information for its probable semantic category. An image is resized to a fixed sized square image, where a set of magnitudes are computed. The original GIST uses the squared magnitude of the Fourier transform (energy spectrum). PCA is used for dimensionality reduction. Averaging the GIST of a large set of images from the same semantic category, creates a vector named the spectral signature of this category. GIST is based on the model of the Spatial Envelope that generates a multidimensional space in which scenes of a semantic category (for example forests, mountains, buildings, coasts) are projected in a similar way. The dimensions of the Spatial Envelope model are naturalness, openness, roughness, expansion and ruggedness of the image scene. The Spatial Envelope properties for a semantic image category can be estimated using linear regression on the spectral information of the GIST features.

G-RIF. Edge orientation and density are combined with color information by the law of Gestalt [141] for proximity and similarity between features in Generalized-Robust Invariant Features descriptor (G-RIF) [67]. G-RIF is a multi-cue contextual descriptor. It uses Gaussian derivatives to approximate Gabor filters for edge description and image hue to define the strength of these edges. G-RIF extracts complementary visual parts for an object for object recognition.

LESH. The Local Energy based Shape Histogram descriptor (LESH) [115] is an invariant descriptor to common changes in face images, such as illumination, skin color, etc. This quality makes it suitable for face pose estimation when plugged into a learning algorithm. LESH uses a local energy model exploiting magnitude and phase of Fourier coefficients. Local histograms are extracted from different patches of the face image using the measurements of this local energy model. The different patch histograms are combined such as to keep the spatial relationship between facial parts.

Steerable filters. The main idea of steerable filters [40] is to compute a set of easily computed filters in different orientations capturing shape from different angles. In the original paper, the frequency response of second order Gaussian derivatives is used. Steerable filters can be also computed on a multi-scale space providing shape information with respect to both orientation and scale.

2.4.3 Classification and taxonomy by application of existing methods.

A collective table of all the aforementioned description methods is presented first in this section, table 1. The five left columns next to the descriptors column respect the classification scheme of the previous section and the rest of the columns provide particular information about the properties and the use of each method. By looking at the table, it is obvious that there are certain trends in image description:

- Discrimination is more important to global descriptors while invariance is generally more important to local descriptors.

- Illumination invariance is important for both local and global descriptors while rotation and scale invariance is more important for local descriptors.
- Gradient and intensity based methods are the majority.
- Laplacian or second order partial derivatives, though used, are not common in the proposal of new methods. Researchers prefer to propose new methods that are derived from already well performing approaches, which use gradients and intensities.
- Spectrum based descriptors are mostly preferred for global and robust description.

The table shows the qualities of descriptors but in the same time it shows what has been already tried and where there is room for exploration.

A taxonomy tree is made next, presented in figure 8, associating methods to applications according to the theory on which these methods are based. For the taxonomy we use only preliminary applications; matching, object/shape/texture detection and recognition. These are substandard to all other applications like indexing, tracking, 3D reconstruction, etc. It was seen that it is meaningful to build such a tree starting with the same classification of methods as in the previous section, first according to the support area and then according to the theory they are based on. The ending nodes are the relevant substandard applications together with the names of the methods that address them. The reason for the structure of the tree is that we wanted to illustrate how applications drive the creation of descriptors that will meet their requirements.

It is made obvious that the same type of applications can be addressed in a different manner by more than one of kinds of methods. A conclusion drawn from the tree is that descriptors are well customized for particular applications. A closer look to the previous section, reveals that descriptors are perhaps too customized. They focus to only a few applications, or even sometimes just a fraction of an application, and are very hardly generalized without important modifications to their theoretical approach.

2.5 A FLEXIBLE NEW APPROACH FOR A GENERAL PURPOSE DESCRIPTOR

This chapter reviews the state of the art on image description focusing on the lowest rank of approaches and applications. Despite image description being a broader field, the aim of this thesis is to propose a new theoretical approach that works for 2D images and which can be later expanded to more dimensions and advanced applications.

After organizing and studying the state of the art, the well defined objective of a new approach can be established. The variety of methods reveal that they tend to be customized for a very narrow scope. This leads to the disability to re-use or share information between systems. Instead, different types of data from the same image need to be computed distinctively for every method using the same pixels. A visual system that has to solve several tasks on the

Table 1: Classification table for existing image descriptors. The classification is made according to the theory they are based and the qualities of the descriptors. The descriptors are presented in alphabetical order. Each one can be found on the previous sections by consulting the first column "SUPPORT AREA" (Local or Global) and then the group of columns "APPROACH" (the type of theoretical approach that the descriptor is based on.)

		APPROACH				INVARIANCE						
DESCRIPTORS	**SUPPORT AREA**	**Intensity Based**	**Gradient or 1st derivative Based**	**Laplacian or 2nd derivative Based**	**Spectrum Based**	**Scale**	**Rotation**	**Affine**	**Bluring/ JPEG compresion**	**Illumination**	**Robustness to Clutter/ Partial Occlusion**	**COMMON APPLICATION DOMAIN**
BRIEF [19]	Local	X				X			X	X		Matching
BRISK [76]	Local	X				X	X		X	X		Matching
CS-LBP [51] & CS-LTP [46]	Local	X				X	X		X	X		Matching
Ferns [106]	Local	X				X	X	X				Matching
FREAK [133]	Local	X				X	X	X	X	X		Matching
GIH [79]	Local	X				X	X	X				Matching
LIOP [139]	Local	X					X	X	X	X		Matching
MRRID [32]	Local	X				X	X	X	X	X		Matching
ORB [112]	Local	X				X	X		X	X		Matching
OSID [124]	Local	X				X	X	X	X	X		Matching
Self-similarity descriptor [120]	Local	X				X		X	X	X	X	Pattern detection
Shape Context [11]	Local	X					X	X	X	X	X	Object recognition and matching
SILT [65]	Local	X				X						Line matching
Spin images [59]	Local	X					X	X				Matching
SYMD [49]	Local	X				X	X		X	X		Matching
ASIFT [96]	Local		X			X	X	X	X	X		Matching
BOLD [127]	Local		X			X	X				X	Line matching (Textureless objects)
CARD [6]	Local		X			X	X		X	X		Matching
CHOG [23]	Local		X			X	X		X	X		Matching
Colour Contour Frames [38]	Local		X			X	X					Templatematching
CSIFT [16]	Local		X			X	X		X	X		Matching
DAISY [126]	Local		X			X	X		X	X	X	Matching
GLOH [91]	Local		X			X	X		X	X		Matching
HALCON [144]	Local		X			X	X	X	X	X	X	Line matching (Textureless objects)
KAZE [4] and A-KAZE [5]	Local		X			X	X	X	X			Matching
LHS [119]	Local		X							X		Texture and facial analysis
MROGH [32]	Local		X			X	X	X	X	X		Matching
MSLD [140]	Local		X				X	X	X	X	X	Line matching
PCA-SIFT [64]	Local		X			X	X			X		Matching
PD [58]	Local		X			X			X	X		Matching
PVEP [98]	Local		X							X		Texture recogn. and face detect.
RIFF [122]	Local		X			X	X		X	X		Recognition and tracking
RIFT [75]	Local		X			X	X		X	X		Matching
SIFT [84]	Local		X			X	X		X	X		Matching
SIFT-GC [97]	Local		X			X	X		X	X		Matching
SYM-FISH [21]	Local		X			X			X	X	X	Template matching
CMD [130]	Local	X	X	X	X							Texture recogn. and facial analysis
SURF [9]	Local			X		X	X		X	X		Matching
SID [70]	Local				X	X	X			X		Matching
CBP [42]	Global	X							X	X		Facial expresion recognition
CENTRIST [142]	Global	X								X		Scene Recognition
Eigenfaces [128]	Global	X										Face detection and etection
FAsT-Match [73]	Global	X				X	X	X	X	X	X	Template matching
Fisherfaces [10]	Global	X								X		Face detection and recognition
FTS [8]	Global	X										Texture recognition
Grid shape descriptors [86]	Global	X				X	X					Shape recognition
HAAR [107]	Global	X										Object recognition
LBP [101]	Global	X					X			X		Matching
LTP [123]	Global	X								X		Matching
Moments [53, 66, 37]	Global	X				X	X					Shape recognition
Color Invariant HOG [137]	Global		X						X	X	X	Object detection and recognition
DSIFT [134]	Global		X						X	X	X	Object detection and recognition
Fast HOG [146]	Global		X						X	X	X	Object detection and recognition
GF-HOG [54]	Global		X						X	X	X	Object detection and recognition
GIST in color [57]	Global		X						X	X	X	Scene Recognition
GLAC [68]	Global		X							X	X	Texture recogn. and human detect.
HOG [30]	Global		X						X	X	X	Object detection and recognition
LARK [118]	Global		X							X		Object and recognition
LINE [52]	Global		X				X		X	X	X	Line Matching (Textureless objects)
NSD [18]	Global		X			X	X	X	X	X		Matching
PHOG [12]	Global		X						X	X	X	Object detection and ecognition
PHOW [13]	Global		X			X			X	X	X	Object detection and recognition
Polar-HOG [77]	Global		X				X		X	X	X	Object detection and recognition
Rotation-Invariant HOG [81]	Global		X				X		X	X	X	Object detection and recognition
Scene-Tensor Duplets [121]	Global		X									Object and recognition
WLD [24]	Global		X						X	X		Texture recogn. and face detect.
Gaussian Derivatives [39]	Global		X	X		X	X					Object detection and recognition
G-RIF [67]	Global			X		X			X	X	X	Matching
Snakes [63]	Global	X		X								Shape recognition
GABOR[55],GEF[103]&GaFour[87]	Global				X							Texture recogn. and facial analysis
GIST [102]	Global				X				X	X	X	Scene Recognition
LESH [115]	Global				X					X	X	Face pose estimation
Steerable Filters [40]	Global				X							Shape recognition

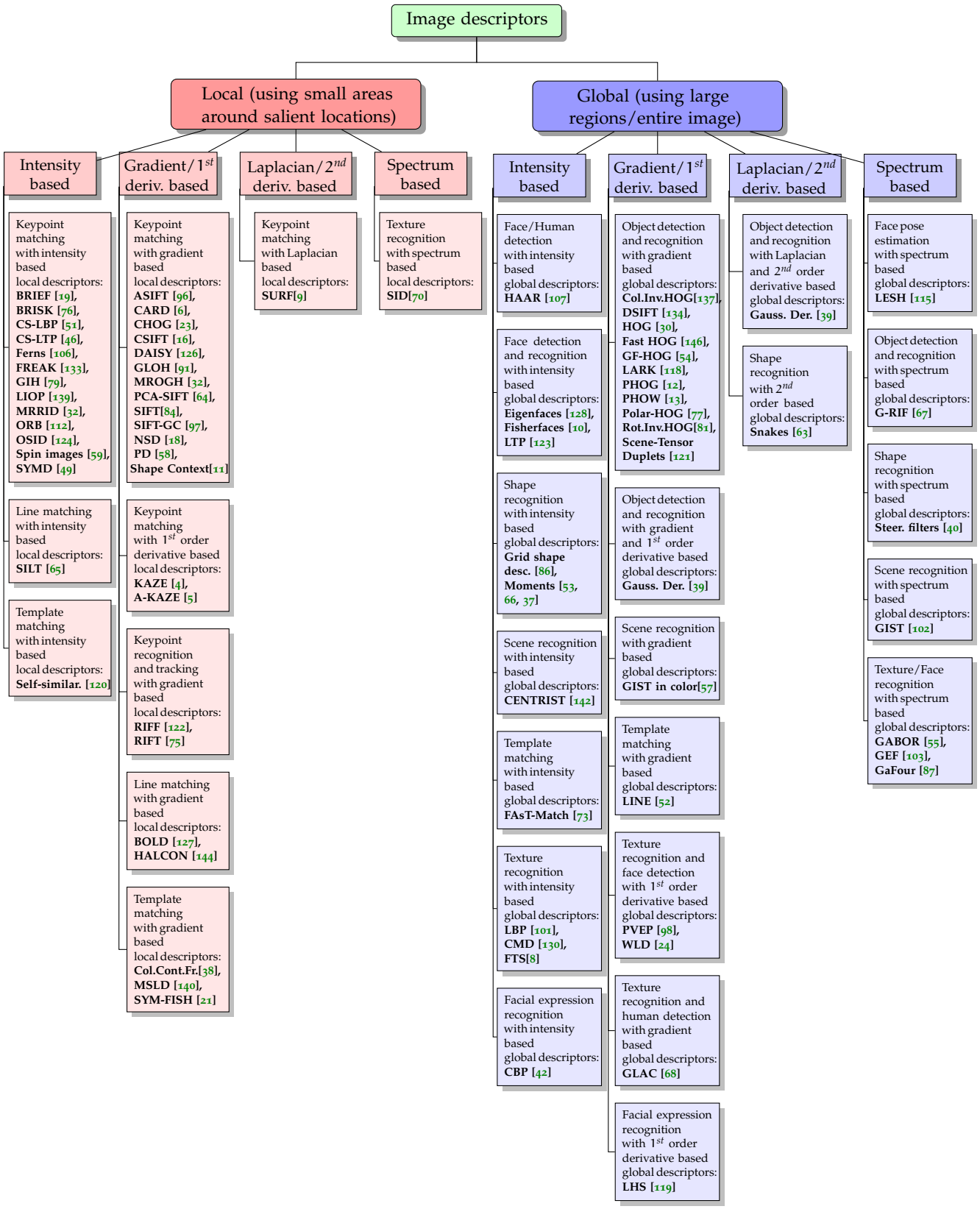

Figure 8: Taxonomy tree of the most substandard application per descriptor. The taxonomy first separates descriptors by the size of their support area and then by the theory they are based. It becomes clear that the support area of a descriptor depends on the size of the visual details that matter for an application and that certain theoretical approaches are preferred to address particular applications.

same images needs to compute different types of features, which is costly for computations and for memory use and storage. Additionally, we can see that the same bottom ideas are recycled and other approaches are left aside, though they are proven to be mathematically very efficient for describing signals.

We propose the creation of a description method that can be easily adjustable to several visual tasks, concerning both local and global description. The objectives we set for this method are:

- The descriptor should have a variable size to fit the needs of different tasks but always encode information in the same manner.
- In the same time, as real time applications become more and more popular, this method should be as compact as possible in order to produce small descriptor vectors while remaining discriminative.
- Robustness should be one of its qualities as its goal is to generalize the most possible. The good aspects of robustness is that it allows the use of smaller amounts of information (meaning smaller descriptor vector lengths) while retaining discrimination power.
- Rotation and scaling are the two major problems of descriptors that can be hardly overcome by robustness, so it is important to see into them more carefully.

To respect these objectives, we thought of creating a method totally depending on pure mathematical approaches. Approaches that have been used and proven efficient for capturing signal information. But, they were never combined into a flexible descriptor for both local and global description. The mathematical approaches we use are the Laplacian Profile, a multi-scale vector of Laplacians, and the Fourier Transform.

In the rest of this thesis, the proposed approach is extensively explained and experiments reveal its efficiency. In chapter 3, the proposed method is carefully established and reasoned through simple tests. In chapter 4, the proposed method is used as a local descriptor for experiments on matching. In the contrary, chapter 5 shows experiments where the method is used as a global descriptor on image patches. In chapter 6, the proposed method is used on a more sophisticated application for reflection symmetry detection. This last experimentation chapter, though also based on matching, shows the flexibility of our approach and the easiness of generalization to diverse visual tasks.

PROFIL LAPLACIEN ET TRANSFORMEÉ DE FOURIER RADIALE POUR LA DESCRIPTION DE L'IMAGE

Ce chapitre étudie un nouveau type de fonctionnalités robustes basées sur l'exploitation purement mathématique du signal d'image. Les descripteurs sont généralement développés pour être bons dans un genre très limité de tâches. L'objectif de cette thèse est de promouvoir l'idée que la description d'images peut être assez robuste et adaptable à de nombreuses applications plutôt que très compétitive pour une petite fraction seulement des applications. En même temps, les descripteurs doivent exiger aussi peu de coût de calcul et de mémoire que possible. La méthode que nous proposons est capable d'exprimer l'information visuelle de manière robuste et très compacte.

A partir de là, nous avons décidé d'utiliser deux outils mathématiques qui sont éprouvées pour capturer efficacement des informations de signal. Le premier outil est le laplacien du gaussien du signal d'image sous la forme d'un vecteur multi-échelle, le Profil Laplacien. Ce vecteur peut être considérée comme la colonne vertébrale du descripteur. Le deuxième outil est la transformée de Fourier calculée radialement autour du Profil Laplacien. Cette partie du descripteur permet l'extension de la surface d'appui pour une meilleure discrimination. Le vecteur de descripteur final est un vecteur de caractéristiques multi-échelle robuste qui peut devenir extrêmement faible si nécessaire, tout en conservant suffisamment de puissance de discrimination.

3

LAPLACIAN PROFILE AND RADIAL FOURIER TRANSFORM FOR IMAGE DESCRIPTION

3.1 CONCEIVING A NEW METHOD

This chapter investigates a new type of robust features based on the pure mathematical exploitation of the image signal. Descriptors are usually developed to be good for a very limited kind of tasks. The aim of this thesis is to promote that description can be rather robust and adaptable to many applications than just being very competitive for only a small fraction of applications. In the same time, descriptors should be as less inexpensive as possible. The method we propose is able to be robust while providing very small descriptor vectors.

We decided to use two mathematical approaches that are proved to capture signal information very efficiently. The first approach is the Laplacian of Gaussian that we use to create a multi-scale vector called the Laplacian Profile. This vector is the spine of the descriptor. The second approach is the Fourier Transform calculated radially around the Laplacian Profile. This part of the descriptor serves the extension of the support area for better discrimination. The final descriptor vector is a robust mutli-scale feature vector that can become extremely small if necessary while retaining enough discrimination power.

This chapter is organized as follows. First, section 3.2 provides an brief review of the new description method. Section 3.3 introduces the Half-Octave Gaussian Pyramid algorithm for fast calculation of a logarithmic scale scape and looks into the efficiency of the different scale spaces that can be produced by this algorithm using different σ. Section 3.4 shows how the Laplacian Profile is extracted form the image pyramid. Section 3.5 explains the possible ways of the Radial Fourier Transform computation and the setting of important relevant parameters. Finally, section 3.6 summarizes the aspects of the investigated method.

3.2 OVERALL APPROACH

Multi-scale image description on logarithmic scale spaces provides strong features that resemble the human way of vision [133]. Visual information in the center of focus is perceived with more detail and while moving away the information is captured with less detail. This way of perception provides important details for a location in an image and extra information about the surrounding area. This approach makes it easier to discriminate between two image locations that locally appear similar but in a wider range their difference is obvious. This is depicted in figure 9. The two ways to create a scale space representation have been explained in subsection 2.3.3. We use an image pyramid to create the necessary scale space, as illustrated in figure 10. The way of the pyramid

structure that we use to create a scale space, explained in detail in section 3.3, allows Gaussian derivatives to be easily computed in different scales.

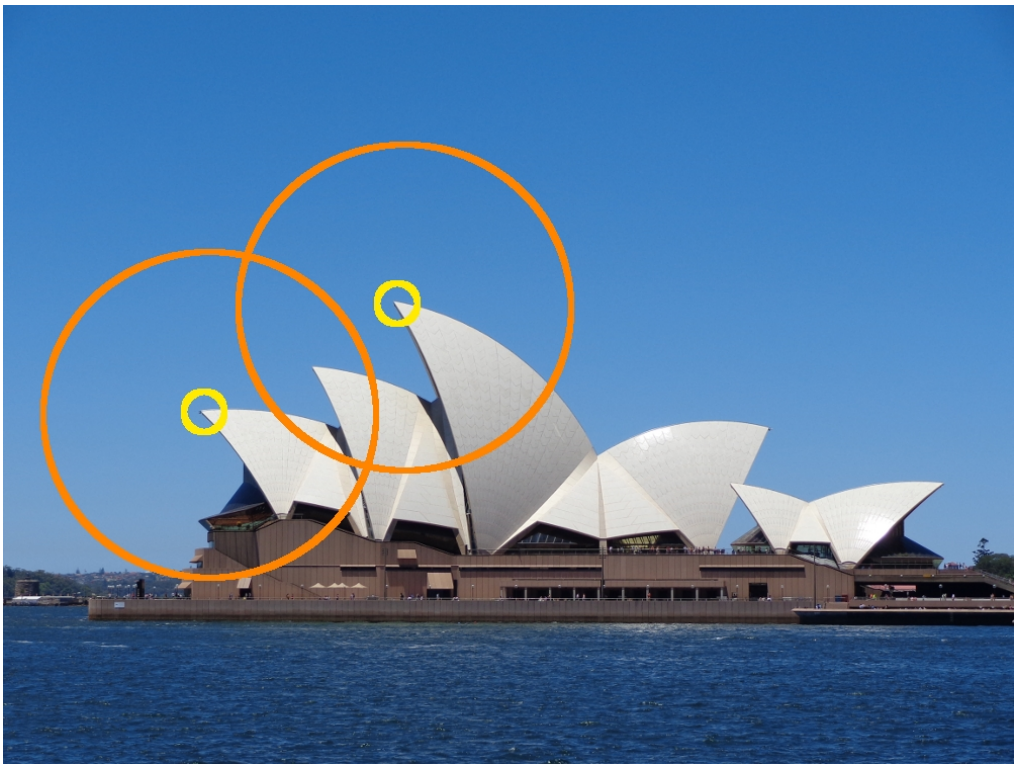

Figure 9: If two descriptor features are extracted only inside the small support areas in the yellow circles, the features will look very similar for the two salient locations. If the features use more information from the areas defined by the orange circles, the descriptor features will look different as the appearance of the two wider areas is much different.

The use of Gaussian derivatives has inspired several algorithms, as seen in the previous chapter. Though it is known for its properties, see paragraphs 2.4.2.2 and 2.4.2.3, Laplacian derivatives have not been used very often to propose new methods. On the contrary, gradient descriptors have dominated the field. This presents an opportunity for research as the Laplacian of Gaussian has the properties that we need for the new method. The Laplacian of Gaussian captures information in the form of extrema in intensity perturbations. Additionally, the Laplacian of Gaussian is by default rotation invariant. So, it does not need any orientation normalization as gradients do. Moreover, computationally it is easy to compute. Therefore, we use the Laplacian of Gaussian on a set of different scales of the image in order to achieve rotation-invariant multi-scale image description as the main part of our descriptor.

The Fourier Transform and formulas derived from it have been used a lot for image description [110, 147, 70, 87, 136, 81]. The Fourier transform by defining several frequencies in an image gives the ability to choose those are interesting for a particular task and discard the rest. High frequencies are usually noisy and do not need to be kept for our goal. By keeping only low frequencies,

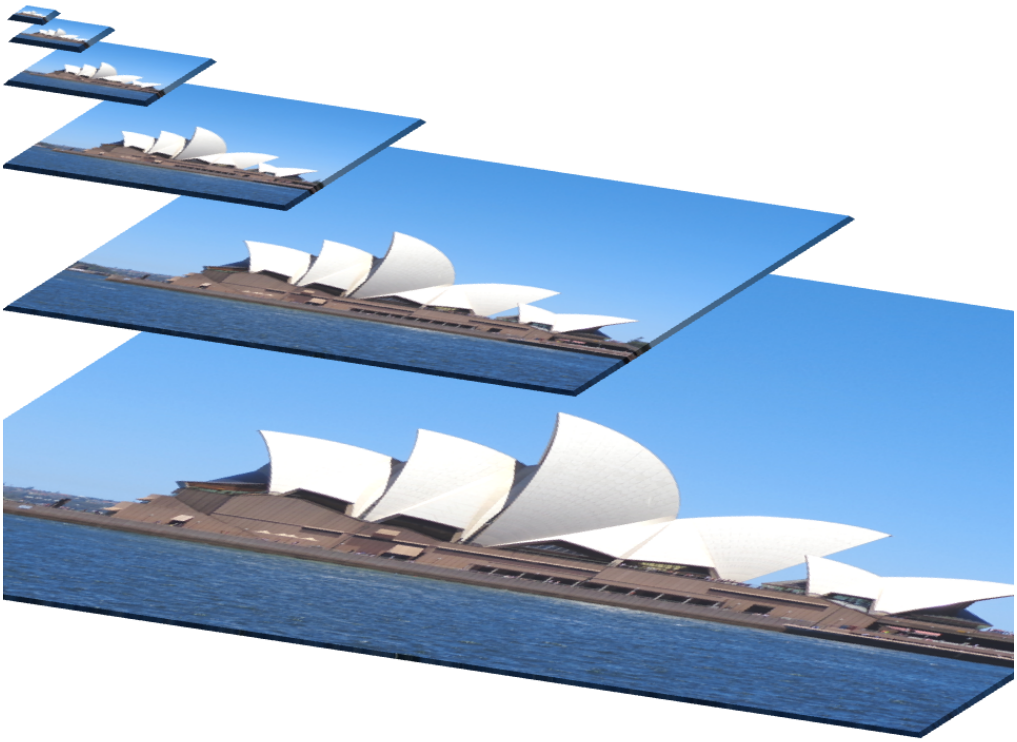

Figure 10: A scale space represented by an image pyramid.

we extend the idea of collecting low resolution information in a large support area to assist the keypoint description. Also, frequencies can be separated in magnitude and phase. Magnitudes are independent of orientation as they just show the quantity of a frequency. On the other hand, phase can give indication about the signal orientation on a round area (disk or circle) [147]. We can calculate these kind of information at any scale around the location of the selected Laplacian of Gaussian values. The Fourier Transform has all the demanded properties for extending the support area of our method. Figures 11 and 12 give a preliminary view of the methods before being thoroughly explained in the next sections.

