CONTRIBUTIONS TO THE COMPLETENESS AND COMPLEMENTARITY OF LOCAL IMAGE FEATURES

PEDRO JOSÉ MENDES MARTINS

Tese de Doutoramento em Engenharia Informática
orientada pelo Professor Doutor Paulo Fernando Pereira de Carvalho
e apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia
da Universidade de Coimbra

Julho 2013

ACKNOWLEDGMENTS

It would not have been possible to conclude the writing of this dissertation without the aid and support of countless people, to only some of whom it is possible to give particular mention here.

First and foremost, I would like to express my gratitude to Dr. Paulo de Carvalho, my thesis advisor, who has supported me with his guidance, knowledge, and encouragement.

I am also indebted to Dr. Carlo Gatta, who was an invaluable support. The good advice, motivation, knowledge, and friendship of Dr. Carlo Gatta have certainly made my research a fulfilling and rewarding experience.

I am very grateful to the Institute for Interdisciplinary Research of the University of Coimbra (III-UC) and to the ArthroNav project (funded by the Portuguese Foundation for Science and Technology under the grant PTDC/EEA-ACR/68887/2006) for their financial support, without which it would not have been possible to complete the present document.

I am also grateful to the anonymous reviewers who have read my work with professional care and attention and have offered valuable comments and suggestions.

I would also like to thank my colleagues at the Center for Informatics and Systems (CISUC), especially Dinesh Kumar, with whom I shared the joys, struggles, and miseries of doing research.

I am equally grateful to my colleagues at the Computer Vision Center (Universitat Autònoma de Barcelona) for providing a stimulating environment during my short visit as a PhD student.

Finally, I would like to express my gratitude to my family and friends for their unconditional support and encouragement. I am particularly grateful to Cristina for her extraordinary patience and for always being there for me.

ABSTRACT

Local image feature detection (or extraction, if we want to use a more semantically correct term) is a central and extremely active research topic in the fields of computer vision and image analysis. Local features have been used as the basis for solutions to prominent problems such as matching, content-based image retrieval, object recognition, and symmetry detection.

It is widely accepted that a good local feature detector is the one that efficiently retrieves distinctive, accurate, and repeatable features in the presence of a wide variety of photometric and geometric transformations. However, these requirements are not always the most important. In fact, not all the applications require the same properties from a local feature detector. We can distinguish three broad categories of applications according to the required properties. The first category includes applications in which the semantic meaning of a particular type of features is exploited. For instance, edge or even ridge detection can be used to identify blood vessels in medical images or watercourses in aerial images. Another example in this category is the use of blob extraction to identify blob-like organisms in microscopic images. A second category includes tasks such as matching, tracking, and registration, which mainly require distinctive, repeatable, and accurate features. Finally, a third category comprises applications such as object (class) recognition, image retrieval, scene classification, and image compression. For this category, it is crucial that features preserve the most informative image content (robust image representation), while requirements such as repeatability and accuracy are of less importance.

Our research work is mainly focused on the problem of providing a robust image representation through the use of local features. The limited number of types of features that a local feature extractor responds to might be insufficient to provide the so-called robust image representation. It is fundamental to analyze the completeness of local features, i. e., the amount of image information preserved by local features, as well as the often neglected complementarity between sets of features.

The major contributions of this work come in the form of two substantially different local feature detectors aimed at providing considerably robust image representations. The first algorithm is an information theoretic-based feature extraction that responds to complementary local structures that are salient (highly informative) within the image context. This method represents a new paradigm in local feature extraction, as it introduces context-awareness principles. The second algorithm extracts Stable Salient Shapes, a novel type of regions, which are obtained through a feature-driven detection of Maximally Stable Extremal Regions (MSER). This method provides compact and robust image representations and overcomes some of the major shortcomings of MSER detection.

We empirically validate the methods by investigating the repeatability, accuracy, completeness, and complementarity of the proposed features on standard benchmarks. Under these results, we discuss the applicability of both methods.

RESUMO

A detecção de características locais (*local features*) de uma imagem é um tema central nas áreas de visão por computador e análise de imagem. Soluções eficazes para a resolução de problemas como a correspondência de imagens, recuperação de imagens baseada no conteúdo, reconhecimento de objectos e detecção de simetria são frequentemente suportadas pela detecção de características locais.

É comum considerar-se como um bom detector de características locais aquele que apresenta resultados repetíveis e exactos na presença de diversas transformações geométricas e fotométricas. Além disso, o detector deverá ser capaz de detectar estruturas relativamente distintas entre si. Contudo, estas propriedades não são sempre as mais desejadas; nem todas as aplicações exigem as mesmas propriedades a um detector. É possível identificar três grandes categorias de domínios de aplicação de acordo com as propriedades que estas requerem aos detectores. A primeira inclui aplicações onde o significado semântico de um dado tipo de característica local é explorado. A título de exemplo, a detecção de arestas e linhas pode ser utilizada na identificação de vasos sanguíneos em imagens médicas, assim como na identificação de cursos de água em imagens aéreas. Um outro exemplo pertencente a esta categoria é a extracção de *blobs* em imagens microscópicas com o objectivo de identificar organismos com a mesma forma. Uma segunda categoria integra tarefas como a correspondência de imagens, o seguimento e o co-registo, que requerem sobretudo características locais que sejam repetíveis, exactas e distintas entre si. A terceira categoria compreende aplicações como o reconhecimento de objectos (ou de categorias de objectos), a recuperação de imagens e a compressão. Nesta categoria é crucial que as características locais preservem o conteúdo mais relevante da imagem de forma a constituírem uma representação robusta da imagem. Neste caso, propriedades como a repetição e a exactidão tendem a tornar-se requisitos secundários.

O trabalho de investigação aqui apresentado centra-se maioritariamente no problema de representar robustamente uma imagem através de características locais. O número limitado de tipos de estruturas a que um detector habitualmente responde poderá ser insuficiente para obter uma representação robusta. Torna-se relevante estudar a completude das características locais, i. e., a quantidade de informação da imagem que é preservada por estes elementos, assim como a complementaridade entre diferentes tipos de características locais. A nossa investigação desenvolve-se em torno destas métricas, tendo como resultado a apresentação de dois algoritmos para a extracção de características locais que conseguem assegurar uma representação robusta da imagem, sem descurar outros requisitos relevantes.

O primeiro algoritmo, baseado em teoria de informação, extrai partes da imagem que são salientes (altamente informativas) segundo o contexto da imagem. O método em questão representa um novo paradigma na extracção de características locais, uma vez que recorre ao contexto da imagem. O segundo algoritmo extrai regiões denominadas de *Stable Salient Shapes* que resultam da extracção de Regiões Extremas Maximamente Estáveis (ou MSER) em mapas de saliência onde as características semi-locais presentes definem a forma e o tamanho das primeiras. Este método assegura uma representação compacta e robusta da imagem e reduz algumas das maiores limitações do detector de Regiões Extremas Maximamente Estáveis.

Validamos experimentalmente ambos os métodos através da análise de critérios como a repetição, exactidão, completude e complementaridade em vários *benchmarks* padrão. Sob estes resultados, discutimos a aplicabilidade dos algoritmos.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

MAIN NOTATION

$\mathbb{R}$	real numbers set
$\mathbb{R}^+$	positive real numbers set
A^T	transpose of (matrix) A
A^{-1}	inverse of (matrix) A
$\det(\cdot)$	determinant
$\operatorname{trace}(\cdot)$	trace
μ	structure tensor matrix
$\mathcal{H}$	Hessian matrix
∇f	gradient of f
$\nabla^2 f$	Laplacian of f
σ_I	integration scale
σ_D	differentiation scale
$*$	convolution operator
χ^2	χ-squared distribution
$\max$ / $\max\{\cdot\}$	maximum
$\min$ / $\min\{\cdot\}$	minimum
argmax	argument of the maximum
argmin	argument of the minimum
argmaxlocal	argument of the local maximum
$\exp(\cdot)$	exponential
$\log(\cdot)$	natural logarithm
$\log_k(\cdot)$	logarithm base k
$\lvert\mathbf{x}\rvert$	absolute value of $\mathbf{x}$
$\lvert S\rvert$	cardinality of (set) S
$\lVert\cdot\rVert$	Euclidean norm
$\lVert\cdot\rVert_p$	Schatten p-norm

ACRONYMS AND INITIALISMS

CAKE	Context-Aware Keypoint Extraction
Caltech	California Institute of Technology
DC	Direct Current
DCT	Discrete Cosine Transform
DoG	Difference of Gaussians
EBR	Edge-Based Regions
FAST	Features from Accelerated Segment Test
FAST-ER	Features from Accelerated Segment Test - Enhanced Repeatability
HARAFF	Harris-Affine
HARLAP	Harris-Laplace
HESAFF	Hessian-Affine
HESLAP	Hessian-Laplace
IBR	Intensity Extrema-Based Regions
JPEG	Joint Photographic Experts Group
KDE	Kernel Density Estimation
LED	Light-Emitting Diode
LoG	Laplacian of Gaussian
MSER	Maximally Stable Extremal Region(s)
PCA	Principal Component Analysis
PCBR	Principal Curvature-Based Regions
PDF	Probability Density Function
PHOW	Pyramid Histogram Of Visual Words
Pixel	Picture Element
SAF	Stable Affine Frames
SFOP	Scale Invariant Feature Operator
SIFT	Scale Invariant Feature Transform
SSS	Stable Salient Shape(s)
SUSAN	Smallest Univalue Self Assimilating Nucleus
TNT	Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung
UBC	University of British Columbia
USAN	Univalue Self Assimilating Nucleus
[eigSTM]-CAKE	Context-Aware Keypoint Extraction based on the eigenvalues of the Structure Tensor Matrix
[HES]-CAKE	Hessian-based Context-Aware Keypoint Extraction

1 INTRODUCTION

1.1 MOTIVATION AND PROBLEM FORMULATION

> "There is no such thing as generic keypoints. They should be selected specifically for the use to which they will be put, using a purpose-designed detector and parameters." (Triggs, 2004, p. 102)

> "Even though a lot of progress has been made in the domain of feature extraction – especially with respect to the level of invariance –, and even though impressive applications have been built using local features, they still have a number of shortcomings." (Tuytelaars & Mikolajczyk, 2008, p. 83)

> "Overcomplete representations, which result from the simultaneous use of multiple detectors provide a temporary solution only in spite of efficient multi-type feature detectors. However, an efficient combination of complementary detectors or a multi-type detector providing complementary features for compact representation would be much more useful given the increasing amounts of data to process." (Tuytelaars & Mikolajczyk, 2008, p. 84)

Local image feature detection has been successfully used to solve a wide range of problems, including wide-baseline stereo matching (Baumberg, 2000; Matas et al., 2002; Tuytelaars & Gool, 2004), content-based image retrieval (Schmid & Mohr, 1997; Tuytelaars & Gool, 1999; Mirmehdi & Periasamy, 2001), object recognition (Dorkó & Schmid, 2003; Mikolajczyk et al., 2006; Schnitzspan et al., 2010), camera calibration (Förstner et al., 2009a), and symmetry detection (Loy & Eklundh, 2006; Deng et al., 2007).

Using sparse sets of locally salient image patches (the so-called local image features) is usually an efficient and robust solution to various problems. Efficiency is a natural consequence of discarding a major part of the image. Robustness is gained through the existence of redundant elements. Furthermore, local features can provide a compact representation of the image content.

Local feature detection is a mature research topic, which has seen an increasing popularity and prominence through its existence. Early algorithms were for the most part relatively simple solutions aimed at detecting keypoints, such as corner points (e. g., Moravec, 1977; Beaudet, 1978). Matching, i. e., the task of establishing correspondences between images, was one of the earliest applications to take advantage of local features (e. g., Förstner, 1986). The use of local features provided an efficient mechanism for matching tasks. However, local feature detectors were required to find "the same" keypoints in different images, regardless of the image transformation (Triggs, 2004). In other words, detectors were required to find keypoints in a repeatable and accurate manner.

There has been a significant research effort to develop algorithms that retrieve repeatable and accurate features. Current state-of-the-art solutions detect features with a relatively high repeatability rate. This is mainly achieved by using techniques that provide robustness against small image deformations and invariance/covariance with respect to larger image deformations, such as illumination changes or viewpoint changes.

The maturity that characterizes local feature detection is not synonym of a research without major open issues. Algorithms have been often presented as a generic tool, without taking into account the conflicting properties required by different applications. Performance evaluation has been mainly based on the repeatability criterion, which is not fully sufficient to reflect the usefulness of the algorithms. It is crucial to consider other criteria in the evaluation in order to define the proper application domain(s) for a given algorithm.

There was also a paradigm shift in local feature detection with the introduction of robust local descriptors (e. g., Lowe, 1999). Descriptors computed on local features are not just a robust and compact characterization of image content, they are also a powerful tool to classify scenes and recognize objects without the need for semantic-level segmentation (Tuytelaars & Mikolajczyk, 2008).

With the change in paradigm, aspects such as completeness, i. e., the amount of image information preserved by the local features, and complementarity between features became more relevant. Despite their importance, these two properties are often neglected in the design of feature detection algorithms (Dickscheid et al., 2011).

1.2 CONTRIBUTIONS AND RELEVANCE

This dissertation focuses on the study and on the design of local feature extractors, where aspects such as the completeness and the complementarity of features are taken into consideration. Our fundamental purpose is to study ways of ensuring a robust and compact image representation through the use of local features, without neglecting other fundamental aspects such as repeatability and distinctiveness. As a result, we propose two different algorithms for feature extraction. The first one, named Contex-Aware Keypoint Extractor (CAKE), represents a new paradigm in local feature extraction. The idea is to retrieve salient locations within the image context, which means no assumption is made on the type of structure to be detected. This scheme is designed to provide a robust image representation, with or without the contribution of other local features. The second algorithm detects a novel type of features, coined as Stable Salient Shapes (SSS). The new features are obtained through a feature-driven Maximally Stable Extremal Regions (MSER) detection (Matas et al., 2002). The feature-driven approach provides suitable domains for MSER detection. Such domains can be viewed as saliency maps in which features related to semantically meaningful structures, e. g., boundaries and symmetry axes, are highlighted and simultaneously delineated under smooth transitions. In comparison with MSER, SSS are more robust to blur and show substantially higher completeness values.

While both algorithms can achieve robust image representations, they represent two different ways of accomplishing such goal. The context-aware extractor is a tool explicitly designed to optimize the coverage of salient image parts, either at a local level or at a global level. As for the SSS detector, it is a computationally efficient method that simultaneously retains the noteworthy properties of the MSER detector and provides robust image representations. The major advantage of the second method over the first one is the computational complexity. On the other hand, the SSS detector makes assumptions on the image content, namely the existence of a well-structured scene. Since structural information is used to delineate the features, one can expect a less robust representation when structural information is less present in the image.

The performance of both methods was evaluated in a comparative study where state-of-the-art algorithms were included and different criteria were taken into consideration in order to derive more sustainable conclusions on the usefulness of features.

1.3 A NOTE ON TERMINOLOGY: DETECT(OR) VS. EXTRACT(OR)

The term detector has been widely used as the term to refer to an algorithm that extracts local features. As noted by Tuytelaars & Mikolajczyk (2008), the term detector only makes sense if one clearly knows what the features are so one can infer about misdetections or false detections. In fact, the term extractor would be semantically more appropriate. Throughout this dissertation, we will use both terms interchangeably, although a preference will be given to the former.

1.4 More notes on notation

Throughout this dissertation, we adopt the following notation: boldface lowercase letters indicate (column) vectors (e. g., $\mathbf{x} = [x\ y]^T$); I stands for a single-channel image, while L denotes an image resulting from the convolution of I with a Gaussian kernel G, i. e., $L(\mathbf{x}, \sigma) = G(\mathbf{x}, \sigma) * I(\mathbf{x})$, with $G(\mathbf{x}, \sigma) = \frac{1}{2\pi\sigma^2} \exp(-\frac{\|\mathbf{x}\|^2}{\sigma^2})$. To simplify the notation, the dependence of G on $\mathbf{x}$ will be left implicit. The gradient of a function f is represented by the row vector ∇f. Sometimes, the gradient of $L(\mathbf{x}, \sigma)$ is alternatively represented by $\nabla_\sigma L(\mathbf{x})$. The first order partial derivative of f with respect to u is denoted by f_u. Second order partial derivatives of f with respect to u and v are denoted by f_{uv}. Unless otherwise stated, matrices are represented by uppercase letters. $\lambda_1(A)$ and $\lambda_2(A)$ denote the minimum and maximum eigenvalues of a given matrix A, respectively. The dependence of λ_1 and λ_2 on A will be sometimes omitted.

1.5 Outline

The remainder of this dissertation is organized as follows:

Chapter 2 – Local feature detection: a review.

> This chapter introduces the basic concepts and definitions surrounding local feature detection and provides a review of local feature detectors. Given the number of methods available in the literature, our review is mainly focused on seminal algorithms and state-of-the-art-algorithms. We complement this information with an analysis of the main challenges and limitations of current local feature detection.

Chapter 3 – Evaluation of local features.

> In this chapter, we describe and discuss the different criteria, datasets, and benchmarks used in the evaluation of local feature detectors.

Chapter 4 – Context-aware features for robust image representation.

> Chapter 4 introduces the Context-Aware Keypoints Extractor, a context-aware local feature detector aimed at providing a robust image representation. This chapter also includes an experimental validation of the proposed algorithm.

chapter 5 – stable (salient) shapes: feature-driven maximally stable extremal regions.

In this chapter, we introduce the Stable Salient Shapes detector. This work is aimed at simultaneously overcoming significant shortcomings of the Maximally Stable Extremal Regions detector and providing robust image representations. The chapter is complemented with an experimental validation of the method.

chapter 6 – conclusions and perspectives.

Chapter 6 finalizes the dissertation by summarizing the main findings and analyzing further lines of research.

2 LOCAL FEATURE DETECTION: A REVIEW

The purpose of this chapter is to introduce fundamental concepts and definitions surrounding local feature detection and to provide a review of local feature detectors. Given the relevance and the maturity of the topic, we mainly focus our review on seminal algorithms and on established state-of-the art algorithms.

2.1 LOCAL FEATURES: PRELIMINAIRIES

A number of computer vision and image analysis tasks are based on the detection of local features, i. e., parts of an image that are more visually salient than their immediate surroundings. The key idea underlying the use of local features is to provide a representation of the image content by using a sparse set of locally salient parts. By discarding most of the image content, one saves computation and improves robustness, as there are redundant local image patches rather than a limited number of global cues (Triggs, 2004).

Depending on the application domain, local feature detectors are required to present different properties. For instance, in some cases, it is fundamental to perform a stable detection when geometric and photometric perturbations occur in order to provide repeatable and accurate features. In other cases, it is more critical to provide a fast detection to fulfill the time constraints imposed by the application.

The different requirements imposed by diverse applications have made local feature detection a very active and prolific research topic both in the area of computer vision and image analysis. In this chapter, we overview local feature detection, giving a special emphasis to seminal algorithms and current state-of-the-art solutions. For a more comprehensive review, we refer the reader to the works of Tuytelaars & Mikolajczyk (2008) and Szeliski (2010).

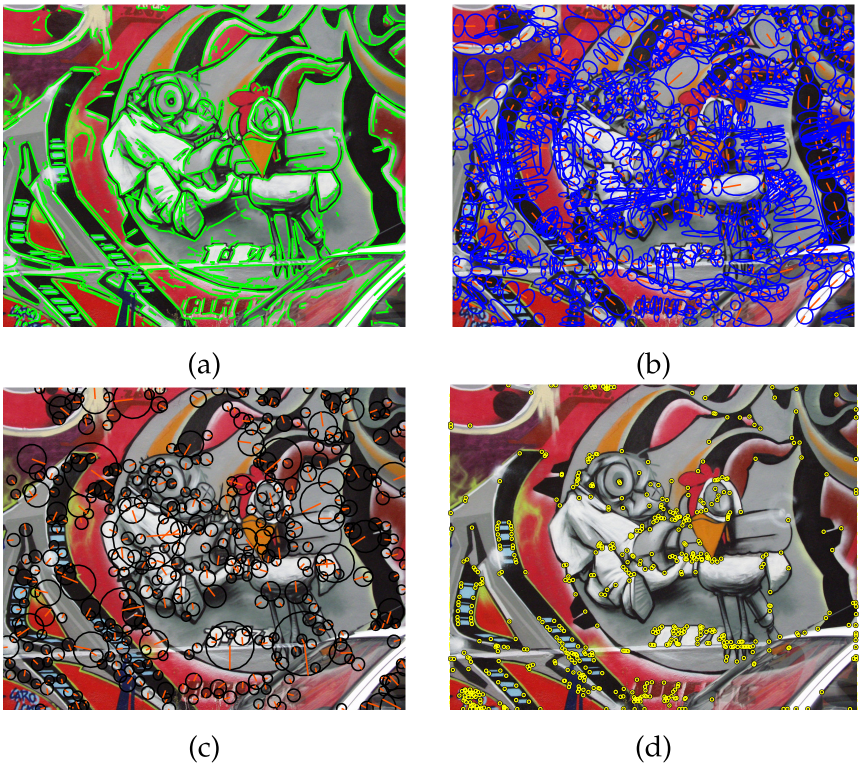
2.1.1 *Local features: a few examples*

In the context of human vision, visual perception is a multistage process comprehending several visual representations presented with an increasing degree of complexity (Marr, 1982). The early representation, known as the primal sketch, mainly consists of information about low-level features, such as edges, ridges, blobs, corners, and junctions. Low-level image features may either have a strictly local or a semi-local structure. Edges and ridges are examples of semi-local features due to their segmental structure. The remaining ones are classified as strictly local features.[1]

Corners are image locations that have significant intensity changes in more than one direction. Besides being perceptually relevant, corners have a well-defined location, which suggests a stable and robust detection.

Blobs are regions that are darker or brighter than the immediate surroundings. Their perceptual relevance is also high but in a different manner from corners: while corners represent fine details on the image, blobs convey most of the image information (Dickscheid et al., 2011).

Figure 2.1 depicts some examples of primal-sketch priors (Kokkinos et al., 2006), including corners and blobs.


(a)

(b)

(c)

(d)

Figure 2.1: Examples of local and semi-local features: (a) edges; (b) ridges; (c) blobs; (d) corners.

[1] This classification is not consensual. Usually, a blob is considered as a local structure. However, some authors consider it as semi-local one (e. g., Lindeberg, 1993).

2.1.2 *Ideal properties of a detector*

Local feature detectors are often presented as generic ones, i. e., as a multipurpose tool. However, the idea of a generic detector entails a certain impracticability (Triggs, 2004), as different applications might require different and sometimes conflicting properties from the detector. We enumerate the desirable properties of the generic local feature detector, which are supposed to fulfill the requirements of the majority of applications. We also illustrate the unfeasibility of a generic detector by presenting some conflicting requirements.

REPEATABLE

Features should be repeatable, that is, a detector should be able to detect "the same" features on two different images of the same scene, regardless of the underlying image transformation. A repeatable output is dependent on the robustness and on the invariance/covariance properties shown by the detector.

1. A local feature detector is considered to be robust against a minor image deformation – typically, a relatively small photometric distortion such as blur and compression artifacts – when detection is only slightly affected by the deformation, i. e., there is only a negligible loss of accuracy.

2. To achieve a repeatable detection when large deformations occur, one should design a method that shows an invariant/covariant response with respect to these deformations. A covariant response is required when feature detection has to change according to the transformation (e. g., a geometric transformation). An invariant response occurs when the detection is not affected by the transformation. For instance, invariance to illumination changes only occurs if we detect the same features under various lighting conditions. In other words, we can view invariance as optimal robustness.

ACCURATE

It is desirable to accurately retrieve the location as well as the scale and the shape of local features (Tuytelaars & Mikolajczyk, 2008).

DISTINCTIVE

The patterns or the structures in the immediate surroundings of the detected features should show a considerable degree of variation among themselves. Such property is fundamental for matching tasks, where features have to be easily distinguished (through description) in order to be matched.

COMPLETE

A detector should not only provide distinctive features, it should also provide a complete set of features, that is, the amount of image information preserved by a set of features should be maximized, without sacrificing the inherent sparseness of the set (Dickscheid et al., 2011). In other words, it is desirable that the cardinality of the set of features reflects the image information.

EFFICIENT

A local feature detector should be computationally efficient. This property becomes crucial in applications where local feature extraction appears as one of a series of tasks to be sequentially performed and a large amount of data is used.

2.1.2.1 *Conflicting requirements*

We can assume an ideal local feature detector, with no particular application in mind, as a computationally efficient algorithm that yields the highest repeatability rate under a large class of image transformations, provides an accurate location of the features, and efficiently covers the most informative content (without a biased preference for a given type of structures). However, these requirements tend to be conflicting.

Achieving invariance/covariance with respect to a given transformation may mean sacrificing the distinctiveness or even the completeness of the features. Invariant descriptors with respect to a certain transformation can only be effectively used if they are computed on features showing invariance (or covariance) with respect to the same transformation. However, such invariance is sometimes achieved by a less distinctive description. In addition, invariant/covariant features with respect to severe image transformations are sometimes the result of a convergence process in which various candidate features are discarded. As a result, completeness is reduced. Similar observations hold if we replace invariance/covariance with robustness.

Efficiency and completeness may also be conflicting requirements. Providing a denser set of features may substantially increase the computation time. Similarly, a denser set of features means less distinctive features, as there are repeated structures in the set.

Different applications require different properties from a local feature detector. It is fundamental to design a feature detector with an application domain in mind. In this way, the performance of the detector is optimized for a specific range of tasks.

2.1.3 *Applications*

Local features have been successfully used in a wide range of applications. In most cases, local features play an extremely crucial role, as their detection serves as a basis for subsequent tasks. Given the multitude of tasks in which local feature detection is used, we only describe a few typical applications which are known for their prominence.

2.1.3.1 *Matching*

(Feature) matching is a typical computer vision task based on the detection of local features. It can be performed either as an isolated task or it can be part of a more complex task. As the name suggests, its purpose is to find corresponding features in two or more images. The process is usually divided into three steps. Feature detection is the first one. The second step is called description, which involves the computation of descriptors for each feature previously extracted. Descriptors are summarized (geometric or photometric) representations of a given image patch. The third and final stage is known as matching and it consists of comparing descriptors by means of a given distance in order to determine matches.

Robust wide-baseline stereo matching (Pritchett & Zisserman, 1998) is a well-known example of a matching problem. Its purpose is to find corresponding features in pairs of overlapping images taken from significantly different viewpoints (Dickscheid, 2011). This problem has been one of the motivations to devise affine covariant methods (Baumberg, 2000; Matas et al., 2002; Tuytelaars & Gool, 2000). An affine invariant description would be ineffective if the detection was not affine covariant. For this kind of task, it is crucial to provide repeatable, accurate, and distinctive features.

2.1.3.2 *Camera calibration*

Camera calibration is the process of determining parameters such as focal length and lens distortion, which are responsible for defining the relationship between 3D world coordinates and 2D image coordinates. A feature-based camera calibration usually occurs in two stages. First, local features are extracted from a calibration object (e. g., a checkerboard calibration grid). Then, the camera model parameters are estimated from the extracted features. This task requires local features to be extracted from the different images in a repeatable and accurate manner (Mühlich & Aach, 2007). There is no need to describe features, only their location is relevant. As a result, keypoints (e. g., corner points and centers of mass of blobs) are the most suited type of features for camera calibration, as long as they are repeatable and accurate. For a reliable parameter estimation, it is also important not to have a reduced number of features. SFOP features (Förstner et al., 2009a) are an example of features proposed for camera calibration.