Invariance to scale and rotation changes are the two most complicated types of invariance, as we can see by the literature in chapter 2. We can address invariance to rotation by the properties of the Laplacian of Gaussian and frequency magnitudes. But in order to address scale invariance, we need to exploit the scale space appropriately. The number of possible scales, where the proposed method can be extracted, is variable. It depends on the total number of resampled images (pyramid levels) that can be created by the used pyramid algorithm and the number of levels we want to use in order to create descriptor vectors. If we use a small subset of the created pyramid levels for the computation of descriptor vectors, we can compute descriptor vectors at different heights on the pyramid, as shown in figure 13. If the descriptors are extracted starting at different scales, we achieve scale-invariant image descriptor vectors: on a higher pyramid of a larger image, the descriptors may be found on a

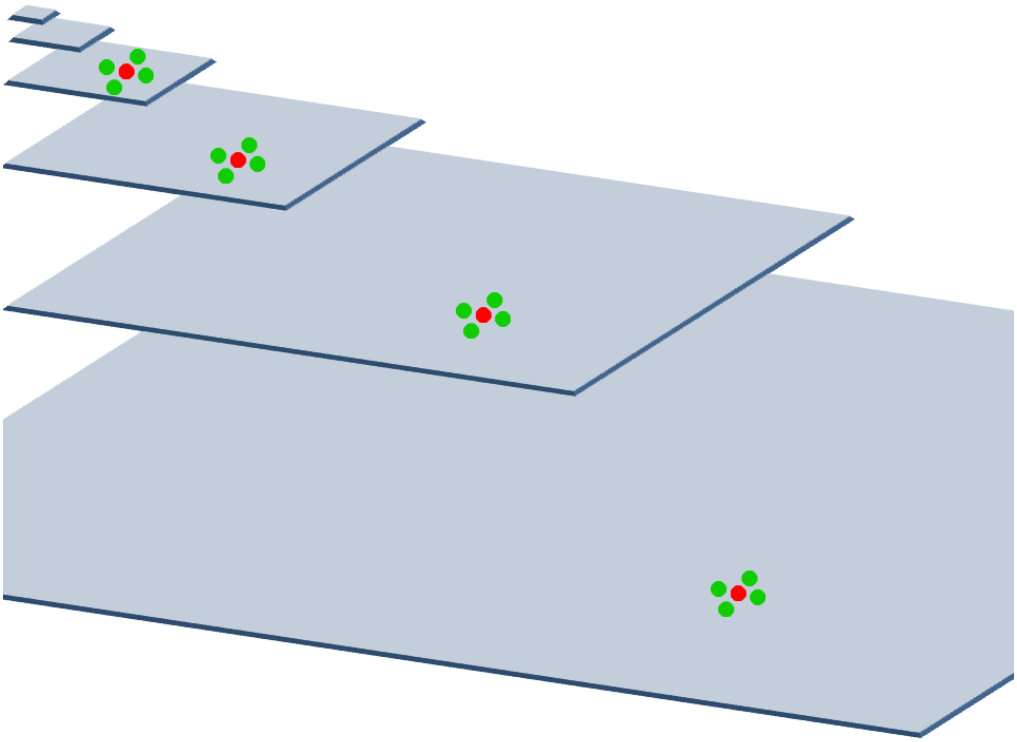

Figure 11: Example of a possible descriptor feature extracted with the proposed method on an image pyramid. The red dots represent the Laplacian of Gaussian values on the respective samples. The green dots represent the samples of a possible support area where the Fourier Transform can be computed.

different height. Further we test different ways to normalize the energy of the final vector by appropriate ways proposed in the literature to increase tolerance to chromatic changes. The L_2-norm normalization is the most effective. What we expect from this descriptor is invariance to scale and rotation while being highly discriminative due to its multi-scale nature and well established theory that it is based on.

3.3 IMAGE PYRAMID

The scale space we use for our method is approximated by a pyramidal structure of smoothened and resampled images. The algorithm we use to create this structure is the Half-Octave Gaussian Pyramid [29]. The use of the pyramid serves three reasons: a) creating a multi-scale descriptor, b) creating a descriptor that is scale-invariant and c) fast and easy computation of the Laplacian of Gaussian.

The scaling of the image to create the pyramid levels is done by the convolution with a Gaussian filter. As already seen in the previous section, Gaussian filtering is generally preferred for the creation of smoothened and resampled

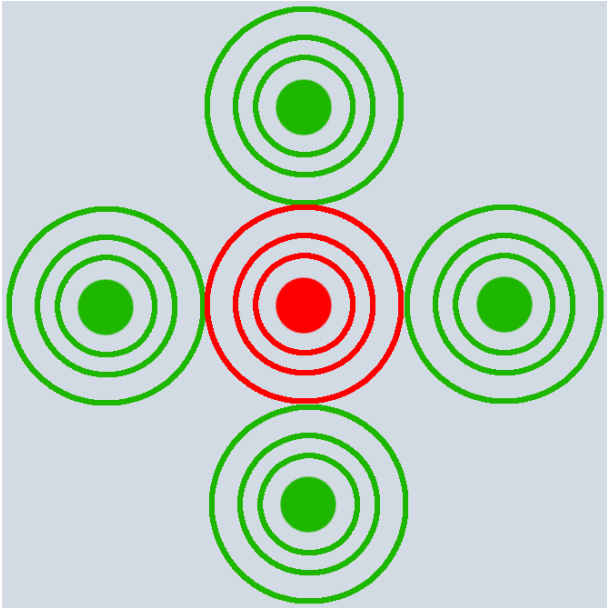

Figure 12: Example of a possible descriptor vector of the proposed method projected on the lowest pyramid level that is used to extract the vector. The red dot represents the Laplacian of Gaussian value on the respective sample on this level. The green dots represent the samples of a possible support area where the Fourier Transform can be computed. The rings around the dots represent the corresponding regions on this level from the respective samples collected on higher scales.

images as the smoothing it offers reduces the appearance of artifacts. In practice, the Gaussian filter is a sampled form of a normalized Gaussian function:

$$G(x, y, \sigma) = W_N(x, y)\frac{1}{A}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{13}$$

where x and y correspond to the integer values of the pixel addresses and σ is the scale of the pyramid level (smoothened and resampled image). $W_N(x, y)$ corresponds to a window of size $N \times N$ that limits the spacial extent of the sampled Gaussian, where N should be greater than or equal to $8\sigma + 1$. The letter A corresponds to the sum of the coefficients of the Gaussian that normalizes the gain of the filter to 1, assuring a scale invariant impulse response.

The half-octave Gaussian pyramid is composed of K resampled images (pyramid levels), each of which has been convolved with a Gaussian filter $G(x, y, \sigma)$ and resampled with a sample distance of

$$s_k = 2^{(k-1)/2} \tag{14}$$

A sample distance of $\sqrt{2}$ is obtained by sampling along the diagonal direction for even valued k. Because we make sure that the sample size is always fixed with respect to the scale σ of the Gaussian, each of the sampled images of the pyramid has an identical impulse response when expressed in pyramid samples. Impulse responses grow exponentially with k, when projected back to image pixel coordinates (x, y). We will use (x, y) to refer to the original image

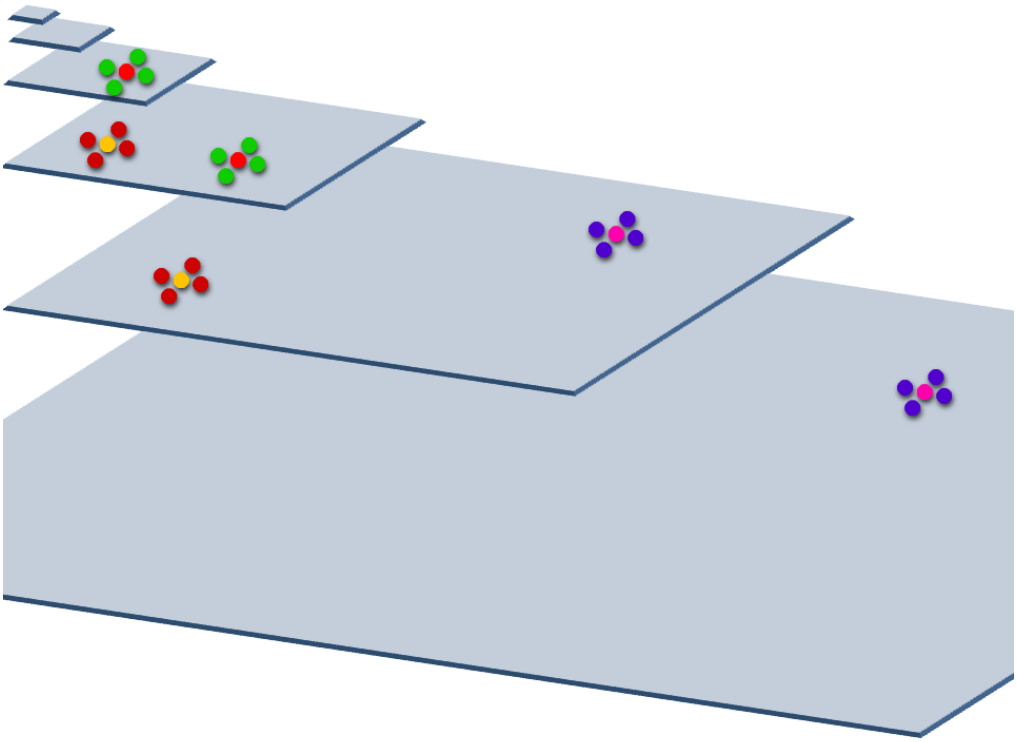

Figure 13: Example of a possible descriptor feature extracted with the proposed method on an image pyramid. If we choose to extract the descriptor on different subsets of pyramid levels, we have a multi-scale descriptor that is scale invariant.

coordinates. We will refer to a sample on a pyramid level as $P(x, y, k)$, so that $P(x, y, k)$ is the pyramid sample of level k that corresponds to the pixel (x, y) in the original image [113]. We can equally express sample $P(x, y, k)$ in the continuous scale space as $P(x, y, \sigma_k)$, with σ_k referring to a scale factor. The number of pyramid levels depends on the size of the original image. For a $W \times H$ image, the algorithm can create a pyramid of $K = 2 \times \mathrm{Log}_2(\min(W, H))$ levels. For k created pyramid levels, $k = 1$ to K, we can have have at most $P = 2 \times W \times H$ samples in total. The actual number of samples will be slightly smaller in practice, because the impulse response at the top levels of the pyramid is larger than the original image. The smoothened and resampled images at these levels are dominated by the filtering border effects and can be discarded.

3.3.1 Gaussian pyramids vs Binomial pyramids

A Gaussian pyramid can be computed with two ways. One way is to use a Gaussian filter with float numbers and the other way is to use a binomial filter that resembles a Gaussian filter [29]. A binomial filter has the advantage of using integer numbers. Therefore, the binomial filter can allow the creation of algorithms that use integers and perform fixed-point operations with shifts in the bit level. This lighter form of pyramid algorithms can be easily integrated

Figure 14: Original image for testing different Gaussian pyramid filtering versions. Its size is 400×300.

in low computational systems such as mobile phones or cameras. While binomial filters need less computation power, the Gaussian filters provide smoother images. The smoother images, that actual Gaussian filters create, result to even less artifacts and better descriptors and higher performance.

The pyramids created with either Gaussian or binomial filters are not very different in practice though. There may be a small improvement in performance with Gaussian filters from time to time and depending on the images. A demonstration of the two types of pyramid computation is provided in figure 15. The filters used to create this figure is a Gaussian filter that has $\sigma = \sqrt{2}$ and a binomial filter that resembles this Gaussian filter using integer coefficients. It is demonstrated that the impulse response of the two filters is almost identical. The keypoints selected from them are practically the same. The keypoints are collected as Laplacian extrema and those presented are only the most stable ones which remained after Non-Maximum Suppression. In conclusion, the use of both ways of filtering is equally successful and the choice depends on the limitations of applications.

3.3.2 Analysis of the scale factor parameter

The scale factor σ of the levels in the Gaussian pyramid is another important parameter. Gaussian filters with small σ offer more levels and can produce higher pyramidal structures that Gaussian filters with larger σ. Larger σ on the other hand offers smoother images and less artifacts. Also, Gaussian filters with higher σ are more expensive computationally as they are wider and have more important border effects at the smoothened images. The same rules are valid for the equivalent binomial filters. Figure 16 shows the comparison between two Gaussian pyramids made with a Gaussian filter of $\sigma = \sqrt{2}$ and a Gaussian filter of $\sigma = 2$ respectively. The figure shows the first level of each pyramid. Keypoints collected on these first levels of the two pyramids are also shown. As previously,the keypoints are collected as Laplacian extrema and those presented are only the most stable ones which remained after Non-Maximum Suppression. We can see that the pyramid where the levels are created with Gaussian filtering of a larger σ is much smoother but provides less keypoints. The decision of the best σ depends on the task we want to perform and the quality of the images.

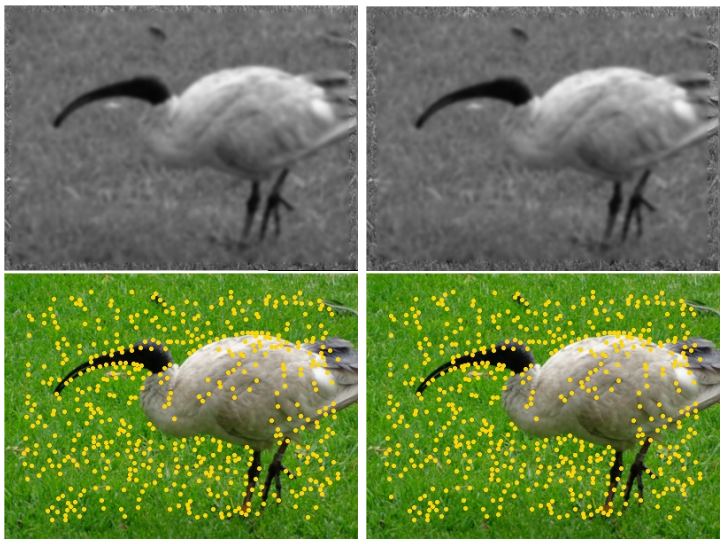

Figure 15: **Left:** Scaling images with an integer binomial filter that resembles a Gaussian filter of $\sigma = \sqrt{2}$. **Right:** Scaling images with a Gaussian filter of $\sigma = \sqrt{2}$. It is obvious that both filters work efficiently for the scaling of the image.
Above left and right: Digital display of the first level of a Half-Octave Gaussian pyramid of image 14. We can see the borders of filtering around each level image.
Below left and right: Under each display of the first level of the pyramid, there is a figure with keypoints found on it. Keypoints are found as Laplacian extrema and away from the borders (unfiltered area). These are the most stable keypoints kept after Non-Maximum Suppression.

3.4 THE LAPLACIAN PROFILE VECTOR

Gaussian derivatives

Gaussian derivatives can be easily computed as weighted differences of adjacent samples on the levels of the Half-Octave Gaussian pyramid [17, 27, 28]. The Gaussian derivatives of an image are commonly computed by constructing a filter by sampling and windowing the derivative of the Gaussian function, and convolving this filter with the image. When computed in this way, derivatives exist over the full range of $sigma$ used in the Gaussian.

$$P_x(x, y, \sigma_k) = P * G_x(x, y, \sigma_k) \tag{15}$$

An alternative method for computing image derivatives is to convolve difference operators in the row and column directions with each level of a Gaussian pyramid [29]. A very close approximation to the gradient (as a combination of first derivatives) can be provide by convolving with the difference operator $[1, 0, -1]$ in the row and column directions over samples at that level of the pyramid. The second derivatives can be provided by convolutions with $[1, -2, 1]$ in the row and column directions of each pyramid level. Similar operators exist for higher order derivatives.

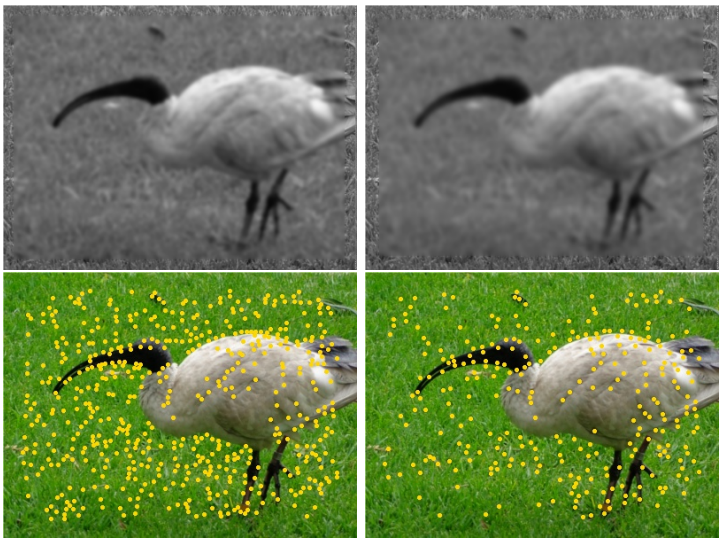

Figure 16: **Left:** Scaling images with a Gaussian filter of $\sigma = \sqrt{2}$. **Right:** Scaling images with a Gaussian filter of $\sigma = 2$. The filter with the larger σ offers better smoothing but less keypoints.
Above left and right: Digital display of the first level of a Half-Octave Gaussian pyramid of image 14. We can see the borders of filtering around each level image.
Below left and right: Under each display of the first level of the pyramid, there is a figure with keypoints found on it. Keypoints are found as Laplacian extrema and away from the borders (unfiltered area). These are the most stable keypoints kept after Non-Maximum Suppression.

The Laplacian of the image $\nabla^2 p(x, y)$ is the sum of the second derivatives in the row and column. When the image derivatives are computed using Gaussian derivatives, this function exists over a range of scales:

$$LP_{xy}(\sigma_k) =< \nabla^2 G(x, y, \sigma_k), P(x, y) > \tag{16}$$

where "$< -, - >$" refers to the inner product operator. We refer to the function $LP_{xy}(\sigma_k)$ as the Laplacian Profile (LP) [29], see figure 17. The LP is invariant to rotation and can be computed at every pixel in an image. When the LP is computed over a logarithmic scale space, it is equivariant to scale [78], see section 2.3. Equivariance in scale means that a change in scale of a pattern in an image will result in a shift of its LP along the σ_k axis. Thus, a sampled LP provides a rotation invariant feature vector that can be used to recognize patterns independent of scale and also used to determine local characteristic scale. The local extrema in the LP over x, y and k corresponds to the keypoints employed by SIFT [84].

The Laplacian Profile (LP)

A close approximation to the Laplacian of Gaussian can be provided by the difference of samples from adjacent pyramid levels in a Half-Octave Gaussian pyramid. For each pyramid sample at levels $k = 2$ to K, a Laplacian can be

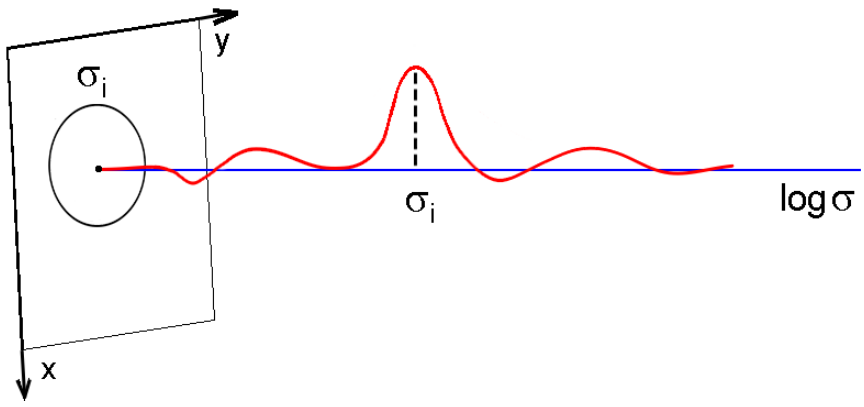

Figure 17: An example of a Laplacian Profile extracted from a continuous scale space. Laplacian values are collected in a vector at every scale σ. The higher the scale is, the larger the corresponding neighborhood described by the Laplacian value is on the original image.

computed by subtracting the pyramid sample at the same image position in level $k-1$.

$$LP(x, y, k) = P(x, y, k) - P(x, y, k-1) \tag{17}$$

The samples in this vector can be interpolated to provide a continuous LP for each sample if desired [29]. Using differences of Gaussians, a Gaussian pyramid composed of $P = 2 \times W \times H$ samples will provide $N = W \times H$ overlapping LP vectors, with lengths ranging from 1 to $K-1$. For level $k = 1$ we cannot use the manner of differences as there is no level below. Though practically, computing the Laplacian on level $k = 1$ by appropriate filters works very well.

3.5 THE RADIAL FOURIER TRANSFORM

The Radial Discrete Fourier Transform (RDFT)

While LP descriptors are invariant to rotation and equivariant to scale, they provide only limited description for visual patterns. Therefore, we employed the Fourier Transform around the locations where we collect the LP elements on the different scales of the Gaussian pyramid. The Fourier Transform is collected radially around an element of the LP, hence it is a Radial Fourier transform. Consequently, we call this part of the descriptor the Radial Discrete Fourier Transform (RDFT).

The Fourier Transform formula for a continuous signal t using angular frequency is:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \tag{18}$$

where ω is the angular frequency. A Fourier Transform can easily be taken radially in two ways. One way is to collect a circle of samples, then consider them in a linear order and compute the 1D Discrete Fourier Transform. The formula, for a sequence of N complex numbers x_0, ..., x_{N-1} that are transformed into the frequency space sequence of N complex numbers X_0, ..., X_{N-1}, interpret-

ing the angular frequency ω in its discrete equivalent $\omega = \frac{2\pi nk}{N}$ and replacing the integral by summation, is given by:

$$X_k = \sum_{m=0}^{M-1} x_m e^{-2\pi i \frac{km}{M}} \tag{19}$$

for $k = 0, 1, ..., M-1$ and i being the imaginary unit. The second way is to collect a disk of samples, convert the disk from Cartesian to polar coordinates and compute a 2D Discrete Fourier Transform. The formula for this is:

$$X_{kl} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{nm} e^{-2\pi i (\frac{km}{M} + \frac{ln}{N})} \tag{20}$$

for $k = 0, 1, ..., M-1$, $l = 0, 1, ..., N-1$ and i being the imaginary unit. The effectiveness of the two ways is examined in subsection 3.5.2.

The coefficients of the RDFT carry redundant frequency information. For example, when computing the X_N Fourier coefficients of an eight element 1D vector x_N, then the X_1 to $X_{(N/2)-1}$ coefficients have the same magnitude and opposite phase with the $X_{(N/2)+1}$ to X_N coefficients. So, if we keep the coefficients X_1 to $X_{N/2}$, we have enough information. Equivalently, this is the same for the coefficients of the 2D Fourier Transform. The fact that Fourier coefficients carry repeating information allows us to keep less coefficients than created and still have the same quality and amount of information.

The selected coefficients from the RDFT can provide magnitudes and phase information separately. A magnitude value is unaffected by rotation, as it only shows the quantity of a certain frequency in the signal. Phase shows the position of the frequency in the signal. The position of the frequency in polar-coordinates can be interpreted as an indication of the orientation of the described neighborhood [147]. The signs of coefficients X_0 and $X_{(N/2)}$ for 1D RDFT, and the respective for 2D RDFT, can also be a very simple but valuable source of information. Instead of the actual Fourier coefficients, we can exploit their magnitudes, signs and phase accordingly in order to keep the part of information that is necessary for a particular visual task.

3.5.1 Radial Fourier on the Image pyramid or the Laplacian pyramid?

One of the matters that that we examined is if we can use very small support areas for the RDFT. For example, if we want to use the 4 closest neighbors at radius=1 for computing the RDFT, is this neighborhood enough to offer valuable information? Therefore, we test if it is better to use another encoding for the image signal for very small sampling neighborhoods for the RDFT. We use the Laplacians of a sampling neighborhood instead of its original intensity values, despite the fact that the Fourier Transform is usually computed on the image signal. We base this experiment on the fact that the Laplacian can provide more important information as it encodes changes in the pixel intensities. Encoding the changes in pixel intensities gives a description not just for the respective pixel intensities but also the relationship between them

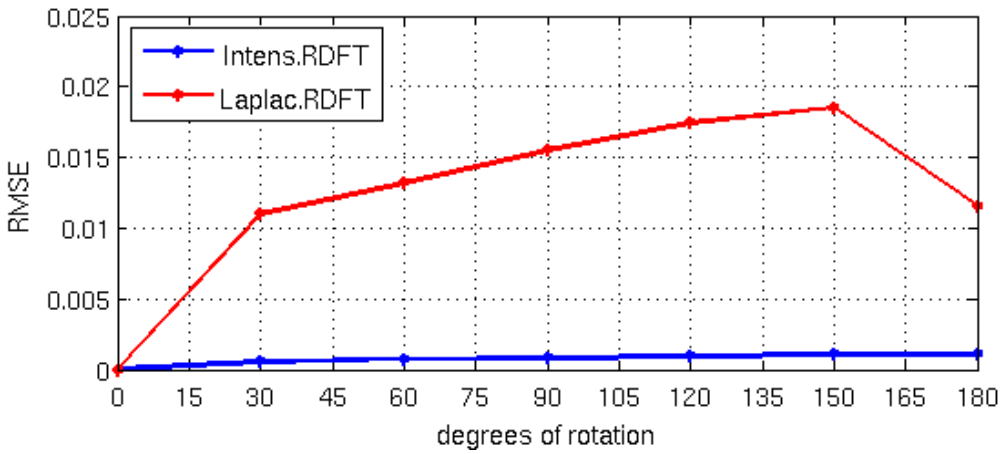

Figure 18: Rotation experiment for the first image from each case folder at Affine Covariant Features dataset [91]. The RMSE of the one-level descriptors of sampling area at radius = 1 for the RDFT, using intensity values (blue line) and Laplacian values (red line). In both cases, we keep only the magnitude values from the RDFT coefficients to make sure we have rotation invariant features. It is shown that according rotation changes, the RDFT is better to be computed on intensity values.

The Laplacian pyramid

and surrounding pixel intensities. We use the term Laplacian pyramid to express that all samples on the Gaussian pyramid can be represented by their Laplacian values and the assemble of them can be seen as a new pyramidal structure. For wider sampling areas, the idea of using the Laplacian pyramid for the computation of the RDFT is not necessary. On wider sampling areas, the original signal can provide enough information, while the Laplacian of it will show only edges and keypoints. Therefore, we do not test the same idea for sampling areas wider than radius = 1. The final concept is that for small neighborhoods, the analysis of the frequencies involved in the Laplacian values can give more important signal information.

The drawback with this idea is that as the Laplacian values in the Gaussian pyramid can be calculated with either DoG or filters. The computations of both the DoG method and the filters is biased due to always using the rows and columns configuration on images for computaions. Therefore, the descriptor will necessarily loose some of its rotation invariance. In figure 18 we perform a test on the rotation of a set of images for evaluating the idea of using Laplacian values of the pyramid samples to compute the RDFT against using the actual values of the samples in the pyramid. We use the Root Mean Square Error (RMSE) to compare the descriptor vectors. Keypoints are collected with DoG within a region in the image that is sure not to suffer by border effects while rotating the image. Descriptor vectors are created using only one pyramid level. Tracing the same keypoints on the rotated images, we collect a Laplacian on each keypoint for the LP and then the 4 closest neighbors around the keypoint for the RDFT. In one case we collect the actual pyramid samples' values of the 4 neighbors and in the other case the Laplacian values of the 4 neighbors. We compute the 1D RDFT, considering the 4 values as if they were

in a line, as explained later in subsection 3.5.2. In both cases, we keep only the magnitudes of the appropriate subset of Fourier coefficients that do not involve repeating informations (see the introductory paragraph of section 3.5 for explanation). The descriptor vectors in both cases have in total 5 elements. In figure 18 for the results on the test for rotation changes, we can see that there is an important deterioration to the performance of descriptors using Laplacian values for the RDFT. The result of the test for rotation shows that the idea of using Laplacian values for the computation of the RDFT is not appropriate for local description as it undermines rotation invariance. On the other hand, the smaller tolerance to rotation changes is a hind that Laplacian samples for the RDFT on a small area might be strong in discrimination power. In chapter 5, we experiment with these two approaches on global image description with a task where orientation information accumulated in the descriptor vectors improves performance.