2.1.3.3 *Object recognition*

Object recognition is a complex and challenging computer vision task. In general terms, the goal of object recognition is to recognize an object in an image or video collection, regardless of the geometric or photometric transformations that objects may suffer. The problem of object recognition can be presented in several forms (Szeliski, 2010). If the object is known, the problem is called object detection and the task of recognition is to determine where a match may occur. In other words, recognition is mainly a matching problem. If the object to be recognized is a rigid object, local features can be used to verify the alignment. Sometimes, the goal is to recognize instances of different classes of objects (object class recognition). A common solution to the problem of object class recognition is to use bags of words (Lazebnik et al., 2006). Given an image, the algorithm determines the distribution of the visual words (descriptors) computed on local features and compares this distribution to others computed in the training images. Object class recognition is an example of a task that requires sets of features with a good coverage of informative image parts. Features are not supposed to be matched on an individual basis, the goal is to analyze and compare their statistics (Tuytelaars & Mikolajczyk, 2008). In this case, local features are used to provide a robust image representation. Repeatability and accuracy become secondary requirements.

2.2 A REVIEW OF LOCAL FEATURE DETECTORS

It is commonly accepted that if we trace back local feature detection to its roots, neglecting preliminary works that have mainly established the theoretical foundation of local feature detection (Attneave, 1954; Fend & Pavlidis, 1973; Rosenfeld & Thurston, 1971), it will lead us to Moravec's detector (Moravec, 1977, 1980), which was aimed at providing the core for a navigation system of a mobile robot through a clustered environment. In his solution, the local saliency measure relied on the minimum intensity variation value computed over unitary shifts of a window centered at a location $\mathbf{x}$, performed across the four principal directions. This detector exhibited a higher response to corners and isolated pixels. Additionally, Moravec coined these locations as points of interest, which has been the terminology adopted by the vision community and used interchangeably with others such as keypoints, interest points, or even corner points.

From Moravec's solution to what we can regard as today's state-of-the-art detectors, there are considerably relevant differences along with a strong correlation among them concerning the basic steps of detection. These differences started being partially delineated by the seminal studies on scale-space theory (Witkin, 1983; Lindeberg, 1994), which had a major impact on local feature detection. They were the inspiration to develop scale covariant methods and to further study scale-space or affine-space representations (e. g., Baumberg, 2000). The scale-space representation, i. e., the representation of an image as a collection of (Gaussian) smoothed images parametrized by the size of the kernel, allowed detectors to define a size (scale) and a shape for features, which ensured scale covariance. Current state-of-the-art algorithms are even more effective. They are able to provide a covariant detection when in the presence of large image deformations, such as viewpoint changes.

We provide a review of local feature detection. Our intent is to give to the reader an explanatory overview on the topic by describing and classifying the fundamental algorithms for feature detection.

We adopt the more generic term keypoint instead of corner point. Although many algorithms were introduced as corner detectors, these detectors also respond to other conspicuous locations, namely isolated pixels of extremum intensity values and pixels in highly textured areas.

We consider three broad categories of local feature detectors using the type of detected feature as the criterion: (i) keyoint detectors, (ii) keypoint-based region detectors, and (iii) region detectors. We overview keypoint detectors and keypoint-based region detectors in the same subsection since the latter is an extension of the former that usually takes into account the scale of the keypoint to delineate a region.

2.2.1 *Keypoint detectors and keypoint-based region detectors*

A keypoint is a well-defined representative of a locally salient image part. Hence, a keypoint can be described as the most salient location of a locally salient region. Given its conspicuity, a keypoint is a local feature itself.

Despite the variety of solutions, a common framework for keypoint extraction can be encountered. In a generic and simplistic manner, we can view the kernel of a keypoint detector as a function (operator) that takes image locations as arguments and maps them into something that measures the saliency of the given locations taking into consideration their surroundings. From this measure, keypoints are commonly retrieved by a local maxima search and by defining a threshold for the minimum saliency measure value. The latter operation is aimed at preserving "the most interesting" keypoints, which are also expected to be the most robust in the presence of deformations. Algorithm 1 outlines the main steps of the majority of keypoint detectors. We can argue that the main differences between two arbitrary keypoint extractors lie on the way we represent the surroundings of a given location, i. e., how we describe the surroundings and how we measure the saliency of patterns from such representation.

Algorithm 1 Keypoint detector (generic framework)

```
for each pixel location $\mathbf{x}$ do
    Compute the local saliency measure $f$ at $\mathbf{x}$.
end for
Select keypoints (locations at which $f$ attains a local maximum).
Select keypoints $\mathbf{x}^\star$ with $f(\mathbf{x}^\star) \geqslant T$ ($T$ is the threshold).
```

2.2.1.1 *Moravec keypoint detector*

In Moravec's seminal algorithm (Moravec, 1977), the local saliency measure corresponds to the minimum intensity variation value computed over unitary shifts of a local window of size $k \times l$ centered at a given pixel location $\mathbf{x} = [x\ y]^T$, performed across the four principal directions. The sum of squared differences is used to determine the variation:

$$\begin{aligned}
\nu_V &= \frac{1}{p(q-1)} \sum_{i=-k}^{k} \sum_{j=-l}^{l-1} (I([x+i\ y+j]^T) - I([x+i\ y+j+1]^T))^2 \\
\nu_H &= \frac{1}{(p-1)q} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l} (I([x+i\ y+j]^T) - I([x+i+1\ y+j]^T))^2 \\
\nu_{D_1} &= \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (I([x+i\ y+j]^T) - I([x+i+1\ y+j+1]^T))^2 \\
\nu_{D_2} &= \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (I([x+i+1\ y+j]^T) - I([x+i\ y+j+1]^T))^2,
\end{aligned} \tag{2.1}$$

with $p = 2k+1$ and $q = 2l+1$. The local saliency measure coincides with the minimum variation:

$$\nu_{min} = \min\{\nu_V, \nu_H, \nu_{D_1}, \nu_{D_2}\}. \tag{2.2}$$

Since keypoints are the pixel positions where V_{min} attains a local maximum, these locations correspond to corners or isolated pixels. This detector provides an anisotropic response, which is result of using a fixed number of directions. By considering the minimum intensity variation, this detector becomes particularly sensitive to noise.

2.2.1.2 *Beaudet keypoint detector*

Beaudet (1978) proposes a second order differential operator for keypoint extraction. The basis of Beaudet's operator is the Hessian matrix of the image, i.e., a symmetric matrix of second-order derivatives of a given image I:

$$\mathcal{H}(\mathbf{x}) = \begin{bmatrix} I_{xx}(\mathbf{x}) & I_{xy}(\mathbf{x}) \\ I_{yx}(\mathbf{x}) & I_{yy}(\mathbf{x}) \end{bmatrix}. \tag{2.3}$$

The saliency measure is the determinant of $\mathcal{H}$,

$$\det(\mathcal{H}) = I_{xx}I_{yy} - I_{xy}^2, \tag{2.4}$$

which has a local maxima near corner points. This operator is quite sensitive to image noise, as it includes the computation of second order derivatives.

This operator is also used in more complex detectors, such as the Hessian-Laplace (Mikolajczyk & Schmid, 2001), and in the KAZE descriptor (Alcantarilla et al., 2012).

2.2.1.3 *Harris-Stephens keypoint detector*

Förstner (1986) and Harris & Stephens (1988) were the first authors to propose operators based on the structure tensor matrix. The Harris-Stephens detector was presented as an improvement over Moravec's algorithm. Instead of using a fixed number of shifted patches, it uses the structure tensor matrix to detect responses at any shift, which allows corners to be more accurately detected. The structure tensor matrix of an image I at $\mathbf{x}$ is given by

$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = G(\sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_xL_y(\mathbf{x}, \sigma_D) \\ L_yL_x(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}, \tag{2.5}$$

where the parameters $\sigma_I > 0$ and $\sigma_D > 0$ denote the differentiation and derivation scales, respectively. μ represents the averaged outer product of the image gradients and its spectral structure captures the local signal changes along the principal directions: the order of magnitude of the eigenvalues is proportional to the gradient variation along the principal directions. By analyzing the spectrum of μ, three cases can be considered:

Proposition 2.1 (Harris & Stephens, 1988). *Let λ_1 and λ_2 be the eigenvalues of the structure tensor matrix μ at $\mathbf{x}$, with $\lambda_1 \leqslant \lambda_2$ ($\lambda_1, \lambda_2 \geqslant 0$, as μ is positive semi-definite).*

1. *if $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, then $\mathbf{x}$ lies on a region of approximately constant intensity.*

2. *if* $\lambda_2 \gg \lambda_1$ *and* $\lambda_1 \approx 0$*, then* $\mathbf{x}$ *lies on an edge.*

3. *if* λ_1 *and* λ_2 *are both large values, then* $\mathbf{x}$ *lies on a corner.*

The inferences made in Proposition 2.1 are the support for the Harris-Stephens saliency measure, which is given by

$$f_{HS}(\mathbf{x}) = \prod_{i=1}^{n} \lambda_i - \kappa(\sum_{i=1}^{2} \lambda_i)^2 = \det(\mu) - \kappa \operatorname{trace}^2(\mu), \tag{2.6}$$

where $\kappa > 0$ is an adjustable parameter for tuning the sensitivity of the detector. Keypoints are found at locations at which f_{HS} attains a local maximum. Note that the Harris-Stephens algorithm does not have to compute the spectral decomposition of μ; it suffices to evaluate the determinant and the trace of μ.

The features shown in Fig. 2.1 (d) correspond to Harris-Stephens keypoints, which are mainly corner points.

2.2.1.4 *Förstner keypoint detector*

Förstner (1986) proposes a keypoint detector based on the structure tensor matrix, which is the kernel of a feature-based matching algorithm. The purpose of the detection stage is to select points with a "promising matching accuracy". The local saliency measure is solely based on trace of the inverse of μ, the image structure tensor matrix:

$$f_F(\mathbf{x}) = \frac{1}{\operatorname{trace}(\mu^{-1}) + \epsilon} = \frac{1}{\sum_{i=1}^{2} \frac{1}{\lambda_i} + \epsilon}, \tag{2.7}$$

where ϵ is an arbitrary small positive constant.

2.2.1.5 *Noble keypoint detector*

Noble (1989) suggests a modified version of the Harris-Stephens detector that does not contain the original tuning parameter, overcoming the need of manually tuning it:

$$f_N(\mathbf{x}) = \frac{\det(\mu)}{\epsilon + \operatorname{trace}(\mu)} = \frac{\prod_{i=1}^{2} \lambda_i}{\epsilon + \sum_{i=1}^{2} \lambda_i}, \tag{2.8}$$

where ϵ denotes an arbitrary small positive constant.

2.2.1.6 *Shi-Tomasi keypoint detector*

The Shi-Tomasi algorithm (Shi & Tomasi, 1994) is another algorithm based on the structure tensor matrix. It detects keypoints via Eq. (2.9). By imposing $f_{ST}(\mathbf{x}) \geqslant \lambda$, where $\lambda > 0$ is a sufficiently large threshold, we are selecting points whose corresponding structure tensor matrix exhibits eigenvalues that do not differ by several orders of magnitude, which in terms of visual patterns, corresponds to selecting those that show high intensity variations in several directions:

$$f_{ST}(\mathbf{x}) = \min\{\lambda_1, \lambda_2\} = \lambda_1. \tag{2.9}$$

2.2.1.7 *Rohr keypoint detector*

By neglecting the trace of the structure tensor matrix, Rohr (1997) proposes an alternative operator relying exclusively on the determinant of this matrix:

$$f_R(\mathbf{x}) = \det(\mu) = \prod_{i=1}^{2} \lambda_i. \tag{2.10}$$

The aforementioned measure implies the exclusion of points that convey a less discriminant information: those exhibiting a large difference between the eigenvalues. The deletion of those points is the result of applying the following threshold:

$$\frac{\det(\mu)}{(\frac{1}{2}\,\mathrm{trace}(\mu))^2} = \frac{\prod_{i=1}^{2} \lambda_i}{(\frac{1}{2}\sum_{i=1}^{2} \lambda_i)^2}. \tag{2.11}$$

2.2.1.8 *Kenney et al. keypoint detector*

The Kenney et al. detector (Kenney et al., 2003) for the Schatten p-norm, with $p \in [1, \infty)$, identifies keypoints using the following function:

$$f_{K,p}(\mathbf{x}) = \frac{1}{\|\mu^{-1}\|_p} = \frac{1}{\sqrt[p]{\sum_{i=1}^{2} \frac{1}{\lambda_i^p}}}. \tag{2.12}$$

The latter detector arises as an explicit attempt to select locations that convey a better repeatability in the presence of image rotations and translations. The method selects points that have a small condition number with respect to translations, and consequently, to rotations (Zuliani et al., 2004).

It is worth mentioning that f_F, $\sqrt{f_R}$, f_{ST}, and f_k are equivalent modulo the choice of a convenient matrix norm (Kenney et al., 2005):

Lemma 2.1 (Kenney et al., 2005)**.** *The Förstner, Shi-Tomasi, and Kenney et al. operators are equivalent modulo the choice of a suitable matrix norm.* $\sqrt{f_R}$ *(modified Rohr operator) is equivalent to Kenney et al. detector in a limit sense (via a normalization constant).*

$$\begin{aligned} f_F &= f_{K,1}(\textit{with } \epsilon = 0); \\ f_{ST} &= \lim_{p\to\infty} f_{K,p}; \\ \sqrt{f_R} &= \lim_{p\to 0} \frac{1}{\sqrt[p]{2}} f_{K,p}. \end{aligned}$$

Figure 2.2 depicts keypoints extracted by the aforementioned algorithms. The operators produce very similar results.

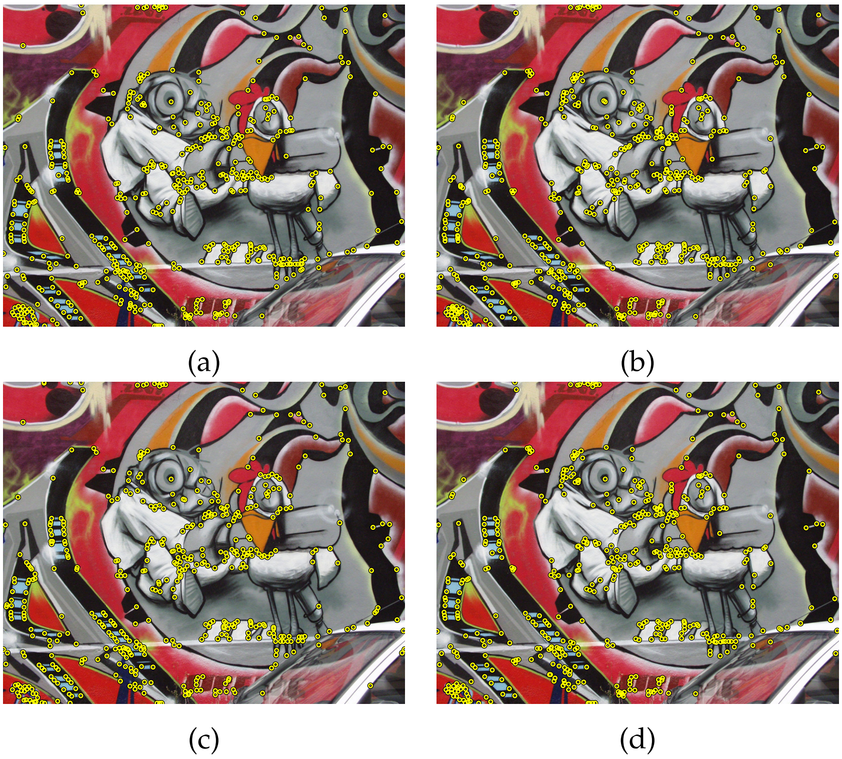

(a)
(b)
(c)
(d)

Figure 2.2: Keypoint extraction: (a) Förstner; (b) Shi-Tomasi; (c) Rhor; (d) Kenney (p=3).

2.2.1.9 *Triggs keypoint detector*

A generalization of the Harris-Stephens detector is the one suggested by Triggs (2004). It employs a multi-scale detection which responds to maximally stable locations with respect to affine deformations and slight illumination changes. This detector is based on the operator

$$f_T = \lambda_1 - \alpha\lambda_2, \tag{2.13}$$

where the parameter α is usually set to 0.06.

2.2.1.10 *SFOP detector*

The Scale Invariant Feature Operator (SFOP) (Förstner et al., 2009a) responds to corners, junctions, and circular features in a scale covariant manner. The explicitly interpretable and complementary detection results from a unified framework that extends a gradient-based detection (Förstner, 1994; Parida et al., 1998) to a scale-space representation. In this algorithm, the general spiral feature model (Bigün,

1990) allows the fusion of the different detectors into one.

An image patch around a given point $\mathbf{x}$ has a spiral structure when the edge direction at a neighboring point $\mathbf{y}$ has a constant angle with the radius vector. The algorithm measures the consistency of the neighborhood of a point $\mathbf{x}$ with the feature model by computing the distances d_n to the edge line through a point $\mathbf{y}_n$ having angle α with respect to the gradient direction at $\mathbf{y}_n$. Such distance is given by

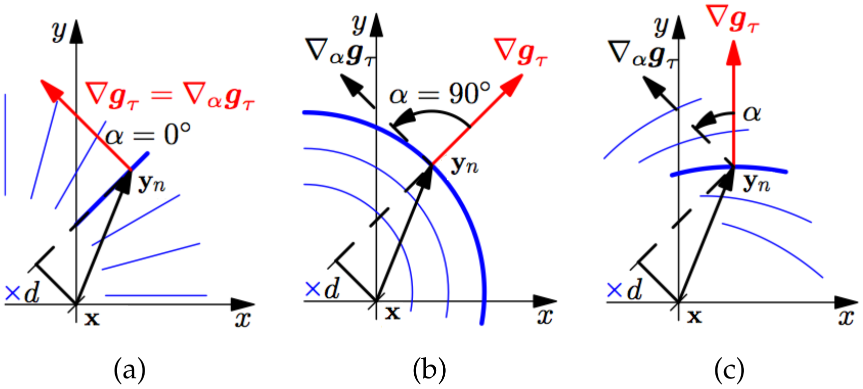


Figure 2.3: Measuring distances in a spiral feature. d corresponds to the distance from $\mathbf{x}$ to an edge line. The constant angle α is measured between the tangential and radial directions. (a) Junction ($\alpha = 0°$); (b) circle ($\alpha = 90°$); (c) logarithmic spiral feature (arbitrary α). Adapted from Förstner et al. (2009a).

$$d_n(\mathbf{x}, \mathbf{y}_n, \alpha, \sigma_D) = \frac{(\mathbf{y}_n - \mathbf{x})^T R_\alpha \nabla_{\sigma_D} L(\mathbf{y}_n)}{|\nabla_{\sigma_D} L(\mathbf{y}_n)|}, \qquad (2.14)$$

where $\nabla_{\sigma_D} L(\mathbf{y}_n)$ is the gradient of a Gaussian smoothed version of the image using σ_D as the differentiation scale and R_α is a rotation matrix of angle α. The SFOP algorithm searches for different spiral structures whose location is determined with highest precision. The basis for detection is the following structure tensor matrix:

$$\mu(\mathbf{x}, \alpha, \sigma_I, \sigma_D) = G(\sigma_I) * (R_\alpha \nabla_{\sigma_D} L(\mathbf{x}) \nabla_{\sigma_D} L(\mathbf{x})^T R_\alpha^T). \qquad (2.15)$$

The smallest eigenvalue of μ is used to define the precision

$$w(\mathbf{x}, \alpha, \sigma_I, \sigma_D) = (N(\sigma_D) - 2) \frac{\lambda_1(\mu(\mathbf{x}, \alpha, \sigma_I, \sigma_D))}{\Omega(\mathbf{x}, \alpha, \sigma_I, \sigma_D)}, \qquad (2.16)$$

where $N(\sigma_D)$ represents the number of pixels in the neighborhood of $\mathbf{x}$ defined by the Gaussian $G(\sigma_D)$ and Ω is the negative log-likelihood function to be minimized:

$$\Omega(\mathbf{x}, \alpha, \sigma_I, \sigma_D) = N(\sigma) \operatorname{trace}(R_\alpha \nabla_{\sigma_I} \nabla_{\sigma_I}^T R_\alpha^T * \mathbf{x}\mathbf{x}^T G(\sigma_D)). \qquad (2.17)$$

To avoid keypoints caused by noise, a threshold is defined for λ_1:

$$T_\lambda(s^2, \sigma_D, \sigma_I, S) = \frac{N(\sigma_D)}{16\pi\sigma_I^4} s^2 \chi^2_{2,S}, \tag{2.18}$$

where s^2 represents the noise variance and S the significance level.

The main steps of the algorithm are outlined in Algorithm 2.

Algorithm 2 SFOP detector.

```
for each integration scale $\sigma_I$ do
  for each pixel location $\mathbf{x}$ do
    Compute gradient $\nabla_{\frac{\sigma_I}{3}} L(\mathbf{x})$.
    Compute $\lambda_1$, the smallest eigenvalue of the following structure tensor matrix
        $\mu(\mathbf{x}, \alpha, \sigma_I, \sigma_D) = G(\sigma_I) * (R_\alpha \nabla_{\sigma_D} L(\mathbf{x}) \nabla_{\sigma_D} L(\mathbf{x})^T R_\alpha^T)$.
    for each angle $\alpha \in \{0°, 30°, 60°\}$ do
      Compute $\Omega(\mathbf{x}, \alpha, \sigma_I)$.
    end for
    Determine $\alpha_0 = \underset{\alpha \in \{0°,30°,60°\}}{\operatorname{argmin}} \Omega(\mathbf{x}, \alpha, \sigma_I)$.
    Compute precision $w(\mathbf{x}, \alpha, \sigma_I)$.
  end for
end for
Detect local maxima in a 26-neighborhood of $w$.
Select keypoints $\mathbf{x}$ with $\lambda_1 > T_\lambda$.
Perform non-maxima suppression.
Interpolate $w$.
```

SFOP features are displayed in Fig. 2.4. Sets of SFOP features usually exhibit a low density, yet they tend to provide a good coverage of the most informative content. Besides retrieving complementary and interpretable features, SFOP detector has also an accurate response.

2.2.1.11 *SUSAN*

SUSAN (Smith, 1992, 1996; Smith & Brady, 1997), which stands for Smallest Univalue Segment Assimilating Nucleus, is a morphological operator suggested for edge detection as well as corner detection.

For each pixel $\mathbf{x}$ in the image, a circular mask M – whose nucleus is $\mathbf{x}$ – is computed. Then, for every pixel $\mathbf{y} \in M \setminus \{\mathbf{x}\}$, its intensity value is compared to the one of $\mathbf{x}$ using the following function:

$$c(\mathbf{y}, \mathbf{x}) = \exp(-(\frac{I(\mathbf{x}) - I(\mathbf{y})}{t})^6), \tag{2.19}$$

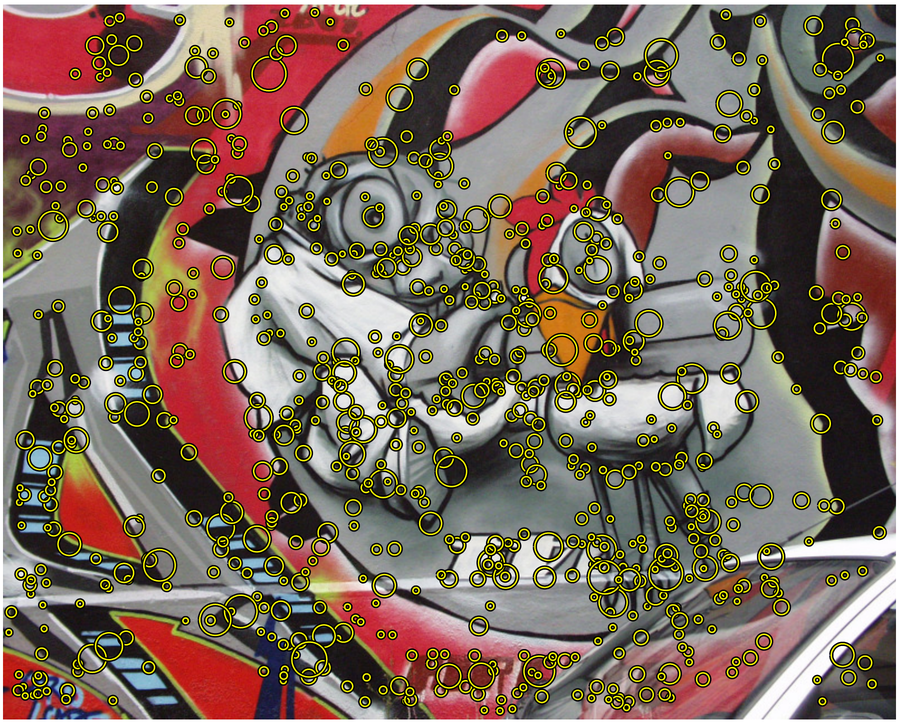


Figure 2.4: SFOP features.

where t determines the radius. The intensity comparison allows us to obtain the Univalue Segment Assimilating Nucleus (USAN) area for $\mathbf{x}$, which is given by

$$n(\mathbf{x}) = \sum_{\mathbf{y} \in M} c(\mathbf{y}, \mathbf{x}). \tag{2.20}$$

The USAN area succinctly describes the structure in the neighborhood of $n(\mathbf{x})$: n is maximum when $\mathbf{x}$ lies in a flat region; in case of edges, the area is half of its maximum; for corners, $n(\mathbf{x})$ is even lower. The corner measure used by the SUSAN algorithm is based on the previous inferences:

$$c(\mathbf{x}) = \begin{cases} \frac{n_{max}}{2} & \text{if } n(\mathbf{x}) < \frac{n_{max}}{2} \\ 0 & \text{otherwise} \end{cases}, \tag{2.21}$$

where n_{max} is the maximum area of the USAN.

Figure 2.5 depicts examples of SUSAN feature points.