3.5.2 RDFT from sampling on a circle or on a disk?

We experiment with two sampling patterns for the RDFT, a set of samples on the periphery of a circle and a disk of samples. Each of the two ways showed different qualities. When sampling on a circle, we can treat the RDFT as an 1D function exploiting the collected samples in a line. When sampling on a disk, the samples are mapped to the polar coordinate system and the 2D RDFT can be computed there as usual. The circle sampling is faster than the disk sampling for the RDFT but disk sampling encodes information more densely.

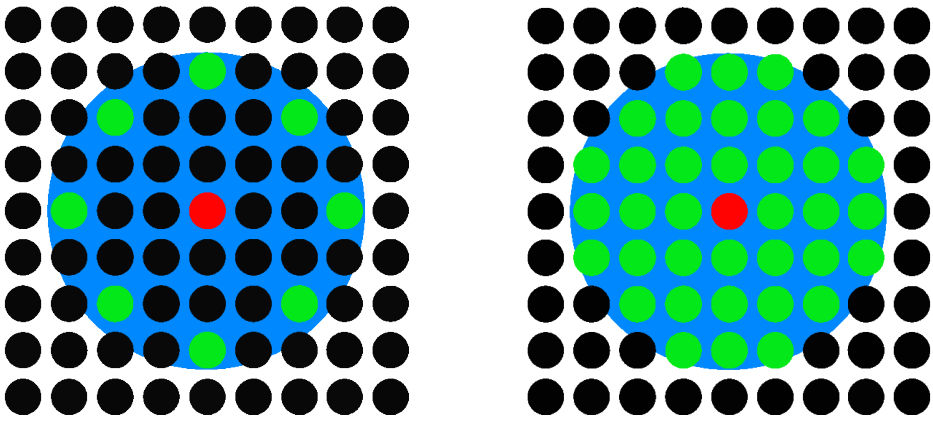

Figure 19: The two tested RDFT sampling areas of the descriptor on one level. The central red dot in each pattern represents an LP element (as the location on a region of a pyramid level where this Laplacian of Gaussian was computed). The green surrounding dots represent the sampling areas for the RDFT around an LP element. The radius of the circle where they are sampled is the same for both sampling patterns, the circle and the disk. In this example, we collect eight samples on the periphery of the circle.

We use the RMSE to compare descriptors with sampling on a circle or on a disk for the RDFT. The RDFT in both cases was computed using the actual

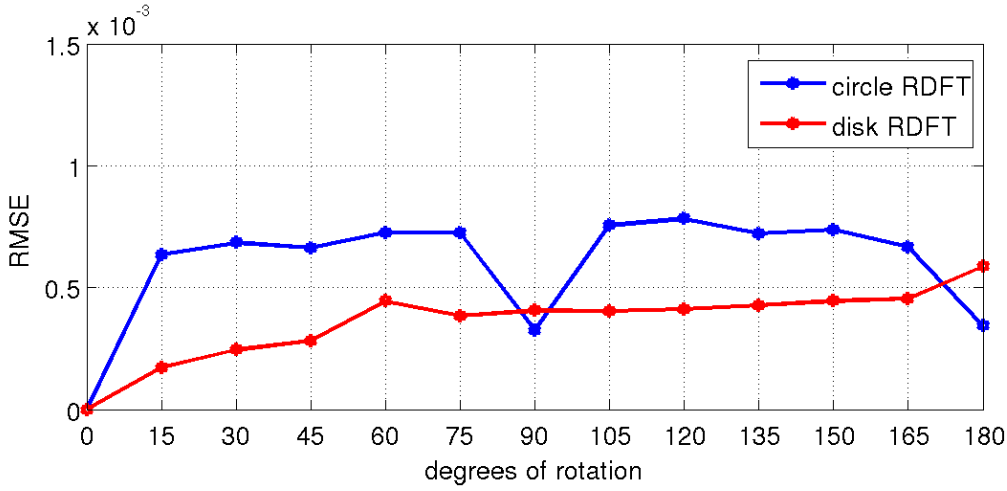

Figure 20: Rotation experiment for the first image from each case folder at Affine Covariant Features dataset [91]. The RMSE of the one-level descriptors with sampling area at radius = 5 for the RDFT, performing sampling on the periphery of the circle or in the entire disk area. In both cases, we kept only the magnitude values from the RDFT coefficients to make sure we have rotation invariant elements. It is shown that according rotation changes, the RDFT from samples on a disk can handle better the situation but it requires a lot more vector elements.

values of the Gaussian pyramid samples. For the case of sampling on the periphery of a circle, we collect 8 samples as seen in figure 19. Keypoints are collected with DoG within a region in the image that is sure not to suffer by border effects while rotating the image. Descriptor vectors are created using only one pyramid level. The elements in both types of the descriptor vectors consist of one Laplacian value for the LP and the magnitudes of the appropriate subset of Fourier coefficients that do not involve repeating informations (see the introductory paragraph of section 3.5 for explanation). Consequently, the descriptor vectors made with the sampling on the circle periphery for the RDFT have a total of 6 vector elements, while the descriptors using sampling on a disk have 17 vector elements. Figure 20 shows that the disk sampling performs better description. Sampling on the circle periphery is less expensive in time and memory cost but performs less good. The decision for the best choice is not simple, especially if we consider the difference in the vector sizes of the two different descriptors. The final decision is better to be taken upon the particular limitations of an application, where the cost of computation can drive to the use of slightly less but more compact (meaning shorter) descriptor vectors. Circle sampling for the RDFT is probably the wisest solution as the aim of this thesis is to make a flexible descriptor that is easily used to several applications including those on machines with low memory capacity and computation power.

3.5.3 Size of the RDFT sampling area

The support area of the proposed descriptor is variable and can be set according to the needs of an application. The intuition behind this arrangement is that the type of visual information needed to perform a task changes according to the task. For example, the visual information needed to perform keypoint matching is very localized and should be described in detail, while that for scene recognition is spread all around the image and very small local details do not matter. As we want to make a descriptor that is flexible to fit to different tasks, we can test several possibilities for the radius of the RDFT sampling area that extends the support area of the descriptor around the LP.

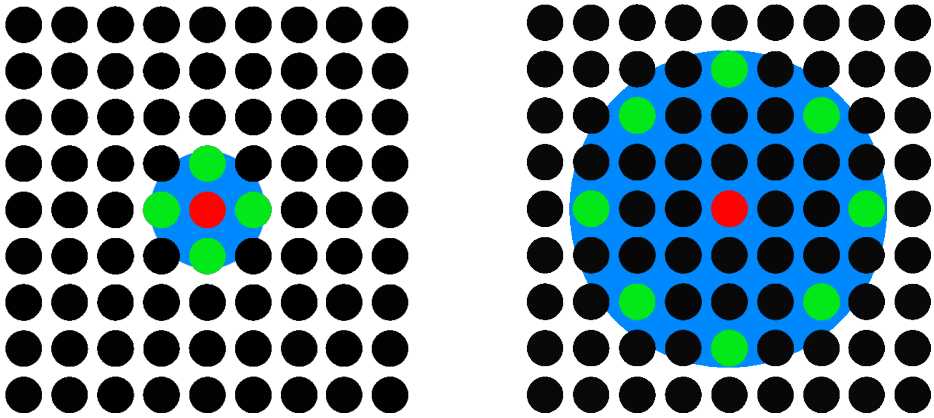

Figure 21: Two possible RDFT sampling areas of the descriptor on one level. The central red dot in each pattern represents an LP element (as the location on a region of a pyramid level where this Laplacian of Gaussian was computed). The green surrounding dots represent the sampling areas for the RDFT around the LP elements. **Left:** 4 neighbors are taken in linearly around an LP element for the 1D RDFT, representing the circle with the smallest possible radius. **Right:** A circle with wider radius is taken around an LP element for the 1D RDFT.

Figure 21 shows two possible sampling ways at different radius collecting different number of samples. Very small sampling areas for the RDFT, for example 4 closest neighbors of a pixel's location, can create very small descriptor vectors. These small vectors are not expected to be very strong in discrimination power but plugged into an appropriate learning method (for example SVM, Adaboost, etc.) can be efficient for pattern/object recognition (the same way as Haar wavelets or Gaussian derivatives). We expect that tasks with low resolution image signal require wider sampling areas for the RDFT. The reason why wider sampling areas should work better for low resolution, is that low resolution misses details but contains smoother shape information. Smooth shapes without details (for exmaple, as in textureless images) need to be described by a method that can extend enough on the image to describe them correctly, so for this case relatively wider sampling areas for the RDFT are expected to perform better.

3.6 CONCLUSIONS: THE LP-RDFT IMAGE DESCRIPTOR

In this chapter, we presented our proposed method for image description. The main purpose of our proposed method is to be easily adjustable to diverse applications, while existing descriptors usually need a fundamental change in order to be adjusted to diverse tasks. We propose a method that offers mathematically correct image description and features that can be adjusted to different needs. The benefits for such a method is that the produced information can be adapted by different systems for different tasks and that an application can perform different visual tasks by extracting only one type of features.

Laplacian Profile and Radial Discrete Fourier Transform (LP-RDFT)

We showed in this section how the Laplacian of Gaussian can be combined with the Fourier Transform in a new flexible description method. We name the resulting descriptor Laplacian Profile and Radial Discrete Fourier Transform descriptor (LP-RDFT). The LP length, the RDFT sampling area, the RDFT information to be kept and eventually the vector length are all variable and adjustable. Consequently, the support area of the descriptor is variable. The generalized procedure for the extraction of one local LP-RDFT descriptor vector is presented with algorithm 1. Though the parameters can be set according to the requirements of applications, a particular set of well performing values can be recommended after deeper experimental exploration. Therefore, for the recommendation of particular values for the parameters of the descriptor, we will perform a series of experiments in the following chapters.

In the next chapter, the first of the experimental chapters of this thesis, we will see keypoint matching experiments. We will experiment on the well known and widely used Affine Covariant Features dataset and textureless images from the MIRFLICKR Retrieval Evaluation dataset [56]. We show how our proposed method can be adjusted to different matching tasks and perform equally to the state of the art with extremely small vector length.

Parameters:
vector v_{LP} = the LP vector with all the collected Laplacian of Gaussian values for a descriptor vector;
length L_{LP} = the length of v_{LP} (L_{LP} is a variable for LP-RDFT that defines the final descriptor vector);
set EL = the set of Laplacian of Gaussian elements of v_{LP} around where the RDFT will be computed;
radius RR = the radius of the sampled area for the computation of RDFT;
pyramid sample a = a sample on the pyramid defined by the level (scale) k_a and location (x_a, y_a) on this level;
coordinates (x_a, y_a, k_a) = the coordinates of a;
Input data: an image;
Output: a local LP-RDFT descriptor vector $v_{LP-RDFT}$

compute a Half-Octave Gaussian pyramid from the input image;
for *each adjacent level from* k_a *until* $k_a + L_{LP}$ **do**
  compute the Laplacian of Gaussian value of the pyramid sample that correspond to a;
  add the computed Laplacian of Gaussian to v_{LP}
end
initiate $v_{LP-RDFT}$ with v_{LP};
for *each element* $e \in EL$ **do**
  sample an area of radius RR at the location and scale where e was computed ((x_a, y_a, k_a));
  compute RDFT on the sampled area (section 3.5);
  keep frequency information in a subvector v_{sub};
  concatenate v_{sub} to $v_{LP-RDFT}$;
end
normalize the $v_{LP-RDFT}$;
return $v_{LP-RDFT}$;

Algorithm 1: The generalized procedure of extracting a local LP-RDFT descriptor vector on a selected location in an image.

LA MISE EN CORRESPONDANCE DE POINTS-CLES AVEC LP-RDFT

Les résultats expérimentaux pour la correspondance de point-clés sur les images texturées de la base de données Affine Covariant Features benchmark ont montré que LP-RDFT fonctionne efficacement tout en ayant une très petite longueur du vecteur. LP-RDFT surpasse l'état de l'art pour les changements d'échelle et les changements pertinentes de l'image à l'échelle, comme l'augmentation de flou et la compression JPEG. D'autres tests sur les images sans texture de la base de données MIRFLICKR Retrieval Evaluation ont montré que LP-RDFT bat l'état de l'art pour de petites valeurs de rotation et mise à l'échelle mais ses performances se détériorent pour des valeurs plus grandes. Le fait le plus important est que la taille du vecteur de LP-RDFT pour les tests d'images sans texture est particulièrement faible avec seulement 7 éléments réels. Les résultats montrent que LP-RDFT est une bonne solution pour la description locale lorsque le coût de la mémoire est une question importante.

4 KEYPOINT MATCHING WITH LP-RDFT

4.1 SEARCHING FOR THE DETAILS THAT MAKE THE DIFFERENCE

Using salient locations in images means preselecting the important details of an image. This preselection is called detection of salient locations and it is followed by the process of description. This type of image description is local, as explained in section 2.2. The description at the salient locations has the purpose to attribute each on of the locations with a characteristic vector in order to be recognized again in other images. A comparison measure decides which are the similar descriptor features between two images and matches them in a pair. The matched salient locations in two images are considered to be the same locations in the 3D world. A simplified complete visual system representation that performs keypoint is given in figure 22.

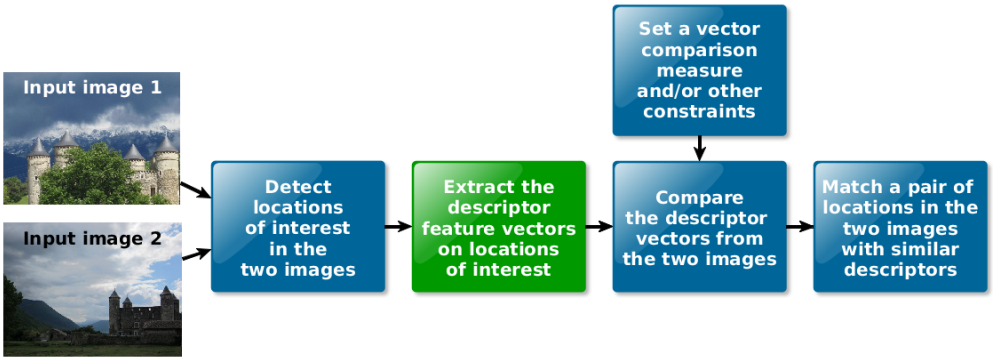

Figure 22: A simplified representation of the visual system for keypoint detection. The green node is the step of the system that the proposed descriptor is involved in.

No matter how strong a description method is, there are always some hard cases and ambiguities that cannot be easily overcome by a simple descriptor vector matching. Figure 23 shows that salient locations can be lost or new can appear in images that have undergone transformations. This is the results of the digital exploitation of the images. Rounding and approximations can alter slightly the the signal information. The treatment of the image always in rows and columns is not tolerant enough to these changes. Another problem is ambiguities in the selection of the salient locations due to the same reasons. salient locations can be collected slightly translated or in different scales. Figure 24 shows cases where the salient locations can be found shifted on the x and y directions and on different scales. The final decision of a good match depends on the objectives of the application. For example, if the purpose of an application is to classify images by their content using local descriptors, then

the salient locations do not need to be extremely precise. On the other hand, an application that creates long panorama images, by stitching many shorter ones, need very precise salient locations. Otherwise the panorama image will look like a terrible patchwork. The result of matching salient locations can vary in precision according to the constraints set by applications.

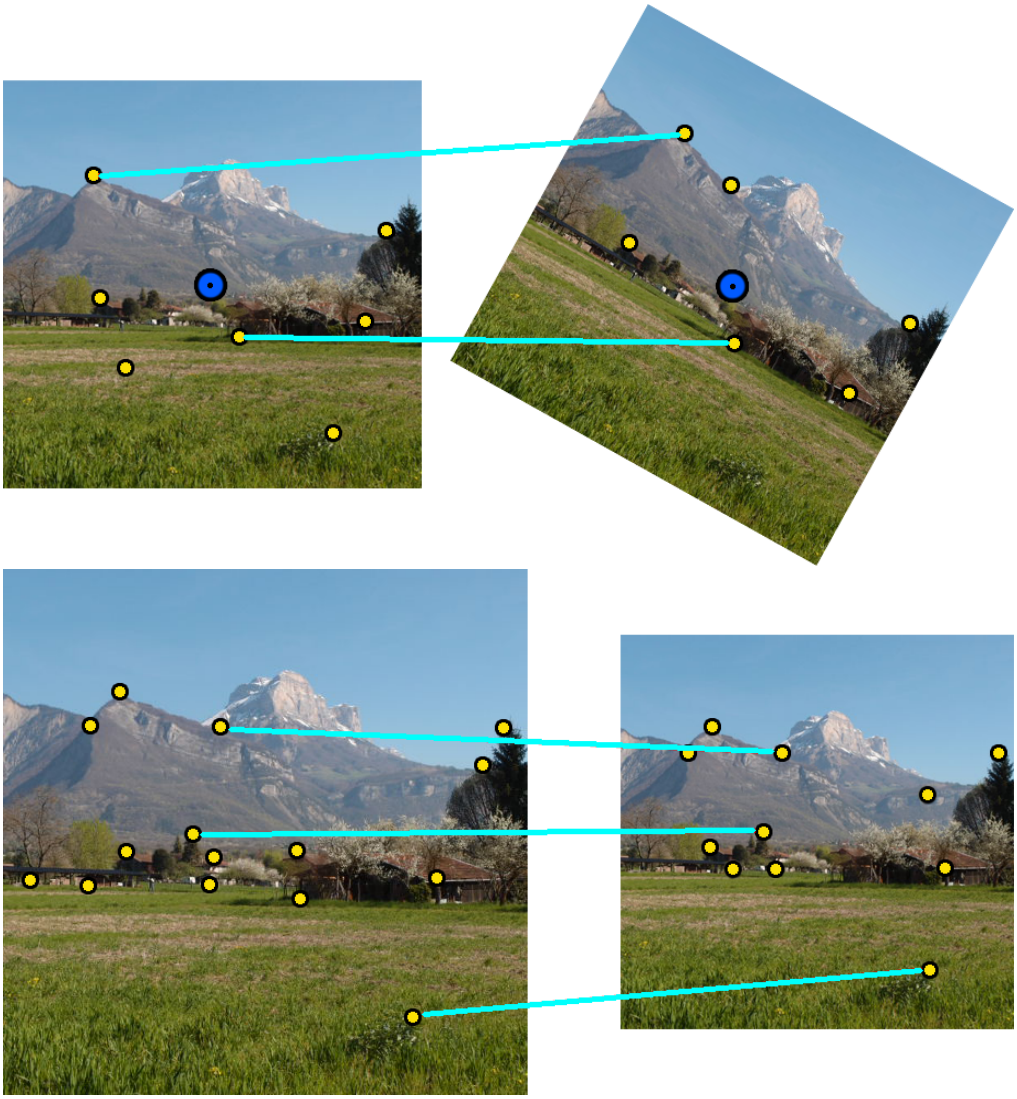

Figure 23: Image transformation can make salient locations disappear or new to appear. **Above:** Rotation of the image with rotation center the center of the image (blue target in the center). The rotation of the image causes new salient locations to appear and other to disappear. **Below:** Scaling of the image. Scaling usually causes salient locations to disappear as high frequencies are successively removed or new to appear as large regions become small points.

Image descriptors are designed to be invariant and robust, but the lose or appearance of new visual details and ambiguities (caused by the digital treatment of images) make the situation harder. A usual approach is to use as much detail as possible around the salient location, by either expanding the support area of the descriptors or calculating very detailed costly functions than need

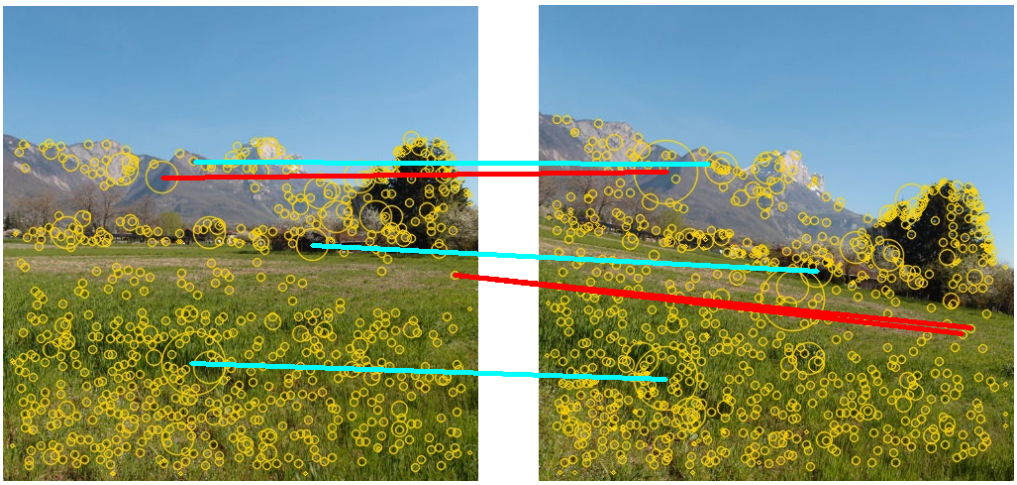

Figure 24: Are all the above location matches valid? The size of the above yellow circles is relevant to the scale where the salient location was detected. Image transformation can cause salient location in different images to be found shift in the x and y directions and on different scales. The salient locations that seem the same to a human can be considered as different to a program. The decision if a match is good or not depends on the constraints of an application.

dimensionality reduction. As embedded computing and mobile computing applications for computer vision increase in popularity, memory requirements have emerged as an important issue. Binary descriptors [76, 112, 19, 18] make a change to this concept by calculating small vectors of binary elements that are easily stored in single bits. Sparse coding is another answer to reduce the cost of descriptors by re-encoding vectors more compactly. Our concept for salient location matching is that we can use our proposed theory to make a descriptor that is already capturing the visual signal in a compact form. This way, there is no need for further reduction in their size, for example with sparse coding or PCA. Additionally, binarization could be made later given an appropriate functions to further reduce the cost of the descriptor. Consequently, we work on developing the LP-RDFT descriptor able to encode image signal information, apart from correctly, very compactly in the first place.

4.2 EXPERIMENTS

4.2.1 Descriptors for comparison

In order to make an evaluation of our method we choose a set of well known or newly proposed image descriptors and we compare to LP-RDFT on keypoint matching. The methods are SIFT, SURF, ORB, BRISK, FREAK, BRIEF and NSD. The NSD descriptor is a new descriptor so we decided to use it in both its real and binary form, named the Seed of Life and the Binary Seed of Life. For details on the descriptors, subsection 2.4.1 provides a summary of their theory and subsection 2.4.3 arrange them into the general plane of image description. SIFT (subsection 2.4.1.2) was selected first of all the other descriptors for the *SIFT*

comparison tests. This descriptor and its variants have dominated the field for fifteen years. SIFT uses orientations of image gradients for local edge de-

SURF

scription. SURF (subsection 2.4.1.3) is a good alternative of SIFT and also well established. It finds orientation from first order Gaussian derivatives and describes the region with second order Gaussian derivatives. It competes well

BRISK, FREAK

with SIFT in robustness thought still SIFT shows better invariance. BRISK and FREAK (subsection 2.4.1.1) are recent well performing and included in many comparison tests in the literature. They are both intensity based descriptors

ORB, BRIEF

but use gradient orientations to achieve rotation invariance. BRIEF and ORB (subsection 2.4.1.1) are two binary and intensity based descriptors. ORB is a version of BRIEF that is better performing against rotation changes. Though both being binary, they are proven to perform well compared to the rest of the state of the art while having the advantage of low memory needs. Finally, NSD is very new but it seems as a very promising new type of descriptors that shares the idea of adjustability with the proposed descriptor, though perceived in different way. NSD starts from a keypoint and spreads its support region as fas as possible according to a preset plan. But at the end only a part of the

SOL, BiSOL

whole possible support region is kept according to a decision measure. Due to being a new descriptor, we use NSD (subsection 2.4.1.2) in both its proposed forms, the real SOL (Seed of Life) and the and binary BiSOL (Binary Seed of Life), in order to be more objective with its performance. In the experiments, we compare the performance of these descriptors on textured and textureless images.

4.2.2 The LP-RDFT version for matching

In order to create an appropriate version for LP-RDFT, we consider the constrains of our application. We need a highly discriminative robust descriptor vector with as small vector length as possible. From the analysis of the parameters in chapter 3, we conclude that the descriptor can be made by sampling on a circle for the RDFT on the Gaussian pyramid. We create the LP-RDFT using a Half-Octave Gaussian pyramid created with a Gaussian filter $G(x, y, 2^k)$, where k is the number of the pyramid level, as explained in section 3.3.1. We use 8 neighbors sampled on the Gaussian pyramid and calculate an 1D RDFT. From the 8 neighbors x_0,x_1, ..., x_7, we take 8 Fourier coefficients X_0,X_1, ..., X_7. We keep the absolute value (magnitude) of X_0, the sign of X_4 and the magnitudes of X_1,X_2 and X_3. The absolute value of X_0 and the sign of X_4 are a measure of the sum of intensities of the 8 neighbors. The magnitudes of X_1,X_2 and X_3 provide important frequency information. X_5,X_6 and X_7 provide redundant information and are therefore discarded. Also, we discard all phase elements which are sensitive to rotation changes. We finally concatenate everything in a vector and normalize with the L_2-norm.

We perform exhaustive testing in order to find the best combination for the size of the LP and the radius for the RDFT sampling area. Another parameter that we tried in this experiments was how short we can make the descriptor by collecting RDFT information from only a subset of levels around the LP. Exper-

iments showed that collecting RDFT information from the half of the levels, the higher levels, provide a more robust image description. The explanation is that the discrimination power of the descriptor is enhanced by samples collected on high scales where lower frequencies remain and give a general view of the surrounding image signal. In the same time, without the high frequencies which are discarded in higher scales, the descriptor is less subjected to smaller signal details which suffer the most from image transformations. We perform exhaustive experiments separately for the two types of images, textured and textureless, in order to see what is the best choices for each type. The double experimentation on textured and textureless images will lead us realize where the strengths and the weaknesses of the proposed method lie.

Parameters:
list KP_{im} = all keypoints detected in an input image im;
matrix H = the homography matrix from imA to imB;
point kp = a keypoint (x, y, σ) in KP_{im};
Input data: *image* imA = an input image of a scene;
image imB = a different input image of the same scene;
Output: *list* $correct_matches$ = all correctly matched pairs (kp_a,kp_b);

```
for each input image im do
    compute a local descriptor vector for each kp in KP_im;
end
for each kp_a in KP_imA do
    for each kp_b in KP_imB do
        match kp_a to the kp_b with the most similar descriptor vector
        (smallest vector distance);
        if kp_a and kp_b are matched then
            use H to project kp_b on imA kp_b on imA = kp_b * inv(H); if kp_b
            is projected where the kp_a is on imA then
                kp_a and kp_b are a correctly matched pair;
                add matched pair (kp_a,kp_b) to correct_matches;
            end
        end
    end
end
return correct_matches;
```

Algorithm 2: The followed matching procedure between two images. The procedure is the same for LP-RDFT and all the competing descriptors.

4.2.3 The procedure of comparison

The measures we use for the evaluation are repeatability [92, 60] and recall against 1 - precision [91]. Repeatability shows how good is a method to find correct matches considering possible correct matches and it is a percentage.

Recall against 1 - precision plots show how important is the quantity of correct matches found by a method considering the quantity of false matches. Repeatability and recall are relevant measures, though recall is expressed into space $[0, 1]$. 1 - precision is also expressed in space $[0, 1]$. The manner we calculate 1 - precision is different by the way described in [91]. In [91], 1 - precision is calculated with respect to the total number of matches between two images. We reduce this number to the total matches that can occur only with keypoints that exist in both images. This changes the form of the curves that we created for recall versus 1 - precision compared to the ones seen in [91], where the curves tend to expand from low reacal and 1 - precision towards high recall and high 1 - precision. The curves of In [91] show that while more matches than can be really made correctly are allowed to happen, the precision of this matching is low. This is expected because if for example we use keypoints that are not existing in the second image, these keypoints can be matched to other keypoints in the second image incorrectly. The curves we create do not always give smooth curves and they are interpreted by looking at the the area they are spread in the graph. The curves we create show the amount of correct matches that can be obtained with the keypoints existing in both images and with how much precision. Also, it must be mentioned that the number of matches from keypoints existing in both images is more related in size to the real correspondences number than to the total matches number from all keypoints in the images. Though different, the curves we created do not undermine the ranking of descriptors as can be found in the literature.

The formulas for repeatability, recall and 1 - precision (the way we calculate it), are:

$$\text{repeatability} = \frac{\text{correct matches}}{\text{possible matches}} \tag{21}$$

$$\text{recall} = \frac{\text{correct matches}}{\text{true correct matches}} \tag{22}$$

$$1 - \text{precision} = \frac{\text{false matches}}{\text{correct matches} + \text{false matches}} \tag{23}$$

with 1 - precision formula computed only for the keypoints existing in both images.

We use the Euclidean distance for the matching of the descriptor vectors. For all the descriptors except NSD, we use the OpenCV library [14]. For NSD we use the source code provided by the authors. The parameters for the descriptors are kept in their default values, trusting that their authors and developers have made the best choices. For LP-RDFT, after exhaustive experimentation on the dataset with the mentioned measures, we concluded that the best performance was given by combining LP vectors of length 7 (so 7 exploited pyramid levels) with RDFT from samples at radius 5 pixels around the LP coordinates for the higher 4 of the 7 exploited pyramid levels. These choices create a descriptor vector of only 27 elements. In order to compare vector lengths, the smallest descriptors in the rest of the testing set are the binary descriptors ORB and BRIEF with 256 binary elements (bits) stored in 32 bytes (OpenCV implementation). For each descriptor we use the proposed by their authors or developers detector for keypoint detection in the images. For LP-RDFT, we

collected keypoints with the DoG method on the levels of the created image pyramid.

4.2.4 Textured images

We use the Affine Covariant Features benchmark dataset [91] for the experiments. The test is keypoint matching between images. The testing protocol is comparing the first image in each one of the different case folders with the rest of the images in the same case folder. The procedure of matching keypoints between two images is briefly presented with algorithm 2. The images in this dataset are characterized with a lot of texture, meaning that there are a lot of variations on the image signal that provide rich information for its content. The results of keypoint matching is shown in 8 figures, numbers 25 to 32. Each figure has 2 plots, one plot with the repeatability measure for each pair of compared images (image 1 to another image in the same case folder, characterized by an index number) and one plot with the recall against 1 - precision.

As we can see from the figures 25 and 26, for the cases of increasing blur ("bikes"and "trees"), LP-RDFT works very well compared to the state of the art, with competitive rates for both correct and false matches as depicted by the repeatability and recall against 1 - precision plots. Actually in figure 26 for the set "trees", LP-RDFT outperforms the other methods. In both cases, LP-RDFT outperforms SIFT. Figure 32 for the case of increasing JPEG compression ("ubc"), shows that LP-RDFT has a very competitive performance compared to the state of the art, with high readability and not many false matches. For high JPEG compression, LP-RDFT outperforms all other descriptors. These results show that LP-RDFT performs very good when information is lost due to bad resolution or compression, regardless of its very small vector length.

Figures 27 and 28, for the cases of viewpoint changes ("graf"and "wall"), the proposed method works relatively lower compared to the state of the art. Though, LP-RDFT has relatively low false positives as show by the recall against 1 - precision plot. Figures 29 and 30 for zoom and rotation changes ("bark"and "boat"), show also lower results than the state of the art. Finally, figure 31 for the case of decreasing light ("leuven"), shows that the proposed method has lower performance compared to the state of the art.

The conclusion is that LP-RDFT performs competitively to the state of the art with a very small vector of only 27 elements, especially when the higher frequencies of an image are lost. It works well for blur and JPEG compression, which are relevant to scaling of the image. On the contrary, it is weaker than other methods at viewpoint changes and light variations.

4.2.5 Textureless images

Textureless images are a difficult subject in image description and due to the lack of obvious interesting information. Usually, these images are characterized by lack of meaningful texture (they may have noise, which can be considered as meaningless texture), smooth edges and large homogeneous areas. In

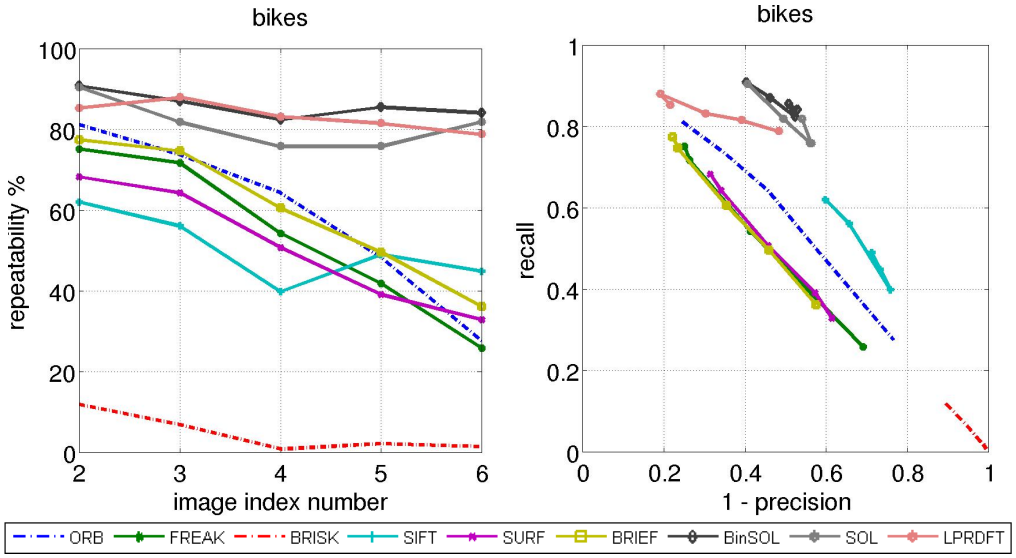

Figure 25: Affine Covariant Features dataset - Blur ("bikes"). Keypoint matching of image 1 to the rest.

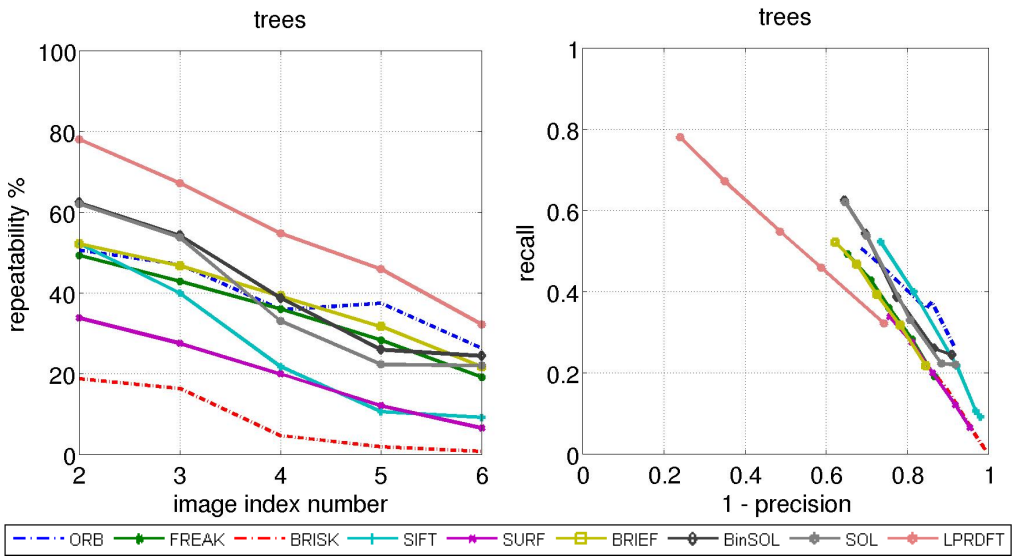

Figure 26: Affine Covariant Features dataset - Blur ("trees"). Keypoint matching of image 1 to the rest.

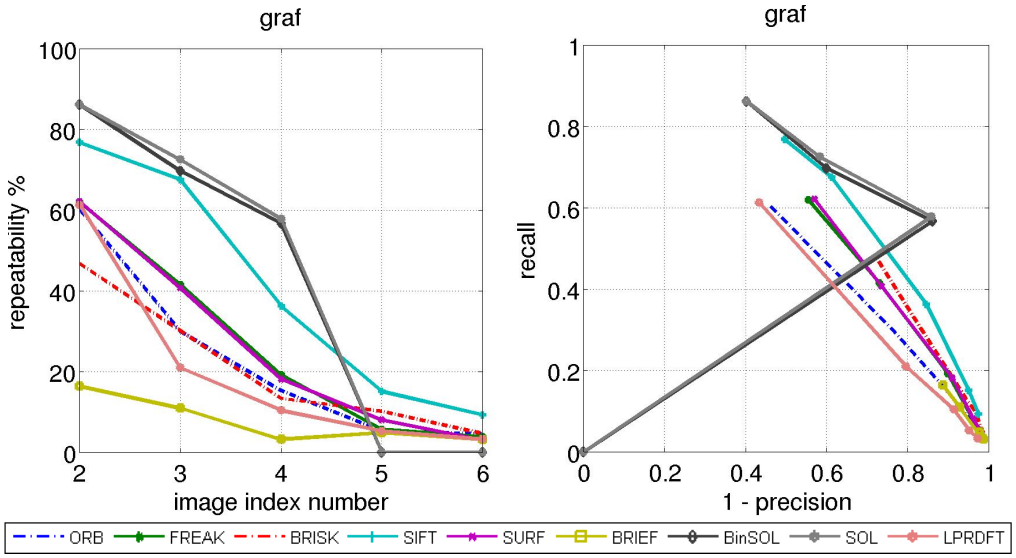

Figure 27: Affine Covariant Features dataset - Viewpoint ("graf"). Keypoint matching of image 1 to the rest.

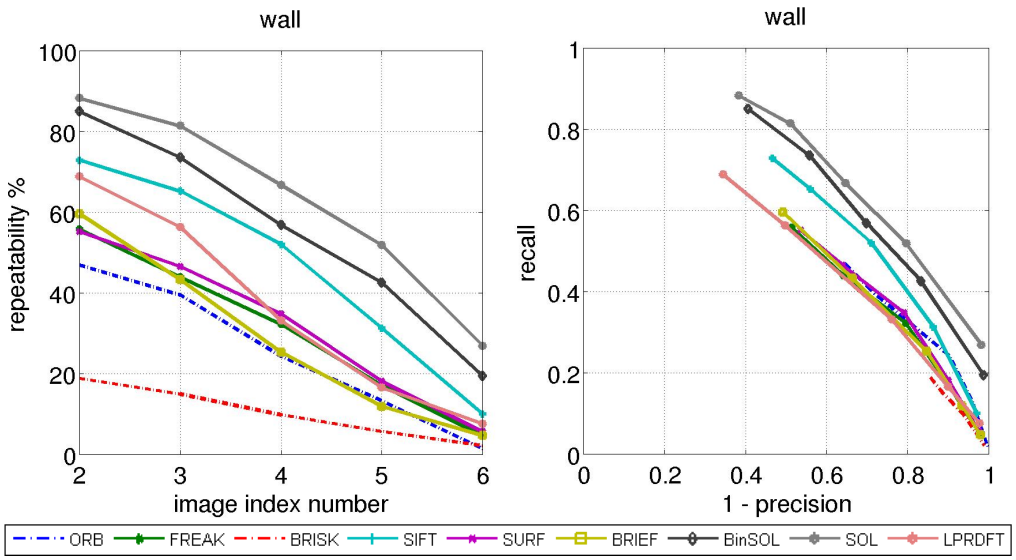

Figure 28: Affine Covariant Features dataset - Viewpoint ("wall"). Keypoint matching of image 1 to the rest.

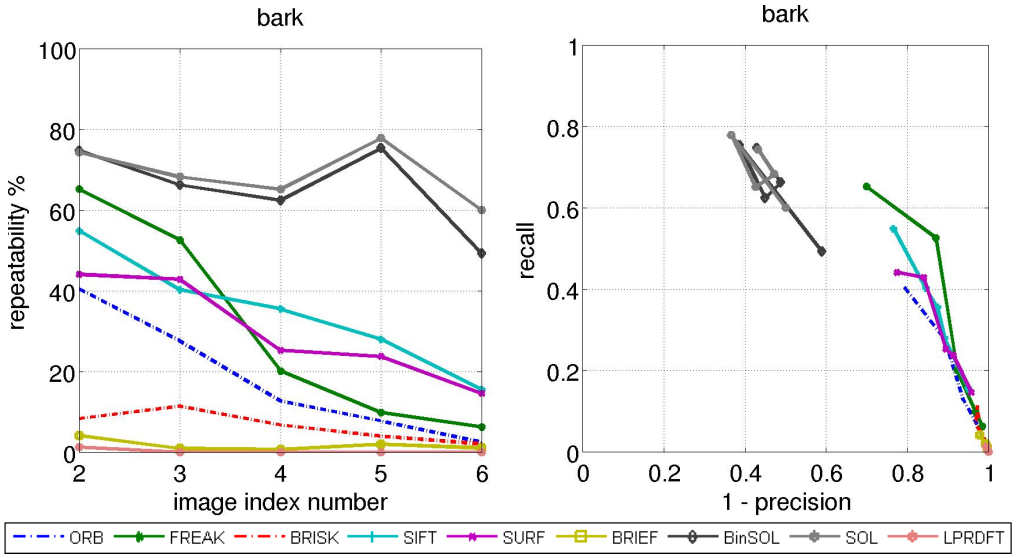

Figure 29: Affine Covariant Features dataset - Zoom + rotation ("bark"). Keypoint matching of image 1 to the rest.

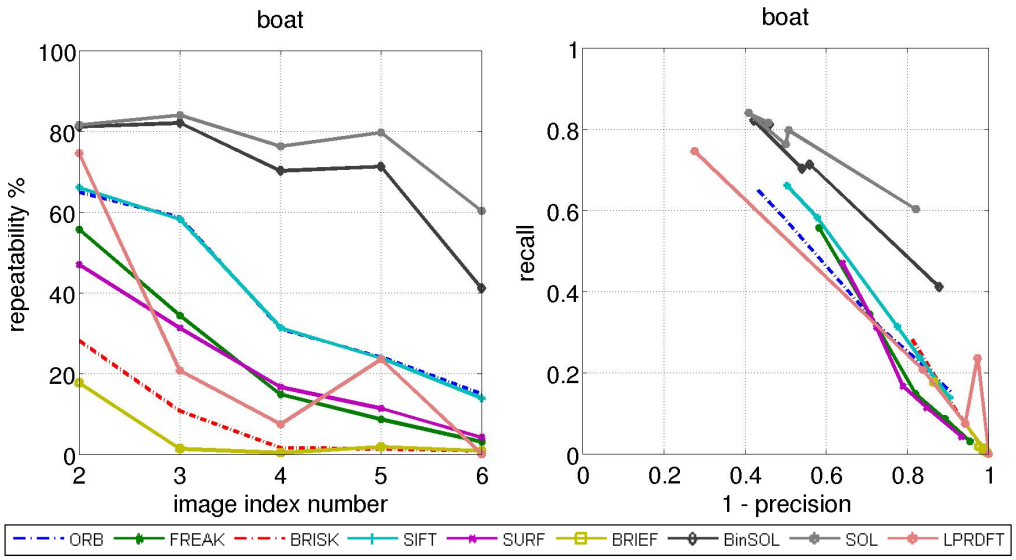

Figure 30: Affine Covariant Features dataset - Zoom + rotation ("boat"). Keypoint matching of image 1 to the rest.

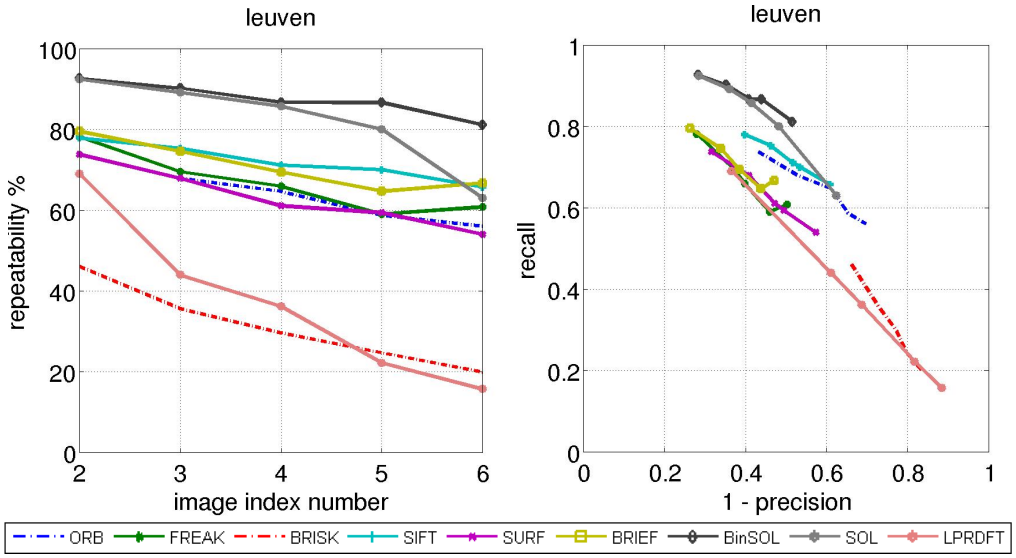

Figure 31: Affine Covariant Features dataset - Light ("leuven"). Keypoint matching of image 1 to the rest.

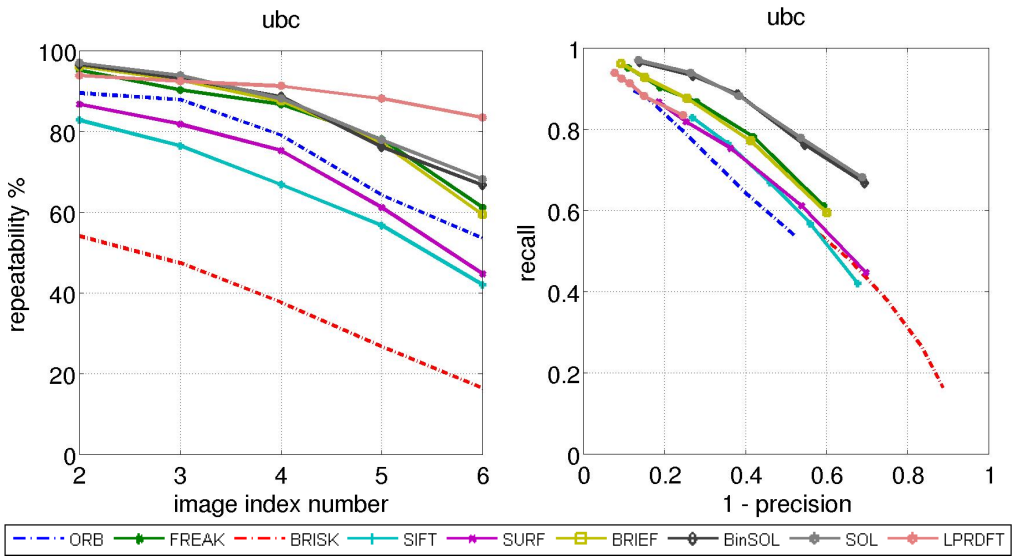

Figure 32: Affine Covariant Features dataset - JPEG compression ("ubc"). Keypoint matching of image 1 to the rest.

order to make a more general testing of the descriptors we have studied in the previous section, we collected a set of images from the MIRFLICKR Retrieval Evaluation dataset [56] with these characteristics. The collected textureless images shown in figure 33.

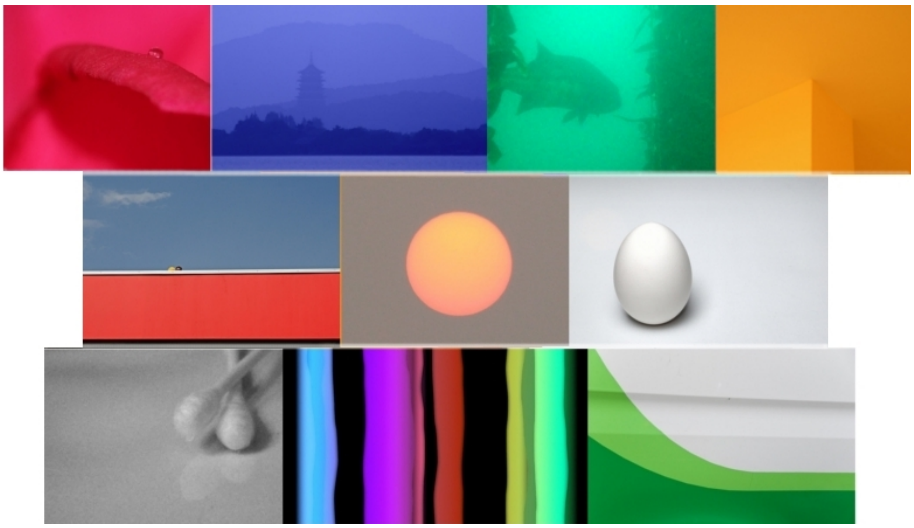

Figure 33: Images from MIRFLICKR Retrieval Evaluation dataset with low texture information.

The tests include rotation of the images from 0° to 180° every 30° and scaling of the images from 4 times bigger to 4 times smaller with scale factor $\sqrt{2}$. The relevant figures are 34 and 35. The same measures are used for evaluation. All compared descriptors are used in the same way as before. Concerning the proposed descriptor, we use another shorter version of LP-RDFT due to the size of the images. Also, the radius, where we collect the samples for the RDFT, is a little bigger; 6 pixels instead of 5. Again, the parameters are chosen after exhaustive tests. We use two element long LP vectors, so two exploited levels on the image pyramid, and the Fourier information from only the one highest of the 2 used pyramid levels. The final vector is a tiny with only 7 elements!

From the plots we see that all methods do not work at their best, with low recall and repeatability. The low recall against 1 - precision measures shows that there are not many keypoints detected, so the matching has been performed with very low numbers of keypoints. Surprisingly, for all descriptors except LP-RDFT, the matching performance on the same image (original image to its self) is very low. This can be explained by the lack of meaningful signal information in this images that causes many of the descriptor features to look alike and cause false matches. The proposed method performs almost perfect for matching on the original image, which shows that it can handle low quality information. Its performance though deteriorates with rotation and scaling. For small rotations and scale changes the repeatability of LP-RDFT is the best among all descriptors.

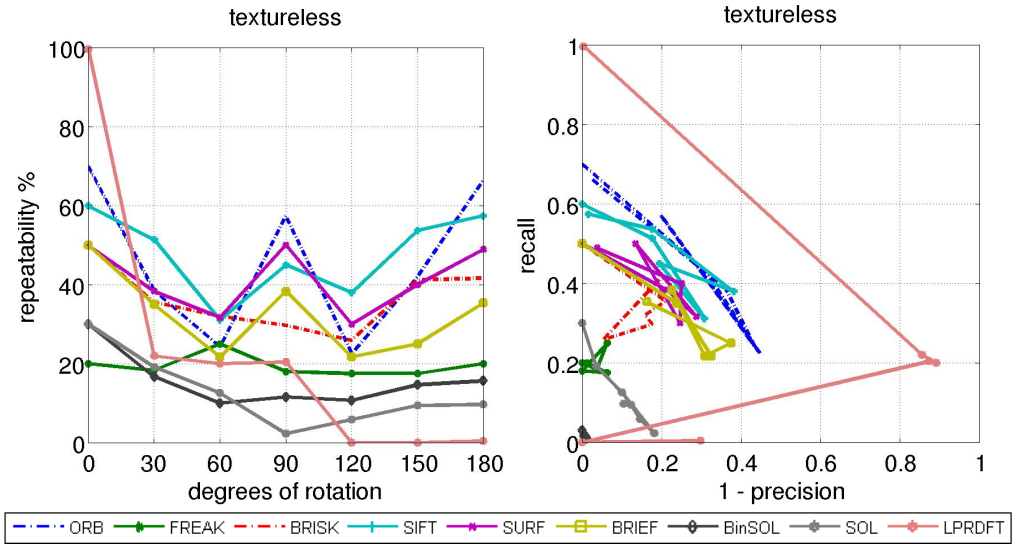

Figure 34: Rotation tests for the collected textureless images from MIRFLICKR dataset.

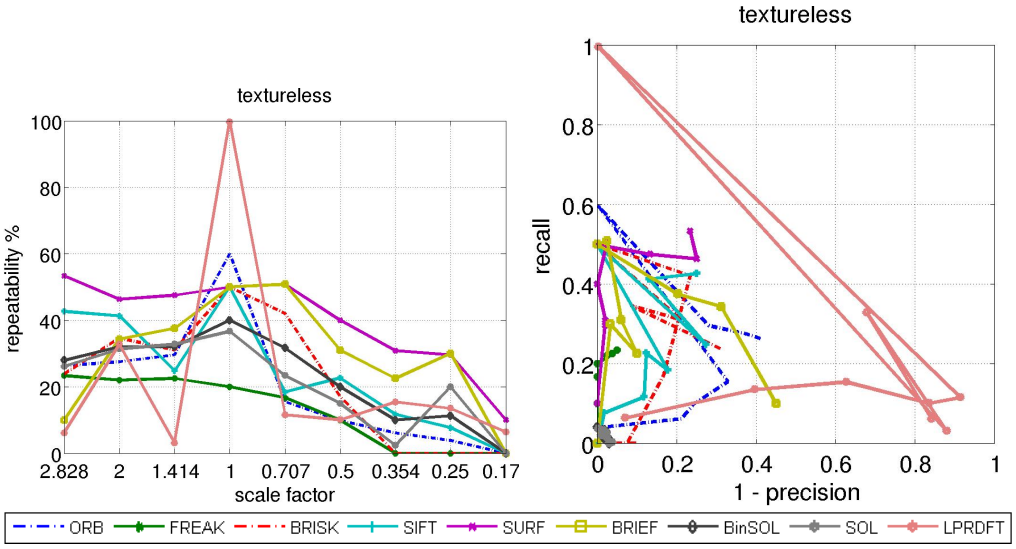

Figure 35: Scaling tests for the collected textureless images from MIRFLICKR dataset.