2.2.1.12 *FAST*

FAST, which stands for Features from Accelerated Segment Test (Rosten & Drummond, 2006), is an algorithm based on the SUSAN criterion which uses machine learning to provide an extremely efficient

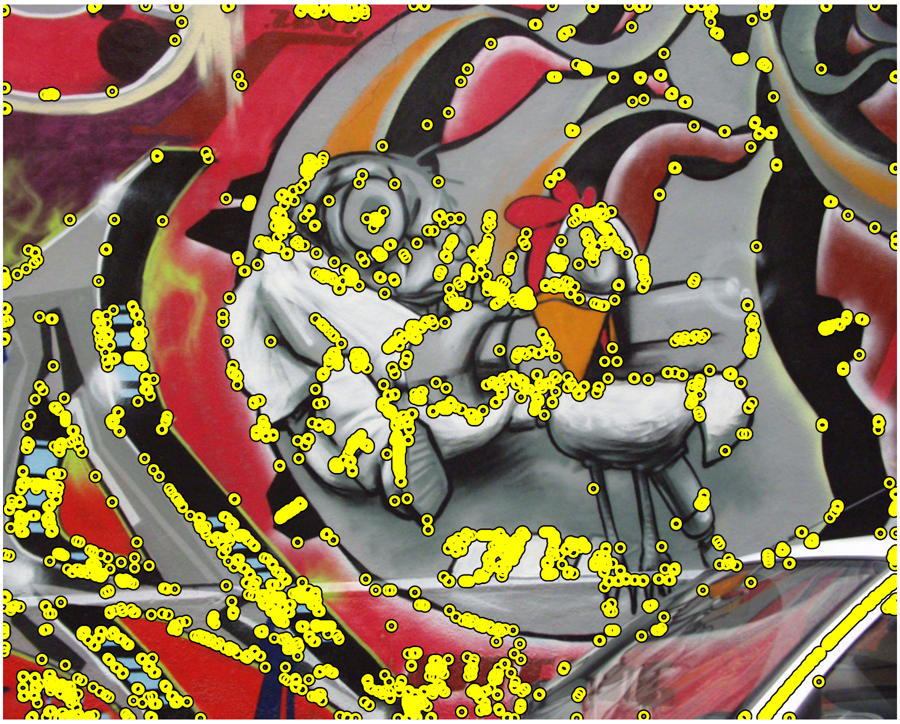


Figure 2.5: SUSAN keypoints.

feature extraction. Pixels are compared on a Bresenham circle of 16 pixels around the keypoint/corner candidate. The idea is to classify groups of adjacent pixel into three categories: brighter, darker, and similar. A given pixel is a corner if there are 12 adjacent pixels that are either brighter or darker than the center. The ID3 (Iterative Dichotomizer 3) algorithm (Quinlan, 1986) is utilized to build a decision tree with the goal of selecting the pixel which yields the most information about whether the candidate pixel is a keypoint/corner, measured by the entropy of the corner classification responses. The resulting decision tree is converted into a long sequence of nested conditional statements written in C language. This source code corresponds to the final detector.

Examples of keypoints extracted by the FAST algorithm are displayed in Fig. 2.6.

FAST-ER (Features from Accelerated Segment Test - Enhanced Repeatability) (Rosten et al., 2010) is an improved version of FAST which takes into account the repeatability of features in order to retrieve points with a high repeatability rate.

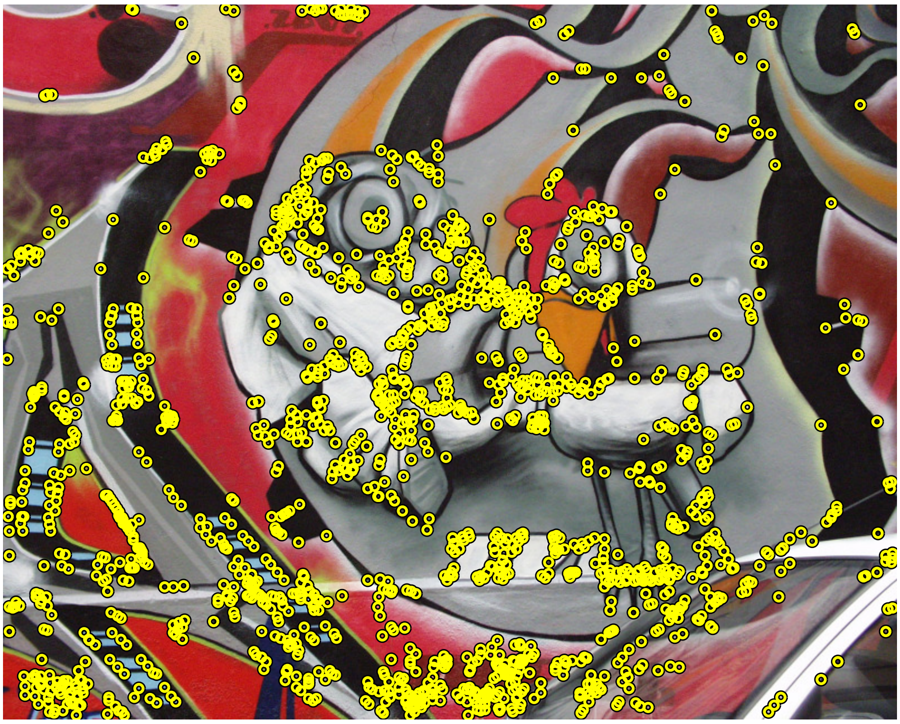


Figure 2.6: FAST keypoints.

2.2.1.13 *Laplacian of Gaussian (LoG) detector*

Keypoints are representatives of visually salient image parts. In some cases, these conspicuous regions around keypoints correspond to blobs. As mentioned earlier, a blob is an image part that is brighter or darker than the surroundings. Blob detection is usually performed in an image scale-space representation in order to determine its scale. Lindeberg (1998) proposes a scale covariant blob detector which is the result of searching for scale-space extrema of (scale) normalized Laplacian of Gaussian (LoG):

$$\sigma^2 \nabla^2 L(\mathbf{x}, \sigma) = \sigma^2 (L_{xx}(\mathbf{x}, \sigma) + L_{yy}(\mathbf{x}, \sigma)). \tag{2.22}$$

This operator has a maximal response at the center of circular blob structures (see Fig. 2.7).

2.2.1.14 *Difference of Gaussians (DoG) detector*

The Difference of Gaussians (DoG) operator is an approximation of the Laplacian operator. In a scale-space, the difference between images at different scales is an approximation of the derivative with respect to scale and the Laplacian corresponds to the image derivative in the scale direction. Therefore, the Laplacian of the Gaussian operator can be approximated by the difference between two Gaus-

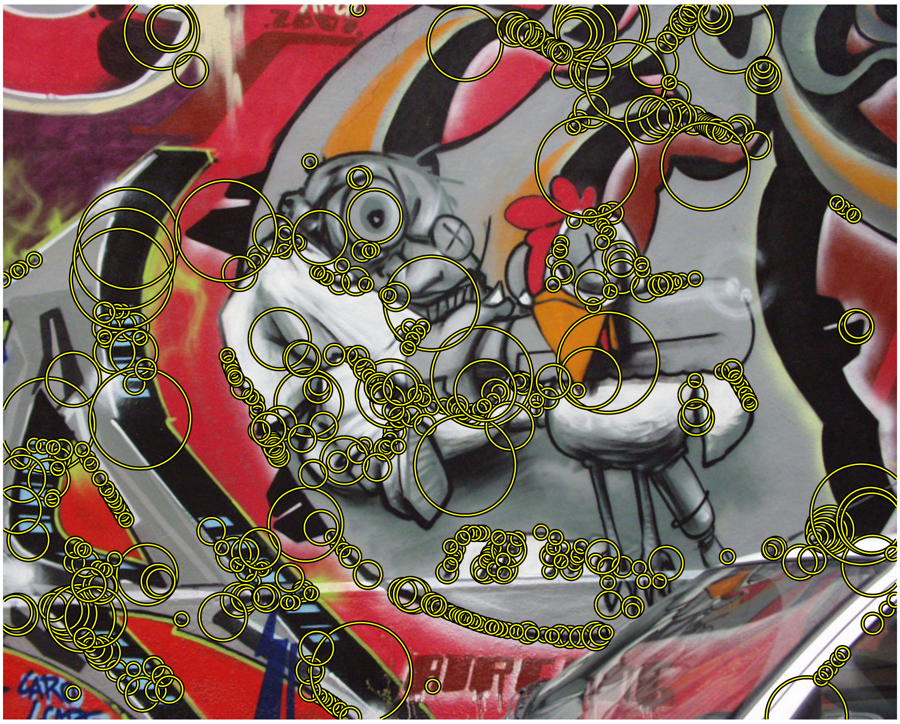


Figure 2.7: Feature extraction using the Laplacian of Gaussian.

sian smoothed images whose scales are separated by a factor of k (Grauman & Leibe, 2011) :

$$D(\mathbf{x}, \sigma) = (G(k\sigma) - G(\sigma)) * I(\mathbf{x}). \tag{2.23}$$

The DoG operator is the basis of the popular Scale-Invariant Feature Transform (SIFT) descriptor (Lowe, 1999, 2004). To construct the descriptors, keypoints are firstly detected in a scale-space. A keypoint is a location at which the DoG attains a local extremum. To characterize the neighborhood of each one of the keypoints, a descriptor is constructed. It consists of 16 gradient orientation histograms with 8 bins each, producing a vector with 128 elements. This descriptor is rotation and scale invariant.

2.2.1.15 *Harris-Laplace*

The Harris-Laplace (Mikolajczyk & Schmid, 2001, 2002, 2004) is a scale covariant detector that results from the combination of the popular Harris-Stephens keypoint detector (Harris & Stephens, 1988) with a Gaussian scale-space representation. It starts with a multi-scale Harris-Stephens keypoint extraction followed by an automatic scale selection (Lindeberg, 1998) defined by a normalized Laplacian operator. In this case, the characteristic scale for a given structure corresponds to the scale where the Laplacian attains a maximum, which is independent of the image resolution, yielding, thereby, a scale covariant response. The algorithm starts by building a scale-space representation for the Harris-Stephens measure using n pre-selected scales $\sigma_i = \xi^{i-1}\sigma_0$, with $\sigma_0 \in \mathbb{R}^+$, $\xi > 1$, and $i = 1, \ldots, n$. At each level (scale) σ_i, keypoints are found by computing the local maxima that are above a given positive threshold T_{HS}:

$$\begin{cases} \mathbf{x}^\star = \underset{\mathbf{x}}{\operatorname{argmaxlocal}}\, f_{HS}(\mathbf{x}, \sigma_I) \\ f_{HS}(\mathbf{x}^\star, \sigma_I) \geqslant T_{HS} \end{cases} . \tag{2.24}$$

The next step in the algorithm is to determine the scale of the keypoints, which is done by finding a local normalized Laplacian (of Gaussian) extrema in a range of scales above a given positive threshold T_{LoG}:

$$\begin{cases} \sigma^\star = \underset{\sigma}{\operatorname{argmaxlocal}}\, \sigma^2(L_{xx}(\mathbf{x}^\star, \sigma) + L_{yy}(\mathbf{x}^\star, \sigma)) \\ {\sigma^\star}^2(L_{xx}(\mathbf{x}^\star, \sigma^\star) + L_{yy}(\mathbf{x}^\star, \sigma^\star)) \geqslant T_{LoG} \end{cases} . \tag{2.25}$$

Algorithm 3 outlines the main steps for the detection of Harris-Laplace regions.

Algorithm 3 Harris-Laplace (HARLAP)

```
for each integration scale $\sigma_I$ do
    for each pixel location $\mathbf{x}$ do
        Compute Harris-Stephens response ($f_{HS}$) at $\mathbf{x}$ with the structure tensor matrix $\mu(\mathbf{x}, \sigma_I, 0.7 \times \sigma_I)$.
    end for
    Detect local maxima in a 8-neighborhood of $f_{HS}$.
    Select keypoints $\mathbf{x}^\star$ with $f_{HS} \geqslant T_{HS}$.
end for
for each integration scale $\sigma_I$ do
    for each keypoint $\mathbf{x}^\star$ detected at scale $\sigma_I$ do
        Let $\mathbf{x}^{(0)} = \mathbf{x}^\star$ and $\sigma^{(0)} = \sigma_I$.
        repeat
            Find the local extremum over scale of the LoG for $\mathbf{x}^\star$ in the range $\sigma^{(k+1)} = t\sigma^{(k)}$, with $t \in [0.7, 1.4]$. Reject $\mathbf{x}^\star$ if the LoG response attains no extremum or if the response is below the threshold $T_{LoG}$.
            At scale $\sigma^{(k+1)}$, detect the location $\mathbf{x}^{(k+1)}$ nearest to $\mathbf{x}^{(k)}$ for which the Harris-Stephens response is a maximum.
        until $\sigma^{(k+1)} = \sigma^{(k)}$ and $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$
    end for
end for
```

An example of Harris-Laplace regions is depicted in Fig. 2.8. Harris-Laplace regions are usually blob-like features due to the use of the Laplacian as a saliency measure. Sets of Harris-Laplace regions usually show a moderate density accompanied with a good coverage of informative image content.

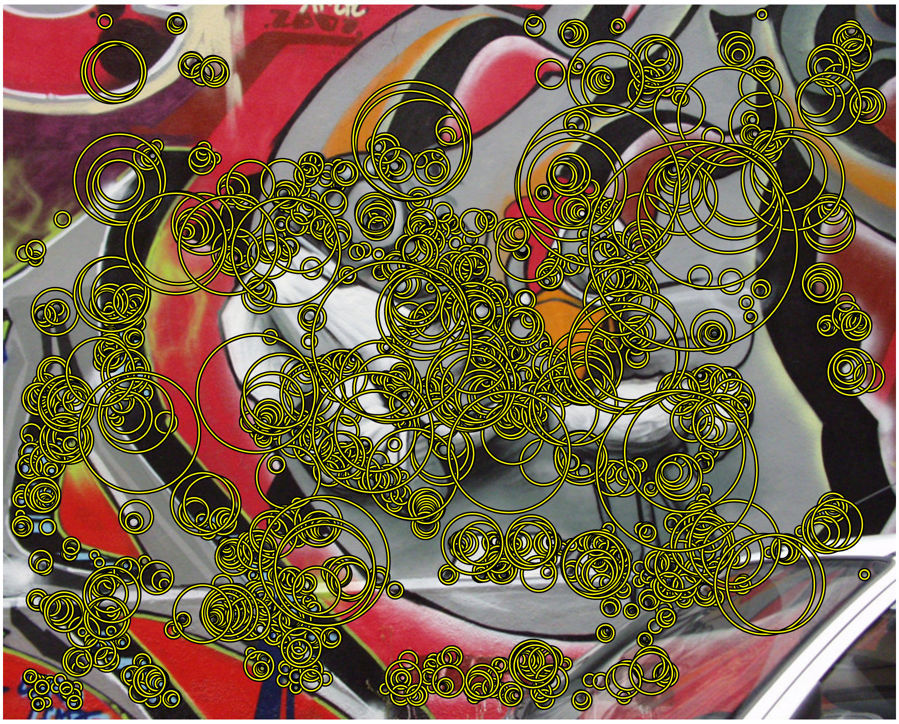


Figure 2.8: Harris-Laplace regions.

2.2.1.16 *Hessian-Laplace*

As the name suggests, the Hessian-Laplace (Mikolajczyk & Schmid, 2001, 2004) is a scale covariant detector that shares a common framework with the Harris-Laplace detector. Instead of using the Harris-Stephens corner measure, it uses the determinant of the Hessian matrix. Figure 2.9 depicts Hessian-Laplace regions, which are mainly blobs.

2.2.1.17 *Harris-Affine*

The Harris-Affine scheme (Mikolajczyk & Schmid, 2002, 2004) is an extension of the Harris-Laplace, which relies on the combination of the Harris-Laplace operator with an affine shape adaptation stage (Lindeberg & Gårding, 1997; Baumberg, 2000). From initial estimates of keypoints detected at their characteristic scales, a convergence to affine covariant keypoints is performed by using the iterative estimation of elliptical affine regions whose shape is determined by the structure

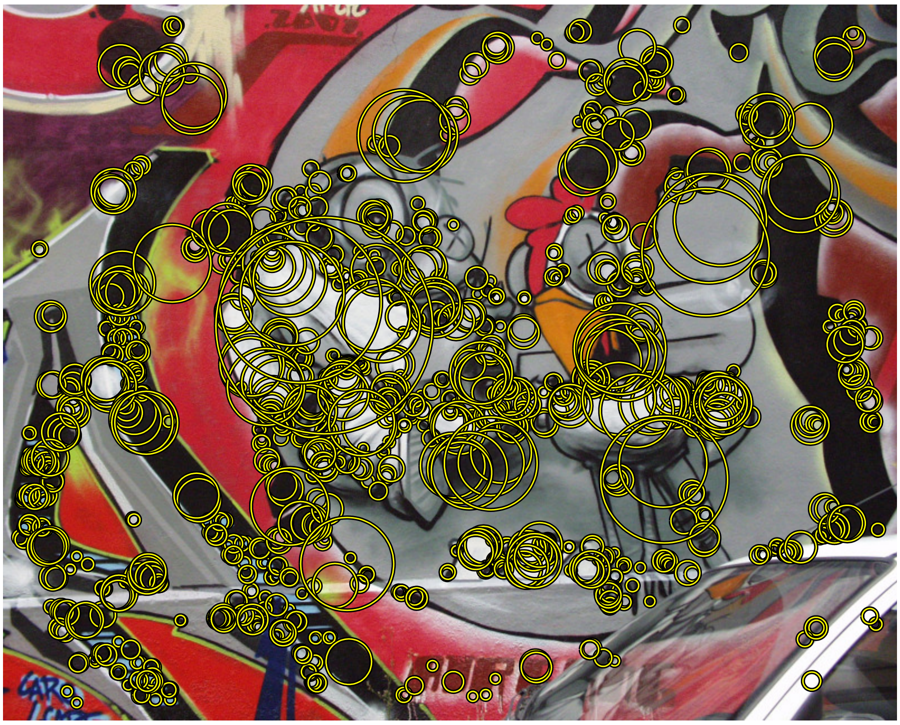


Figure 2.9: Hessian-Laplace regions.

tensor matrix of the image. In the affine scale-space, the structure tensor matrix is given by

$$\mu(\mathbf{x},\Sigma_I,\Sigma_D) = \det(\Sigma_D)G(\Sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x},\Sigma_D) & L_xL_y(\mathbf{x},\Sigma_D) \\ L_yL_x(\mathbf{x},\Sigma_D) & L_y^2(\mathbf{x},\Sigma_D) \end{bmatrix}, \tag{2.26}$$

where Σ_I and Σ_D are covariance matrices responsible for determining the integration and differentiation kernels, respectively. The eigenvalues of μ are used to measure the affine shape of a region around a keypoint. The shape is determined by finding the transformation that projects the affine pattern to one with equal eigenvalues. The transformation corresponds to the square root of the affine scale-space structure tensor matrix. If the neighborhood of two points $\mathbf{x}_L$ and $\mathbf{x}_R$ are normalized by transformations $\mathbf{x}_L^{'} = \mu_L^{\frac{1}{2}}\mathbf{x}_L$ and $\mathbf{x}_R^{'} = \mu_R^{\frac{1}{2}}\mathbf{x}_R$, respectively, then the normalized regions are related by a rotation: $\mathbf{x}_L^{'} = R\mathbf{x}_R^{'}$. The adaption stage is outlined in Algorithm 4.

Examples of Harris-Affine regions are given in Fig.2.10. The number of extracted regions is slightly lower than the one of Harris-Laplace (see Figs. 2.8 and 2.10 for comparison), as initial regions retrieved by the latter may not converge to affine covariant ones.

Algorithm 4 Harris-Affine: Affine shape adaptation

```
1: Perform a scale covariant detection with the Harris-Laplace detector.
2: repeat
3:     Estimate the affine shape using the structure tensor matrix.
4:     Normalize the affine region.
5:     Re-detect the new location and scale in the normalized patch.
6: until $\lambda_1 = \lambda_2$
```

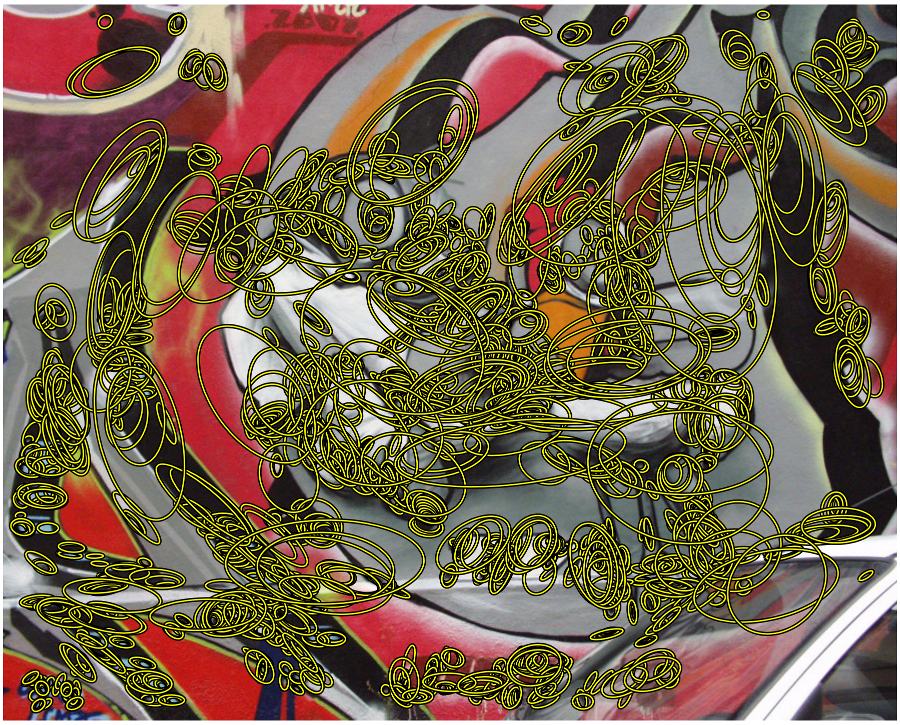


Figure 2.10: Harris-Affine regions.

2.2.1.18 *Hessian-Affine*

The Hessian-Affine (Mikolajczyk & Schmid, 2002, 2004) is the affine-covariant version of the Hessian-Laplace. It follows the affine shape adaptation scheme described in 2.2.1.17. Examples of Hessian-Affine regions are given in Fig. 2.11.

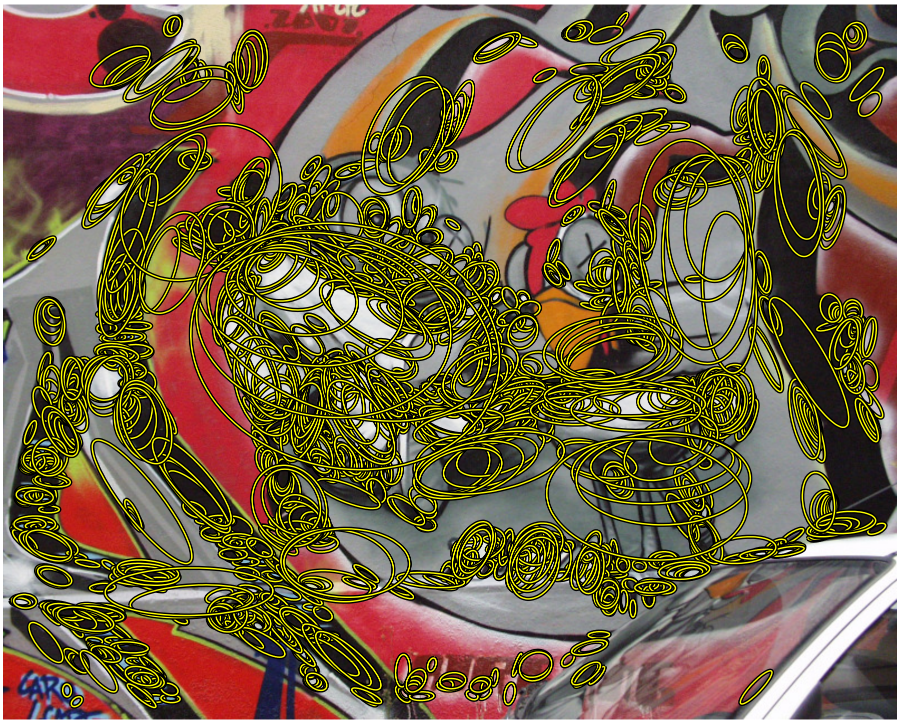


Figure 2.11: Hessian-Affine regions.

2.2.2 *Region detectors*

More recent detectors, especially affine covariant ones, extract regions without relying on a prior keypoint extraction. These regions usually have a irregular shape, which are often replaced by a fitted ellipse.

2.2.2.1 *Salient Regions*

The Salient Regions detector (Kadir & Brady, 2001; Kadir et al., 2004) can be seen as a refinement over the keypoint detector proposed by Gilles (1998). In both cases, saliency is defined in terms of local signal complexity, i. e., Shannon entropy (Shannon, 1948). While Gilles work is mainly an algorithm aimed at matching and registering aerial reconnaissance images by using locally salient patches derived from keypoints, the Salient Regions detector is introduced as a more generic tool and it also addresses the problem of scale selection. Salient Regions are not derived from keypoints, as salient points taken from the estimated entropy maps tend to be sensitive to noise and small deformations in the image.

There are two versions of the algorithm: the scale covariant version (Kadir & Brady, 2001) and the affine covariant one (Kadir et al., 2004). The latter is an extension of the former: it increases the search space required to extract Salient Regions.

Scale covariant salient regions detector For each pixel position, the salient region detector selects the scales at which the entropy of the local intensity histogram is peaked. Entropy is given by

$$H(\mathbf{x}, s) = -\sum_{i \in D} P(i, \mathbf{x}, s) \log(P(i, \mathbf{x}, s)), \tag{2.27}$$

where D is the set of intensity values and $P(i, \mathbf{x}, s)$ denotes the probability of intensity i at pixel $\mathbf{x}$ and scale s. Salient Regions also exhibit self-dissimilarity in the scale-space. The degree of self-dissimilarity is estimated by analyzing the change of the intensity's probability density function in a range of scales around the scale where entropy is at its peak. The measure of self-dissimilarity, which is given by

$$w(\mathbf{x}, s) = \frac{s^2}{2s-1} \left| \sum_{i \in D} P(i, \mathbf{x}, s) - \sum_{i \in D} P(i, \mathbf{x}, s-1) \right|, \tag{2.28}$$

is a weight for the entropy values. The saliency measure is the product of factors w and H:

$$y(\mathbf{x}, s) = w(\mathbf{x}, s) H(\mathbf{x}, s) \tag{2.29}$$

The whole detection process is summarized in Algorithm 5.

Algorithm 5 Salient Regions detector (scale covariant version)

```
for each pixel location $\mathbf{x}$ do
    for each scale $s \in \{s_0, s_2, \dots, s_n\}$ do
        Compute a local descriptor within a window of scale $s$.
        Estimate the local probability function using local intensity
        histograms.
    end for
    Select scales for which the entropy is a local maximum.
    Weight the entropy values at the select scales using $w$.
end for
```

Examples of Salient Regions are depicted in Fig. 2.12. Due to the explicit use of local entropy to define the saliency measure, Salient Regions provide an even coverage of informative parts of the image.

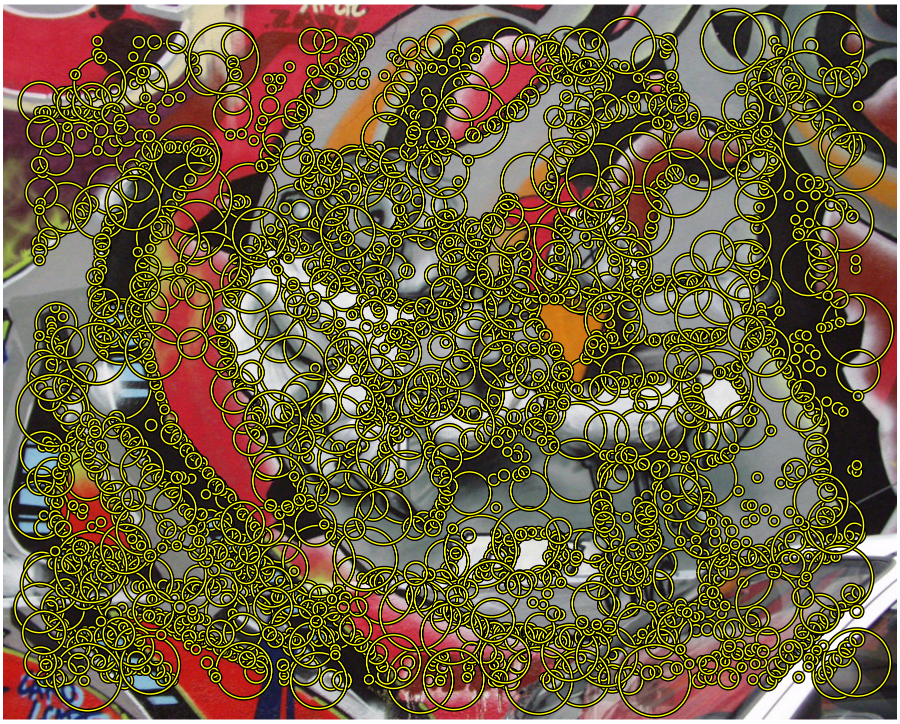


Figure 2.12: Salient Regions (scale covariant version).

AFFINE COVARIANT SALIENT REGIONS DETECTOR In the affine covariant version, a given region R is parameterized by three parameters: s, ρ, and θ, where ρ is the axis ratio and θ is the orientation, i. e., the circular window parametrized by scale s is replaced with an ellipse. The algorithm starts with scale covariant regions which are iteratively deformed into ellipses by searching for the optimal (s, ρ, θ) that maximizes the saliency measure.

2.2.2.2 *Edge-based Regions (EBR)*

The affine covariant Edge-based Regions (EBR) (Tuytelaars et al., 1999; Tuytelaars & Gool, 2004) are drawn from a geometry-based method, which uses a combined multi-scale detection of Harris-Stephens keypoints (Harris & Stephens, 1988) and nearby edges given by the Canny filter (Canny, 1986). The edge information is utilized to define the detected features, whose shape corresponds to a parallelogram.

The motivation for such strategy comes from two observations: (i) edges are stable features that can still be detected under the presence of significant geometric transformations or illumination changes; (ii) corner points are found at edges.

When two points, $\mathbf{x}_1$ and $\mathbf{x}_2$, move away from a corner point $\mathbf{x}$ in both directions along a edge (see Fig. 2.13), their relative speed is coupled through the equality of l_1 and l_2:

$$l_i = \int \left| \det \left[\begin{array}{cc} \mathbf{x}_i^{(1)}(s_i) & \mathbf{x} - \mathbf{x}_i(s_i) \end{array} \right] \right| d\, s_i, i = 1, 2, \tag{2.30}$$

where s_i is an arbitrary curve parameter and $\mathbf{x}_i^{(1)}(s_i)$ is the first order derivative of $\mathbf{x}_i(s_i)$ with respect to s_i.

Let $l = l_1 = l_2$. For each value l, $\mathbf{x}_1(l)$, $\mathbf{x}_2(l)$, and $\mathbf{x}$ define a parallelogram $\Omega(l)$: the one spanned by the vectors $\mathbf{x}_1(l) - \mathbf{x}$ and $\mathbf{x}_2(l) - \mathbf{x}$. The next step in the algorithm is to select a few parallelograms for which the following invariants, $i_1(\Omega)$ and $i_2(\Omega)$, are an extremum:

$$i_1(\Omega) = \left| \frac{\det \left[\begin{array}{cc} \mathbf{x}_1 - \mathbf{x}_g & \mathbf{x}_2 - \mathbf{x}_g \end{array} \right]}{\det \left[\begin{array}{cc} \mathbf{x} - \mathbf{x}_1 & \mathbf{x} - \mathbf{x}_2 \end{array} \right]} \right| \frac{M_{0,0}^1}{\sqrt{M_{0,0}^2 M_{0,0}^0 - (M_{0,0}^1)^2}} \tag{2.31}$$

$$i_2(\Omega) = \left| \frac{\det \left[\begin{array}{cc} \mathbf{x} - \mathbf{x}_g & \mathbf{y} - \mathbf{x}_g \end{array} \right]}{\det \left[\begin{array}{cc} \mathbf{x} - \mathbf{x}_1 & \mathbf{x} - \mathbf{x}_2 \end{array} \right]} \right| \frac{M_{0,0}^1}{\sqrt{M_{0,0}^2 M_{0,0}^0 - (M_{0,0}^1)^2}}, \tag{2.32}$$

where $M_{p,q}^n$ is the n-th order, $(p+q)$-th degree moment computed over $\Omega(l)$, i. e.,

$$M_{p,q}^n = \int_\Omega I^n(x, y) x^p y^q \, d\, x\, d\, y, \tag{2.33}$$

$\mathbf{y}$ is the vertex of the parallelogram opposite to the corner point $\mathbf{x}$ (see Fig. 2.13), and $\mathbf{x}_g$ is the center of gravity of $\Omega(l)$:

$$\mathbf{x}_g = \left(\frac{M_{1,0}^1}{M_{0,0}^1}, \frac{M_{0,1}^1}{M_{0,0}^1}\right). \tag{2.34}$$

EBR features are shown in Fig. 2.14. Usually, the EBR algorithm produces a dense set of features, with high redundancy and very few semantically interpretable features.

2.2.2.3 *Intensity Extrema-based Regions (IBR)*

Intensity Extrema-based Regions (IBR) (Tuytelaars & Gool, 2000, 2004) are affine covariant patches whose construction comprehends a multiscale detection of points at which intensity attains a local extremum and a subsequent definition of feature regions via the search of maxima of an intensity function f_I along the rays emanating from the previously detected locations. In the end, an ellipse is fitted to the

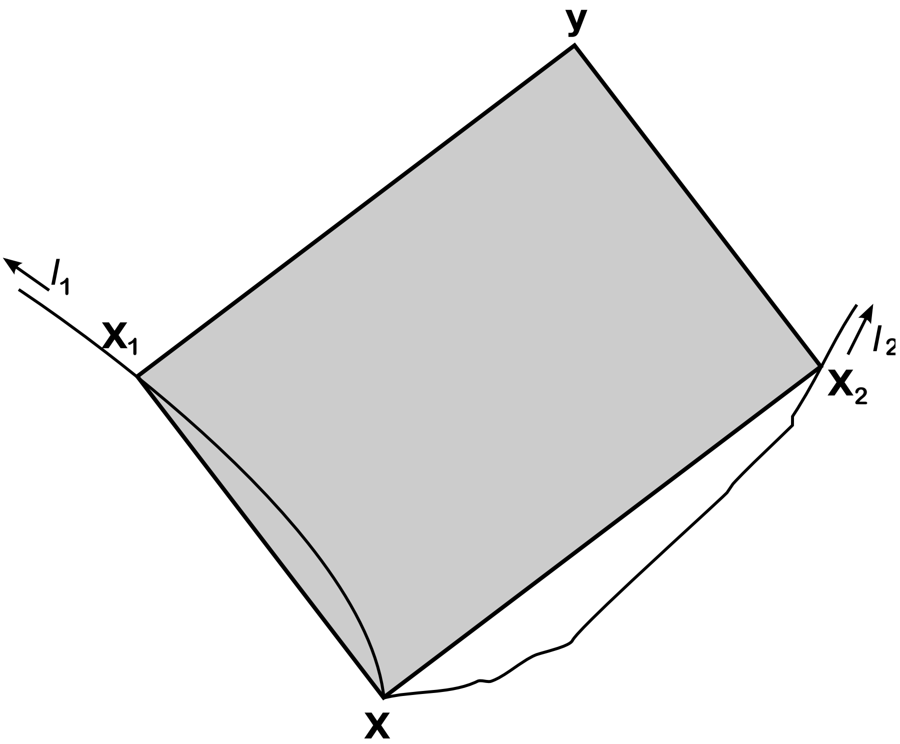


Figure 2.13: Construction of Edge-based Regions: Adapted from Tuytelaars & Gool (2004).

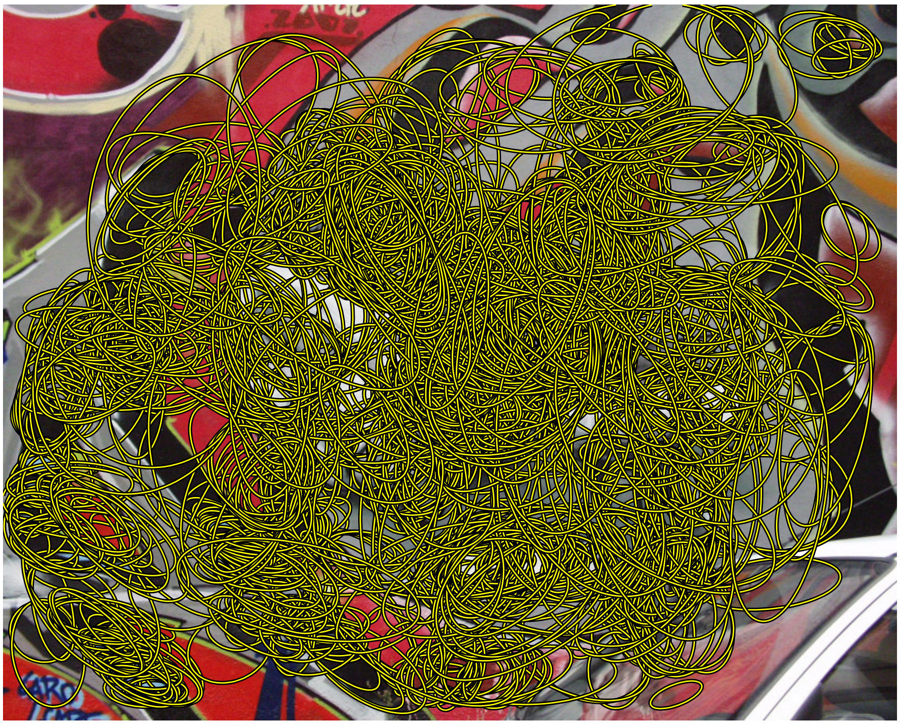


Figure 2.14: Affine covariant features extracted by the EBR algorithm.

delineated salient region.

The first step of the algorithm consists of a multi-scale detection of pixels for which the intensity is a local extremum (maximum or minimum). Given a local intensity extremum $I^{\star}$, the function f_I is evaluated along each ray:

$$f_I(t) = \frac{|I(t) - I^{\star}|}{\max\left(\frac{\int_0^t |I(t) - I^{\star}| \, dt}{t}, d\right)}, \tag{2.35}$$

where t is an arbitrary parameter representing the Euclidean arc-length along the ray and d is a small positive number to prevent a division by zero. This analysis is depicted in Fig. 2.15. Next, an affine covariant region is defined by linking the locations corresponding to maxima of f_I along the rays originating from the same anchor point. The resulting irregularly-shaped region is replaced by an ellipse, which has the same shape moments up to second order. In the end, the area of the fitted ellipse is doubled to cover more distinctive patches, which facilitates the matching process. The construction of IBR features is summarized in Fig. 2.16. The output of an IBR detection is depicted in Fig 2.17. In the given example, one can see that IBR features are characterized by some redundancy and they cannot be straightforwardly interpreted.

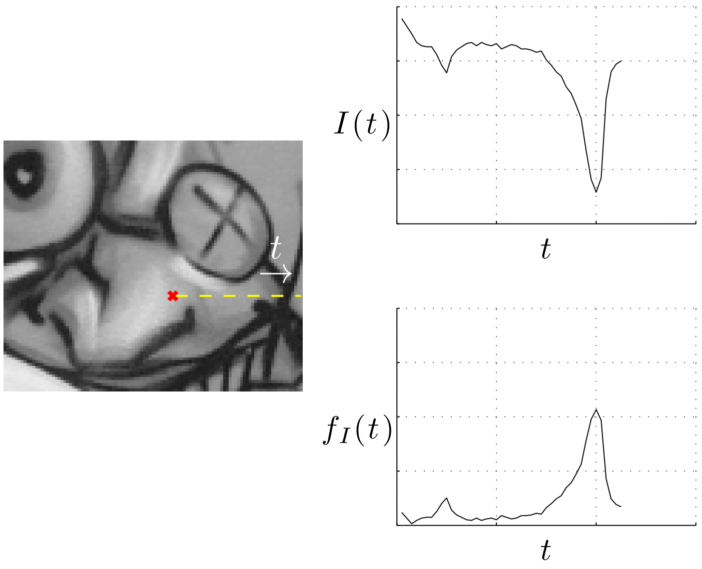


Figure 2.15: The IBR algorithm analyzes the intensity pattern along rays emanating from a point where intensity reaches an extremum.

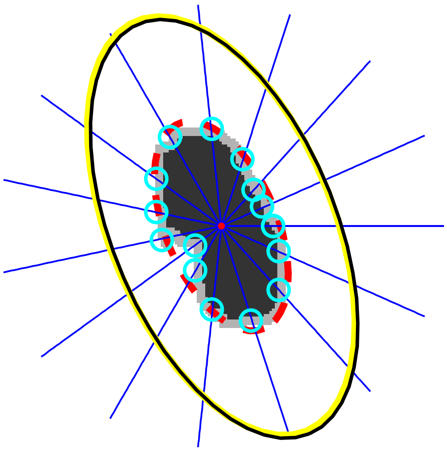


Figure 2.16: Construction of Intensity Extrema-based Regions. Adapted from Tuytelaars & Gool (2004).

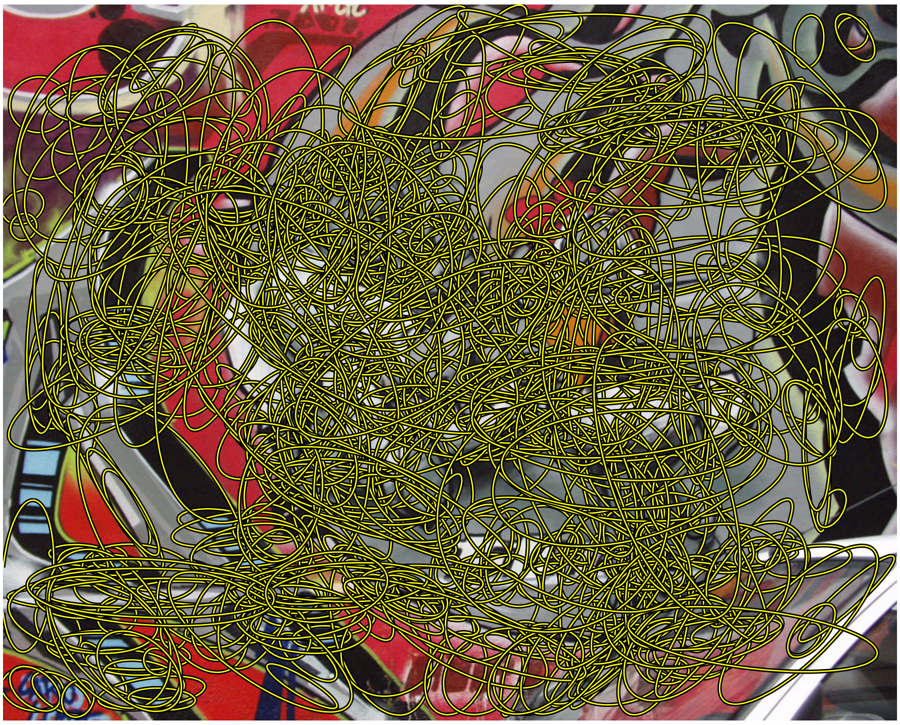


Figure 2.17: Affine covariant features extracted by the IBR algorithm.

2.2.2.4 *Maximally Stable Extremal Regions (MSER)*

Affine covariant regions can be derived from extremal regions. In the image domain, an extremal region corresponds to a connected component whose corresponding pixels have either higher or lower intensity than all the pixels on its boundary. Extremal regions hold two important properties: the set of extremal regions is closed under continuous transformations of image coordinates as well as monotonic transformations of image intensities. The Maximally Stable Extremal Regions (MSER) detector (Matas et al., 2002) responds to extremal regions that are stable with respect to intensity perturbations. For a better understanding of the MSER detector, we introduce the formal definitions of connected component and extremal regions.

A connected component (or region) $\mathcal{Q}$ in $\mathcal{D}$ is a subset of $\mathcal{D}$ for which each pair of pixels $(\mathbf{p}, \mathbf{q}) \in \mathcal{Q}^2$ is connected by a path in $\mathcal{Q}$, i. e., there is a sequence $\mathbf{p}, \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m, \mathbf{q} \in \mathcal{Q}$ such that $\mathbf{p} \sim \mathbf{a}_1, \mathbf{a}_1 \sim \mathbf{a}_2, \ldots, \mathbf{a}_m \sim \mathbf{q}$, where $\sim$ denotes the equivalence relation defined by $(\mathbf{p} \sim \mathbf{q}) \iff \max\{|p_1 - q_1|, |p_2 - q_2|\} \leqslant 1$ (8-neighborhood).

We define the boundary of a region $\mathcal{Q}$ as the set $\partial\mathcal{Q} = \{\mathbf{p} \in \mathcal{D} \backslash \mathcal{Q} : \exists \mathbf{q} \in \mathcal{Q} : \ \mathbf{p} \sim \mathbf{q}\}$. A connected component $\mathcal{Q}$ in $\mathcal{D}$ is an extremal region if $\forall p \in \mathcal{Q}, \mathbf{q} \in \partial\mathcal{Q} : I(\mathbf{p}) < I(\mathbf{q})$ or $I(\mathbf{p}) > I(\mathbf{q})$.

Let $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_{i-1}, \mathcal{Q}_i, \ldots$ be a sequence of extremal regions such that $\mathcal{Q}_k \subset \mathcal{Q}_{k+1}, k = 1, 2, \ldots$. We say that $\mathcal{Q}_i$ is a maximally stable extremal region if and only if the stability criterion

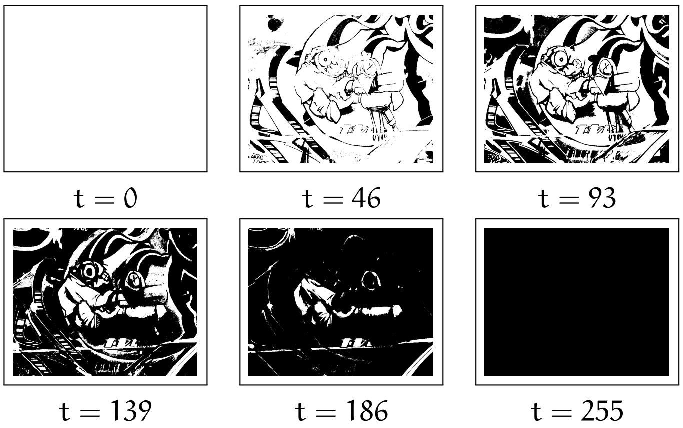
$$\rho(k, \Delta) = \frac{|\mathcal{Q}_{k+\Delta} \setminus \mathcal{Q}_k|}{|\mathcal{Q}_k|}, \tag{2.36}$$

attains a local minimum at i, where Δ is a positive integer denoting the stability threshold. As area ratios are preserved under affine transformations, ρ is an affine invariant measure. Consequently, MSER features are covariant with these geometric transformations.

Note that MSER detection is related to image thresholding, since every extremal region is a connected component of a thresholded image (Tuytelaars & Mikolajczyk, 2008). In fact, we can alternatively describe MSER detection as a process that considers all the possible thresholdings of an intensity image (see Fig. 2.18), yielding different binary images B_t:

$$B_t(\mathbf{x}) = \begin{cases} 1 & \text{if } I(\mathbf{x}) \geqslant t \\ 0 & \text{otherwise} \end{cases} . \tag{2.37}$$

An extremal region in B_t is considered maximally stable if it shows a small area change across several thresholdings (Forssén, 2007). By



$t = 0$ $t = 46$ $t = 93$

$t = 139$ $t = 186$ $t = 255$

Figure 2.18: Sequence of binary images produced by different thresholdings of the same image.

increasing the threshold, we detect MSER+, which correspond to dark regions with brighter boundaries. MSER−, which are brighter regions with dark boundaries, can be obtained through the same process by inverting the input intensity image.[2]

In the original implementation, the MSER detector enumerates extremal regions using a union-find algorithm, whose complexity is $\mathcal{O}(n \log \log n)$, where n denotes the number of pixels. The low computational complexity of the algorithm along with the high repeatability rates shown by the MSER detector in structured images and the suitability of MSERs to be described either by photometric or by shape descriptors (Moreels & Perona, 2007; Forssén & Lowe, 2007) have made the MSER detector a prominent reference in the literature. MSER features are more present in well-structured scenes with homogeneous regions (such as the example depicted in Fig.2.19), while textured scenes provide a reduced number of these features. In addition, MSER detection is sensitive to image blur, as it severely affects the stability criterion.

The MSER detector has known several extensions and refinements. It has been extended to deal with color images (Forssén, 2007) as well as video sequences (Donoser & Bischof, 2006). Forssén & Lowe (2007) proposes an alternative MSER detector that makes use of a multi-scale pyramid representation with one octave between scales. This multi-resolution approach detects MSER at each resolution and duplicated regions at consecutive scales are removed by discarding fine

[2] Our definition of MSER+ and MSER− features differs from the original one. Matas et al. (2002) define MSER+ features as brighter regions with dark boundaries. However, several other authors adopted the definitions given herein.

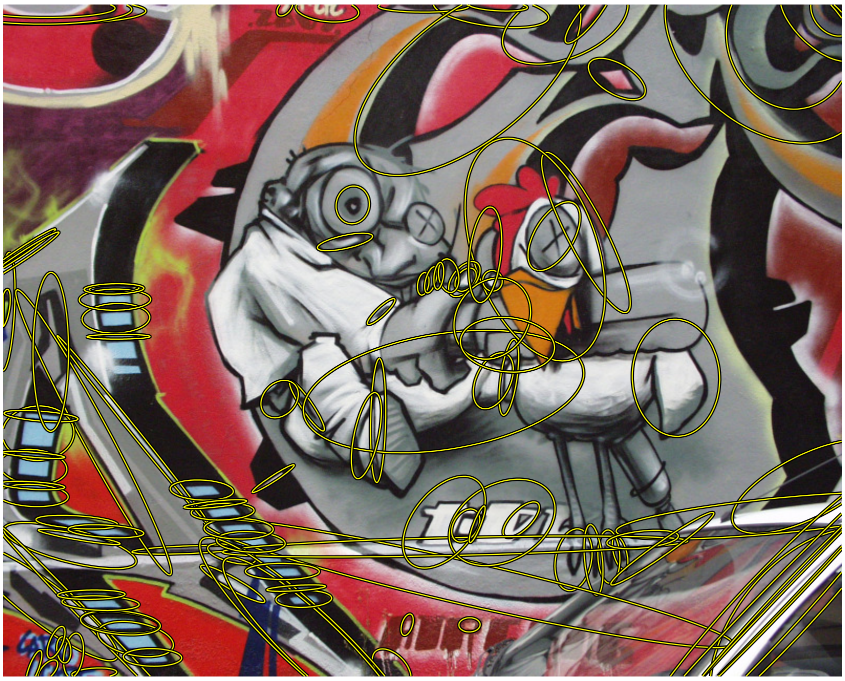


Figure 2.19: Maximally Stable Extremal Regions.

scale features with similar locations and sizes as regions detected at the next coarser scale. The multi-resolution MSER detector produces a higher number of regions and is more robust to image blur and scale changes. On the downside, it requires a detection at each scale, which increases the computational complexity of the algorithm. The algorithm for the detection of Stable Affine Frames (SAF) (Perdóch et al., 2007) can be regarded as a refinement of the MSER detector. SAF features lie on the boundary of extremal regions. Unlike MSER, the stability of SAF with respect to intensity perturbations is measured locally, i. e., we do not require the whole boundary to be stable to intensity changes. This algorithm produces a higher number of features and covers more evenly the image content. Moreover, SAF are more repeatable in the presence of image blur. On the downside, the method requires a considerably higher computational effort. Kimmel et al. (2011) free the MSER detector from the preference towards regular shapes by presenting several redefinitions of the stability criterion, which prefer irregular shapes and are still affine invariant. The main goal of these reinterpretations is to define more distinctive shape descriptors.

2.2.2.5 *Principal Curvature-Based Regions (PCBR)*

The Principal Curvature-Based Regions (PCBR) detector (Deng et al., 2007) uses structural information to detect affine covariant features.

The idea is to use edges and lines to construct structure-based regions. The structural information is obtained from the principal curvature image, which is given by either

$$P(\mathbf{x}) = \max(\lambda_2(\mathcal{H}(\mathbf{x})), 0) \tag{2.38}$$

or

$$P(\mathbf{x}) = \min(\lambda_1(\mathcal{H}(\mathbf{x})), 0), \tag{2.39}$$

where $\lambda_1(\mathcal{H}(\mathbf{x}))$ and $\lambda_2(\mathcal{H}(\mathbf{x}))$ denote, respectively, the minimum and maximum eigenvalues of the image Hessian matrix at $\mathbf{x}$. The principal curvature images are calculated in a scale-space. The first image in the scale space, $I_{1,1}$, has double the size of the input image, while the subsequent images, $I_{1,j}$, correspond to increasingly Gaussian smoothed images with scales $\sigma = (2^{1/3})^{j-1}$, with $j = 2, \ldots, 6$. Then, image $I_{1,4}$ is down-sampled to half of its size to yield a new image, $I_{2,1}$, which is the first image in the second octave. The process is repeated until the creation of $\log_2(\min(n, m)) - 3$ octaves, where n and m are the width and height of the doubled image, respectively. The maximum curvature over each set of three consecutive principal curvature images in the scale-space is computed, which will produce four new images for each one of the octaves. Stable regions will be defined from these regions, using a watershed algorithm. Since watershed segmentation is sensitive to noise, the authors precede the segmentation with a grayscale morphological closing and a eigenvector-flow guided hysteresis thresholding to provide cleaner maps. Finally, stable regions are selected across local scale changes. The selection is based on the computation of the regions overlap error computed across each triplet of consecutive scales. When the overlap error is greater than 90%, one region is kept (the one at the smaller scale). If the error is less than 70%, all regions are discarded. Otherwise, all regions are kept.

PCBR features are shown in Fig. 2.20. The use of structural information makes this algorithm more suitable to deal with well-structured scenes, which reflects the initial purpose of the algorithm: to support object recognition an symmetry detection tasks. Further, the detection of overlapping regions at different scales induces some variation, which helps object recognition since it provides several descriptions of the same pattern.