4.2.6 How different is the LP-RDFT detector from the SIFT detector?

Both LP-RDFT and SIFT descriptors create a scale space with a Gaussian pyramid algorithm. SIFT is the oldest descriptor in our experimental list, but still one of the most successful in the state of the art, that also uses a Gaussian pyramid and the DoG method to collect keypoints. Therefore, we take a closer look to the detection method of SIFT in order to compare it with the one of LP-RDFT.

As the two pyramid algorithms are different, it is expected from the start that the keypoints will be different in quantity. But how much different in quantity are the collected keypoints for LP-RDFT and SIFT? We make a simple experiment and we can easily see the difference: for the same image, taken from the Affine Covariant Features dataset, SIFT finds more keypoints. This

can be easily seen by comparing figures 36 and 37. The pyramid algorithm of SIFT allows the calculation of keypoints very close to the borders of the image, while the pyramid algorithm we use for LP-RDFT does not allow it due to the border effects(section 3.3). But most importantly, we can see that SIFT can find keypoints in more scales than LP-RDFT. This is true because SIFT uses octaves of scales, each octave having several intervals of intermediate scales. An octave corresponds to a group of adjacent scales so that the last scale in this group is double the last scale of the previous octave. Each interval of an octave creates a different pyramid level. The more intervals used per octave, the more levels are created for the Gaussian pyramid of SIFT. Due to the particular structure of the pyramid of SIFT, it can offer more keypoints than the pyramid of LP-RDFT.

Apart from their number, the quality of the keypoints is another important issue. From a quick look on figures 36 and 37, we could say that the keypoints of LP-RDFT seem to be more meaningful than those of SIFT. This can be more obvious when zooming on a part of these two figures. In figure 36 for LP-RDFT, we can see that the keypoints of LP-RDFT are either closer or exactly on edges, corners or blobs. On the other hand, in figure 37 for SIFT, we can see that some of the keypoints of SIFT appear scattered on image neighborhoods that seem to be rather homogeneous for intensity. Although from the human point of view these might seem strange, the detection procedure of SIFT is able to capture very small perturbations of the image intensity and the description procedure can recognize them correctly with significant probability. The ability of the SIFT detector to find very small signal discontinuities is possible due to the many levels of the SIFT pyramid with small difference in scale compared to the levels of the LP-RDFT pyramid.

The efficiency of the keypoints should not be judged generically. The fact that a detector can find very small perturbations of the image intensity does not necessarily means that it suffers from noise in all cases. With images of a modest resolution and clear content, SIFT has been proven to work very efficiently by finding and matching keypoints even where humans fail to see the resemblance between the two matched neighborhoods, as can be seen in figure 40. SIFT is capable to recognize very small local patterns correctly. When details are important, for example in the stitching of images for an image panorama, this ability is very valuable. Then, if details are not an important matter but computational time and storage in memory are more important, LP-RDFT is a better choice. The pyramid of LP-RDFT can have much less levels than the pyramid of SIFT and still provide enough keypoints and descriptor vectors that are competing while significantly smaller than those of SIFT.

4.3 CONCLUSIONS ON LOCAL DESCRIPTION

The experimental results on keypoint matching for textured images of the Affine Covariant Features benchmark dataset showed that LP-RDFT works efficiently having a very small vector length. LP-RDFT outperforms the state of the art for scale changes and image changes relevant to scaling, like increasing blur and JPEG compression. Further testing on the textureless images of

Figure 36: LP-RDFT keypoints. The keypoints are represented as yellow circles with their diameter relevant to the scale where they are detected.

Figure 37: SIFT keypoints. The keypoints are represented as yellow circles with their diameter relevant to the scale where they are detected.

Figure 38: A closer look to LP-RDFT keypoints. This is a part from the upper left side of figure 36. The keypoints detected on the Gaussian pyramid of LP-RDFT capture important intensity perturbations. Therefore, they are more often closer or on edges, corners or blobs. The keypoints are represented as yellow circles with their diameter relevant to the scale where they are detected.

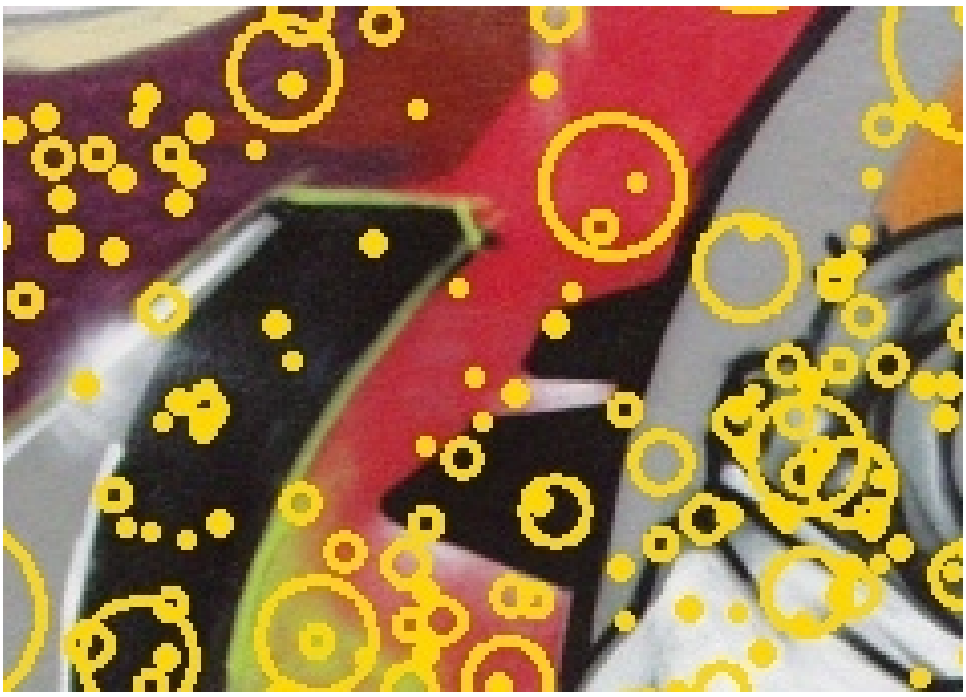

Figure 39: A closer look to SIFT keypoints. This is a part from the upper left side of figure 37. The keypoints detected on the Gaussian pyramid of SIFT can capture either important or very small intensity perturbations. Therefore, keypoints can be found either closer or on edges, corners and blobs but also on areas that for humans look rather homogeneous. The keypoints are represented as yellow circles with their diameter relevant to the scale where they are detected.

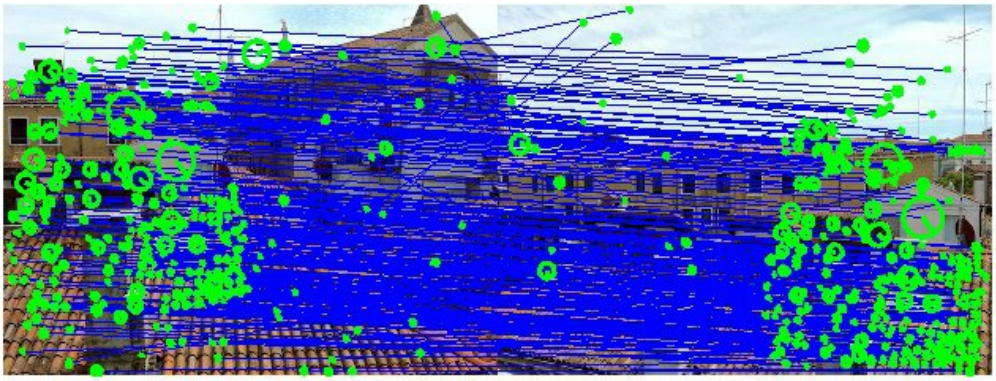

Figure 40: This image is an example of the SIFT implementation from the VLFeat library. The green circles are the keypoint with their diameter being relevant to the scale where they were found and the green line in each circle indicates the orientation of the keypoint. SIFT can find and match keypoints that are hard for a human to identify, for example at the region of the image where there is the sky. Image taken from [135]

the MIRFLICKR Retrieval Evaluation dataset showed that LP-RDFT beats the state of the art for small values of rotation and scaling but its performances deteriorates for larger values. The most important fact is that the vector size of LP-RDFT for the textureless images tests is particularly tiny having just 7 elements.

The results reveal that LP-RDFT can be a solution for local description when the memory cost is a very important issue. It is obvious that every description method has its advantages and disadvantages, working very good in some cases and less good in others, always in relation to the rest of the tested methods. Therefore, a method that works with similar efficiency to the state of the art but with small vector lengths can be rather useful.

In the next chapter, we will view LP-RDFT working on a shape description task in order to be tested for its efficiency and ability to adjust to global description.

DETECTION DE FORMES AVEC LP-RDFT

Nous évaluons le pouvoir discriminant du descripteur proposé en l'utilisant dans un détecteur Adaboost pour piétons, que nos comparons à un détecteur Adaboost utilisant des descripteurs HOG. Les résultats montrent que notre méthode peut avoir un taux de détection compétitif pour la base de données INRIA Person, pour laquelle le descripteur HOG s'est montré particulièrement efficace. Notre méthode fonctionne avec des descripteurs de longueur de vecteur 8 fois plus petite, ce qui la rend appropriée pour les applications à faible puissance de calcul.

5 PATTERN DETECTION WITH LP-RDFT

5.1 DETECTING THE VISUAL PATTERN OF PEOPLE

In this chapter, we will test the capability of LP-RDFT to be used as a global descriptor for the detection of visual patterns, i.e. shapes of objects, in images. We will make experiments with a very usual but still not easy class of objects, which is people in images. Figure 41 is a generalized representation of system constructed for a visual pattern detection.

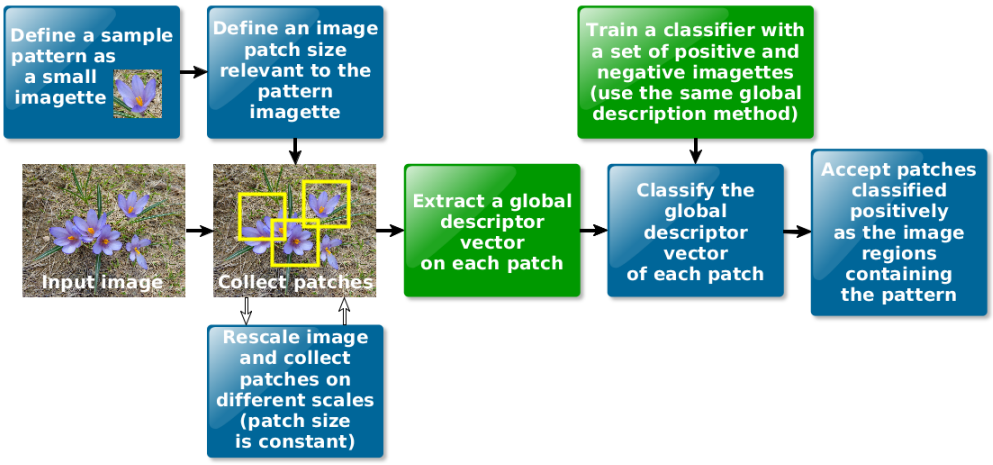

Figure 41: A visual system for pattern detection. The green nodes are the steps of the system where LP-RDFt is involved.

Our experiments will involve "pedestrian"detection, meaning people standing and walking. Detecting people in images is difficult because of the large variety of appearances that can result from variation in pose, clothing and illumination. For many applications, people must be detected within complex real world backgrounds in the presence of occlusions, diverse illumination conditions and viewing angles. While it is possible to construct a detector using a very large set of non-invariant local features that cover the space of possible appearances, the very large number of features required for such a detector results in a very high computational cost. An alternative approach is to project the overall appearance of an object at multiple scales onto a feature vector that is robust to appearance changes (small rotations, illumination, partial occlusion, etc.). When dense descriptors are used for global description (global because we capture an object's shape as a whole on a window with fixed size), the drawback is the lack of invariance. The use of local descriptors overcomes the invariance problems but increases the computational cost as the local descriptors must be shown to be correlated in a particular manner in order to describe a true human form. For the experiments, we create a detector that

captures the appearance of humans based on global form rather than matching individual parts.

We employ LP-RDFT descriptor to construct a detector with Adaboost for detecting people in images. We compare the resulting detector to a detector constructed using HOG (subsection 2.4.2.2) on the INRIA Person dataset [30]. Some sample images are shown in figure 42. Our experiments show that a detector with LP-RDFT descriptor can perform similarly to a HOG detector using descriptor vectors that are more than 8 times shorter.

Figure 42: Some of the test images from INRIA Person dataset. As it can be seen by the example images, a person's pattern, meaning the shape of a person, can be very diverse and also very "mixed"with the background. These facts make this detection problem hard to solve.

5.2 THE HISTOGRAM OF ORIENTED GRADIENTS DESCRIPTOR FOR PEOPLE DETECTION

We use the OpenCV library version of HOG giving 37804 elements for a HOG descriptor vector. In experiments, Dalal and Triggs found that the most effective descriptor for detecting humans was obtained using histograms of 9 gradient orientations computed within a 3×3 grid of 6×6 cells. This provides a vector of 2916 features. Though, usually the existing implementations concerning the INRIA Person dataset provide vectors of 3780 elements extracted on a grid of 16×16 blocks of 8×8 cells. We use the default values for the parameters from the OpenCV library implementation which were customized for the INRIA Person dataset. Additionally, the HOG descriptor we use makes gamma correction the image before extracting features. In comparison to the LP-RDFT descriptor that we use, the final descriptors of LP-RDFT are more than 8 times smaller in length and do not make any gamma correction to the image before being extracted.

5.3 DECIDING ON A DETECTOR

Two popular approaches for building detectors are SVM [25] and Adaboost [41]. The usual scheme for object detection is to slide a window over an image and collect subimages known as patches. Patches are the input test images to classification methods and have the same size as the training images. The coordinates of the collected patches are known. When a patch is classified, we know if we found the desired object according to the probability indicated by the classification method. The SVM is a non-probablistic binary classifier [25] that

is well suited for detecting visual classes in images. The original learning algorithm for support vector machines assumes the existence of a set of labeled training data such that the feature vectors for the two classes are separable by a set of linear surfaces (hyperplanes). The SVM learning algorithm selects a minimal subset of the training samples to define a separating hyperplane that provides the largest margin between the two sets. The method can be applied to any two separable classes by using a kernel function to project the features into a space where the classes can be separated by a hyperplane. Recent extensions using "soft margins"allow this the learning algorithm to use non-separable training data [43]. The primary advantage of SVM is that it is well suited to problems with very large training sets, as the resulting classifier uses only a small subset of the training data and requires relatively low computational power compared to more complex techniques. When used for detecting visual classes, an important disadvantage of SVM is that the trade-off between false positive and false negative detections is difficult to control.

An alternative to the SVM is provided by the AdaBoost learning algorithm [41]. Adaboost sequentially learns a committee of linear classifiers, such that each classifier is based on a weighted version of the training data, such that the weights increase the importance of training samples that were misclassified by the previous classifiers. It can be shown that any arbitrary ratio of false positive and false negative can be achieved by continuing to add linear classifiers to the committee. Therefore, Adaboost has the advantage to create a better optimized final classifier/ detector.

In our experiments we use an Adaboost based algorithm named Boostexter. Its open-source implementation is named ICSIBoost [33], which implements Adaboost over one-level decision trees and can work for either discrete and continuous attributes. This Adaboost algorithm is a well performing and fast method that has the advantage to create efficient detectors in order to perform experiments very easily [61, 71, 143, 1]. Our focus has been to compare the effectiveness of our image descriptor to that of detectors constructed with HOG features on pedestrian detection. As the HOG descriptor is proven very efficient for this purpose on INRIA Person dataset, our aim is to see how the proposed method can work on this task compared to HOG.

5.4 EXPERIMENTAL METHOD AND DATA

5.4.1 Building the detection procedure

We used the positive and negative training images from the INRIA Person dataset to train each of the two methods. Images of humans in this dataset are represented by patches of 64×128 pixels. We generated additional negative patches, so we have 2416 positive patches (images containing pedestrians) and 12180 negative images. We separate 600 positive and 2000 negative patches for composing a development image set (a validation set that helps to optimize the detector parameters) in order to better train the detector and we use the rest as the training set. We create all detectors with the same number of iterations

(weak classifiers), in particular 100 iterations for each detector. All trained detectors have the same length but the descriptor features used are not the same length, with LP-RDFT having significantly less vector elements than HOG.

The HOG descriptor for a patch

We then use HOG and LP-RDFT to extract descriptor vectors that describe each patch. The descriptor vector is extracted by using the full sized patch, so we have global description of the patches. As we mention above, we use the OpenCV library version of HOG in its default values which gives 3780 elements for a 64×128 patch (section 5.2). We examine two versions of the LP-RDFT. The two versions differ on collecting samples for the RDFT on the Gaussian or the Laplacian pyramid, as explained in section 3.5.1. Eventually, we use the Receiver Operating Characteristic (ROC) [34] curves to compare the performance of each method.

5.4.1.1 *LP-RDFT for pedestrian detection*

The global LP-RDFT descriptor for a patch

In order to create the LP-RDFT patch descriptors, we first created a Gaussian pyramid for each patch using a Gaussian filter of $\sigma = \sqrt{2}$, as explained in section 3.3. For a 64×128 patch we receive a pyramid of 7 levels. We keep until the fifth level and discard the rest on top as they are dominated by border effects. For the rest of the 5 levels we create a grid of 16×16 cells while keeping in mind to avoid the border effects. For each cell we find the center. Staring from a cell on a level, we collect an LP vector of length 3 elements from the sample at the center of this cell and on corresponding coordinates at two more adjacent levels. For the first 3 levels, we calculate the 3-element-long LP vectors, starting from a level towards the higher levels. For the rest 2 higher levels, we calculate the 3-element-long LP vectors starting form a level towards the lower levels of the pyramid. This way we impose very dense description on the patch by using several local descriptors with overlapping support areas. Around each LP element, we calculate the RDFT at the closest possible neighborhood of samples, which is the 4 neighbors at radius = 1. From the 4 neighbors x_0,x_1, x_2, x_3, we take 4 Fourier coefficients X_0,X_1, X_2, X_4. We collect the magnitudes of X_0, X_1, X_2 and additionally we collect the phase element of X_1. For further explanations, RDFT is explained in section 3.5. The phase elements for each local cell descriptor will offer additional discrimination power concerning the shape segment of the human shape that each one of them describe. Each cell descriptor is normalized separately from the others with the L_2-norm. The ensemble of the local cell descriptors for each patch gives the final LP-RDFT patch descriptor.

5.4.2 Results on INRIA Person dataset

The ROC curve in figure 43 shows the results for the three detectors. Testing was performed with the testing set image from the INRIA Person dataset. The ROC curves were made counting the True positive (TP) and False positive (FP) rates. The TP rate is also known as the detection rate of a detector.

We used a sliding window approach for detection, searching for people at multiple resolutions with a window size of 64×128 pixels. The sliding window

was moved horizontally and vertically in steps of 8 pixels. The test images were repeatedly resized by 90% and tested until the resized image became smaller than 64×128 pixels. When a window is identified as positive, it is compared to the groundtruth data and if the overlap at least 60%, it is recorded as a true positive. Otherwise, it is recorded as false positive. We ensure that all windows covering the same person are counted only once. A generilized version of the procedure is summarized with algorithm 3.

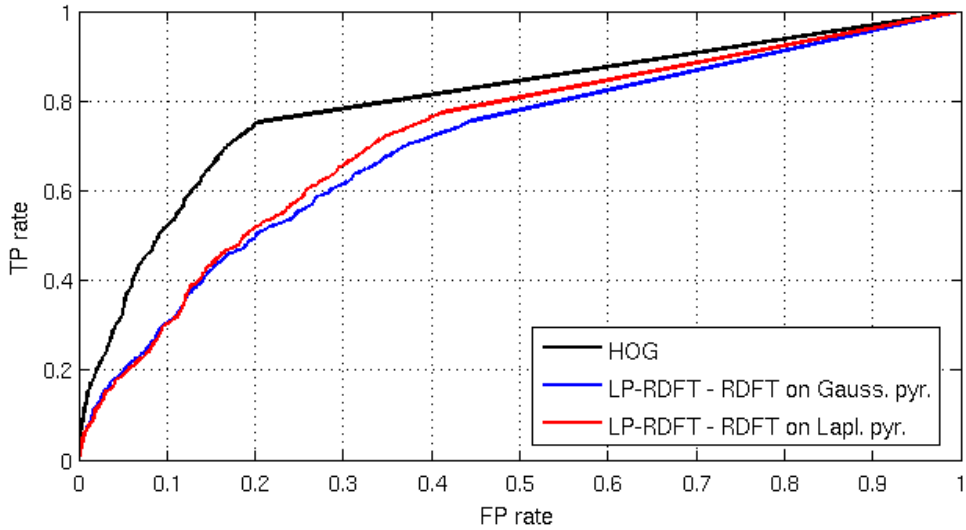

Figure 43: ROC for the techniques. Detection rate refers to true positive detections and False positive rate refers to false positive.

Figure 43 reveals that the detector using HOG and the detectors using the two versions of LP-RDFT have similar TP rates but the LP-RDFT detectors have less good FP rates. Of course, the values of TP rates and FP rates can be raised for all detectors if we change the thresholds of the detection, like the overlapping percentage with the groundtruth or the detection step, but this is not the purpose of this experiment. Although the FP rates are less good for the detectors of LP-RDFT, the length of the descriptors used in the detectors compensate for the performance. The HOG descriptors used have 3780 elements per patch and the LP-RDFT descriptors used have 466 elements per patch each. This is 8.11 times smaller vectors for the LP-RDFT descriptors per patch. This advantage could be very valuable for applications that need high detection rates while accuracy is not that much of a problem and involve low computational power.

Comparing the results of the two detectors using LP-RDFT we see that the results are pretty similar. LP-RDFT with sampling on the Laplacian pyramid for the RDFT has better FP rate values for higher values of the TP rate. This was expected as Laplacian values contain more information than simple intensity values, as explained in section 3.5.1 for further explanation. Though sampling an area on the Laplacian pyramid for the RDFT can be used only for applications as this one, where the sampling neighborhoods are minimal. The problem for wider sampling areas the Laplacian of it will show only edges

and keypoints. On the other hand, LP-RDFT with sampling on the Gaussian pyramid for the RDFT has only slightly less good performance and can be consider equally useful. The two version of LP-RDFT show similar performance and can be both similarly used. Though in order to respect the principal we have set to create a unique method for all applications, we propose that the version of LP-RDFT with sampling on the Gaussian pyramid for the RDFT should be preferred.

5.4.3 Conclusions on Pattern Detection

We have evaluated the discriminant power of the proposed descriptor by using it to compute two different Adaboost detector for pedestrians in images and comparing it to an Adaboost detector using the HOG descriptor. The results revealed that our proposed method can have competitive detection rate to the HOG descriptor on the INRIA Person dataset, for which the HOG descriptor have been proved extremely efficient. Our method performs with more than 8 times small descriptor vector length, a fact which makes it suitable for applications with low computational power.

In the next chapter we see experiments on reflection symmetry detection with a technique that integrates our proposed method.

Parameters:
size SZ = the size of the desired pattern;
Data types:
image p = a patch (cropped imagette) of size SZ;
% p_{pos} (positive) contains the desired pattern and p_{neg} (negative) does not. %

Training
Input data: a set TRAINSET of positive and negative p;
Output: a trained CLAS = a binary classifier that evaluates a p as positive or negative;
for *each* p *in* TRAINSET **do**
  compute a global descriptor vector (subsection 5.4.1);
end
train CLAS with the global descriptors from the TRAINSET;
return CLAS;

Testing
Additional parameters:
counter TP = No. of p_{pos} evaluated as positive (True Positives);
counter FP = No. of p_{neg} evaluated as positive (False Positives);
counter TN = No. of p_{neg} evaluated as negative (True Negatives);
counter FN = No. of p_{pos} evaluated as negative (False Negatives);
Input data: *image* im = an input image of a scene;
Output: TP, FP, TN, FN;
while *the* im *size is bigger than* SZ **do**
  scale im to smaller sizes;
  for *each* d *pixels in both directions of* im **do**
    crop a p;
    compute a global descriptor vector for the cropped p;
    if CLAS *evaluates the cropped* p *as positive* **then**
      if *the groundtruth indicates the cropped* p *as positive* **then**
        increment TP;
      else
        increment FP;
      end
    else
      if *the groundtruth indicates the cropped* p *as negative* **then**
        increment TN;
      else
        increment FN;
      end
    end
  end
end
return TP, FP, TN, FN;

Algorithm 3: The followed procedure for detecting a pattern in an image. The procedure is the same for both LP-RDFT and HOG.

DÉTECTION DE SYMETRIE DE RÉFLEXION AVEC LP-RDFT

Utilisant le descripteur LP-RDFT, nous avons créé une nouvelle technique pour la détection de symétrie de réflexion dans les images. LP-RDFT décrit la forme indépendamment de l'orientation, et conserve les informations d'orientation séparément afin de mesurer la quantité de symétrie de la zone décrite. Comparé à l'état de l'art, la notre technique est la première qui n'utilise pas le gradients de l'image. Elle est aussi la première à trouver des symétries sans calcul explicite de l'image inversée. En outre, nous avons introduit des contraintes capables de filtrer les meilleures et plus proches correspondances de point-clés qui correspondent aux axes de symétrie possibles. De plus, nous avons fourni la formule RSM qui est adaptée à LP-RDFT. Les résultats suivent les performances de l'état de l'art. Cependant, nous pensons que les performances médiocres des algorithmes automatiques s'expliquent par le biais des opérateurs humains qui ont fourni la vérité de terrain. Notre principale contribution concerne à la vue générale du problème avec l'introduction de nouveaux critères pour la détection de réflexion de symétrie.

6 REFLECTION SYMMETRY DETECTION WITH LP-RDFT

6.1 INTRODUCTION TO REFLECTION SYMMETRY

Symmetry generally refers to the repetition of a pattern and can appear in all possible scales and patterns in nature. It plays an important role for human vision by helping to extract meaningful information from the background, for example recognize a skyscraper as a structured set of single windows or tell a butterfly from a flower [82]. Symmetry detection has proved to be a highly demanding problem for computer vision, despite the fact that there are already a number of successful algorithms for image description and pattern recognition. Even though, several techniques have been proposed specifically for this purpose, there is still much room for further improvement.

There are four basic types of symmetry: reflection symmetry, rotation symmetry, translation symmetry and glide-reflection symmetry [82]. We focus on reflection symmetry, also known as mirror symmetry. In simple words, this type of symmetry refers to the opposite similarity of the half part of a pattern to the other half of this same pattern, across an imaginary line segment that separates the two halves. This line segment is called a symmetry axis. Figure 45 gives an outline of reflection symmetry detection system and the steps of the system where we propose new ideas.