2.3 Discussion and concluding remarks

Despite the maturity that local feature detection deservedly claims, it is equally valid to consider it as a subject with crucial open issues, demanding the definition of new research directions. The evolution of the topic has been partially dictated by the applications. Early algo-

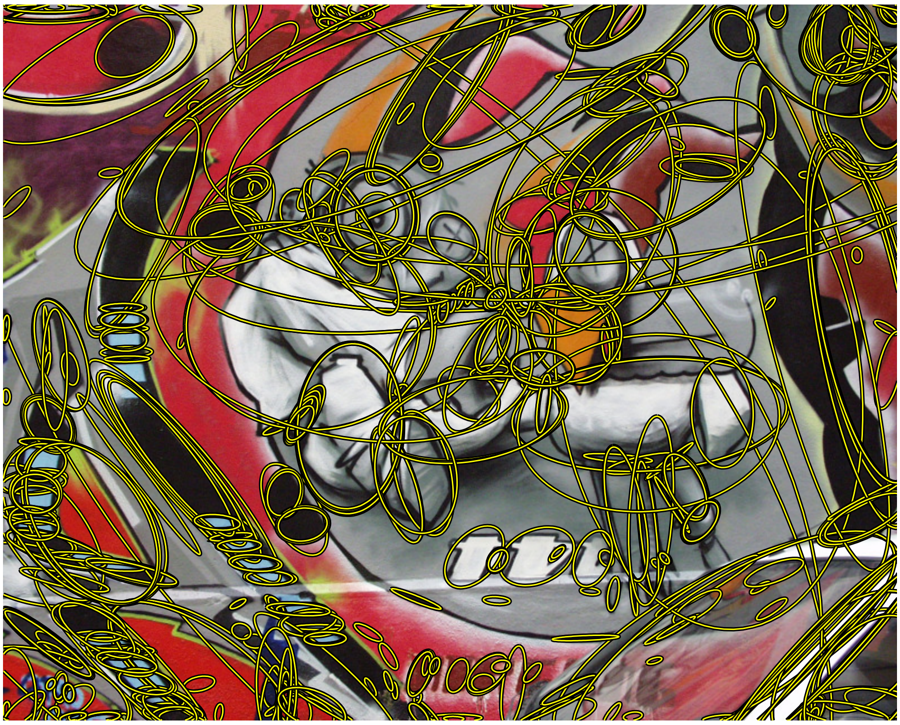


Figure 2.20: Principal Curvature-Based Regions.

rithms were used in matching as well as in tracking, and camera calibration problems. Initially, the levels of invariance/covariance and robustness were more relaxed. As solutions were found to the different problems, more complex tasks were proposed. Robust wide-baseline stereo matching is a clear example of a problem which has only been successfully solved with the introduction of affine covariant features. The studies on scale-space theory have set a milestone in local feature detection. The use of scale-space representations has contributed to increase the level of covariance: various methods have gained covariance with respect to similarity transformations or even affine transformations. Scale-space representation has equally contributed to set another milestone: the introduction of invariant feature descriptors, namely the SIFT descriptor. The combination of local features and local descriptors has opened a new and promising direction for local features. Robust and compact image representations were now possible, which allowed local features to be used in an even wider range of applications. Local feature detection became a reliable basis for solving problems in which a semantical interpretation was involved, such as the tasks of recognizing objects, classifying scenes, and retrieving semantically equivalent images. Furthermore, the use of local features in real-time applications has required the design of extremely efficient algorithms. The FAST algorithm provides a clear example of an efficient algorithm which has been successful in real-time applications.

Regardless of such evolution, there are open issues that need to be addressed in the future.

biased performance evaluation

The evaluation of local features is mainly based on the repeatability criterion. Repeatability is an important requirement for most applications. However, it is insufficient to evaluate a local feature detector solely based on repeatability. An evaluation should take into consideration other criteria in order to assess the suitability of algorithms for a category of applications or even for a type of images. In order to provide more reliable validations, it is crucial to include other criteria in the evaluation process.

reduced repeatability

The repeatability scores achieved by state-of-the-art algorithms are usually high. However, there is still some room for improvement, namely in terms of affine covariance or invariance with respect to illumination changes.

low completeness and neglected complementarity

Using a dense sampling grid of photometric descriptors (Bosch et al., 2007; Liu et al., 2011) is a common and successful strategy in object class recognition. A more interesting solution would be to efficiently extract local features representing the most informative content. Detectors are usually designed to extract one or two types of features. Although local features are, by definition, informative parts of an image, the extraction of a reduced number of types of structures does not ensure a robust image representation. The completeness of sets of features can be easily increased by using a combined feature extraction, where different detectors retrieve complementary features. The major downside of this approach is its computational complexity. Additionally, the different detectors do not provide fully complementary features. The existence of redundant features is not necessarily a downside if they present a slight variation, which is advantageous to perform recognition tasks. While a substantial effort has been put to improve repeatability, only a few authors have focused on the problem of retrieving complete sets of features. Given the current application domains of local features, it is imperative to study the completeness of features as well as their complementarity.

3 EVALUATION OF LOCAL FEATURES

We give an overview of some of the main benchmarks and datasets used in the evaluation of local feature detectors. Although it is not a comprehensive review, it covers commonly used protocols, namely the ones that are seen as standard benchmarks. The main purpose of this analysis is to highlight the advantages and disadvantages of current evaluation protocols.

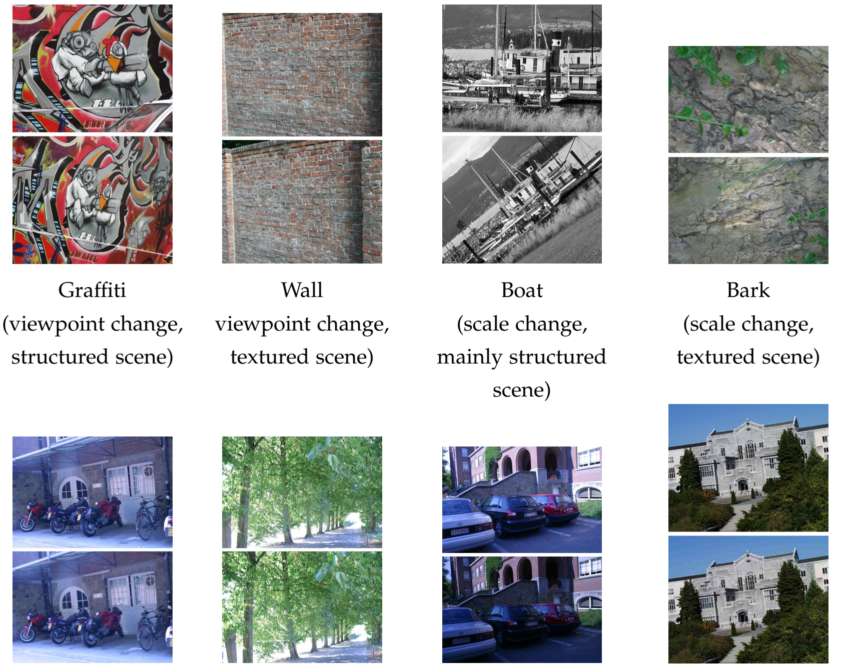
3.1 INTRODUCTION

How to evaluate the performance of local feature detectors? Which criteria should we use? Regardless of the relevance of such questions, one cannot provide simple and straightforward answers. A review of the literature shows us that repeatability is indubitably the preferred criterion to evaluate local feature detectors. The tendency is to regard a detector as a generic tool and measure a property that appears relevant in most cases. Repeatability is indeed one of the most important properties, but it neither reflects the usefulness of features nor guarantees full effectiveness in a given application domain (Tuytelaars & Mikolajczyk, 2008; Rosten et al., 2010). Furthermore, there are no generic detectors (Triggs, 2004) and the evaluation of repeatability will only provide an upper bound on performance (Rosten et al., 2010). On the other hand, the evaluation of a given algorithm in a specific application is reductive in the sense that such analysis does not consider a wider range of applications. It is therefore crucial to consider other properties or criteria rather than repeatability. By taking into consideration other properties, a more comprehensive evaluation could be achieved, which would help us to assess the suitability of the detectors in several application domains. For example, for recognition tasks, it is more crucial to analyze the reconstruction capability of features rather than repeatability (Tuytelaars & Mikolajczyk, 2008).

3.2 BENCHMARKS AND DATASETS

3.2.1 *The Oxford benchmark*

The repeatability evaluation protocol proposed by Mikolajczyk et al. (2005) has become the de facto benchmark. It is mainly a tool to evaluate the repeatability of affine covariant features. However, other properties can be easily assessed, namely the accuracy and the distinctiveness of features. The benchmark is supported by the Oxford image dataset, which comprises 8 sequences with 6 images each, showing 5 different changes in imaging conditions: viewpoint changes, scale changes (with rotation), blurring, illumination changes, and JPEG compression. The images are of medium resolution ($\approx 800\times640$ pixels) and the sequences depict either different views of a planar scene or fixed-camera scenes. The dataset contains two different scene types: structured and textured. The former contains homogeneous regions delineated by well-defined boundaries; the latter consists of several repeated textures. Most of the above-mentioned changes in imaging conditions are applied to both scene types, which means that such changes can be analyzed for each scene type. Figure 3.1 depicts some images of the different sequences in the Oxford dataset. In Fig. 3.2, we depict all the images from one of the sequences: the Graffiti sequence. In the presented sequence, a viewpoint change occurs, starting from a fronto-parallel view.



Graffiti (viewpoint change, structured scene)

Wall viewpoint change, textured scene)

Boat (scale change, mainly structured scene)

Bark (scale change, textured scene)

Bikes (de-focus blur, structured scene)

Trees (de-focus blur, textured scene)

Leuven (illumination change, structured scene)

UBC (JPEG compression, mainly structured scene)

Figure 3.1: First (reference) and third images from each sequence of the Oxford dataset.

The core of the evaluation is a repeatability test that assigns a score to a given detector. The score is computed for each pair of images $(I_{1,s}, I_{k,s})$, where $k \in \{2, 3, \dots, 6\}$ denotes the image position in the sequence and $s \in \{1, 2, \dots, 8\}$ denotes the sequence. The repeatability score is based on an overlap error, which requires the features to be replaced by approximating ellipses. In the case of affine co-

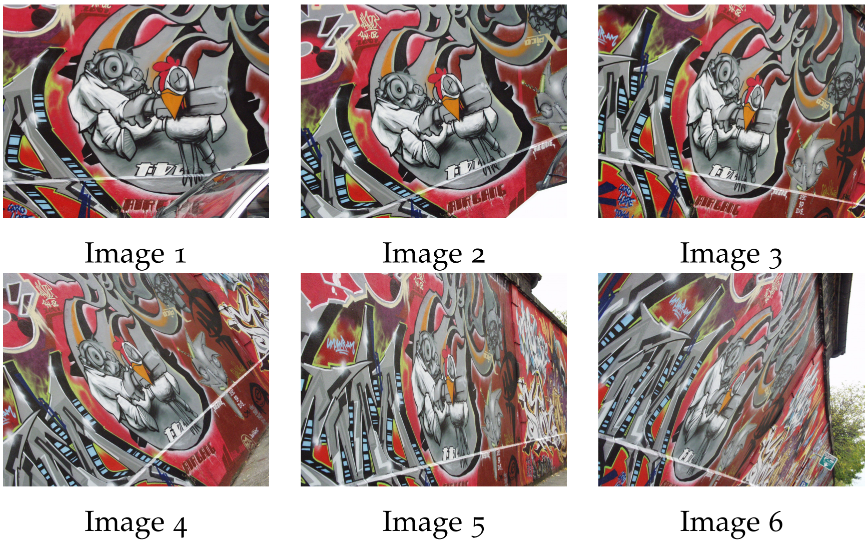


Image 1

Image 2

Image 3

Image 4

Image 5

Image 6

Figure 3.2: Graffiti sequence.

variant features, some features already correspond to ellipses (e. g., Hessian-Affine), while others, such as MSER, have arbitrary shapes which have to be approximated by ellipses.

To compute the repeatability score between regions detected on two image pairs, 2D homographies are used as a ground truth. The idea is to map the features detected on an image to the reference image (first image in the sequence). A homography is a projective transformation that will provide such mapping. Figure 3.3 shows two views of a scene (images I_1 and I_2) acquired with cameras C_1 and C_2. A point $\mathbf{x}$ in I_1 can be associated with its corresponding point $\mathbf{x}'$ in I_2 through one of the homography matrices $H_{I_1 I_2}, H_{I_2 I_1} \in \mathbb{R}^{3\times 3}$, using homogeneous coordinates:

$$\begin{bmatrix} w\mathbf{x}' \\ w \end{bmatrix} = H_{I_1 I_2} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}; \tag{3.1}$$

$$\begin{bmatrix} w\mathbf{x} \\ w \end{bmatrix} = H_{I_2 I_1} \begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix}, \tag{3.2}$$

with $w \in \mathbb{R} \setminus \{0\}$ and $H_{I_2 I_1} = H^{-1}_{I_1 I_2}$.

Two features (regions) are deemed as corresponding and, therefore, repeated, with an overlap error of $\epsilon_0 \times 100\%$ if

$$1 - \frac{\left|\mathcal{R}_{\mu_1} \cap \mathcal{R}_{(H^\mathsf{T} \mu_2 H)}\right|}{\left|\mathcal{R}_{\mu_1} \cup \mathcal{R}_{(H^\mathsf{T} \mu_2 H)}\right|} < \epsilon_0, \tag{3.3}$$

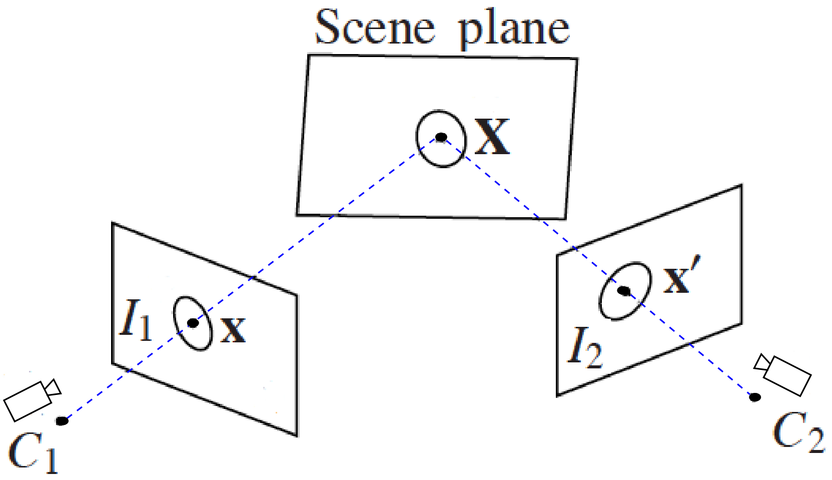


Figure 3.3: Projection of an ellipse around the point $\mathbf{X}$ to the images I_1 and I_2 acquired with the respective cameras C_1 and C_2. $\mathbf{x}$ and $\mathbf{x}'$ are corresponding points. Adapted from Schmid et al. (2000) and Cordes et al. (2011).

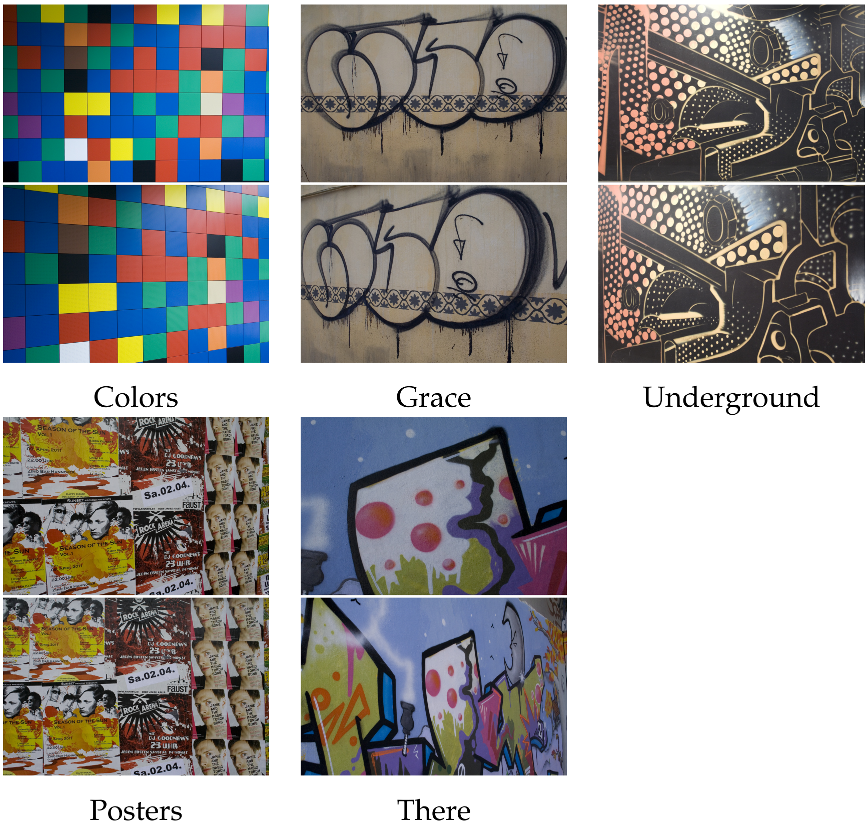
where R_μ denotes the set of image points in the elliptical region verifying $\mathbf{x}^T \mu \mathbf{x} \leqslant 1$ and H is the homography that relates the two input images. For a given pair of images and a given overlap error, the repeatability score corresponds to the ratio between the number of correspondences between regions and the smaller of the number of regions in the pair of images. Only regions that are located in parts of the scene that are common to the two images are considered. In fact, this procedure can give us two repeatability measures: a relative one (the repeatability score) and the absolute repeatability (the number of correspondences). These results are usually reported for an overlap error of 40%.

To get an idea of the accuracy of the detectors, one can analyze the repeatability rates for different overlap error thresholds (the evaluation protocol computes the repeatability for overlap errors of 10%, 20%, . . . , 60%). A less accurate detector will show higher variations in the repeatability score as we change the overlap error.

An important feature of the Oxford benchmark is the ability to perform a matching test, which is similar to the one defined for repeatability. This matching test gives an idea of how distinctive features are. The detected regions provide the image patches that will be described and matched. The matching score between the reference image and the other images in a sequence is computed as the ratio between the number of correct matches and the smaller number of detected regions in the pair of images. The Euclidean distance is used to compare descriptors. A match corresponds to the nearest neighbor in the descriptor space. A maximum overlap error of 40% is allowed for matched regions. A comparison of repeatability and matching scores allows us to infer about the distinctiveness of features: a matching score that significantly differs from the repeatability score for a given feature detector suggests less distinctive features. As the authors note, these results are mainly indicative rather than quantitative.

3.2.2 *The TNT dataset (an extension to the Oxford dataset)*

The TNT (acronym for Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung) dataset (Cordes et al., 2011) is an extension to the Oxford dataset. It contains sequences of 6 high resolution images (1534×1024 pixels) in which the main transformation is a viewpoint change. The first and third images from each sequence in the dataset are depicted in Fig. 3.4. To illustrate the image transformations, we show the whole Colors sequence in Fig. 3.5.



Colors

Grace

Underground

Posters

There

Figure 3.4: First (reference) and third images from each sequence of the TNT dataset.

The authors who proposed the TNT dataset have also improved the accuracy of the ground truth homographies provided for both sets. The updated homographies have shown to provide higher re-

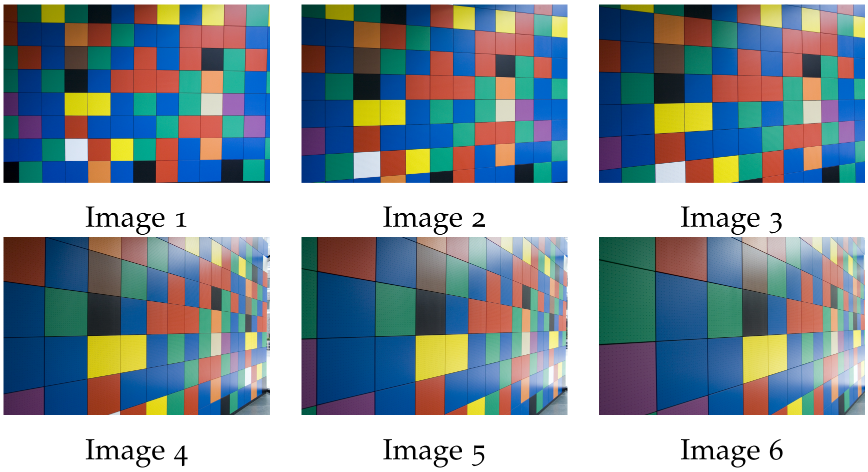


Image 1

Image 2

Image 3

Image 4

Image 5

Image 6

Figure 3.5: Colors sequence.

peatability scores. In some cases, the authors reported an increase of 20%.

3.2.3 *The Robot benchmark*

The Robot dataset was created by Aanæs et al. (2012) at the Department of Informatics and Mathematical Modeling of the Technical University of Denmark. The large-scale dataset was proposed as a precise ground truth for benchmarking local feature detection. It contains 135.660 high resolution color images (1200×1600 pixels), depicting 60 different scenes. For each scene, images were acquired from 119 positions, and for each position, the scene was illuminated by 19 white LEDs. Figure 3.6 shows two scenes from the dataset, taken from two different viewpoints and with two different lighting conditions. To provide a very accurate camera positioning, the acquisition was performed with a camera mounted on a 6-axis robot. For each scene, the sequence of camera positions follows a predefined path, which was chosen relative to a central image position (key frame).

The Robot dataset makes the evaluation of the performance of local features possible to a very high degree of accuracy. The authors presented a comprehensive study that evaluates the performance of different methods in terms of repeatability and accuracy, reflecting the covariance of features with respect to scale and viewpoint changes, as well as the invariance with respect to light changes.

The ground truth for the evaluation is provided by the geometry of the 3D scene surfaces and the camera positions. To obtain a 3D surface reconstruction, the different scenes were surface scanned using structured light. Both the surface information and the camera posi-

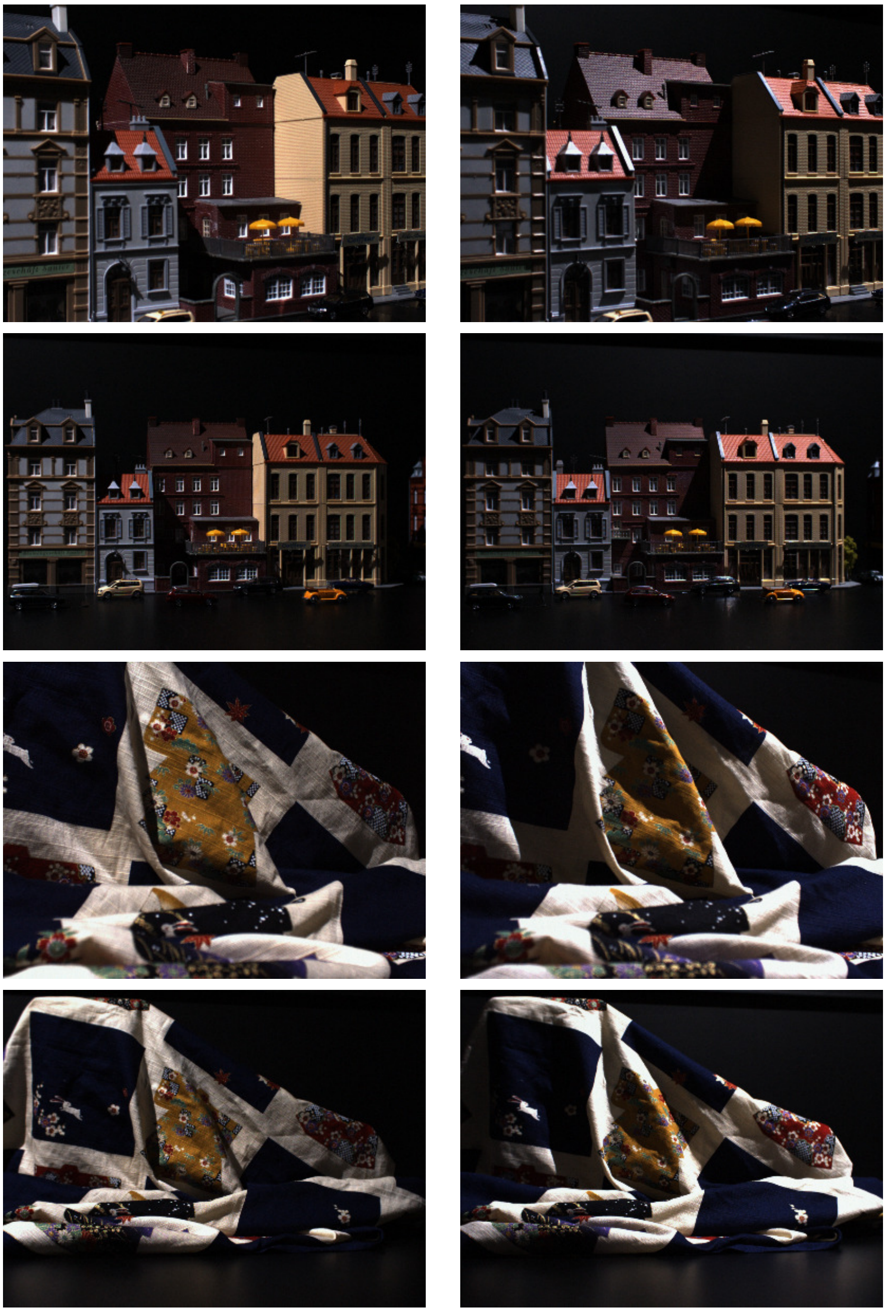


Figure 3.6: Scenes 1 (first and second row) and 6 (third and fourth row) from the Robot dataset. Each row shows a scene acquired from a different viewpoint. For each viewpoint, a pair of images with different lighting conditions is shown.

tions form the basis of the evaluation, which is essentially a matching test performed for each keypoint in the key frame. Figure 3.7 schematizes the three criteria which have to be fulfilled to accept the correspondence between points as a potential match.

The authors use a recall rate as a performance measure, which corresponds to the ratio

$$\mathtt{recall} = \frac{\#\mathtt{potentialMatches}}{\#\mathtt{keypoints}}, \tag{3.4}$$

where $\#\mathtt{potentialMatches}$ indicates the number of keypoints from the key frame fulfilling the three criteria illustrated in Fig. 3.7 and $\#\mathtt{keypoints}$ represents the number of keypoints in the key frame.

The authors also propose a complementarity measure. The complementarity between two sets of keypoints $\mathcal{X}$ and $\mathcal{Y}$ is measured by computing the distance from each point in the structured light scan $\mathcal{S}$ to the nearest point in $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{X} \cup \mathcal{Y}$. Then, an average of the distributions is constructed:

$$D_{\mathcal{X}} = \frac{1}{n} \sum_{i=1}^{n} \min_{j} \left\| \mathbf{x}_j - \mathbf{s}_i \right\|, \tag{3.5}$$

$$D_{\mathcal{Y}} = \frac{1}{n} \sum_{i=1}^{n} \min_{j} \left\| \mathbf{y}_j - \mathbf{s}_i \right\|, \tag{3.6}$$

$$D_{\mathcal{X} \cup \mathcal{Y}} = \frac{1}{n} \sum_{i=1}^{n} \min(\min_{j} \left\| \mathbf{x}_j - \mathbf{s}_i \right\|, \min_{j} \left\| \mathbf{y}_j - \mathbf{s}_i \right\|), \tag{3.7}$$

where n denotes the number points in $\mathcal{S}$ and $\mathbf{x}_i$, $\mathbf{y}_i$, and $\mathbf{s}_i$ represent points in $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{S}$, respectively. These distance distributions provide the following complementarity measure:

$$\mathtt{comp}(\mathcal{X}, \mathcal{Y}) = \frac{2 \frac{D_{\mathcal{X} \cup \mathcal{Y}}}{\sqrt{n_{\mathcal{X}} + n_{\mathcal{Y}}}}}{\frac{D_{\mathcal{Y}}}{\sqrt{n_{\mathcal{X}}}} + \frac{D_{\mathcal{Y}}}{\sqrt{n_{\mathcal{Y}}}}}, \tag{3.8}$$

where $n_{\mathcal{X}}$ and $n_{\mathcal{Y}}$ indicate the number of keypoints in $\mathcal{X}$ and $\mathcal{Y}$, respectively.