6.1.1 Image description for Reflection Symmetry detection

The most recent state of the art for reflection symmetry consists of 5 techniques. The technique of Loy et al [85] is the oldest but still the most effective. Loy's technique has been used as the baseline algorithm for reflection symmetry at the IEEE CVPR2013 Competition [80]. The technique creates symmetric combinations of keypoint matches with SIFT [84], which the authors name constellations of features. Keypoints are firstly collected on the original image. Then the image is flipped about the y (or x) axis and new descriptors are collected.

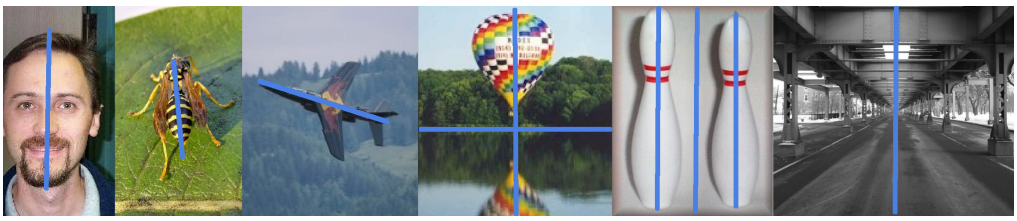

Figure 44: Groundtruth examples of reflection symmetry from the IEEE CVPR2013 Competition [82]. The blue line segments are the reflection symmetry axes that separate the symmetric patterns into the two parts, the one part being the reflection of the other.

The matches for the constellations are made between the two sets of keypoints. The most symmetric matches are selected by measuring the amount of symmetry that they include. The measurement is made by a formula that evaluates the angle, the scale difference and the distance between the two keypoints of a match. The result of the formula is named symmetry magnitude M_{ij}, where i and j refer to the two keypoints of a match. Every match votes for a potential symmetry axis in Hough space, each vote weighted by the M_{ij} magnitude of each match. The most popular axes are accepted as possible symmetry axes. We used this technique for experimental comparison.

The other four state of the art techniques, compared with the baseline algorithm, have sometimes slightly better performance but most of the time they are less good. Michaelsen et al. [89] propose the evaluation of clusters of Gestalten [141] using SIFT features in order to search for reflection symmetry axes in images. Patraucean et al. [109] use the SIFT constellations of the baseline algorithm on candidate patches selected by a statistical procedure based on the *a contrario* theory. Kondra et al. [72] measure the correlation of patches taken on SIFT keypoints. Adluru et al. [2] create star shaped directed graphs of symmetric edgelets in an attempt to encode a symmetric object by its edges. All of these techniques, the same as the baseline, are gradient based. None of them can clearly outperform the baseline algorithm, especially with the most updated symmetry detection test dataset from [80].

The new proposed technique for detecting reflection symmetry in images employs LP-RDFT. The descriptor is calculated on keypoints in an image and performs matching among these keypoints (keypoint matching on the same image). Avoiding the flipping of the image and the recomputation of new descriptors reduces complexity for the overall technique. This is possible because LP-RDFT can capture shape independently of orientation. Another advantage for using LP-RDFT, is that it captures shape in different proportions and resolutions on a range of different image scales. Therefore, it is expected to capture symmetry as symmetry is mainly defined by reflecting shapes. The major innovation towards reflection symmetry detection is the use of an image descriptor not based on gradients. LP-RDFT uses Laplacian of Gaussian and frequencies. As suggested in [85], we also share the belief that a descriptor encoding shapes is suitable for reflection symmetry detection and so we tested LP-RDFT on this task.

The detection of reflection symmetry is made by detecting the possible axes of reflection symmetry. The axes detection is made by a carefully structured procedure inspired by the properties of reflection symmetry. The principal behind using LP-RDFT for this task is that the automatic tuning of parameters to an image, which is easy with LP-RDFT, practically fits an algorithm interactively to each new image and gives more chances to correct detections. Accordingly, the technique is designed to consider the image size and number of keypoint matches, as explained later in this chapter. We follow the concept of the symmetry magnitude from [85] while replacing SIFT with LP-RDFT. We thus achieve comparable performance to the state of the art with a new technique that brings several innovations to the subject of reflection symmetry detection.

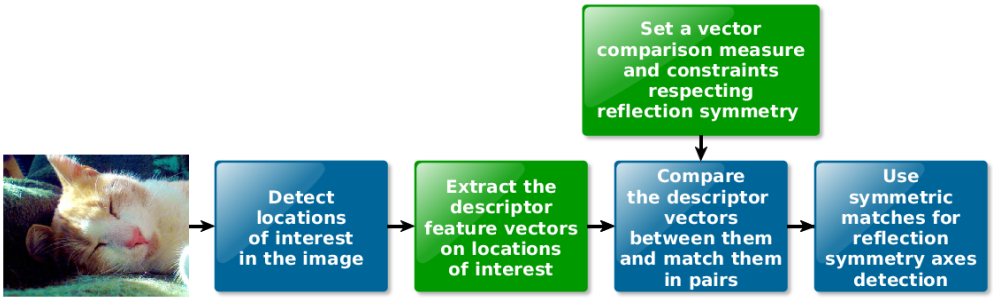

Figure 45: A simplified representation of the visual system for reflection symmetry detection. The green nodes are the steps of the system that the proposed technique is involved in.

6.2 A NEW TECHNIQUE FOR REFLECTION SYMMETRY DETECTION

6.2.1 Adjusting the LP-RDFT features

The main advantages of LP-RDFT are that it is easily adjustable to several description tasks and can have a very compact vector which makes it suitable for real time applications with low memory capacities. Another reason why we test LP-RDFT for reflection symmetry detection is that it can describe shape independently of orientation while keeping separately the orientation information in the feature vector for later using it to determine the amount of symmetry involved. Therefore, LP-RDFT is a good candidate for this task. We can collect feature vectors on an image, make the matching of these features among them without needing to flip the image and recompute features as the baseline algorithm, and then use the orientation information to find out which of the matches are symmetric in order to propose a good symmetry axis.

To create an LP-RDFT feature vector, we use a Gaussian pyramid with a Gaussian filter $G(x, y, 2^{k/2})$, where k is the number of the pyramid level. In this case, we use 8 neighbors around each LP value and we compute linearly the RDFT on these 8 neighbors. Also, the 8 neighbors were collected on the Gaussian image pyramid. Exhaustive experiments showed that the RDFT of 8 neighbors at radius 5 collected on the Gaussian pyramid provided the best results. From the 8 neighbors x_0,x_1, ..., x_7, we take 8 Fourier coefficients X_0,X_1, ..., X_7. From these coefficients, we collect the absolute value of X_0, the sign of X_4 and the magnitudes of X_1,X_2 and X_3. The collection of these values is meaningful due to the properties of these coefficients. The absolute value of X_0 (which is actually its magnitude) and the sign of X_4 are a measure of the sum of intensities of the 8 neighbors, while the magnitudes of X_1,X_2 and X_3 provide important frequency information. The X_5,X_6 and X_7 coefficients were not used due to the properties of the Fourier Transform that make them carry the same information as the X_1,X_2 and X_3. We discard the phase elements which are sensitive to rotation changes in the image signal but we need to keep at least

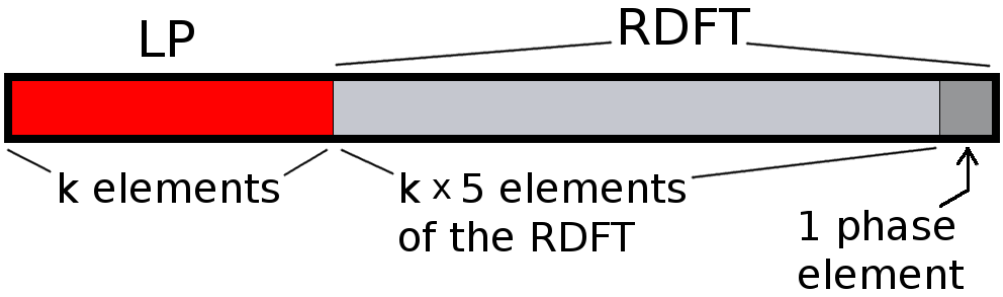

Figure 46: The creation of an LP-RDFT feature vector. The final descriptor vector consists of the LP, magnitude information from the RDFT around each LP element and one phase element. The vector is normalized with the L_2-norm to improve robustness to illumination except the phase element at the end of the vector.

one phase element in order to detect the existence of reflection symmetry. To do this, we gather all the phase elements computed with all the magnitudes of X_1,X_2 and X_3 coefficients around all the elements of the LP vector. Then we keep only one of these elements, the one that belongs to the Fourier coefficient with the highest magnitude value. This phase element is an indication of the orientation of the support are of an LP-RDFT vector.

The final descriptor vector is the concatenation of the LP vector, the RDFT magnitude information and the largest phase element. We normalize the vector with the L_2-norm to improve robustness to illumination but we exclude from the normalization the phase element at the end of the vector. We need the phase element with its original value. Figure 46 is an illustration of the construction of an LP-RDFT descriptor vector constructed as explained.

6.2.2 The overall technique using LP-RDFT

6.2.2.1 *Collecting keypoints on an image.*

We collect keypoints on the original image with the DoG method on the Half-Octave Gaussian pyramid. In comparison with the baseline algorithm, we do not need to flip the image and recompute descriptors in order to find symmetric matches. This fact reduces complexity in the technique.

6.2.2.2 *Computing LP-RDFT feature vectors on the keypoints.*

LP-RDFT features capture shape independently of orientation information while keeping a phase element (that indicates orientation) separately at the end of the vector. This phase element can then be used for selecting the most symmetric matches. The size of the LP-RDFT features used in each image is variable. It depends by the size of the image and the Gaussian pyramid that is created by this image. A larger image creates a pyramid with more levels. There, it is possible to create longer LP-RDFT vectors, and consequently more powerful. Experiments show that a logical descriptor length with good per-

formance is taken when using $\lceil K/3 \rceil$ levels of the pyramid to construct the descriptor, where K is the total number of created levels in the pyramid.

6.2.2.3 *Performing matching between the LP-RDFT feature vectors.*

We perform the keypoint matching using the Euclidean distance among the same set of collected keypoints on the image. The matching is performed with the LP-RDFT features vectors without using their last element, which is the phase element that indicates orientation.

6.2.2.4 *Keeping the closest best matches than the first best.*

At this point, we introduce the idea of keeping the second best match for a keypoint if the distance (in pixels) between the keypoints of this match is smaller. This idea derives from the fact that human perception makes more sense from symmetric patterns when they are closer to the symmetry axis [131]. In the case of more that one symmetry axis existing in an image, this procedure of selecting less similar keypoints but closer to each other provides denser clusters of symmetric matches that indicate more precisely the respective symmetry axes. This assumption works very well when there are more than one symmetry axes in an image because it helps to clear up the clusters of matches that belong to each axis. In case of a single symmetry axis in an image, this assumption might not be very helpful, especially in case of images that the symmetric pattern stretches in the whole image, like the last image in figure 44.

6.2.2.5 *The matches must not be too close or too far.*

Another constraint we set is that the keypoints of a match must not be too close to each other, so their descriptor vectors could overlap, and not too far away, for example close to the edges of the image. If we let the descriptor vectors' support areas of two matched keypoints overlap, the support area around these two keypoints would not be symmetric but just similar. Matched keypoints that are too far away on the image tend to be less significant for symmetry [131], as already discussed in the previous paragraph. Therefore, we further threshold the matches by discarding those matches whose descriptor vectors' support area overlap and matches that their keypoints are more distant than the half of the image diagonal. Finally, a third constraint according to the proximity of keypoints for matching concerns the difference in scale where the two keypoints can be found. We decided that a reasonable difference in scale would be at most one time the resizing factor of the pyramid algorithm, so a difference of one pyramid level.

6.2.2.6 *Introducing the Reflection Symmetry Magnitude measure.*

After the collection of keypoint matches, we need to discard those matches that do not include any symmetry. In order to measure symmetry, [85] proposed a formula for calculating a symmetry magnitude geared to SIFT vectors. We adapted this idea and adjust it to the LP-RDFT descriptor vectors. We refer

to the new proposed measure as Reflection Symmetry Magnitude (RSM). We use letters a and b to refer to the two keypoints of a match. The RSM_{ab} of a match is the product of three values, F_{ab} which measures angular symmetry indicated by phase elements, S_{ab} which measures scale similarity and D_{ab} which gives a weight to the distance between keypoint a and keypoint b. The formula is:

$$RSM_{ab} = F_{ab} \times S_{ab} \times D_{ab} \tag{24}$$

where F_{ab}, S_{ab} and D_{ab}:

$$F_{ab} = \frac{\pi - |ph_a - ph_b|}{\pi} \tag{25}$$

where ph_a and ph_b are the phase elements in radians from the LP-RDFT feature vectors on keypoint a and keypoint b. The phase elements must not be the same or too close, because thius means that the described areas around two keypoints are identical, not symmetric. The two phase elements should have a difference of half a circle in order to describe symmetric signals.

$$S_{ab} = \frac{1}{1 + |\sigma_a - \sigma_b|} \tag{26}$$

where σ_a and σ_b refer to the level of the image pyramid where the keypoints a and b were collected. This way, matches on the same level are favored.

$$D_{ab} = \frac{diag}{dist_{ab}} \tag{27}$$

where $diag$ is the image diagonal and $dist_a b$ is the distance between keypoint a and keypoint b. After calculating the RSM for each match, we keep only a subset of the matches with the best magnitude. A meaningful subset would be $1/3$ of the matches. If the $1/3$ for an image is a very small number we keep all of the matches.

6.2.2.7 *Using the Hough space for lines to detect Symmetry Axes.*

We use the matches from the previous step to vote for possible symmetry axes in a Hough space [31]. Each match proposes a line that passes perpendicularly from the midpoint between its two keypoints. This line can be represented in the $r\theta$ polar coordinate system and vote this way in a Hough table. Each vote is weighted by the respective RSM_{ab} of a match. We follow the manner of [85] and collect the local maxima of the Hough table after smoothing it with a Gaussian filter. The maxima in the Hough table are the possible reflection symmetry axes of the image.

6.2.2.8 *Confidence score to select the best detected axes.*

We assign a confidence score to each detected axis. The confidence score for the detected axes of the baseline algorithm is the value of the local maxima of the Hough table normalized by the value of the global maximum. We use

another computation for the confidence score. We find the local maxima in Hough space and we check again which matches indicate the detected axes. The reason is that the votes of some matches were lost in the Gaussian smoothing of the Hough table. These matches with the lost votes can now re-vote for the detected axes from the Hough space. The confidence score is the number of votes per detected axis normalized by the largest number of votes given for an axis. Each match is allowed to vote only for one of the detected axes. The confidence scores are necessary for creating the precision-recall curves in the experiments section. We use the confidence scores to control the number of correct detections by manually tuning a threshold in the space $[0, 1]$. A summary of the proposed procedure for reflection symmetry detection is given with algorithm 4.

6.3 EXPERIMENTS

6.3.1 Validation framework

In this section we demonstrate the results of the proposed technique against the baseline algorithm. Judging from the literature, the proposed baseline algorithm in general outperforms the other methods. In a few cases during the experiments shown, there are some small exceptions by one or the other technique where they work slightly better. Still the baseline algorithm works the best in most of the cases and especially with the most updated symmetry detection test dataset from [80]. Althought its results still can be improved, it is a good competitor. The test dataset, the testing protocol and the source code for the baseline algorithm are taken by the web site of the Symmetry Detection from Real World Images Competition 2013 [80]. For each detected symmetry axis DA, we measure the angle and the distance between it and each groundtruth axis GT. The DA is matched to a GT if the angle between them is $angle_{DA,GT} <= 10°$ and the distance between them is $dist_{DA,GT} <= 0.2 \times min\{length(DA), length(GT)\}$. A long GT can be matched by many small DAs but one DA can only match the closest GT. True positives (TP) are considered all the GTs that are matched by at least one DA, false positives (FP) are considered the DAs that cannot match any GT and false negatives (FN) are the GTs that are not matched by any DA. This three values are used to calculate recall and precision by the following formulas:

$$precision = TP/(TP + FP) \tag{28}$$

$$recall = TP/(TP + FN) \tag{29}$$

In order to create points of the precision-recall curves in figures 47 and 49, we manually tune a threshold in $[0, 1]$ and each time we consider only those DAs whose confidence scores are allowed.

Parameters:
list KP_{im} = all keypoints detected in an input image im;
point kp_a = a keypoint (x_a, y_a, σ_a) in KP_{im};
list KP_{im}' = KP_{im} excluding kp;
point kp_b = a keypoint (x_b, y_b, σ_b) in KP_{im}';
Input data: *image* im = an input image of a scene;
Output: reflection symmetry axes in the image (the axes are line segments);

```
compute LP-RDFT on all kp in KP_im;
for each kp_a in KP_im do
    for each kp_b in KP_im' do
        compare descriptor vectors of kp_a and kp_b;
        if kp_a and kp_b are matched AND respect the characteristics of reflection
        symmetry (subsections 6.2.2.4 and 6.2.2.5) then
            keep the pair (kp_a,kp_b);
        end
    end
end
for each kept pair (kp_a,kp_b) do
    compute the RSM_ab (subsection 6.2.2.6);
end
select only a part of the (kp_a,kp_b) with the best RSM_ab;
for each selected pair do
    cast a vote in Hough space for a possible reflection symmetry axis,
    weighted by the RSM_ab;
end
detect possible axes as maxima in the Hough space;
for each detected axis do
    compute a confidence score;
end
return detected axes with the highest confidence scores;
```

Algorithm 4: The proposed procedure for reflection symmetry axes detection in an image with LP-RDFT. The procedure for the baseline algorithm requires the flipping of the image and the recomputation of descriptors in the new image. The matching of symmetric pairs of keypoints for the baseline happens between the two images. LP-RDFT does not require flipping of the image because it allows the descriptor vectors to be independent of orientation. A separate phase element kept for each descriptor vector can later be used in the RSM formula to measure symmetry between matched keypoints.

6.3.2 The IEEE CVPR2013 Competition dataset

The dataset contains four different sets of images concerning reflection symmetry. There are two sets with images containing a single symmetry axis, a training set and a test set, and two sets with images containing multiple symmetry axes, again training set and test set. We used them appropriately. It is worth mentioning that the groundtruth takes into account the human perception of symmetry. This means that for deciding on the existence of a symmetry axis, the creators of the dataset considered those who make more sense to humans. This constraint has an important impact on the results, seriously undermining the performance of any algorithm [80, 89]. In figures 51 and 52 for the proposed technique, we can see in some cases that the technique identifies symmetry very differently from what humans think is symmetry. Still, this is the most updated dataset for symmetry detection in images.

The creation of a more objective dataset for symmetry detection is a very complicated matter. A dataset can be made considering symmetry more relevant to the way computer algorithms identify symmetry. But then, even if the typical measures for performance would show high values, the symmetry detected in the images would probably make little sense to humans. For the time being, it would be better if the evaluation of symmetry detection algorithms should not be based only on the values of usual measures but also on the actual results drawn on the images.

6.3.3 Results

We can see the precision-recall curves in figures 47 and 49. The baseline algorithm has generally a better performance. But for the highest recall values that the new technique reaches, it outperforms the baseline in both recall and precision. In figure 47, the new technique reaches high recall values with slightly better precision than the baseline. It is also seen that when the baseline works with higher but still modest precision, the recall is lower and also modest. In figure 49, for medium recall the new technique works also with better precision. In the same figure, the baseline reaches higher recall but with very low precision, which shows that there is very low certainty of the detections. These results show that the proposed technique can compete with the baseline algorithm though further improvement is necessary.

Regarding the idea of keeping the second best match (paragraph 6.2.2), this worked correctly for multiple axes detection. This constraint helped to gather symmetric matches closer together and indicate symmetry axes more precisely. Examples of detections with both techniques can be seen in figure 50. For the detections of the baseline, the respective keypoint matches are shown as dots of the same color as the detected axis (source code given [80]). For the proposed technique, we indicate the support area of each axis (area of the respective matches) because we use more matches than the baseline and the images are too crowded otherwise, as can be seen in figures 51 and 52. For the single axis detection tests, the idea of keeping the second best match deteriorated

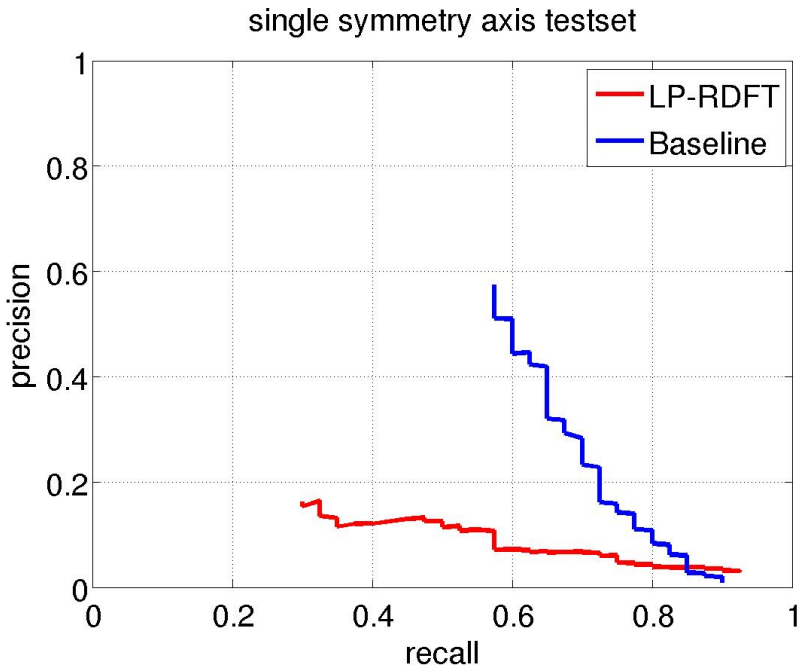

Figure 47: Precision-recall curves for images with a single symmetry axis. For the highest possible recall that the new technique reaches, it outperforms the baseline in both recall and precision. For images containing one symmetry axis, the proposed technique reaches higher recall values with better precision that the baseline algorithm.

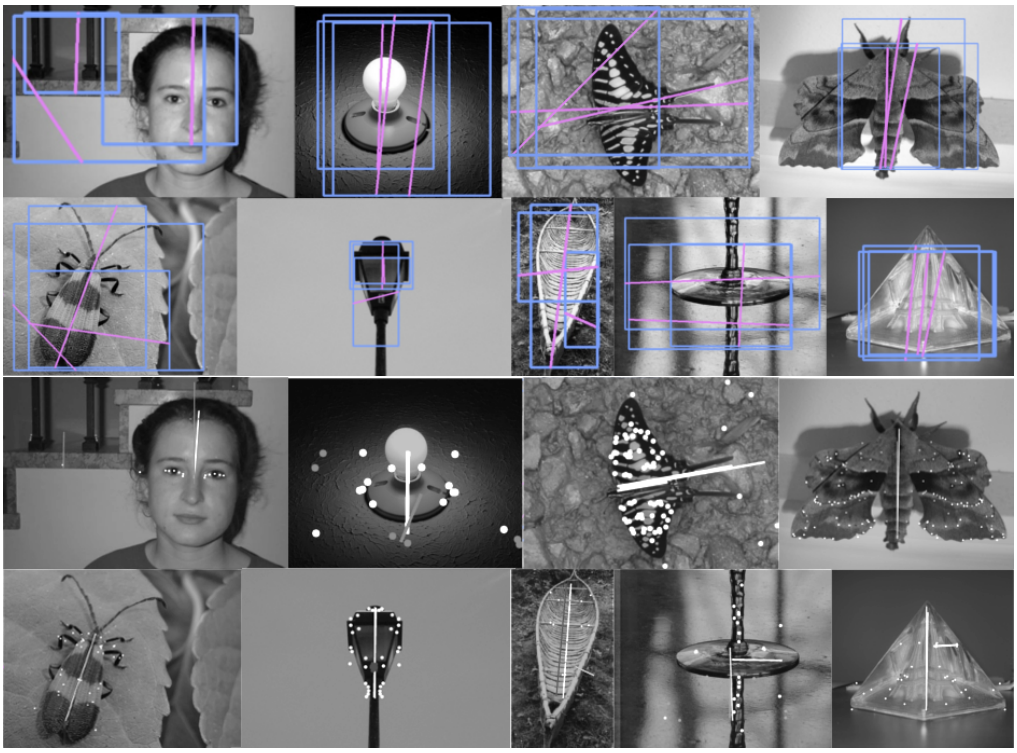

Figure 48: Detected axes in images containing a single symmetry axis. **Above:** the new technique. **Below:** the baseline algorithm. We keep at most three detections per image with the highest confidence score. For the baseline algorithm, the symmetric matches that indicate a axis are given with dots of the same color. We use blue rectangles to indicate the support area (symmetric matches) for each axis.

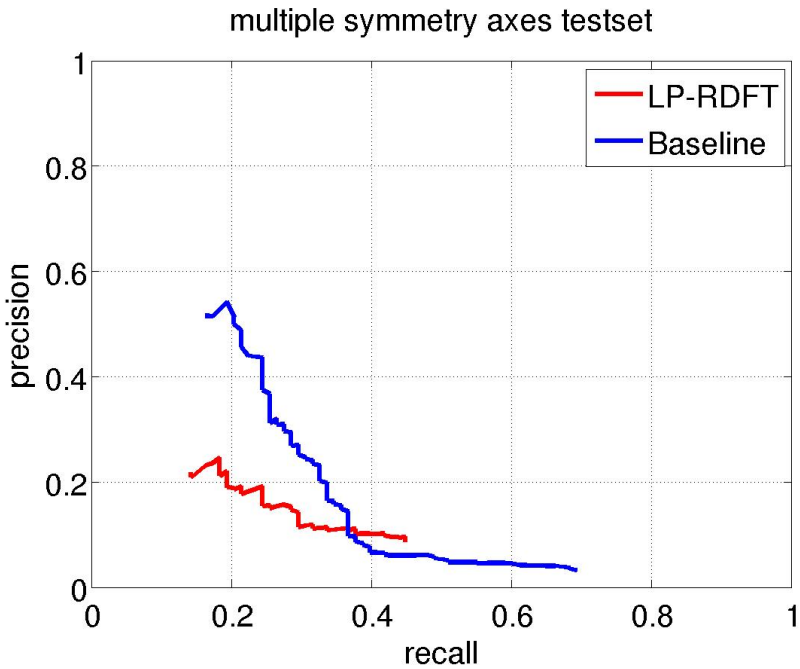

Figure 49: Precision-recall curves for images with multiple symmetry axes. For the highest possible recall that the new technique reaches, it outperforms the baseline in both recall and precision. For images containing multiple symmetry axes, the baseline algorithm reaches better recall values but with very low precision.