If there is a high complementarity between $\mathcal{X}$ and $\mathcal{Y}$, the average distance from $\mathcal{X} \cup \mathcal{Y}$ to the structured light scan will be low.

3.2.4 *The completeness (and complementarity) benchmark*

A measure of completeness becomes particularly useful if we want to assess the suitability of local features in efficiently summarizing the

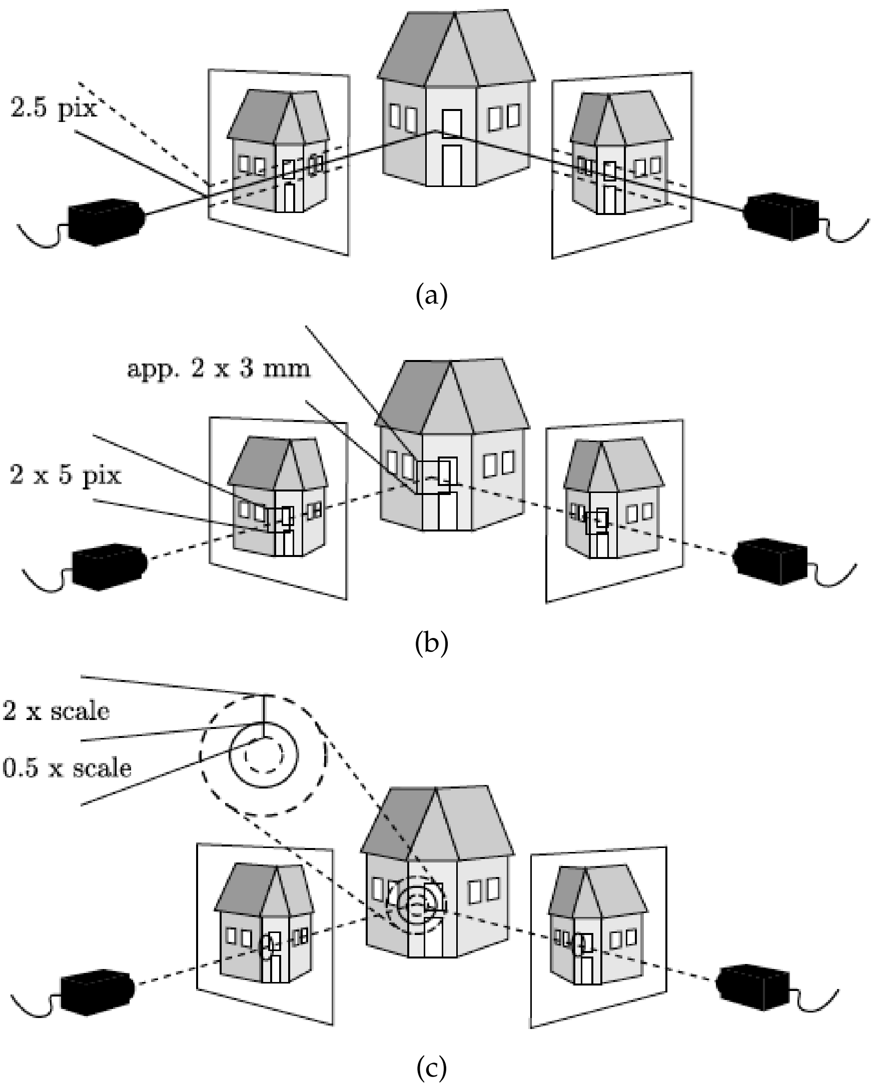
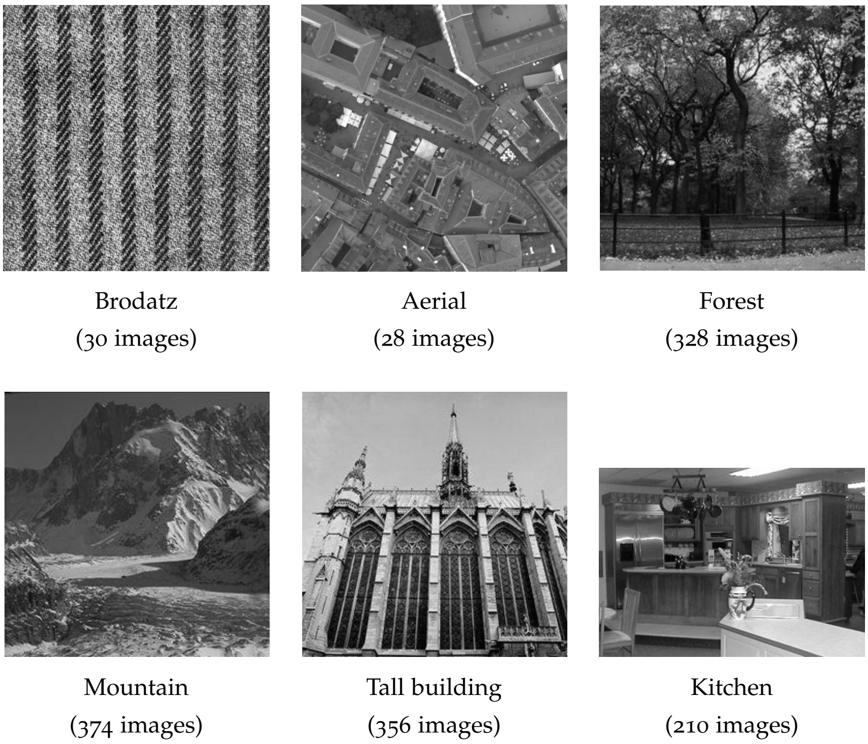


Figure 3.7: Criteria for repeatability/matching evaluation. (a) Criterion 1: corresponding descriptors should be within 2.5 pixels from the epipolar line. (b) Criterion 2: window of interest with a radius of 5 pixels corresponding descriptors should be within the window; (b) Criterion 3: corresponding descriptors are within a scale range factor of 2 from each other. Source: Aanæs et al. (2012).

relevant image content. Despite the need and importance of quantifying completeness, the recent work of Dickscheid et al. (2011) can be seen as the first attempt to tackle this problem. The authors proposed metrics for completeness and complementarity which were used in a large-scale test comprising different types of features. The goal was not only to find the most complete type of features but also to study the complementarity among different types of features.

The dataset used in the evaluation (Fig. 3.8) comprised four categories of natural scenes (Li & Perona, 2005; Lazebnik et al., 2006), the Brodatz texture collection (Brodatz, 1966), a set of aerial images, and a collection of cartoon images (not depicted in Fig. 3.8).



Brodatz
(30 images)

Aerial
(28 images)

Forest
(328 images)

Mountain
(374 images)

Tall building
(356 images)

Kitchen
(210 images)

Figure 3.8: Example images from the categories in the dataset for completeness and complementarity evaluation.

To measure completeness, Dickscheid et al. (2011) compute an entropy density $p_H(\mathbf{x})$ based on local image statistics and a feature coding density $p_c(\mathbf{x})$ derived from a given set of features. The (in)completeness measure corresponds to the Hellinger distance between the two densities:

$$d_H(p_H, p_c) = \sqrt{\frac{1}{2} \sum_{\mathbf{x} \in \Phi} (\sqrt{p_H(\mathbf{x})} - \sqrt{p_c(\mathbf{x})})^2}, \tag{3.9}$$

where Φ is the image domain.

The entropy density p_H is computed from local image patches with different sizes (scales). The authors assume that these patches represent a larger image, i. e., an $N \times N$ patch is part of a periodic image with period N in both directions. In addition, an image is considered to be a noisy version of a Gaussian process. From these assumptions, the entropy of an image patch g can be derived as follows:

$$H(g) = \frac{1}{2} \log_2(2\pi \exp(\frac{\det \Sigma_{gg}}{\sigma_n^2})), \tag{3.10}$$

where Σ_{gg} represents the covariance matrix of the intensity values in g and σ_n^2 is the noise variance. The determinant of Σ_{gg} is derived from the power spectrum $P(\mathbf{u}) = |DCT(g(\mathbf{x}))|^2$, i. e., $\det \Sigma_{gg} = \prod_{\mathbf{u} \setminus \{\mathbf{o}\}} P(\mathbf{u})$, where $\mathbf{o}$ is the DC coefficient. By assuming that the power spectrum is additively composed of the power spectra of the signal and the noise, the following estimate of the power spectrum can be used:

$$\hat{P}(\mathbf{u}) = \max(P(\mathbf{u}) - \sigma_n^2, 0), \tag{3.11}$$

and (3.10) becomes

$$H(g) = \frac{1}{2N^2} \sum_{\mathbf{u} \setminus \{\mathbf{o}\}} \max(\log_2(2\pi \exp(\frac{\hat{P}(\mathbf{u})}{\sigma_n^2})), 0). \tag{3.12}$$

The entropy at a pixel $\mathbf{x}$ will be obtained from the patches entropy. If $H(\mathbf{x}, N)$ is the entropy of a pixel $\mathbf{x}$ based on a patch of size N, the entropy at pixel $\mathbf{x}$ is

$$H(\mathbf{x}) = \sum_{s=1}^{S} H(\mathbf{x}, 1 + 2^s), \tag{3.13}$$

where $s \in \{1, \ldots, S\}$ denotes the scale. Finally, the density p_H is computed through normalization:

$$p_H(\mathbf{x}) = \frac{H(\mathbf{x})}{\sum_{\mathbf{y} \in \Phi} H(\mathbf{y})}. \tag{3.14}$$

The feature coding density p_c is computed for a given set of features $\mathcal{F}$. It is assumed that a feature $f \in \mathcal{F}$ can be characterized by its location $\mathbf{m}_f$ and its scale σ_f (or Σ_f in the case of affine covariant features). A Gaussian distribution spreading over the image domain is used to represent a region covered by a local feature. It is equally assumed that $c(f)$ bits are required to represent a feature f. Such assumptions lead to the coding map

$$c(\mathbf{x}) = \sum_{f \in \mathcal{F}} c(f) G(\mathbf{x}, \mathbf{m}_f, \Sigma_f), \tag{3.15}$$

where G denotes an anisotropic Gaussian kernel. The final coding density p_C is the result of a normalization:

$$p_c(\mathbf{x}) = \frac{c(\mathbf{x})}{\sum_{y \in \Phi} c(\mathbf{y})}, \tag{3.16}$$

which allows us to compare both densities (Fig. 3.9).

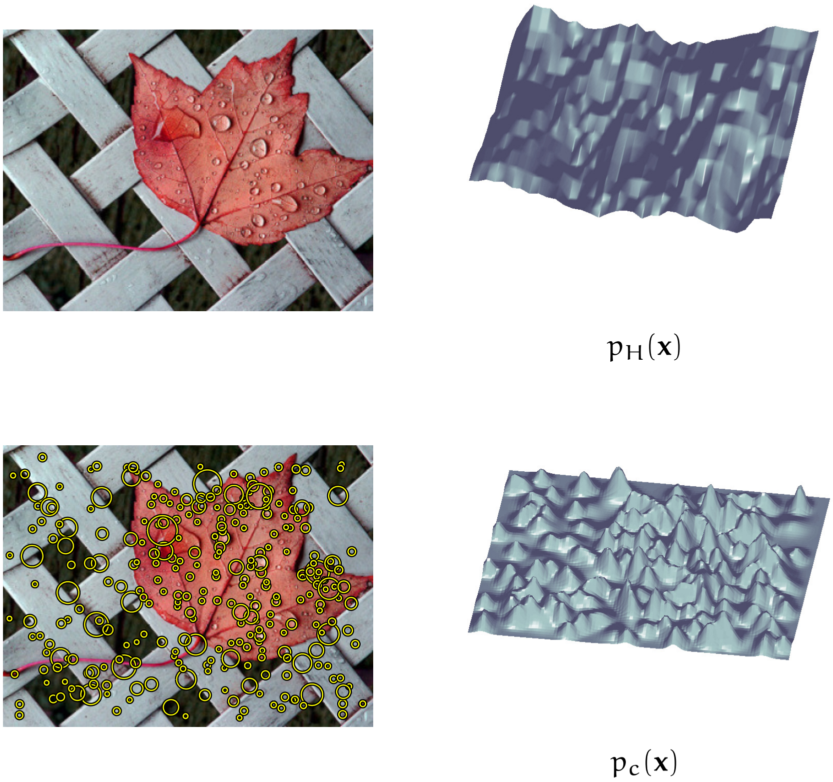
When p_H and p_c are very close, the distance d_H will be small, which means the set of features with a coding density p_c effectively covers the image content (the set of features has a high completeness). Such metric penalizes the use of large scales (a straightforward solution to achieve a full coverage) as well as the presence of features in pure homogeneous regions. On the other hand, it will reward the "fine capturing" of local structures or superimposed features appearing at different scales.

Complementarity can be measured by considering coding densities of sets which are the result of a combined feature detection, i. e., $\mathcal{F} = \bigcup_{i=1,\ldots,N} \mathcal{F}_i$, where $\mathcal{F}_i$ is a set of features of a given type. A high level of complementarity is achieved when $\mathcal{F}$ produces a significantly lower distance $d_H(p_H, p_c)$ than the ones provided by the subsets $\mathcal{F}_i$, $i = 1, \ldots, N$.

3.3 CONCLUDING REMARKS

Evaluating local feature detectors is a complex task, requiring a detailed analysis of different properties. An assessment relying on the analysis of diverse properties helps us to understand which problems are going to benefit from a local feature-based solution and which are not. Thus far, evaluation protocols have been mainly based on the analysis of a few characteristics of the algorithms, namely repeatability and accuracy. Measuring such properties is important, but it will only provide a superficial idea regarding the suitability of features.

The benchmarks reviewed herein represent a valuable contribution to local feature detection. Perhaps the best example that illustrates such contribution comes from the Oxford benchmark: it became the standard protocol to evaluate detectors that claim covariance with respect to rigid and affine transformations. The majority of recent



$p_H(\mathbf{x})$

$p_c(\mathbf{x})$

Figure 3.9: Entropy density $p_H(\mathbf{x})$ of the “Leaf” image and feature coding density $p_c(\mathbf{x})$ of SFOP features (Förstner et al., 2009a) detected on the same image.

state-of-the-art detectors was validated in the Oxford benchmark.

The Robot benchmark represents an improvement over the one proposed by the Oxford Vision Group. It considers a considerably larger dataset (60 scenes vs. 8 scenes, 135.600 images vs. 48 images). In addition, the ground truth describing the geometry of the scenes is more precise. However, the evaluation is still mainly focused on the repeatability and accuracy of features, exposing the performance of feature detectors in more extreme conditions.

While there is not a unified framework that provides the aforementioned desired evaluation, the combination of several standard benchmarking tools aimed at measuring different properties provides a relatively thorough analysis.

The analysis provided by the completeness (and complementarity) benchmark discloses important properties of local features that tend to be neglected. We argue that a more comprehensive analysis should give metrics for both completeness and complementarity, namely when recognition tasks are a potential application. While an isolated completeness test may not be sufficient to conclude on the usefulness of the features, an evaluation that disregards such property is not going to inform us on the reconstruction capabilities of local features.

4 CONTEXT-AWARE FEATURES FOR ROBUST IMAGE REPRESENTATION

Tasks such as image retrieval and object recognition often make use of local image features, which are mainly intended to provide a reliable and efficient image representation. However, local feature detectors are designed to respond to a limited set of structures (e. g., corners and junctions), which might not be sufficient to capture the most relevant image content.

In this chapter, we discuss the lack of coverage of relevant image information by local features as well as the often neglected complementarity between sets of features. As a result, we propose an information theoretic-based keypoint extraction that responds to locations which are salient within the image context. We empirically assess and discuss the validity of the method by analyzing the completeness, complementarity, and repeatability of our context-aware features on different datasets.

4.1 INTRODUCTION AND MOTIVATION

As evidenced by Chapter 2, the desired properties of a local feature detector are dictated by its application. For instance, matching and tracking tasks mainly require a repeatable and accurate feature detection, as the fundamental objective is to accurately identify the same features across a sequence of images, regardless of the degree of deformation. It does not become relevant if the set of features fails to cover the most informative image content. On the other hand, tasks such as object (class) recognition, image retrieval, and image compression require a robust image representation (Tuytelaars & Mikolajczyk, 2008). In these particular cases, the idea is to analyze the image statistics and use local features to capture informative image content. Here, repeatability and accuracy, despite their relative importance, are not primary requirements.

Local feature detectors tend to be based on strong assumptions on the image content. For example, Harris-Stephens keypoint detector (Harris & Stephens, 1988) and Laplacian-based algorithms (e. g., Mikolajczyk & Schmid, 2002; Lowe, 2004) assume that there exist, respectively, corners and blobs in the image. The MSER detector (Matas et al., 2002) assumes the existence of image regions characterized by stable isophotes with respect to intensity perturbations. All of the aforementioned structures are expected to be related to semantically meaningful parts of an image, such as the boundaries, the vertices of objects, or even the objects themselves. However, we cannot ensure that the detection of a particular feature will cover the most informative parts of the depicted scene. Therefore, if our goal is to provide a robust image representation via local feature detection, a plausible and straightforward strategy will be a combined and complementary feature detection, using two or more detectors. In Fig. 4.1, we depict an example of a combined feature detection utilizing SFOP (Förstner et al., 2009a) and Hesssian-Laplace (Mikolajczyk & Schmid, 2002) features. The SFOP algorithm provides an explicit, interpretable, and already complementary detection. Despite the reasonable degree of complementary among SFOP features, we can achieve a more complete detection by combining SFOP features with Hessian-Laplace regions, i. e., blobs.

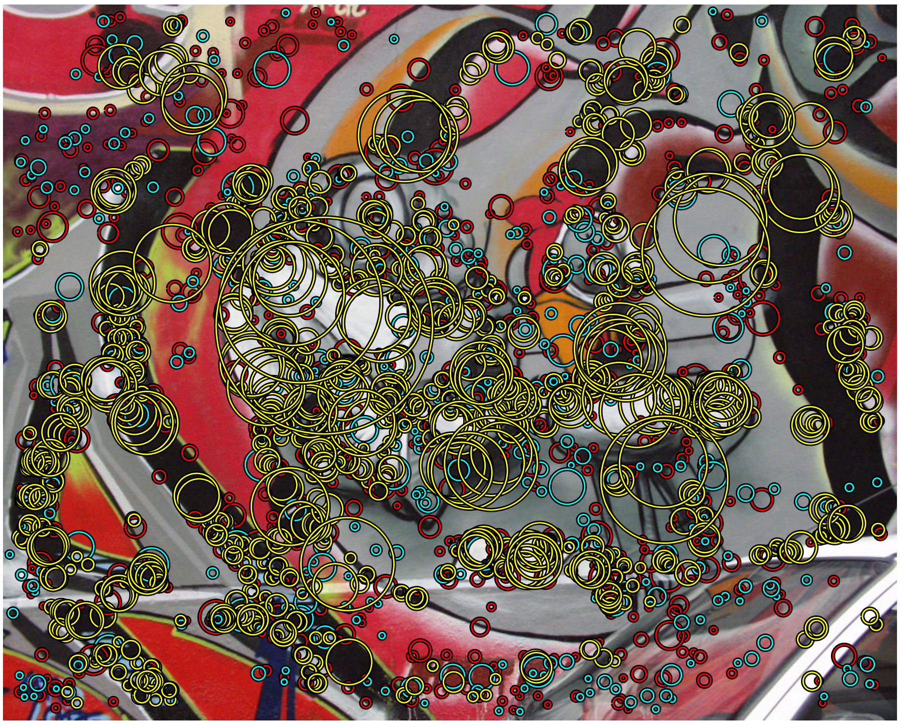


Figure 4.1: Combined feature detection: cyan circumferences enclose SFOP features when $\alpha = 90°$; red circumferences enclose SFOP features when $\alpha = 0°$; yellow circumferences denote the boundaries of Hessian-Laplace regions (blobs). Best viewed in color.

A combined feature detection may not be the ideal solution. In fact, it entails two major shortcomings: first, it implies a higher computational cost due to the use of several detectors instead of one; second, it is not always straightforward to ensure the absence of redundant features, i. e., we cannot guarantee a fully complementary detection.

There are a few attempts in designing a single detector aimed at responding to complementary features (e. g., Förstner et al., 2009a). However, these complementary detections still make strong assumptions on the image content.

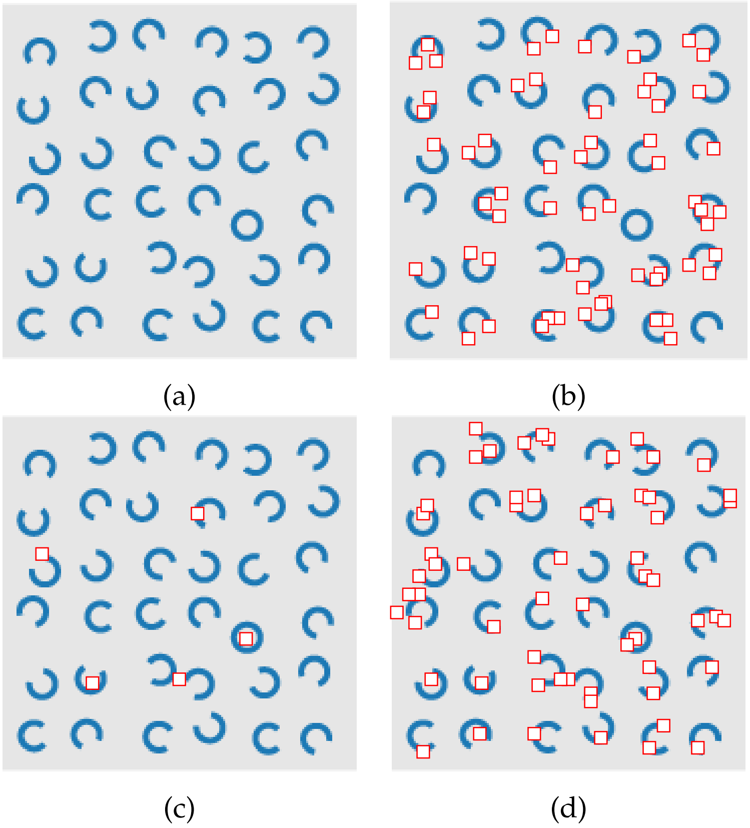
In this chapter, we introduce a novel feature algorithm aimed at providing a robust image representation by responding to complementary features while not making any a priori assumption on the image content. The proposed algorithm, coined as Context-Aware Keypoint Extraction (CAKE), represents a new paradigm in local feature detection: it considers the image context to define salient parts. Figure 4.2 provides an illustrative comparison between our context-aware detection and a strictly local one (Shi & Tomasi, 1994). In the given example, the closed contour, which appears as a relevant object within the image context, is neglected by the Shi-Tomasi algorithm when retrieving the most salient locations (Fig. 4.2 (b)). On the other hand, the proposed context-aware keypoint extraction depicted in the same figure provides a better coverage of relevant content by considering a keypoint inside the closed contour as one of the most salient locations (Fig. 4.2 (c) and (d)).

4.2 CAKE: A CONTEXT-AWARE KEYPOINT EXTRACTOR

4.2.1 *More on motivation*

Our contribution is the CAKE algorithm, a keypoint extractor aimed at covering the most informative image content. The proposed algorithm responds to complementary local structures and is aware of the image composition. We follow an information-theoretic approach by assuming that the so-called salient locations correspond to points within structures with a low probability of occurrence, which is in accordance with a plausible characterization of visual saliency (Bruce, 2005). As we have noted earlier, the majority of local feature detectors tends to make strong assumptions on the image content, which can lead to an ineffectual coverage of the content (Förstner et al., 2009b; Dickscheid et al., 2011). Here, the idea is not to formulate any a priori assumption on the structures that might be salient. Furthermore, our scheme is designed to take advantage of different local representations (descriptors) and the use of information to measure saliency allows us to establish a well-defined hierarchy among features.

Our context-aware extraction can respond to features with a reasonable degree of complementarity as long as they are informative. For images with many types of structures and patterns, one can expect a high complementarity among the features retrieved by a context-aware algorithm. Conversely, images with repetitive patterns inhibit



(a)

(b)

(c)

(d)

Figure 4.2: Keypoints on a psychological pattern: (a) pattern (Julész & Bergen, 1983); (b) 60 most salient Shi-Tomasi keypoints; (c) 5 most salient context-aware keypoints; (d) 60 most salient context-aware keypoints.

context-aware methods from retrieving a clear summarized representation of the content. Nevertheless, in the latter case, the extracted set of features can be complemented with a counterpart that retrieves the repetitive elements in the image. To illustrate the aforementioned advantages, we depict these two considerably different cases in Fig. 4.3.

The image in the top row of Fig. 4.3 shows a context-aware keypoint extraction on a well-structured scene, retrieving the 100 most informative locations. This small number of features is sufficient to provide a reasonable coverage of the content, which includes several types of structures. The image in the bottom depicts the benefits of combining context-aware keypoints with strictly local ones (SFOP keypoints) to obtain a better coverage of textured images. In the latter image, it is important to note the absence of redundant features.

4.2.2 *The algorithm*

Shannon's measure of information (Shannon, 1948) forms the basis of our measure of saliency. If we consider a symbol s, its information is given by

$$I_S(s) = -\log(P(s)), \tag{4.1}$$

where $P(\cdot)$ denotes the probability of a symbol. In the particular case of images, defining symbols is not a straightforward task and using solely the content of a pixel $\mathbf{x}$ is not applicable, whereas the content of a region around $\mathbf{x}$ will be more appropriate. We will therefore consider any local description $\mathbf{w}(\mathbf{x}) \in \mathbb{R}^D$ that represents the neighborhood of $\mathbf{x}$ as a viable codeword.[1] This codeword can be seen as the symbol, so that, with a conceptual shift, we can denote the symbol corresponding to a pixel $\mathbf{x}$ with its codeword $\mathbf{w}(\mathbf{x})$, which allows us to rewrite (4.1):

$$I_S(\mathbf{x}) = -\log(P(\mathbf{w}(\mathbf{x}))). \tag{4.2}$$

In this way, we define the information for each pixel $\mathbf{x}$ using the definition of codeword. However, in Shannon's perspective, a symbol should be a case of a discrete set of possibilities, while we have defined the codeword in $\mathbb{R}^D$. As a consequence, to estimate the probability of a certain symbol, a frequentists approach might be used. In this case, one should be able to quantize codewords into symbols. It is clear that the frequentists approach is inappropriate, and the quantization is a dangerous process applied to a codeword, since the quantization error can induce strong artifacts in the $I(\mathbf{x})$ map, generating spurious local maxima.

[1] For the sake of generality, we are assuming that a codeword is a D-dimensional real vector. However, one can consider other arbitrary domains, including discrete ones.