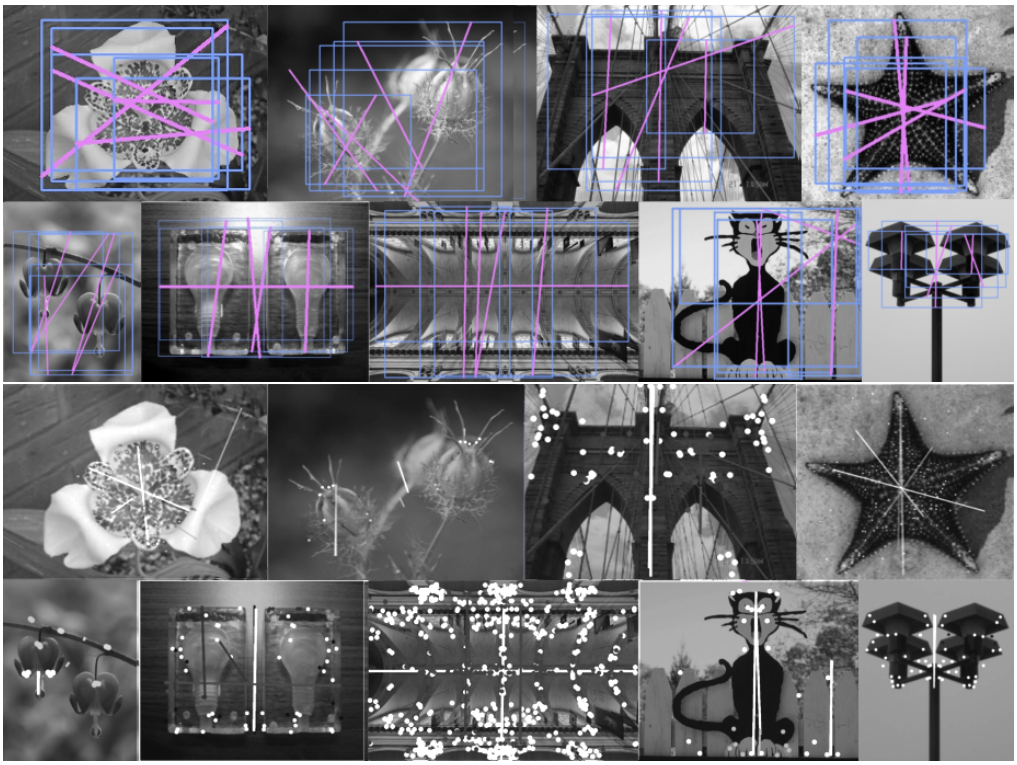

Figure 50: Detected axes in images containing multiple reflection symmetry axes. **Above:** the new technique. **Below:** the baseline algorithm. We keep at most 5 detections per image with the highest confidence score. For the baseline algorithm, the symmetric matches that indicate a axis are given with dots of the same color. We use blue rectangles to indicate the support area (symmetric matches) for each axis.

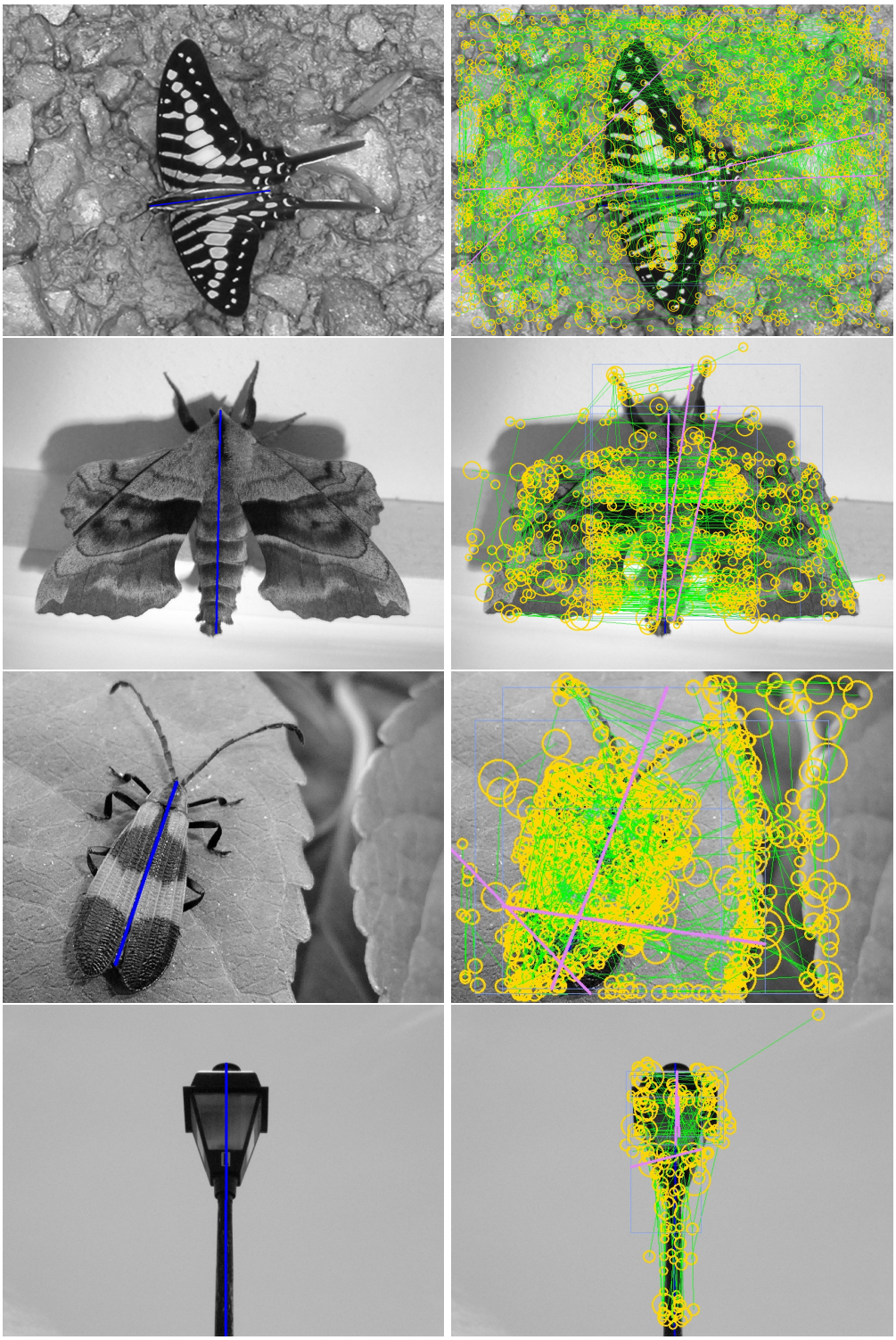

Figure 51: Examples of matches for working on images with a single symmetry axis. **Left:** groundtruth axes (blue line segments). **Right:** symmetric keypoint matches found and the most voted axes (magenta line segments). Each match is shown as a couple of yellow circles representing the keypoints matched, with the radius indicating the scale where the keypoint was detected and a green line segment that connects the two keypoints. In the third case, we can see that the proposed algorithm considers symmetry differently from a human.

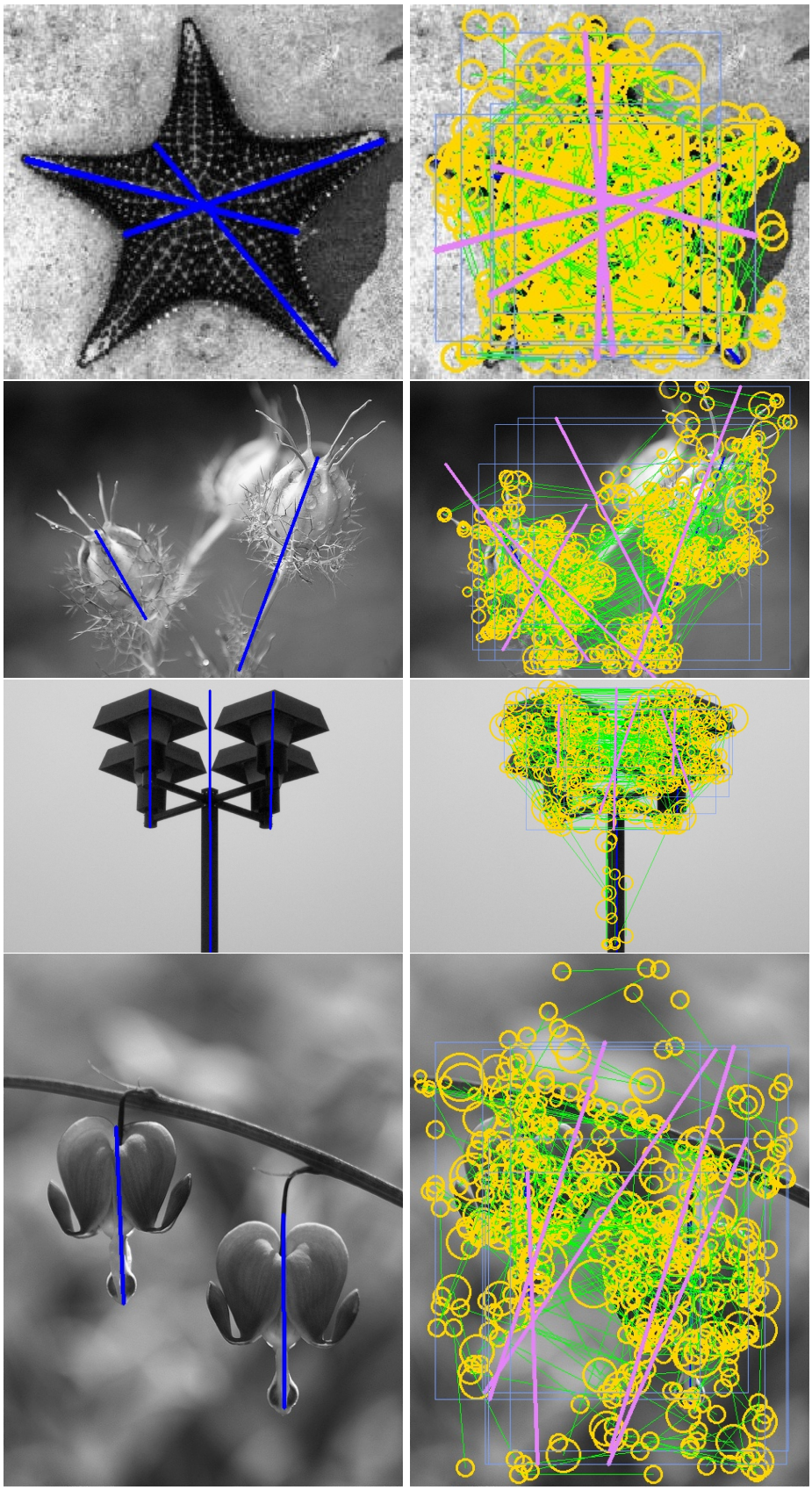

Figure 52: Examples of matches working on images with multiple symmetry axes. **Left:** groundtruth axes (blue line segments). **Right:** symmetric keypoint matches found and the most voted axes (magenta line segments). Each match is shown as a couple of yellow circles representing the keypoints matched, with the radius indicating the scale where the keypoint was detected and a green line segment that connects the two keypoints. In the first, second and fourth case, we can see that the proposed algorithm considers symmetry differently from a human.

the recall about 10%. So, we did not use it for this case. The explanation for this outcome is that despite the assumption that features found closer to a symmetry axis make more sense, the symmetric patterns can spread in the whole image when there is only one symmetry axis (figure 48). By figures 48 and 50 we can see that concerning the best detections of both techniques for the same images, the results are similar.

6.4 CONCLUSIONS ON REFLECTION SYMMETRY DETECTION

We presented a new technique using LP-RDFT for reflection symmetry detection in images. We use LP-RDFT because it captures shape information in an image, a property which is very important for symmetry detection. LP-RDFT describes shape independently from orientation while it keeps a phase element, that indicates orientation, separately in order to measure the amount of symmetry of the described area. The proposed technique is the first to use a description method that does not utilize image gradients. All of the other state of the art techniques are gradient based. Another contribution to the subject is that we are able to find symmetry without flipping the image and regathering features, whereas it is a prerequisite for the baseline algorithm. Also, we introduced some constraints in order to filter the closest best matches of keypoints that correspond to possible symmetry axes. Moreover, we provided the RSM formula which is adapted to LP-RDFT. The results follow the performances of the state of the art.

As the groundtruth is influenced by human preference, we consider logical the mediocre results of our algorithm and the baseline. Nevertheless, the results of the proposed technique with LP-RDFT follow the performances of the state of the art. The main contribution is however to the general view of the problem by introducing new ideas that will boost further improvement to the very demanding subject of symmetry detection in images.

In the next and final chapter of this thesis, the qualities and drawbacks of LP-RDFT are summed up and an overview of the method leads us to conclude on the efficiency, the possible improvement and the use of LP-RDFT.

CONCLUSIONS

Dans cette thèse nous avons exploré une nouvelle approche multi-échelle à la description d'images. Après une recherche volumineuse des approches existantes, nous avons décidé d'utiliser le Laplacien du Gaussien et la Transformée de Fourier, les deux calculés dans un espace d'échelle logarithmique. Avec cette approche, nous avons réalisé un descripteur d'image purement mathématique qui peut être facilement ajusté à plusieurs tâches visuelles et des exigences des petites mémoire.

Les résultats expérimentaux ont montré la robustesse aux variations de l'image, en particulier des changements impliquant la dégradation du signal d'image, et l'efficacité de capture de formes dans les images. Par rapport à l'état de l'art la méthode proposée a des bonnes performances, tout en utilisant des vecteur descripteur de très petite taille, ce qui la rend appropriée pour les systèmes avec une puissance de calcul et/ou une mémoire limitée. L'efficacité de la méthode proposée peut être explorée plus sur les applications en temps réel.

La qualité des vecteurs de caractéristiques par la méthode proposée est approprié pour les applications nécessitant à la fois la description invariante et de l'information d'orientation. Les expériences sur la symétrie de réflexion ont montré que nous pouvons facilement manipuler l'information de signal transportée dans les vecteurs LP-RDFT. Cette capacité du descripteur LP-RDFT, combiné avec ses besoin mémoire limités, peut ouvrir la voie à des applications exigeantes en temps réel.

7 CONCLUSIONS

A new approach for multi-scale image description has been explored in this thesis. After a review of research on existing approaches for image description in chapter 2, we describe a new approach for image description that used the Laplacian of Gaussian to describe intensity variations and the Fourier transform to describe low frequencies at the surrounding area. We compute this descriptor on a logarithmic scale space to provide scale invariance. Rotation invariance is addressed by the properties of the Laplacian of Gaussian and the Fourier transform taken radially. Chapter 3 discusses all the possible options and parameters involved in the definition of the approach. The aim of this approach is to offer a general image description method that can be easily adjusted to several visual tasks. Experimental results revealed robustness to image changes, especially changes that involved deterioration of the image signal, and efficiency of capturing shapes in images. The proposed method can perform sufficiently good with very small vector sizes compared to the state of the art, a fact that makes it suitable for low computational systems and real time applications.

In chapter 4, we show experiments on keypoint matching against a selected subset of existing descriptors that are either well established or newly proposed. Tests on the Affine Covariant Features benchmark dataset showed that the proposed descriptor is very effective with scaling and lower resolution changes in images but less effective in the case of other affine related and illumination changes. On the other hand, this new approach employs a much small vector length compared to the other descriptors making it potentially appropriate for application where memory and power are limited. Further experiments on a set of textureless images from the MIRFLICKR Retrieval Evaluation dataset revealed that the proposed method can be very discriminative when there is very low quality information. In this case, our results showed that the descriptor vector was much smaller than the other methods. In total, the proposed descriptor showed in the experiments that it can compete with the state of the art for local image description with the advantage of low memory requirements.

Chapter 5 presents experiments where the proposed method was used for global description on pattern detection. The human shape was the subject of search in the INRIA Person dataset and the well-known HOG descriptor [30] was the baseline. Using an Adaboost classifier, we created three detectors with the same number of iterations (weak classifiers), one detector using the HOG descriptor and two detectors using two different version of the proposed descriptor. The proposed descriptor vectors were created by using samples either on the Gaussian pyramid for the computation of Radial Fourier Transform or on the Laplacian pyramid. The idea behind testing Laplacian of Gaussian values for the Radial Fourier Transform was introduced in order to compensate

the lack of information due to the small size of sampling area for it. A Laplacian of Gaussian has more information about a neighborhood than an simple intensity value (sample on the Gaussian pyramid). These two versions of the descriptor created two different detectors. Both detectors of the proposed method showed similar detection performance to the HOG detector. The false positive rates for the two detector were higher compared to the HOG detector, but the proposed descriptor vectors (for both versions) were 8.11 times smaller than the HOG descriptor vectors. Such a performance can be valuable for low computational systems where high detection rates are more important than accuracy in detection.

Chapter 6 contains experiments on reflection symmetry detection. The aim of this visual task is to determine the axis of symmetry between two reflecting parts of an image. The proposed method was integrated into a technique for symmetry detection. Keypoints were detected in the image and descriptor features were computed for each of them. The proposed method used magnitudes of the Fourier Transform coefficients for a circle of samples around the Laplacian Profile and one phase element. The phase element belonged to the coefficient with the most important magnitude computed around any of the Laplacians of Gaussians of the Laplacian Profile. This phase element was an indication of the orientation of the local area's intensity. A series of constraints was then employed to find symmetric keypoint matches among the set of keypoints in the image. Each match was tested for describing symmetric image neighborhoods. A part of these constraints were tailored on LP-RDFT but can be adapted. Another part of the constraints are general ideas that we explored about what is important for detecting efficiently mirror symmetric shapes and can be used by any relevant technique. The cloud of all the accepted matches voted for the existence of possible symmetry axes in a Hough space for lines. The experiments on this task were performed by the dataset and rules of the latest milestone on symmetry detection on images, the IEEE CVPR2013 Competition [80]. Compared to the baseline of the contest, the propose method showed interesting results. Our main contribution for this task is to introduce new ideas and constraints about how to better identify reflection symmetry.

The overall conclusion of the thesis is that while initial results are encouraging further investigation is required. The proposed method is indeed compact and can be used for several application of different types. Its performance is in line with the performance of the state of the art. Further improvement of the method can be focus mainly on its robustness towards affine image transformations. All in all, this new descriptor is an appropriate choice for systems of low computational power and limited memory space.

7.1 LESSONS LEARNED

During the exploration of the LP-RDFT descriptor, we made a list of factors that effect its qualities and performance. In this section, we list this factors and we give an intuition about how they should be used efficiently to give the appropriate version of this descriptor.

7.1.0.1 *The appropriate scale space*

The scale factor σ of the Gaussian filtering for the computation of the Half-Octave Gaussian pyramid can be any power of two. Ideally, larger σ can be used for images that contain very sharp edges and corners. Smaller σ values are better for images that already have quite smooth edges and corners. A $\sigma = \sqrt{2}$ is a good choice for any application in general.

7.1.0.2 *The sampling area for the RDFT*

The area for endorsing the LP with frequency details can be either a circle of samples or a disk. The disk sampling area offers more discrimination power but raises the length of the descriptor vector. As we want to have a descriptor that can be used for low computational devices, the circle sampling area is more appropriate. The radius of the sampling area can be any, but we have seen from our experiments that a radius of 5 or 6 pixels is advisable for applications involving local description and a small radius of 1 pixel is enough for dense description.

7.1.0.3 *Fourier coefficients to keep and how*

The Fourier coefficients have redundant information and we can keep only the half of them. keeping their magnitudes, we have rotation invariant information about the frequencies at the described area. Using the coefficients' phase, we have an indication for orientation that can be used to enhance discrimination power relatively to the demands of a visual task.

7.1.0.4 *Length of the LP*

The appropriate length of the LP (the number of pyramid levels used to extract the Laplacians of Gaussians for the LP vector) is also relative to the quality and the size of the image. Images with small size do not allow long LP vectors compared with larger images given a particular σ for the Gaussian filtering in the pyramid structure. Smoother images need longer LP vectors (meaning that they need to be described with more scales) as the information they contain is not well defined by accurate edges and corners.

7.1.0.5 *Length of the final descriptor vector*

RDFT can be computed around every LP element (Laplacian of Gaussian on a level) on the respective level of the pyramid (level where this Laplacian of Gaussian is taken). But it is not necessary to compute it around all LP elements. Experiments on keypoint matching show that computing the RDFT around the LP elements only for half the levels where this LP was extracted provides sufficient signal information. This way, the vector becomes even smaller while remaining still discriminative to complete the task.

7.2 PERSPECTIVES OF THIS STUDY

- The proposed description method can be discriminative enough to be used for local image description but in the same time short enough in size to be used into classifiers and detectors without high cost. Therefore, it can be used to extract one type of descriptors from images that can be used for different visual tasks. This way, we reduce system complexity. Depending on the quality of the images that a system has to exploit, a particular version of the proposed descriptor will be more appropriate for use for both local and global description tasks.

- Low quality images contain low amounts of important details and large descriptor vectors unnecessarily increase the computational and memory cost. A reasonable alternative is a compact multi-scale descriptor as the proposed one that captures enough information with modest cost in computations and storage in memory.

- Symmetry in all forms (reflection, rotation and translation symmetry) is defined by shape [80]. The proposed descriptor captures shape with the Laplacian of Gaussian and additional frequency information about its surrounding area with the Radial Fourier Transform. The technique for reflection symmetry detection using the proposed descriptor can be extended to the other two forms of symmetry with appropriate manipulations. The phase of the Radial Fourier Transform coefficients can be used to define the relationship between the parts of the rotation symmetric shapes. Moreover, the phase of the Radial Fourier Transform coefficients can determine the directions of the repeating components for translation symmetry. Therefore, the proposed descriptor can be used to detect all forms of symmetry in images.

7.3 PROPOSING FUTURE WORK

An idea for further experiments on this work is to create an application that performs multiple tasks using the proposed descriptor. The objective should be to use only one version of LP-RDFT descriptors extracted once from an image and used for a variety of tasks. For example, a possible test application can be to perform image matching and reflection symmetry detection between patterns that are the one (from the first image) the reflection of the other (on the second image). An example of such images may contain the same person that has his/her head turned to a different side in each image. LP-RDFT descriptors as described in chapter 6 can be extracted on keypoints on the two images. First, they can be used without the phase element to perform the matching of the images by their background, with the usual keypoint matching technique. Then, the same descriptors can be used with the phase element to recognize the human face turned at different sides (considering the two turned faces to different directions as symmetric). Moreover, one of the two images can be compared to an image that has the same background but with the human

face looking straight forward. The symmetry will exist between the turned face in the side of the first image and the half of the face in the other image, while in the second image the two halves of the face will be also symmetric between them. This possible application, that performs keypoint matching and detection of symmetric patterns spreading across two images, can be used for detecting robustly the orientation (left-right-forward) of a face in different images with the same background.

A second idea for future work is to extend the reflection symmetry technique with LP-RDFT to the other two types of symmetry by adjusting the technique to their characteristics. The phase of the Radial Fourier Transform coefficients can be used to define the symmetry between the parts of a rotation symmetric pattern and the directions of the repeating patterns for translation symmetry. Moreover, the technique already proposed for reflection symmetry can be further improved. The formula of RSM can be improved to consider more factors, for example inspired by the proposed constraints in chapter 6. Furthermore, other methods apart from the Hough space can be tested for the symmetry axes detection. For example, axes could be more simply detected by averaging the bisectors of the line segments that join the symmetric keypoints into pairs. This method might be more precise than the Hough space. The Hough space suffers from the smoothing before the computation of the maxima when the voting pairs vote for very similar but not exactly the same axes. The LP-RDFT descriptor should be further examined on symmetry as its vector elements, the rotation invariant elements and the phase of the Fourier coefficients that can be stored independently, provides the right kind of features for symmetry.

A third idea for future work, is to use the same version of LP-RDFT descriptor in a real time application for two completely different reasons. For example, the descriptor can be used locally on keypoints for matching to find similar images and then globally on image patches to detect objects. The purpose of this test application should be to show that only a single version of the LP-RDFT can be used to perform both tasks successfully and with low memory cost.

7.4 BRAINSTORMING IN THE AFTERMATH

There is always the argument of whether qualities such as low complexity, low computational cost and low storage memory requirements are useful or even relevant. It is expected that with the advancement of computers, descriptors than are costly today will be easily used in the future. On these grounds, we can say that it should be not that important to make less costly descriptors allowing us to search for "heavier" but more powerful methods. But this is not entirely correct. As computers advance, so does the quantity of data they can handle. It is sure that computers will keep becoming faster and having larger memory capacities. But then people will want to store images and videos with better resolution, better colors, more dimensions, etc. Even if the actual descriptor computations remains stable, the time and the cost of proper image description depends also on the quality and the amount of the data. Therefore,

the cost of descriptors should never be a less important matter that the rest of their characteristics.

Data with more than two dimensions, like depth images or video sequences, require more time and memory to be treated than 2D images. The proposed descriptor can be a good answer to such types of data given its qualities. Both the Laplacian and the Fourier theory can be generalized to many dimensions. Therefore, this descriptor can be extended to work with multidimensional data.

Another argument on how to ameliorate image description is if it is best to keep evolving the existing state of the art methods or keep searching for new ones. It is true that changing existing methods is easier than beginning from point zero. This is undeniably a good solution when the time to complete a project plan is limited, but in the same time this attitude leaves several unexplored ideas still in the dark. There is always the chance that there is nothing more to be done in image description and we have reached a limit to the possibility of further progress, but this still remains just an assumption. It might seem that we have reached a limit, but perhaps it is because it was the research that was limited around what has already been proved to work efficiently. For example, gradients were for long time now proven to be very efficient which naturally led to a lot of research around them. This fact though must not restrain us to consider them as the one or the best option and keep restraining our selves only towards this point of view. For sure, we have to keep making better what we already have but we must not ignore that there might be a better idea that it was not discovered yet. A new idea for better image description can be inspired by theory used in another field but never used for computer vision or a combination of theories in a way that was never tried before, as attempted in this thesis.

[1] Abdulwahab O Adi and Erbug Celebi. Classification of 20 news group with naïve bayes classifier. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pages 2150–2153. IEEE, 2014.

[2] N. Adluru, L.J. Latecki, R. Lakaemper, T. Young, Xiang Bai, and A. Gross. Contour grouping based on local symmetry. In *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007.*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408879.

[3] Timo Ahonen and Matti Pietikäinen. Soft histograms for local binary patterns. In *Finnish Signal Processing Symposium (FINSIG 2007)*, 2007.

[4] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE features. In *European Conference on Computer Vision (ECCV)*, 2012.

[5] P. F. Alcantarilla, J. Nuevo, and A. Bartoli. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. In *British Machine Vision Conference (BMVC)*, 2013.

[6] Mitsuru Ambai and Yuichi Yoshida. CARD: Compact And Real-time Descriptors. *IEEE International Conference on Computer Vision*, 0:97–104, 2011. doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.2011.6126230.