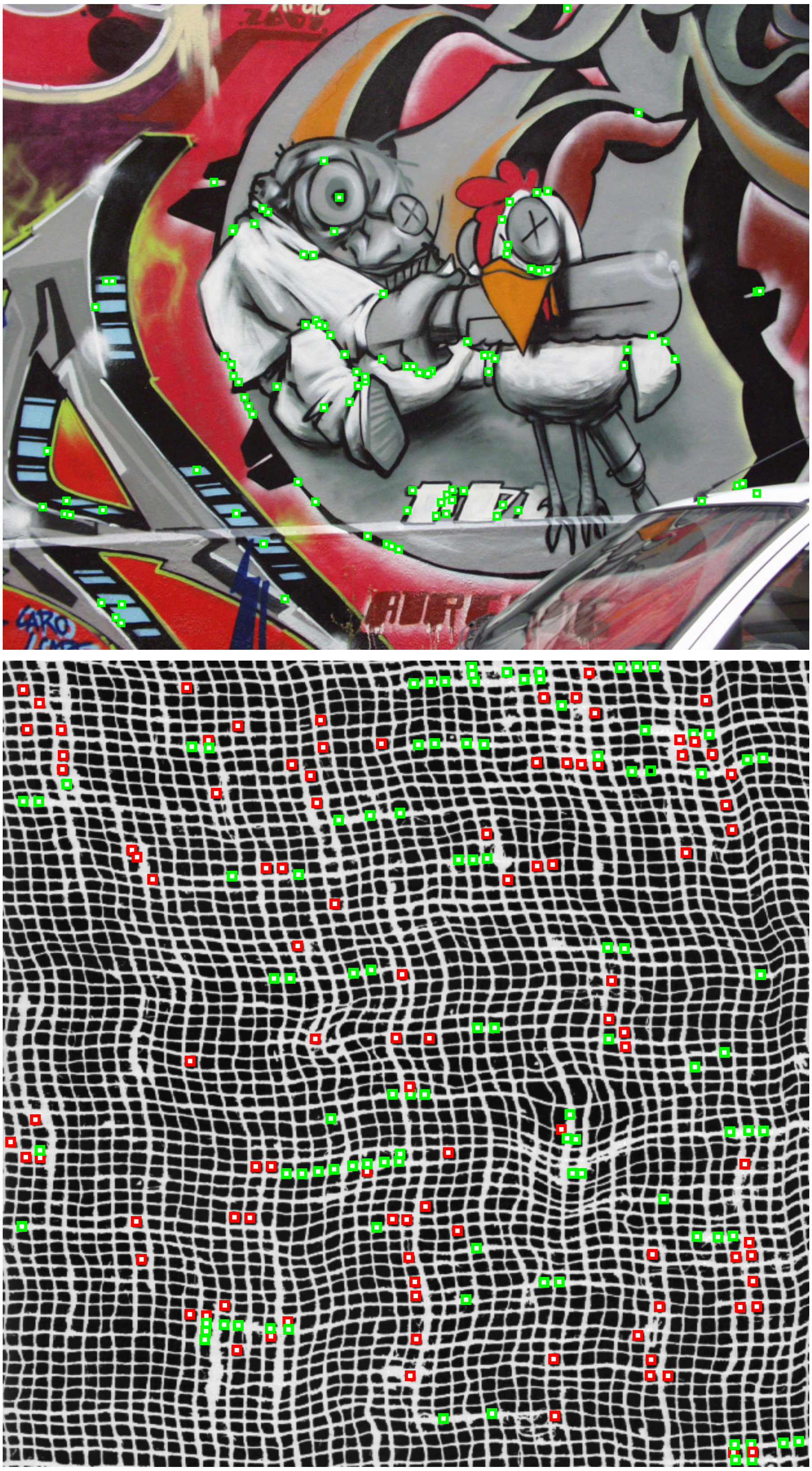


Figure 4.3: Proposed keypoint extraction. Top row: context-aware keypoints on a well-structured scene (100 most informative locations); bottom row: a combination of context-aware keypoints (green squares) with SFOP keypoints (red squares) on a highly textured image (Brodatz, 1966). Best viewed in color.

We abandon the frequentist approach in favor of a Parzen Density Estimation (Parzen, 1962), also known as Kernel Density Estimation (KDE). The Parzen estimation is suitable for our method as it is non-parametric, which will allow us to estimate any probability density function (PDF), as long as there is a reasonable number of samples. Using the KDE, we estimate the probability of a codeword $\mathbf{w}(\mathbf{y})$ as follows:

$$\widehat{P}(\mathbf{w}(\mathbf{y})) = \frac{1}{Nh} \sum_{x \in \Phi} K(\frac{d(\mathbf{w}(\mathbf{y}), \mathbf{w}(\mathbf{x}))}{h}), \qquad (4.3)$$

where K denotes a kernel, d is a distance measure, h is a smoothing parameter called bandwidth and $N = |\Phi|$ is the cardinality of the image domain Φ. The key idea behind the KDE method is to smooth out the contribution of each sample $\mathbf{x}$ by spreading it to a certain area in $\mathbb{R}^D$ and with a certain shape as defined by the kernel K. There is a number of choices for the kernel. Nonetheless, the most commonly used and the most suitable is a multidimensional Gaussian function with zero mean and standard deviation σ_k. Using a Gaussian kernel, (4.3) can be rewritten as

$$\tilde{P}(\mathbf{w}(\mathbf{y})) = \frac{1}{N\Gamma} \sum_{x \in \Phi} \exp(-\frac{d^2(\mathbf{w}(\mathbf{y}), \mathbf{w}(\mathbf{x}))}{2\sigma_k^2}), \qquad (4.4)$$

where h has been replaced by the standard deviation σ_k and Γ is a proper constant such that the estimated probabilities are taken from an actual PDF. Having defined the probability of a codeword, we can define the saliency measure as follows:

$$f_{CAKE}(\mathbf{y}) = -\log(\frac{1}{N\Gamma} \sum_{x \in \Phi} \exp(-\frac{d^2(\mathbf{w}(\mathbf{y}), \mathbf{w}(\mathbf{x}))}{2\sigma_k^2})). \qquad (4.5)$$

In this case, context-aware keypoints will correspond to local maxima of f_{CAKE} that are above a given threshold T.

To complete the description of the proposed method, we have to define the distance measure d and set a proper value to σ_k. Due to the relevance of these two parameters in the process of estimating the PDF, we have decided to discuss them in two separate subsections (4.2.3 and 4.2.4). Nonetheless, the KDE has an inherent and significant drawback: its computational cost. To estimate the probability of a pixel, we have to compute (4.4), which means N distances between codewords, giving a computational cost of $\mathcal{O}(N^2)$ for the whole image. The computational complexity of the KDE is prohibitive for images, where N is of the order of millions. Different methods have been proposed to reduce the computation of a KDE-based PDF. Many methods rely on the hypothesis that the sample distribution forms separated clusters, so that it is feasible to approximate the probability in a certain location of the multivariate space using a reduced set of samples. Other methods have been devised for the purpose of a Parzen classifier, so that the cardinality of the training sample is reduced, without changing significantly the performance of the reduced Parzen classifier. In our case, none of the two aforementioned strategies can be straightforwardly used since (i) we cannot assume that the multivariate distribution forms different clusters, and (ii) we do not have ground truth labels to use the same strategy as the one defined for Parzen classifiers. We propose an efficient method that reduces the number of samples by approximating the full $\mathcal{O}(N^2)$ PDF in (4.4) with a $\mathcal{O}(N \log N)$ algorithm. A detailed explanation of the speed-up method can be found in 4.2.5.

4.2.3 *The distance* d

To completely define a KDE-based approach, we have to define (i) the distance d, (ii) the kernel K, and (iii) the bandwidth h. These three parameters are interrelated since they will form the final "shape" of the kernel. As for the distance function d, we consider the Mahalanobis distance:

$$d(\mathbf{w}(\mathbf{x}), \mathbf{w}(\mathbf{y})) = \sqrt{(\mathbf{w}(\mathbf{x}) - \mathbf{w}(\mathbf{y}))^T \Sigma_W^{-1} (\mathbf{w}(\mathbf{x}) - \mathbf{w}(\mathbf{y}))}, \qquad (4.6)$$

where $W = \bigcup_{\mathbf{x} \in \Phi} \mathbf{w}(\mathbf{x})$ and Σ_W is the covariance matrix of W. By using this distance, any affine covariant codeword will provide an affine invariant behavior to the extractor. In other words, any affine transformation will preserve the order of P. This result is summarized in the following theorem:

Theorem 4.1. *Let* $\mathbf{w}^{(1)}$ *and* $\mathbf{w}^{(2)}$ *be codewords such that* $\mathbf{w}^{(2)}(\mathbf{x}) = T(\mathbf{w}^{(1)}(\mathbf{x})))$*, where* T *is an affine transformation. Let* $P^{(1)}$ *and* $P^{(2)}$ *be the probability maps of* $\mathbf{w}^{(1)}$ *and* $\mathbf{w}^{(2)}$*, i. e.,* $P^{(i)}(\cdot) = P(\mathbf{w}^{(i)}(\cdot))$*,* $i = 1, 2$*. In this case,*

$$P^{(2)}(\mathbf{x}_l) \leqslant P^{(2)}(\mathbf{x}_m) \iff P^{(1)}(\mathbf{x}_l) \leqslant P^{(1)}(\mathbf{x}_m), \forall \mathbf{x}_l, \mathbf{x}_m \in \Phi.$$

Proof. Let us suppose that $P^{(2)}(\mathbf{x}_l) \leqslant P^{(2)}(\mathbf{x}_m)$ (the reasoning will be analogous if we consider the other inequality). From the definition of probability, we have

$$\sum_{j=1}^{N} \exp\left(-\frac{(\mathbf{w}^{(2)}(\mathbf{x}_l) - \mathbf{w}^{(2)}(\mathbf{x}_j))^T \Sigma_{W^{(2)}}^{-1} (\mathbf{w}^{(2)}(\mathbf{x}_j) - \mathbf{w}^{(2)}(\mathbf{x}_l))}{2\sigma_k^2}\right) \leqslant$$
$$\leqslant \sum_{j=1}^{N} \exp\left(-\frac{(\mathbf{w}^{(2)}(\mathbf{x}_m) - \mathbf{w}^{(2)}(\mathbf{x}_j))^T \Sigma_{W^{(2)}}^{-1} (\mathbf{w}^{(2)}(\mathbf{x}_j) - \mathbf{w}^{(2)}(\mathbf{x}_m))}{2\sigma_k^2}\right).$$

Let A be the matrix that represents the transformation T (we assume no translation). Since $\Sigma_{W^{(2)}} = A\Sigma_{W^{(1)}}A^T$, the numerators from the exponents in the first and second members of the inequality can be rewritten as

$$(A(\mathbf{w}^{(1)}(\mathbf{x}_l) - \mathbf{w}^{(1)}(\mathbf{x}_j)))^T (A\Sigma_{W^{(1)}}A^T)^{-1} (A(\mathbf{w}^{(1)}(\mathbf{x}_j) - \mathbf{w}^{(1)}(\mathbf{x}_l)))$$

and

$$(A(\mathbf{w}^{(1)}(\mathbf{x}_m) - \mathbf{w}^{(1)}(\mathbf{x}_j)))^T(A\Sigma_{W^{(1)}}A^T)^{-1}(A(\mathbf{w}^{(1)}(\mathbf{x}_j) - \mathbf{w}^{(1)}(\mathbf{x}_m)),$$

respectively. By simplifying the previous expressions, we have

$$((\mathbf{w}^{(1)}(\mathbf{x}_l) - \mathbf{w}^{(1)}(\mathbf{x}_j)))^T\Sigma^{-1}_{W^{(1)}}(\mathbf{w}^{(1)}(\mathbf{x}_j) - \mathbf{w}^{(1)}(\mathbf{x}_l)))$$

and

$$((\mathbf{w}^{(1)}(\mathbf{x}_m) - \mathbf{w}^{(1)}(\mathbf{x}_j)))^T\Sigma^{-1}_{W^{(1)}}(\mathbf{w}^{(1)}(\mathbf{x}_j) - \mathbf{w}^{(1)}(\mathbf{x}_m))).$$

Thus,

$$P^{(2)}(\mathbf{x}) = \frac{1}{|\det A|}P^{(1)}(\mathbf{x}), \forall \mathbf{x} \in \Phi.$$

From the hypothesis, we have $P^{(1)}(\mathbf{x}_l) \leqslant P^{(1)}(\mathbf{x}_m)$.

□

4.2.4 *The smoothing parameter* σ_k

A Parzen estimation can be seen as an interpolation method, which provides an estimate of the continuous implicit PDF. It has been shown that, for $N \to \infty$, the KDE converges to the actual PDF (Parzen, 1962). However, when N is finite, the bandwidth h plays an important role in the approximation. In the case of a Gaussian kernel, σ_k is the parameter that accounts for the smoothing strength.

The free parameter σ_k can potentially vanish the ability of the proposed method to adapt to the image context. When σ_k is too large, an over-smoothing of the estimated PDF occurs, canceling the inherent PDF structure due to the image content. If σ_k is too small, the interpolated values between different samples could be low, such that there is no interpolation anymore. We propose a method, in the case of univariate distribution, to determine an optimal sigma $\sigma_k^\star$, aiming at sufficient blurring while having the *highest sharpen* PDF between samples. We use univariate distributions, since we approximate the KDE computations of a D-dimensional multivariate PDF by estimating D separate univariate PDFs (see subsection 4.2.5). From N samples w, we define the optimal σ_k for the given distribution as

$$\sigma_k^\star = \underset{\sigma>0}{\operatorname{argmax}} \int_{w_i}^{w_{i+1}} \frac{1}{\sqrt{2\pi}\sigma} \left| \frac{d\left(\exp(\frac{-(w-w_i)^2}{2\sigma^2}) + \exp(\frac{-(w-w_{i+1})^2}{2\sigma^2})\right)}{dw} \right| dw, \qquad (4.7)$$

where w_i and w_{i+1} is the farthest pair of consecutive samples in the distribution. It can be shown that, by solving (4.7), we have $\sigma_k^\star = |w_i - w_{i+1}|$. It can be also demonstrated that for $\sigma < |w_i - w_{i+1}|/2$, the estimated PDF between the two samples is concave, which provides insufficient smoothing. Using $\sigma_k^\star$ as defined above, we assure that we have *sufficient blurring* between the two farthest samples, while, at the same time, providing the *highest sharpen* PDF.

4.2.5 *Reduced KDE*

As shown by Theorem 4.1, applying an affine transformation to the codewords does not change the result of the extractor. We take advantage of this, and perform a principal component analysis (PCA) to obtain a new codeword distribution W_P, where elements are denoted by $\mathbf{w}_P(\mathbf{x})$. In this case, the inverse of the covariance matrix $\Sigma_{W_P}^{-1}$ is a diagonal matrix, where the elements on the diagonal contain the inverse of the variance of every variable of W_P. Consequently, we can rewrite the Gaussian KDE in (4.4), using the Mahalanobis distance $d(\cdot,\cdot)$, as another Gaussian KDE with Euclidean distance:

$$\tilde{p}(\mathbf{w}_P(\mathbf{y})) = \frac{1}{N\Gamma}\sum_{x\in\Phi}\exp(-\frac{\sum_{i=1}^{D} a_i(w_{P,i}(\mathbf{y}) - w_{P,i}(\mathbf{x}))^2}{2\sigma_k^2}), \tag{4.8}$$

where $a_i = \sqrt{\Sigma_{W_P}^{-1}(i,i)}$, i.e.., the square root of the i^{th} diagonal element of the inverse of covariance matrix. Equation (4.8) can be rewritten as

$$\tilde{p}(\mathbf{w}_P(\mathbf{y})) = \frac{1}{N\Gamma}\sum_{x\in\Phi}\prod_{i=1}^{D}\exp(-\frac{a_i(w_{P,i}(\mathbf{y}) - w_{P,i}(\mathbf{x}))^2}{2\sigma_k^2}). \tag{4.9}$$

By assuming that each dimension i provides a PDF that is independent of other dimensions, (4.9) can be approximated as follows:

$$\tilde{p}(\mathbf{w}_P(\mathbf{y})) \simeq \frac{1}{N\Gamma}\prod_{i=1}^{D}\sum_{x\in\Phi}\exp(-\frac{a_i(w_{P,i}(\mathbf{y}) - w_{P,i}(\mathbf{x}))^2}{2\sigma_k^2}) \simeq \frac{1}{N\Gamma}\prod_{i=1}^{D}\tilde{p}_i(w_{P,i}(\mathbf{y})). \tag{4.10}$$

Note that this approximation is only valid if PCA is able to separate the multivariate distribution into independent univariate distributions. This is not always verified. However, the proposed approximation works sufficiently well for convex multivariate distributions, which is the case in all the experiments we have conducted in this chapter. Therefore, we have to compute D one-dimensional KDEs $\tilde{p}_i(w_{P,i}(\mathbf{y}))$, using the Euclidean distance, which reduces a multivariate KDE to D univariate problems. This step simplifies the computation of distances between codewords, but still does not reduce the number of basic product-sum computations. Nevertheless, we can approximate the D one dimensional KDEs to speed-up the process. The fact that we have univariate distributions will be profitably used. For the sake of compactness and clarity, in the next part of the section, we will refer to $\tilde{p}_i(w_{P,i}(\mathbf{y}))$ as $p(w(\mathbf{y}))$. We will also omit the constant $1/N\Gamma$ and the constants a_i.

We can extend the concept of KDE, by giving a weight $\nu(\mathbf{x}) > 0$ to each sample, so that the univariate KDE can be rewritten as a reduced KDE:

$$p_R(w(\mathbf{y})) = \sum_{\mathbf{x} \in \Phi_R} \nu(\mathbf{x}) \exp(-\frac{(w(\mathbf{y}) - w(\mathbf{x}))^2}{2\sigma_k^2}), \tag{4.11}$$

where $\Phi_R \subset \Phi$. This formulation can be seen as a hybrid between a Gaussian KDE and a Gaussian Mixture Model. The former has a large number of samples, all of them with unitary weight and fixed σ_k, while the latter has a few number of Gaussian functions, each one with a specific weight and standard deviation.

The goal of our speed-up method is to obtain a set Φ_R with $|\Phi_R| = N_r \ll N$ samples that approximate the $\mathcal{O}(N^2)$ KDE. The idea is to fuse samples that are close to each other into a new sample that "summarizes" them. Given a desired number of samples N_R, the algorithm progressively fuses pairs of samples that have a minimum distance (see Algorithm 6).

Algorithm 6 Speed-up method

```
1: $\Phi_R \leftarrow \Phi$
2: $\nu(\mathbf{x}) \leftarrow 1, \forall \mathbf{x} \in \Phi$
3: **while** $|\Phi_R| > N_R$ **do**
4:     $\{\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1\} \leftarrow \underset{\mathbf{x}_0, \mathbf{x}_1 \in \Phi_R, \mathbf{x}_0 \neq \mathbf{x}_1}{\operatorname{argmin}} |w(\mathbf{x}_0) - w(\mathbf{x}_1)|$
5:     $\nu(\mathbf{x}_{01}) \leftarrow \nu(\tilde{\mathbf{x}}_0) + \nu(\tilde{\mathbf{x}}_1)$
6:     $w(\mathbf{x}_{01}) \leftarrow \frac{\nu(\tilde{\mathbf{x}}_0) w(\tilde{\mathbf{x}}_0) + \nu(\tilde{\mathbf{x}}_1) w(\tilde{\mathbf{x}}_1)}{\nu(\tilde{\mathbf{x}}_0) + \nu(\tilde{\mathbf{x}}_1)}$
7:     $\Phi_R \leftarrow (\Phi_R \setminus \{\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1\}) \cup \{\mathbf{x}_{\mathbf{01}}\}$
8: **end while**
```

The algorithm uses as input the N samples of the univariate distribution (line 1), giving constant weight 1 to all the samples (line 2). While the number of points is greater than the desired number N_R (line 3 to 8), the algorithm selects the pair of samples that have the minimum distance in the set Φ_R (line 4), and a new sample is created (lines 5 and 6), whose weight ν is the sum of the pair's weights and the value w is a weighted convex linear combination of the previous samples. The two selected samples are then removed by the set Φ_R and replaced by the new one (line 7).

At first sight, the reduction algorithm may appear computationally expensive ($\sim \mathcal{O}(N^3)$), since a minimum distance over N_R^2 pairs of points has to be found. However, $w \in \mathbb{R}$, so that $w(\mathbf{x})$ can be ordered at the beginning of the algorithm (with cost $\mathcal{O}(\lceil N \log N \rceil)$), and the pairs of minimum distance can be computed in N subtractions. Consequently, for each sample $\mathbf{x}$, we have the respective sample at minimal distance $\mathbf{x}_m$ and their distance $d_m(\mathbf{x}) = |w(\mathbf{x}) - w(\mathbf{x}_m)|$. This data can be represented using a self-balancing tree (Koffman & Wolfgang, 2007), allowing us to perform deletion and insertions (line 7), in $\log N$ time. Since the samples are ordered both in terms of $w(\mathbf{x})$ and $d_m(\mathbf{x})$, updating the distances after deletions and insertions can be done in $\mathcal{O}(1)$. Summarizing, we need to perform $2(N - N_r)$ deletions and $N - N_r$ insertions, so that the total cost of the reduction algorithm is proportional to $\lceil N \log N \rceil + 3(N - N_r) \log N$, which is $\mathcal{O}(N \log N)$. The total cost to compute $p_R(w(\mathbf{y}))$ linearly depends on the desired N_R and the number of dimensions D.

To further speed-up the approximation, we can use a reduced number of dimensions $\tilde{D} < D$ such that the first $\tilde{D}$ dimensions of the multivariate distribution W_P cover 95% of the total distribution variance. This is a classical strategy for dimensionality reduction that has provided, in our tests, an average of 3× further speed-up.

4.3 INSTANCES OF THE CONTEXT-AWARE KEYPOINT EXTRACTOR

Different CAKE instances are constructed by considering different codewords. As observed by Gilles (1998) and Kadir & Brady (2001), the notion of saliency is related to rarity. What is salient is rare. However, the reciprocal is not necessarily valid. A highly discriminating codeword will turn every location into a rare structure; nothing will be seen as salient. On the other hand, with a less discriminating codeword, rarity will be harder to find. We present two differential-based instances, which are provided by sufficiently discriminating codewords. The strong link between image derivatives and the geometry of local structures is the main motivation to present two examples of instances based on local differential information.

4.3.1 *[eigSTM]-CAKE*

As seen in Chapter 2, local feature detection based on differential and (implicitly) geometrical information often makes use of the structure tensor matrix, whose spectrum summarily describes the local signal variations along the principal directions. We introduce a plausible CAKE instance, coined as [eigSTM]-CAKE, based on the codeword $\mathbf{w}(\mathbf{x}) = [\lambda_1(\mu(\mathbf{x}))\ \lambda_2(\mu(\mathbf{x}))]^T$, which solely conveys information about the spectral structure of μ, the structure tensor matrix. Under this instance, keypoints will be the locations that show the most improbable local signal changes in orthogonal directions of the image plane. The [eigSTM]-CAKE serves as an introductory example, which, despite its simplicity, can provide a rotation covariant response and a good coverage of relevant image information. Note that a similar instance could have been constructed using the three different components of the structure tensor matrix.

Figure 4.4 depicts the "Needle in a Haystack" image with the overlaid maps representing Shi-Tomasi and [eigSTM]-CAKE saliency measures, as well as the information map provided by the proposed codeword. It is readily seen that our instance provides a better coverage of the most relevant object.

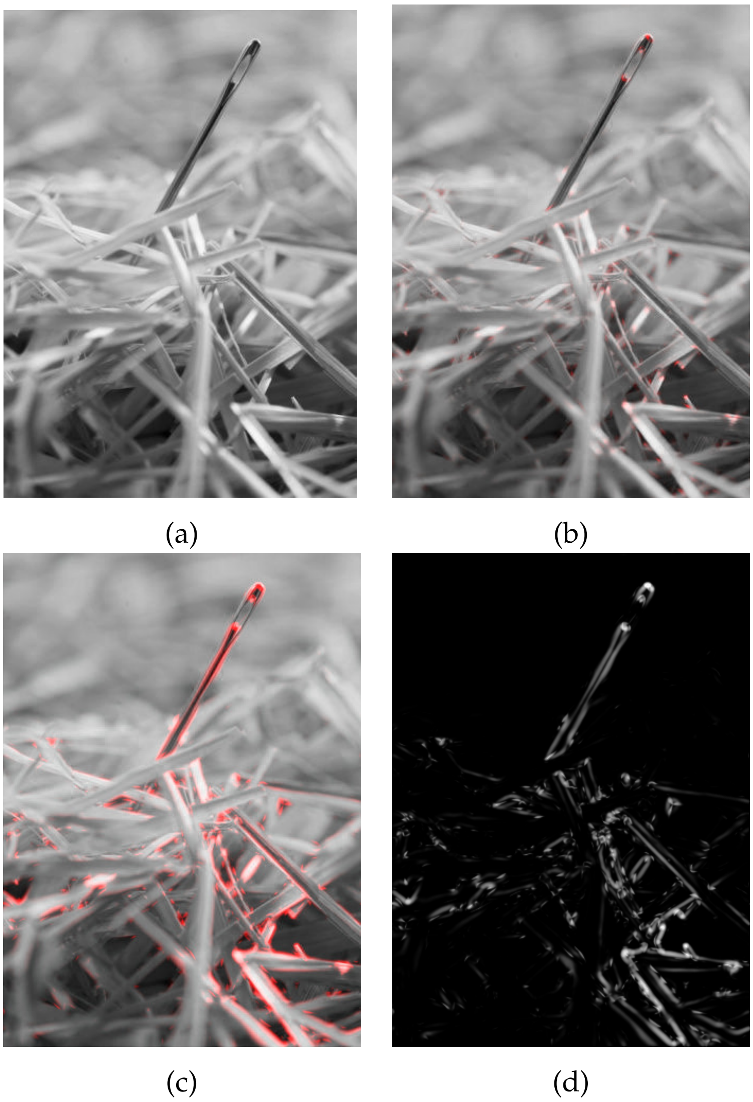


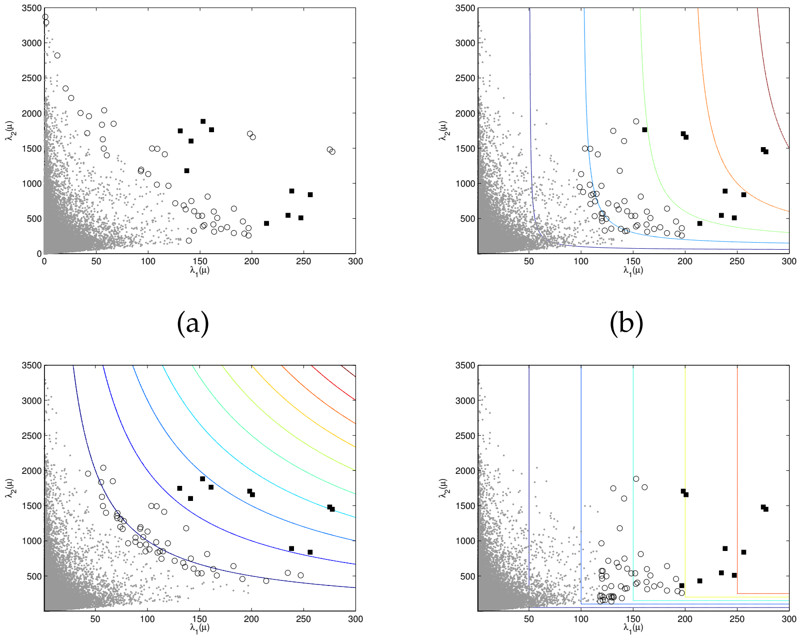
Figure 4.4: Saliency measures as overlaid maps on the "Needle in a Haystack" image: (a) Input image; (b) Shi-Tomasi saliency measure; (c) [eigSTM]-CAKE saliency measure; (d) [eigSTM]-CAKE information map. Best viewed in color.