[7] Kevin P. Balanda and H. L. MacGillivray. Kurtosis: A Critical Review. *The American Statistician*, 42(2):pp. 111–119, 1988. ISSN 00031305. URL http://www.jstor.org/stable/2684482.

[8] Aina Barcelo, Eduard Montseny, and Pilar Sobrevilla. Fuzzy Texture Unit and Fuzzy Texture Spectrum for Texture Characterization. *Fuzzy Sets Syst.*, 158(3):239–252, February 2007. ISSN 0165-0114. doi: 10.1016/j.fss.2006.10.008. URL http://dx.doi.org/10.1016/j.fss.2006.10.008.

[9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision - ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-33832-1. doi: 10.1007/11744023_32. URL http://dx.doi.org/10.1007/11744023_32.

[10] P.N. Belhumeur, J.P. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, Jul 1997. ISSN 0162-8828. doi: 10.1109/34.598228.

[11] S. Belongie and J. Malik. Matching with shape contexts. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 20–26, 2000. doi: 10.1109/IVL.2000.853834.

[12] A. Bosch, A. Zisserman, and X. Munoz. Representing Shape with a Spatial Pyramid Kernel. In *ACM International Conference on Image and Video Retrieval*, 2007.

[13] A. Bosch, A. Zisserman, and X. Muoz. Image Classification using Random Forests and Ferns. In *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007.*, pages 1–8, October 2007. doi: 10.1109/ICCV.2007.4409066.

[14] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[15] M. Brown and S. Susstrunk. Multi-spectral SIFT for scene category recognition. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 177–184, June 2011. doi: 10.1109/CVPR.2011.5995637.

[16] G. J. Burghouts and J. M. Geusebroek. Performance Evaluation of Local Colour Invariants. *Computer Vision and Image Understanding*, 113:48–62, 2009. URL http://www.science.uva.nl/research/publications/2009/BurghoutsCVIU2009.

[17] P.J. Burt and E.H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31: 532–540, 1983.

[18] Jeffrey Byrne and Jianbo Shi. Nested Shape Descriptors. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 1201–1208, Dec 2013. doi: 10.1109/ICCV.2013.152.

[19] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision - ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15560-4. doi: 10.1007/978-3-642-15561-1_56. URL http://dx.doi.org/10.1007/978-3-642-15561-1_56.

[20] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679 –698, November 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767851.

[21] Xiaochun Cao, Hua Zhang, Si Liu, Xiaojie Guo, and Liang Lin. SYM-FISH: A Symmetry-Aware Flip Invariant Sketch Histogram Shape Descriptor. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 313–320, Dec 2013. doi: 10.1109/ICCV.2013.46.

[22] David Casasent and Demetri Psaltis. Position, rotation, and scale invariant optical correlation. *Applied Optics*, 15(7):1795–1799, 1976. URL http://dx.doi.org/10.1364/AO.15.001795.

[23] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod. CHoG: Compressed histogram of gradients A low bit-rate feature descriptor. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2504–2511, June 2009. doi: 10.1109/CVPR.2009.5206733.

[24] Jie Chen, Shiguang Shan, Chu He, Guoying Zhao, M. Pietikainen, Xilin Chen, and Wen Gao. WLD: A Robust Local Image Descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1705–1720, September 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.155.

[25] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995. DOI: 10.1007/BF00994018.

[26] James L. Crowley. *A Representation for Visual Information with Application to Machine Vision*. PhD thesis, The Robotics Institute Carnegie-Mellon University Pittsburgh, Pennsylvania 15213, Pittsburgh, PA, USA, 1982. AAI8305202.

[27] James L. Crowley and Alice C. Parker. A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):156–170, March 1984. ISSN 0162-8828. doi: 10.1109/TPAMI.1984.4767500.

[28] James L. Crowley and Richard M. Stern. Fast Computation of the Difference of Low-Pass Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):212–222, March 1984. ISSN 0162-8828. doi: 10.1109/TPAMI.1984.4767504.

[29] James L. Crowley, Olivier Riff, and Justus H. Piater. Fast Computation of Characteristic Scale Using a Half-Octave Pyramid. In *Scale Space 03: 4th International Conference on Scale-Space theories in Computer Vision, Isle of Skye*, 2002.

[30] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 1, pages 886–893 vol. 1, June 2005. doi: 10.1109/CVPR.2005.177.

[31] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972. ISSN 0001-0782. doi: 10.1145/361237.361242. URL http://doi.acm.org/10.1145/361237.361242.

[32] Bin Fan, Fuchao Wu, and Zhanyi Hu. Rotationally Invariant Descriptors Using Intensity Order Pooling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2031–2045, 2012. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.277.

[33] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet. Icsiboost. http://code.google.come/p/icsiboost, 2007. URL https://github.com/benob/icsiboost.

[34] T. Fawcett. Roc Graphs: Notes and Practical Considerations for Researchers. Technical report, HP Laboratories, 2004. URL http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf.

[35] M. Felsberg and G. Sommer. The monogenic signal. *Signal Processing, IEEE Transactions on*, 49(12):3136–3144, Dec 2001. ISSN 1053-587X. doi: 10.1109/78.969520.

[36] R. A. Fisher. The use of multiple measures in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. ISSN 2050-1439. doi: 10.1111/j.1469-1809.1936.tb02137.x. URL http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x.

[37] Jan Flusser. On the independence of rotation moment invariants. *Pattern Recognition*, 33(9):1405 – 1410, 2000. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/S0031-3203(99)00127-2. URL http://www.sciencedirect.com/science/article/pii/S0031320399001272.

[38] P.-E. Forssen and A. Moe. Autonomous Learning of Object Appearances using Colour Contour Frames. In *The 3rd Canadian Conference on Computer and Robot Vision, 2006.*, pages 3–3, June 2006. doi: 10.1109/CRV.2006.17.

[39] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991. ISSN 0162-8828. doi: 10.1109/34.93808.

[40] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991. ISSN 0162-8828. doi: 10.1109/34.93808.

[41] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, pages 23–37, London, UK, UK, 1995. Springer-Verlag. ISBN 3-540-59119-2. URL http://dl.acm.org/citation.cfm?id=646943.712093.

[42] Xiaofeng Fu and Wei Wei. Centralized Binary Patterns Embedded with Image Euclidean Distance for Facial Expression Recognition. In *Fourth International Conference on Natural Computation, 2008. ICNC '08.*, volume 4, pages 115–119, October 2008. doi: 10.1109/ICNC.2008.94.

[43] H. Funaya and K. Ikeda. A statistical analysis of soft-margin support vector machines for non-separable problems. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1 –7, June 2012. doi: 10.1109/IJCNN.2012.6252443.

[44] T. Gevers. Robust histogram construction from color invariants. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 615–620 vol.1, 2001. doi: 10.1109/ICCV.2001.937575.

[45] Gösta H. Granlund and Hans Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA, 1995. ISBN 0792395301.

[46] R. Gupta, H. Patil, and A. Mittal. Robust order-based methods for feature description. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 334–341, June 2010. doi: 10.1109/CVPR.2010.5540195.

[47] D. Hall and J. L. Crowley. Face detection by robust generic features computed from luminance. *Reconnaissance des Formes et Intelligence Artificiel (RFIA)*, 2004.

[48] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[49] D.C. Hauagge and N. Snavely. Image matching using local symmetry features. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 206–213, June 2012. doi: 10.1109/CVPR.2012.6247677.

[50] Dong-Chen He and Li Wang. Texture Unit, Texture Spectrum, And Texture Analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4): 509–512, July 1990. ISSN 0196-2892. doi: 10.1109/TGRS.1990.572934.

[51] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of Interest Regions with Local Binary Patterns. *Pattern Recognition*, 42(3): 425–436, March 2009. ISSN 0031-3203. doi: 10.1016/j.patcog.2008.08.014. URL http://dx.doi.org/10.1016/j.patcog.2008.08.014.

[52] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient Response Maps for Real-Time Detection of Textureless Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):876–888, May 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.206.

[53] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, February 1962. ISSN 0096-1000. doi: 10.1109/TIT.1962.1057692.

[54] Rui Hu, M. Barnard, and J. Collomosse. Gradient field descriptor for sketch based retrieval and localization. In *2010 17th IEEE International Conference on Image Processing (ICIP)*, pages 1025–1028, September 2010. doi: 10.1109/ICIP.2010.5649331.

[55] D. H. Hubel and T. N. Wiesel. Receptive Fields Of Single Neurones In The Cat's Striate Cortex. *Journal of Physiology*, 148:574–591, 1959.

[56] Mark J. Huiskes and Michael S. Lew. The MIR Flickr Retrieval Evaluation. In *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA, 2008. ACM.

[57] Hueihan Jhuang and Sharat Chikkerur. Video Shot Boundary Detection Using Gist, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.207.6633.

[58] B. Johansson and A. Moe. Patch-duplets for object recognition and pose estimation. In *The 2nd Canadian Conference on Computer and Robot Vision, 2005. Proceedings.*, pages 9–16, May 2005. doi: 10.1109/CRV.2005.58.

[59] Andrew Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[60] Luo Juan and Oubong Gwon. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[61] ElchinS. Julfayev, RyanJ. McLaughlin, Yi-Ping Tao, and WilliamA. McLaughlin. A new approach to assess and predict the functional roles of proteins across all known structures. *Journal of Structural and Functional Genomics*, 12(1):9–20, 2011. ISSN 1345-711X. doi: 10.1007/s10969-011-9105-3. URL http://dx.doi.org/10.1007/s10969-011-9105-3.

[62] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In Tomás Pajdla and Jiří Matas, editors, *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 228–241. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-21984-2. doi: 10.1007/978-3-540-24670-1_18. URL http://dx.doi.org/10.1007/978-3-540-24670-1_18.

[63] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. ISSN 0920-5691. doi: 10.1007/BF00133570. URL http://dx.doi.org/10.1007/BF00133570.

[64] Yan Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–506–II–513 Vol.2, June 2004. doi: 10.1109/CVPR.2004.1315206.

[65] B. Khaleghi, M. Baklouti, and F.O. Karray. SILT: Scale-invariant line transform. In *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 78–83, Dec 2009. doi: 10.1109/CIRA.2009.5423244.

[66] A. Khotanzad and Yaw Hua Hong. Rotation invariant pattern recognition using Zernike moments. In *9th International Conference on Pattern Recognition*, pages 326–328 vol.1, November 1988. doi: 10.1109/ICPR.1988.28233.

[67] Sungho Kim, Kuk-Jin Yoon, and In-So Kweon. Object Recognition Using a Generalized Robust Invariant Feature and Gestalt's Law of Proximity and Similarity. In *Conference on Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06.*, pages 193–193, June 2006. doi: 10.1109/CVPRW.2006.146.

[68] Takumi Kobayashi and Nobuyuki Otsu. Image Feature Extraction Using Gradient Local Auto-Correlations. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 346–358, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_27. URL http://dx.doi.org/10.1007/978-3-540-88682-2_27.

[69] J.J. Koenderink and A.J. Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987. ISSN 0340-1200. doi: 10.1007/BF00318371. URL http://dx.doi.org/10.1007/BF00318371.

[70] I. Kokkinos and A. Yuille. Scale invariance without scale selection. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587798.

[71] Jáchym Kolár and Lori Lamel. Development and evaluation of automatic punctuation for french and english speech-to-text. In *INTERSPEECH*, 2012.

[72] S. Kondra, A. Petrosino, and S. Iodice. Multi-scale kernel operators for reflection and rotation symmetry: Further achievements. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 217–222, June 2013. doi: 10.1109/CVPRW.2013.39.

[73] Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. FastMatch: Fast Affine Template Matching. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1940–1947. IEEE, 2013.

[74] I Kviatkovsky, A Adam, and E. Rivlin. Color invariants for person reidentification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(7):1622–1634, July 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.246.

[75] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Semi-Local Affine Parts for Object Recognition. In *British Machine Vision Conference (BMVC)*, pages 959–968, 2004.

[76] S. Leutenegger, M. Chli, and R.Y. Siegwart. BRISK: Binary Robust invariant scalable keypoints. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011. doi: 10.1109/ICCV.2011.6126542.

[77] Wang Li, Wu Chengdong, Chen Dongyue, and Lu Baihua. Rotation-Invariant Human Detection Scheme Based on Polar-HOGs Feature and Double Scales Direction Estimation. In *2011 Symposium on Photonics and*

Optoelectronics (SOPO), pages 1–4, May 2011. doi: 10.1109/SOPO.2011.5780495.

[78] Tony Lindeberg. On the axiomatic foundations of linear scale-space: Combining semi-group structure with causality vs. scale invariance, 1997. Technical report, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden, August 1994. (ISRN KTH NA/P–94/20–SE). Revised version published as Chapter 6 in J. Sporring, M. Nielsen, L. Florack, and P. Johansen (eds.) *Gaussian Scale-Space Theory: Proc. PhD School on Scale-Space Theory*, (Copenhagen, Denmark, May 1996), Kluwer Academic Publishers, 75–98.

[79] Haibin Ling and D.W. Jacobs. Deformation invariant image matching. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005.*, volume 2, pages 1466–1473 Vol. 2, Oct 2005. doi: 10.1109/ICCV.2005.67.

[80] Jingchen Liu, G. Slota, Gang Zheng, Zhaohui Wu, Minwoo Park, Seungkyu Lee, I. Rauschert, and Yanxi Liu. Symmetry Detection from Real World Images Competition 2013: Summary and Results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 200–205, June 2013. doi: 10.1109/CVPRW.2013.155. URL http://vision.cse.psu.edu/research/symComp13/index.shtml.

[81] Kun Liu, Henrik Skibbe, Thorsten Schmidt, Thomas Blein, Klaus Palme, Thomas Brox, and Olaf Ronneberger. Rotation-Invariant HOG Descriptors Using Fourier Analysis in Polar and Spherical Coordinates. *International Journal of Computer Vision*, 106(3):342–364, 2014. ISSN 0920-5691. doi: 10.1007/s11263-013-0634-z. URL http://dx.doi.org/10.1007/s11263-013-0634-z.

[82] Yanxi Liu, Hagit Hel-Or, Craig S. Kaplan, and Luc Van Gool. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision*, 5(1-2):1–195, 2010. ISSN 1572-2740. doi: 10.1561/0600000008. URL http://dx.doi.org/10.1561/0600000008.

[83] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL http://dx.doi.org/10.1023/B%3AVISI.0000029664.99615.94.

[84] D.G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 2, pages 1150–1157 vol.2, August 1999. doi: 10.1109/ICCV.1999.790410.

[85] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In *Proceedings of the 9th European Conference on*

Computer Vision - Volume Part II, ECCV'06, pages 508–521, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33834-9, 978-3-540-33834-5. doi: 10.1007/11744047_39. URL http://dx.doi.org/10.1007/11744047_39.

[86] Guojun Lu and Atul Sajjanhar. Region-based shape representation and similarity measure suitable for content-based image retrieval. *Multimedia Systems*, 7(2):165–174, 1999. ISSN 0942-4962. doi: 10.1007/s005300050119. URL http://dx.doi.org/10.1007/s005300050119.

[87] Bingpeng Ma, Shiguang Shan, Xilin Chen, and Wen Gao. Head yaw estimation from asymmetry of facial appearance. *Trans. Sys. Man Cyber. Part B*, 38(6):1501–1512, December 2008. ISSN 1083-4419. doi: 10.1109/TSMCB.2008.928231. URL http://dx.doi.org/10.1109/TSMCB.2008.928231.

[88] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput.*, pages 761–767, 2004.

[89] E. Michaelsen, D. Muench, and M. Arens. Recognition of symmetry structure by use of gestalt algebra. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 206–210, June 2013. doi: 10.1109/CVPRW.2013.37.

[90] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 128–142, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-43745-2. URL http://dl.acm.org/citation.cfm?id=645315.649184.

[91] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.188.

[92] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1-2):43–72, November 2005. ISSN 0920-5691. doi: 10.1007/s11263-005-3848-x. URL http://dx.doi.org/10.1007/s11263-005-3848-x.

[93] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000027790.02288.f2. URL http://dx.doi.org/10.1023/B%3AVISI.0000027790.02288.f2.

[94] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. URL http://lear.inrialpes.fr/pubs/2004/MS04.

[95] F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, PAMI-8(1):34–43, January 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767750.

[96] Jean-Michel Morel and Guoshen Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM J. Img. Sci.*, 2(2):438–469, April 2009. ISSN 1936-4954. doi: 10.1137/080732730. URL http://dx.doi.org/10.1137/080732730.

[97] Eric N. Mortensen, Hongli Deng, and Linda Shapiro. A SIFT Descriptor with Global Context. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 184–190, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.45. URL http://dx.doi.org/10.1109/CVPR.2005.45.

[98] S. Murala and Q.M.J. Wu. Peak Valley Edge Patterns: A New Descriptor for Biomedical Image Indexing and Retrieval. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 444–449, June 2013. doi: 10.1109/CVPRW.2013.73.

[99] Klas Nordberg. A fourth order tensor for representation of orientation and position of oriented segments. Technical Report LiTH-ISY-R-2587, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, May 2004.

[100] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing.*, volume 1, pages 582–585 vol.1, October 1994. doi: 10.1109/ICPR.1994.576366.

[101] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.1017623.

[102] Aude Oliva and Antonio Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vision*, 42 (3):145–175, May 2001. ISSN 0920-5691. doi: 10.1023/A:1011139631724. URL http://dx.doi.org/10.1023/A:1011139631724.

[103] Margarita Osadchy, David W. Jacobs, and Michael Lindenbaum. Surface Dependent Representations for Illumination Insensitive Image Comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1): 98–111, 2007. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2007.19.

[104] N. Otsu. A new scheme for practical, flexible and intelligent vision systems. *Proc. IAPR Workshop on Computer Vision, 1988*, pages 431–435, 1988. URL http://ci.nii.ac.jp/naid/80004616356/en/.

[105] N. Otsu. Chlac approach to flexible and intelligent vision systems. In *ECSIS Symposium on Bio-inspired Learning and Intelligent Systems for Security, 2008. BLISS '08.*, pages 23–33, Aug 2008. doi: 10.1109/BLISS.2008.25.

[106] M. Ozuysal, P. Fua, and V. Lepetit. Fast Keypoint Recognition in Ten Lines of Code. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07.*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383123.

[107] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision, 1998.*, pages 555–562, January 1998. doi: 10.1109/ICCV.1998.710772.

[108] Jae-Han Park, Kyung-Wook Park, Seung-Ho Baeg, and Moon-Hong Baeg. pi-sift: A photometric and scale invariant feature transform. In *19th International Conference on Pattern Recognition, 2008. ICPR 2008.*, pages 1–4, December 2008. doi: 10.1109/ICPR.2008.4761181.

[109] V. Patraucean, R.G. von Gioi, and M. Ovsjanikov. Detection of mirror-symmetric image patches. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 211–216, June 2013. doi: 10.1109/CVPRW.2013.38.

[110] E. Persoon and King-Sun Fu. Shape discrimination using fourier descriptors. *IEEE Transactions on Systems, Man and Cybernetics*, 7(3):170–179, March 1977. ISSN 0018-9472. doi: 10.1109/TSMC.1977.4309681.

[111] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33832-2, 978-3-540-33832-1. doi: 10.1007/11744023_34. URL http://dx.doi.org/10.1007/11744023_34.

[112] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *2011 IEEE International Conference on Computer Vision (ICCV)*, Barcelona, November 2011.

[113] J. A. Ruiz-Hernandez, A. Lux, and J. L. Crowley. Face detection by cascade of Gaussian derivates classifiers calculated with a Half-Octave Pyramid. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1 –6, sep 2008. doi: 10.1109/AFGR.2008.4813457.

[114] S. Salti, A. Lanza, and L. Di Stefano. Keypoints from symmetries by wave propagation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2898–2905, June 2013. doi: 10.1109/CVPR.2013.373.

[115] M. Saquib Sarfraz and Olaf Hellwich. Head Pose Estimation in Face Recognition Across Pose Scenarios. In Alpesh Ranchordas and Helder

Araújo, editors, *VISAPP (1)*, pages 235–242. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008. ISBN 978-989-8111-21-0. URL http://dblp.uni-trier.de/db/conf/visapp/visapp2008-1.html#SarfrazH08.

[116] Bernt Schiele and James L. Crowley. Object recognition using multidimensional receptive field histograms. In Bernard Buxton and Roberto Cipolla, editors, *Computer Vision - ECCV '96*, volume 1064 of *Lecture Notes in Computer Science*, pages 610–619. Springer Berlin Heidelberg, 1996. ISBN 978-3-540-61122-6. doi: 10.1007/BFb0015571. URL http://dx.doi.org/10.1007/BFb0015571.

[117] Bernt Schiele and James L. Crowley. Recognition without Correspondence using Multidimensional Receptive Field Histograms. *International Journal of Computer Vision*, 36:31–50, 2000.

[118] Hae Jong Seo and P. Milanfar. Training-Free, Generic Object Detection Using Locally Adaptive Regression Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1688–1704, September 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.153.

[119] Gaurav Sharma, Sibt Ul Hussain, and Frédéric Jurie. Local Higher-Order Statistics (LHS) for Texture Categorization and Facial Analysis. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV 2012 - European Conference on Computer Vision*, volume 7578 of *Lecture Notes in Computer Science*, pages 1–12, Florence, Italie, August 2012. Springer. doi: 10.1007/978-3-642-33786-4_1. URL http://hal.inria.fr/hal-00722819.

[120] E. Shechtman and M. Irani. Matching Local Self-Similarities across Images and Videos. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07.*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383198.

[121] Robert Söderberg, Klas Nordberg, and Gösta Granlund. An Invariant and Compact Representation for Unrestricted Pose Estimation. In JorgeS. Marques, Nicolás Pérez de la Blanca, and Pedro Pina, editors, *Pattern Recognition and Image Analysis*, volume 3522 of *Lecture Notes in Computer Science*, pages 3–10. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26153-7. doi: 10.1007/11492429_1. URL http://dx.doi.org/10.1007/11492429_1.

[122] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. Unified Real-Time Tracking and Recognition with Rotation-Invariant Fast Features. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 934–941, June 2010. doi: 10.1109/CVPR.2010.5540116.

[123] Xiaoyang Tan and B. Triggs. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Transactions*

on Image Processing, 19(6):1635–1650, June 2010. ISSN 1057-7149. doi: 10.1109/TIP.2010.2042645.

[124] Feng Tang, Suk Hwan Lim, N.L. Chang, and Hai Tao. A novel feature descriptor invariant to complex brightness changes. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, pages 2631–2638, June 2009. doi: 10.1109/CVPR.2009.5206550.

[125] M. Tkalcic and J.F. Tasic. Colour spaces: perceptual, historical and applicational background. In *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, volume 1, pages 304–308 vol.1, Sept 2003. doi: 10.1109/EURCON.2003.1248032.

[126] Engin Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587673.

[127] Federico Tombari, Alessandro Franchi, and Luigi Di Stefano. BOLD Features to Detect Texture-less Objects. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 1265–1272, December 2013. doi: 10.1109/ICCV.2013.160.

[128] M.A. Turk and A.P. Pentland. Face Recognition Using Eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91.*, pages 586–591, Jun 1991. doi: 10.1109/CVPR.1991.139758.

[129] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000020671.28016.e8. URL http://dx.doi.org/10.1023/B%3AVISI.0000020671.28016.e8.

[130] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, ECCV'06, pages 589–600, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33834-9, 978-3-540-33834-5. doi: 10.1007/11744047_45. URL http://dx.doi.org/10.1007/11744047_45.

[131] C.W. Tyler, L. Hardage, and R.T. Miller. Multiple mechanisms for the detection of mirror symmetry. *Spatial Vision*, 9(1):79–100, 1995.

[132] K. E A Van de Sande, T. Gevers, and C. G M Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, September 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.154.

[133] P. Vandergheynst, R. Ortiz, and A. Alahi. FREAK: Fast Retina Keypoint. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:510–517, 2012. ISSN 1063-6919. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2012.6247715.

[134] Andrea Vedaldi and Brian Fulkerson. Dense Scale Invariant Feature Transform (DSIFT). From the VLFeat library. URL http://www.vlfeat.org/api/dsift.html.

[135] Andrea Vedaldi and Brian Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[136] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[137] M. Villamizar, J. Scandaliaris, A. Sanfeliu, and J. Andrade-Cetto. Combining color-based invariant gradient detector with HoG descriptors for robust image detection in scenes under cast shadows. In *IEEE International Conference on Robotics and Automation, 2009. ICRA '09.*, pages 1997–2002, May 2009. doi: 10.1109/ROBOT.2009.5152429.

[138] P. Viola and M. Jones. Rapid Object Detection using a Boosted cascade of Simple Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001.

[139] Zhenhua Wang, Bin Fan, and Fuchao Wu. Local Intensity Order Pattern for feature description. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 603–610, November 2011. doi: 10.1109/ICCV.2011.6126294.

[140] Zhiheng Wang, Fuchao Wu, and Zhanyi Hu. MSLD: A Robust Descriptor for Line Matching. *Pattern Recognition*, 42(5):941–953, May 2009. ISSN 0031-3203. doi: 10.1016/j.patcog.2008.08.035. URL http://dx.doi.org/10.1016/j.patcog.2008.08.035.

[141] Max Wertheimer. Untersuchungen zur lehre von der gestalt. ii. *Psychologische Forschung*, 4(1):301–350, 1923. ISSN 0033-3026. doi: 10.1007/BF00410640. URL http://dx.doi.org/10.1007/BF00410640.

[142] Jianxin Wu and J.M. Rehg. CENTRIST: A Visual Descriptor for Scene Categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1489–1501, Aug 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.224.

[143] Puyang Xu and Ruhi Sarikaya. Exploiting shared information for multi-intent natural language sentence classification. In *INTERSPEECH*, pages 3785–3789, 2013.

[144] Xuebin Xu, Xinman Zhang, Jiuqiang Han, and Cailing Wu. HALCON application for shape-based matching. In *3rd IEEE Conference on Industrial Electronics and Applications, 2008. ICIEA 2008.*, pages 2431–2434, June 2008. doi: 10.1109/ICIEA.2008.4582953.

[145] Huixing Ye, Roland Hu, Huimin Yu, and RobertIan Damper. Face Recognition Based on Adaptive Soft Histogram Local Binary Patterns. In

Zhenan Sun, Shiguan Shan, Gongping Yang, Jie Zhou, Yunhong Wang, and YiLong Yin, editors, *Biometric Recognition*, volume 8232 of *Lecture Notes in Computer Science*, pages 62–70. Springer International Publishing, 2013. ISBN 978-3-319-02960-3. doi: 10.1007/978-3-319-02961-0_8. URL http://dx.doi.org/10.1007/978-3-319-02961-0_8.

[146] Qiang Zhu, M.-C. Yeh, Kwang-Ting Cheng, and S. Avidan. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1491–1498, 2006. doi: 10.1109/CVPR.2006.119.

[147] S. Zokai and G. Wolberg. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *IEEE Transactions on Image Processing*, 14(10):1422–1434, Oct 2005. ISSN 1057-7149. doi: 10.1109/TIP.2005.854501.