In Fig. 4.5, we extend the comparison to the other detectors summarized in Table 4.1 by giving a geometrical interpretation of the process of detection. Gray dots show the distribution of the eigenvalues λ_1 and λ_2, while squares and circles indicate respectively the 10 and 60 most salient keypoints. For the CAKE instance, the geometry of the most salient keypoints depends on the data distribution (Fig. 4.5

(a)), while for other methods the geometry of relevant points strictly depends on the respective saliency measure (Fig. 4.5 (b), (c), and (d)).

Table 4.1: Saliency measures for Noble, Rhor, and Shi-Tomasi detectors.

	Noble	Rohr	Shi-Tomasi
f	$\dfrac{\prod_{i=1}^{2} \lambda_i}{\epsilon + \sum_{i=1}^{2} \lambda_i}$	$\prod_{i=1}^{2} \lambda_i$	$\min\{\lambda_1, \lambda_2\}$



(a)

(b)

(c)

(d)

Figure 4.5: Visualizing keypoint detection in the spectrum of μ (without non-maxima suppression): (a) [eigSTM]-CAKE; (b) Noble; (c) Rohr; (c) Shi-Tomasi. Legend: circles – 60 most salient keypoints; squares – 10 most salient keypoints.

4.3.2 *[HES]-CAKE*

Our second instance, which we will refer to as [HES]-CAKE, is based on the Hessian matrix. The idea is to efficiently describe local shape characteristics by means of second order derivatives, which will be computed at multiple scales to produce robustness to scale changes. The use of second order derivatives will allow us to capture structures that carry most image information, such as blobs, as well as structures where the fine details of image can be found, i. e., lines and edges (Lillholm et al., 2003; Dickscheid et al., 2011).

The codeword for the multiscale Hessian-based instance is

$$\begin{aligned}\mathbf{w}(\mathbf{x}) = \Big[\ & \sigma_1^2 L_{xx}(\mathbf{x};\sigma_1) \quad \sigma_1^2 L_{xy}(\mathbf{x};\sigma_1) \quad \sigma_1^2 L_{yy}(\mathbf{x};\sigma_1) \\ & \sigma_2^2 L_{xx}(\mathbf{x};\sigma_2) \quad \sigma_2^2 L_{xy}(\mathbf{x};\sigma_2) \quad \sigma_2^2 L_{yy}(\mathbf{x};\sigma_2) \\ & \dots \\ & \sigma_M^2 L_{xx}(\mathbf{x};\sigma_M) \quad \sigma_M^2 L_{xy}(\mathbf{x};\sigma_M) \quad \sigma_M^2 L_{yy}(\mathbf{x};\sigma_M)\ \Big]^T,\end{aligned} \tag{4.12}$$

where L_{xx}, L_{xy}, and L_{yy} are the second order partial derivatives of L, a Gaussian smoothed version of the image, and σ_i, with $i = 1, \dots, M$, represents the scale.

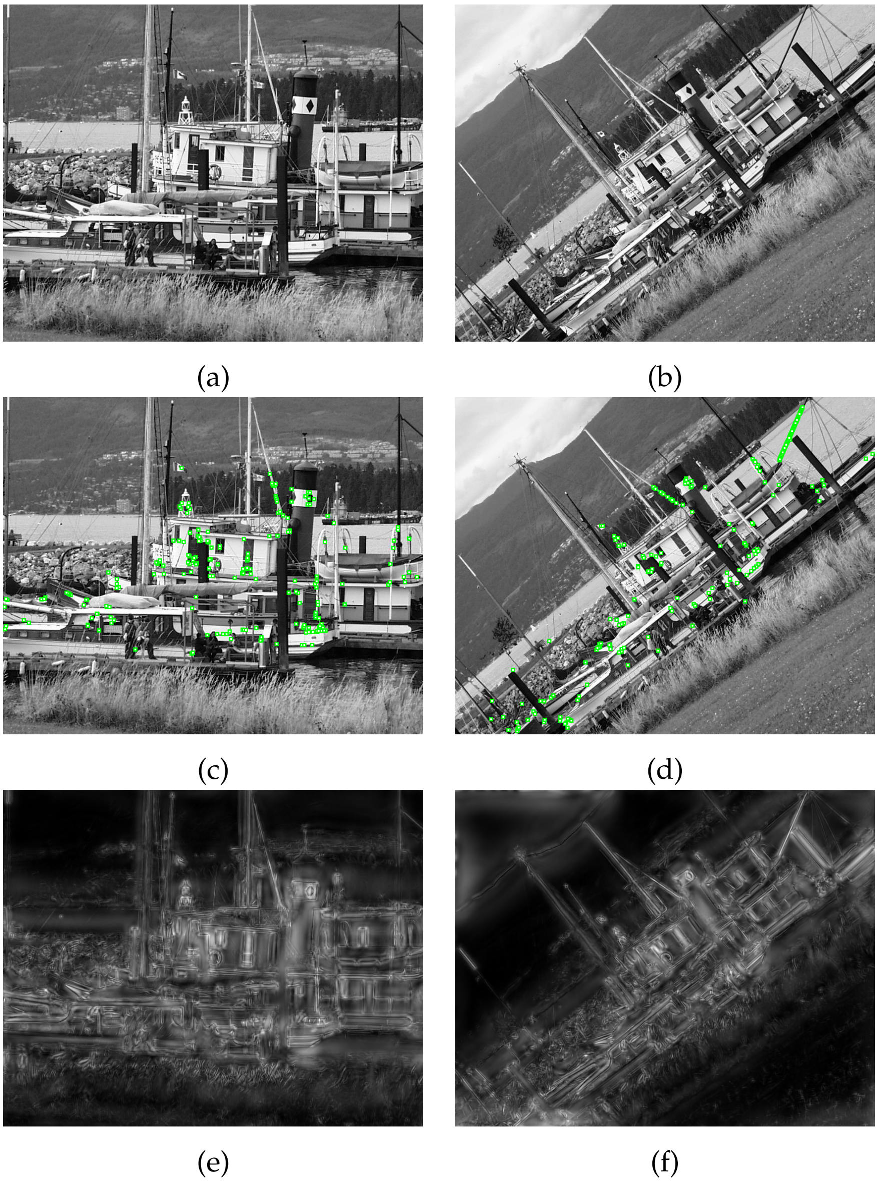
In Fig. 4.6, we show [HES]-CAKE detection (200 keypoints) on the first and third images of the Boat sequence (Oxford dataset). These relatively few features are mainly concentrated in the foreground object, and they are either part of blobs or edges.

4.4 EXPERIMENTAL VALIDATION AND DISCUSSION

We evaluated and compared the performance of the proposed CAKE instances using three criteria: completeness, complementarity, and repeatability. In the context of feature-based robust image representation, completeness and complementarity appear as crucial criteria. Although the presence of repeatable features is not a fundamental requirement for robust image representation, their existence is advantageous: a robust image representation without repeatable features provides an unpredictable coverage of the image content when in the presence of image deformations. Furthermore, a repeatable set of features allows the detector to be used in a wider range of application domains.

[HES]-CAKE is designed to provide a better coverage of informative content. Furthermore, [eigSTM]-CAKE is an instance which has a covariant response to rotations, while the Hessian-based one combines such covariance with robustness to scale changes. Given these differences in terms of robustness and completeness, a special emphasis was given to the Hessian-based instance in our experimental validation.

We followed the evaluation protocol proposed by Dickscheid et al. (2011) to measure the completeness and the complementarity of features. The metric for completeness is based on local statistics, which totally excludes the bias in favor of our context-aware features, as our algorithm is based on the analysis of the codeword distribution



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.6: [HES]-CAKE information maps for the first and third images of the Boat sequence: (a) first image; (b) third image; (c) and (d) [HES]-CAKE features (200 most salient keypoints); (e) and (f) information maps.

over the whole image. In fact, this evaluation gives a hint on the quality of the trade-off between the context-awareness and the locality of context-aware features. However, it does not provide a hint on how features cover informative content within the image context. If we take the "Needle in a Haystack" image depicted in Fig. 4.2 as an example, we can claim that strictly local features can show high completeness scores without properly covering the most interesting object in the scene. Note that such image representation, despite its considerable robustness, might be ineffectual if the goal is to recognize the salient object. Therefore, for a better understanding of the performance of our method, we complemented the completeness analysis with a qualitative evaluation of context-awareness.

4.4.1 *Repeatability evaluation*

4.4.1.1 *[eigSTM]-CAKE repeatability evaluation*

We evaluated the repeatability of [eigSTM]-CAKE features on a dataset containing 5 axial slices of synthetic T1 MRI of the human brain[2] with a size of 181×217 pixels and other images that are the result of applying different levels of Gaussian noise (5% and 7%) or rotation (2° and 5°) to the original slices. To provide an acceptable coverage of the brain anatomy, we selected slices #25, #50, #100, and #150 from the original database. Figure 4.7 depicts some of the test images. For comparison, implementations of the keypoint extractors mentioned in Table 4.1 were included in the evaluation. The derivation and integrations scales were set to 1.5 and 3, respectively. To estimate the information maps, we used 200 samples.

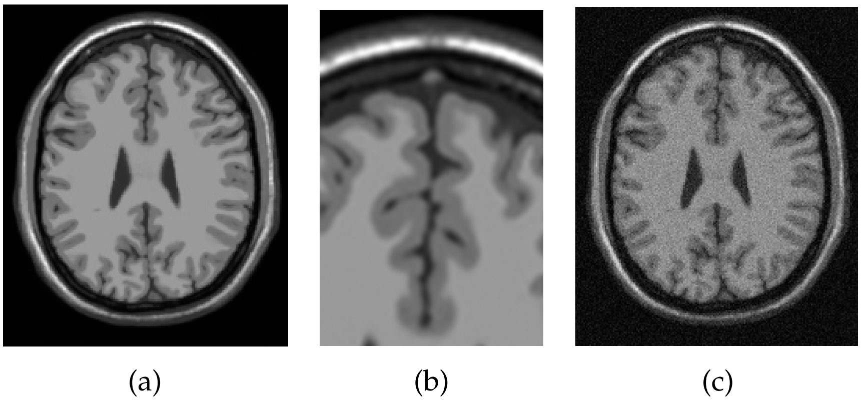


Figure 4.7: BrainWeb test Images: (a) normal slice; (b) a region of interest in (a); (c) noisy slice (7%).

[2] Available at `http://www.bic.mni.mcgill.ca/brainweb/`.

The repeatability rates for the first 20 and 100 keypoints within a 1.5-neighborhood were conjointly computed with the mean localization error. These results are outlined in Tables 4.2–4.5. The asterisk denotes that the difference between the indicated result and the result of the proposed method is statistically significant (Wilcoxon rank-sum test). Figure 4.8 contains specific plots of repeatability and mean localization error as a function of the number of detected points for slice #100 with 7% of noise.

Table 4.2: Repeatability rate results in a 1.5 neighborhood for the 20 most salient keypoints.

	Noise		Rotation	
	5%	7%	2°	5°
[eigSTM]-CAKE	90%±7%	**86.3%**±12.5%	78.8%±4.8%	15%±9.1%
Noble	* *68.8%*± 7.5%	72.5%± 9.6%	* *55%*± 11.6%	*13.8%*± 6.3%
Rohr	85%±9.1%	80%±7%	**82.5%**±9.6%	**28.8%**±7.5%
Shi-Tomasi	**92.5%**±8.7%	83.8%±6.3%	77.5%±6.5%	18.8%± 8.5%

Table 4.3: Repeatability rate results in a 1.5 neighborhood for the 100 most salient keypoints.

	Noise		Rotation	
	5%	7%	2°	5°
[eigSTM]-CAKE	83.5%± 7.9%	76.5%± 11.2%	77.8%± 12%	18.3%±6.4%
Noble	* *63.5%*± 5%	*61.3%*± 6.7%	* *51.5%*± 5%	*14.3%*± 4.2%
Rohr	83.8%±%3.8	76% ± 5%	**88.3%** ± 2.8%	23.3%± 8.2
Shi-Tomasi	**86.8%**±4.1%	**80.3%**±1%	82.5%± 3.7%	**24.3%**±7.5%

For this particular type of images, these detectors show a similar performance, although the Shi-Tomasi algorithm shows a higher repeatability when dealing with noisy images, whereas the Rhor extractor has the preferred performance when in the presence of minor rotations. Our proposed instance shows repeatability rates and localization errors slightly inferior to the Shi-Tomasi algorithm. The differences become more discrepant when more points are considered, which is explained by the existence of repeating structures along the images. The accuracy of our context-aware keypoints is worth of note, as it shows that the proposed method alleviates the sensitivity to noise that characterizes information theoretic-based algorithms,

Table 4.4: Mean localization error results in a 1.5 neighborhood for the 20 most salient keypoints.

	Noise		Rotation	
	5%	7%	2°	5°
[eigSTM]-CAKE	**0.2** ± 0.1	0.4±0.0	0.9±0.0	*1.1*± 0.2
Noble	* *0.6* ± 0.2	* *0.6* ± 0.3	0.9 ± 0.1	**1.0**±0.2
Rohr	**0.2** ±0.1	**0.3**±0.1	0.9±0.0	**1.0**±0.2
Shi-Tomasi	**0.2** ± 0.0	0.4 ± 0.0	0.9 ± 0.0	**1.0** ±0.3

Table 4.5: Mean localization error results in a 1.5 neighborhood for the 100 most salient keypoints.

	Noise		Rotation	
	5%	7%	2°	5°
[eigSTM]-CAKE	**0.3** ± 0.1	**0.5** ± 0.1	0.9 ± 0.0	*1.1* ± 0.0
Noble	* *0.6*±0.1	* *0.7*±0.1	0.9± 0.0	**1.0** ± 0.0
Rohr	0.4±0.1	**0.5**±0.1	0.9±0.0	*1.1* ± 0.1
Shi-Tomasi	**0.3**±0.0	**0.5** ± 0.0	0.9 ± 0.0	*1.1*±0.0

namely when keypoints are used instead of regions (Kadir & Brady, 2001).

4.4.1.2 *[HES]-CAKE repeatability evaluation*

We followed the evaluation protocol proposed by Mikolajczyk et al. (2005) to evaluate the repeatability of [HES]-CAKE features. Since the Hessian-based CAKE instance is mainly a keypoint extractor, we make use of the normalized Laplacian operator, $\nabla^2 L_n = \sigma^2(L_{xx} + L_{yy})$, to determine the characteristic scale for each extracted keypoint, which, in this case, corresponds to the one at which the normalized Laplacian attains an extremum. This scale defines the radius of a circular region centered about the keypoint (Fig. 4.9). The repeatability of regions is computed within an overlap error of 40% ($\epsilon_R = 0.4$).

The Hessian-based CAKE instance does not solely respond to blob-like keypoints. In fact, this instance can capture other structures where scale selection can be less reliable, i. e., edges. Nevertheless, the combination of [HES]-CAKE with the normalized Laplacian operator can provide robustness to scale changes, despite the resulting method not being entirely scale covariant.

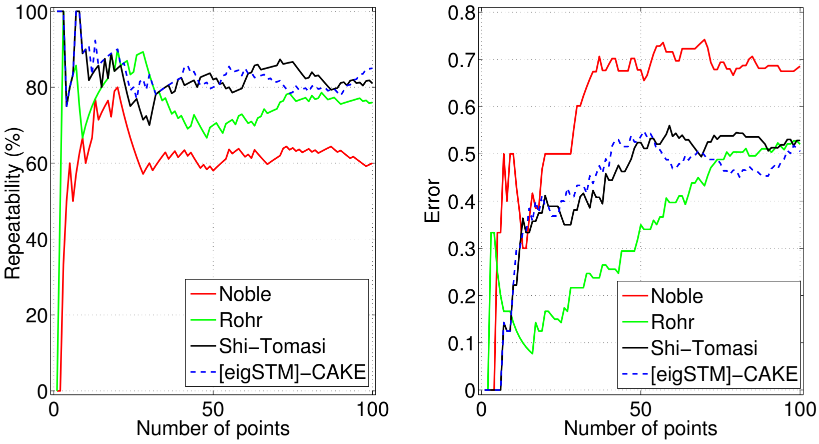


Figure 4.8: Repeatability rate and mean localization error as a function of the number of extracted points for a noisy slice.

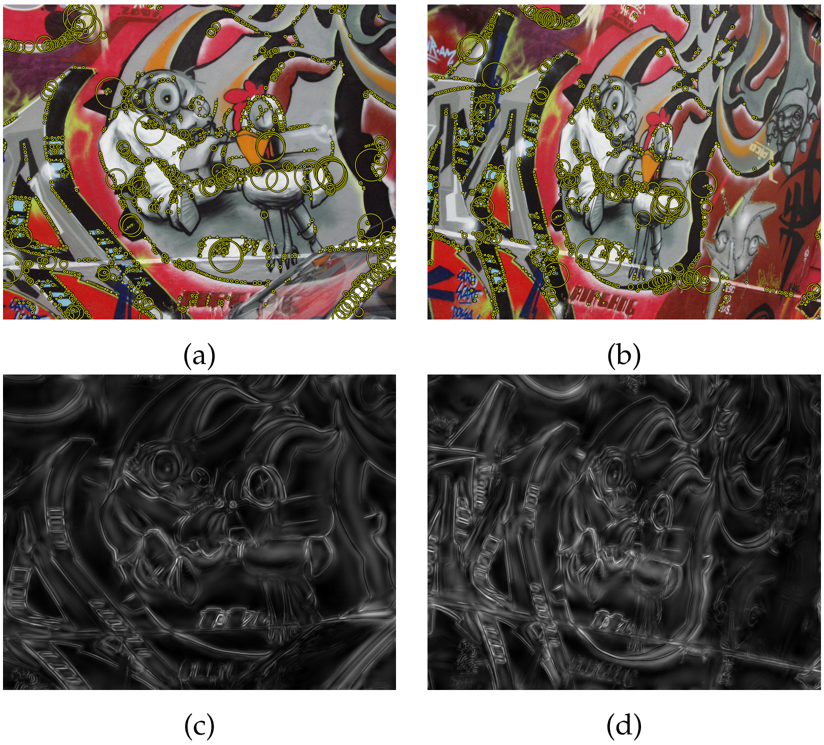

(a)
(b)
(c)
(d)

Figure 4.9: [HES]-CAKE regions extracted from the first and third images from the Graffiti sequence: (a) and (b): extracted regions (from the 1500 most informative keypoints); (c) and (d): information maps.

We compared the repeatability of keypoints retrieved by [HES]-CAKE and the ones given by some of the leading algorithms on scale covariant feature extraction: Hessian-Laplace (HESLAP), Harris-Laplace (HARLAP), the Scale Invariant Feature Operator (SFOP), and the Salient Regions detector (scale covariant version). We also analyzed the repeatability of Maximally Stable Extremal Regions (MSER), a type of affine covariant features that are not derived from keypoints. In all cases, the implementations are the ones given and maintained by the authors and default parameters were used.

A few important remarks should be made about the choice of the algorithms in the comparative study. First, affine covariant region detectors are more appealing than the scale covariant ones. However, since [HES]-CAKE is designed to have a quasi-scale-covariant response, our decision was to compare it directly to detectors showing a similar level of covariance. Moreover, as reported by Mikolajczyk & Schmid (2004), scale covariant methods such as HARLAP and HESLAP yield better repeatability results in the presence of scale change than its affine covariant derivations.

Table 4.6 outlines the parameter settings for [HES]-CAKE. We note that our algorithm retrieves more features than its counterparts. For a fair evaluation of repeatability, we defined a threshold to avoid a considerable discrepancy in the number of features.

Table 4.6: Parameter settings for [HES]-CAKE.

[HES]-CAKE	
Number of scales	12
t_{i+1}/t_i (ratio between successive scale levels)	1.19
t_0 (initial scale)	1.4
Non-maximal suppression window	3×3
T (threshold)	12 (or 3000 keypoints)
σ_k	optimal
N_R (number of samples)	200

Figures 4.10 to 4.17 depict the relative and absolute repeatability of regions for the different sequences, and Figure 4.18 gives a summarized version of these results. Figures 4.10 to 4.17 also include the repeatability score for the third image with respect to the first one as a function of the overlap error, which will give us an idea of the accuracy of the detectors. The computation of repeatability only takes into account the regions in common parts between the images. Among scale covariant features, HESLAP regions exhibit a slightly better overall repeatability score, namely in well-structured scenes (e. g., Bikes) where blob-like features are more present and well-defined. HARLAP has a similar performance, yielding the most

repeatable results in textured scenes. The repeatability scores of SFOP and [HES]-CAKE are similar, yet the latter responds to a higher number of features. Aside from viewpoint changes, the repeatability of MSER tends to be lower than its counterparts. In a direct comparison of information theoretic-based methods, we observe that [HES]-CAKE features are more repeatable than Salient Regions. The only two exceptions to this observation are the results for Trees and Wall sequences. Such results are explained by the fact that both sequences depict highly textured scenes, providing denser sets of Salient Regions. As for scale changes (Boat and Bark sequences), [HES]-CAKE regions show a sufficiently robust behavior. In the case of the Bark sequence, only HESLAP features are more repeatable than the proposed regions.

In terms of accuracy, we observe a similar performance among all methods, although Salient Regions are less accurate in sequences such as Wall, Trees, or UBC. This is explained by the presence of a higher number of Salient Regions in these sequences, which yields a higher number of correspondences for larger overlap errors. Note that in the original evaluation performed by Mikolajczyk et al. (2005), the authors decided to use a reduced number of Salient Regions. As a result, these features showed a lower repeatability for the default overlap error as well as lower repeatability variations as the overlap error was changed, i. e., they showed a higher accuracy. Here, we considered a higher number of Salient Regions in order to make this cardinality comparable to the one of [HES]-CAKE features.

4.4.2 *Completeness and complementarity evaluation*

Six of the seven image categories used by Dickscheid et al. (2011) in the original evaluation were also used in our evaluation. The categories are the ones depicted in Fig. 3.8. The seventh category, which is comprised of different cartoon images, was not made publicly available and, therefore, it was not included in our dataset.

The cardinality of the sets influences the completeness scores, as sparser sets tend to be less complete. While it is interesting to analyze the completeness of sets with comparable sparseness, one cannot expect similar cardinalities when dealing with different features types. We took such facts into consideration and, as a result, we performed two different tests. The first one corresponds to the main completeness test, which does not restrict the number of features. The second one allows us to make a direct comparison between our method and the Salient Regions algorithm by using the same number of features. Let $\mathcal{F}_{\mathrm{[HES]-CAKE}}(I)$ and $\mathcal{F}_{\mathrm{Salient}}(I)$ be the respective

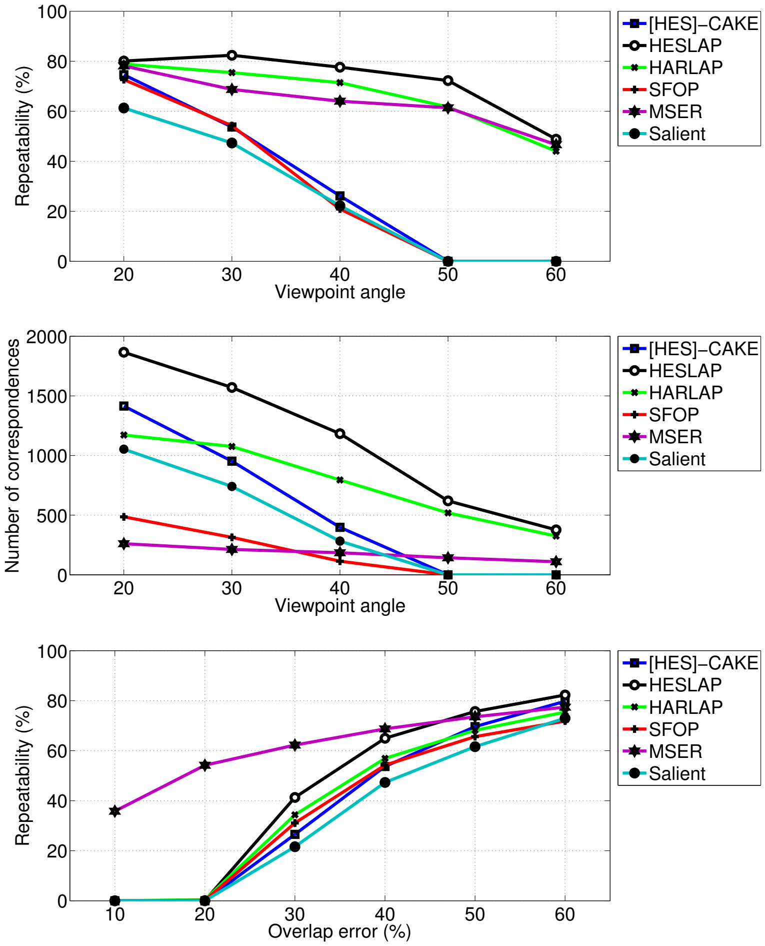


Figure 4.10: Repeatability results for the Graffiti sequence (viewpoint change). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

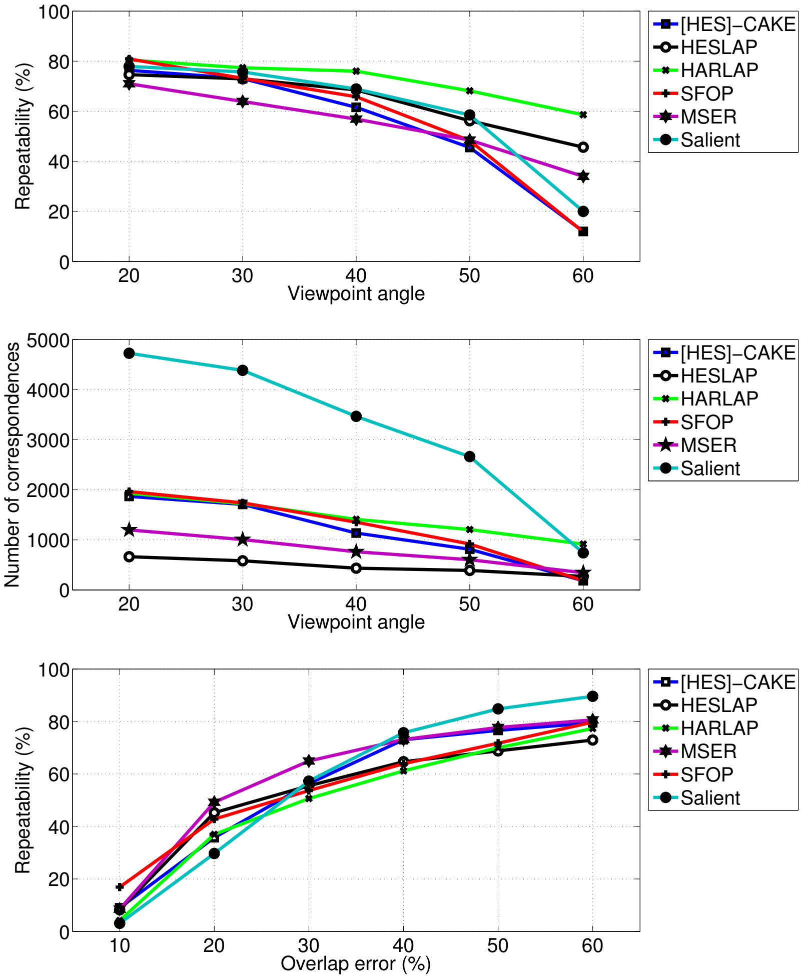


Figure 4.11: Repeatability results for the Wall sequence (viewpoint change). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

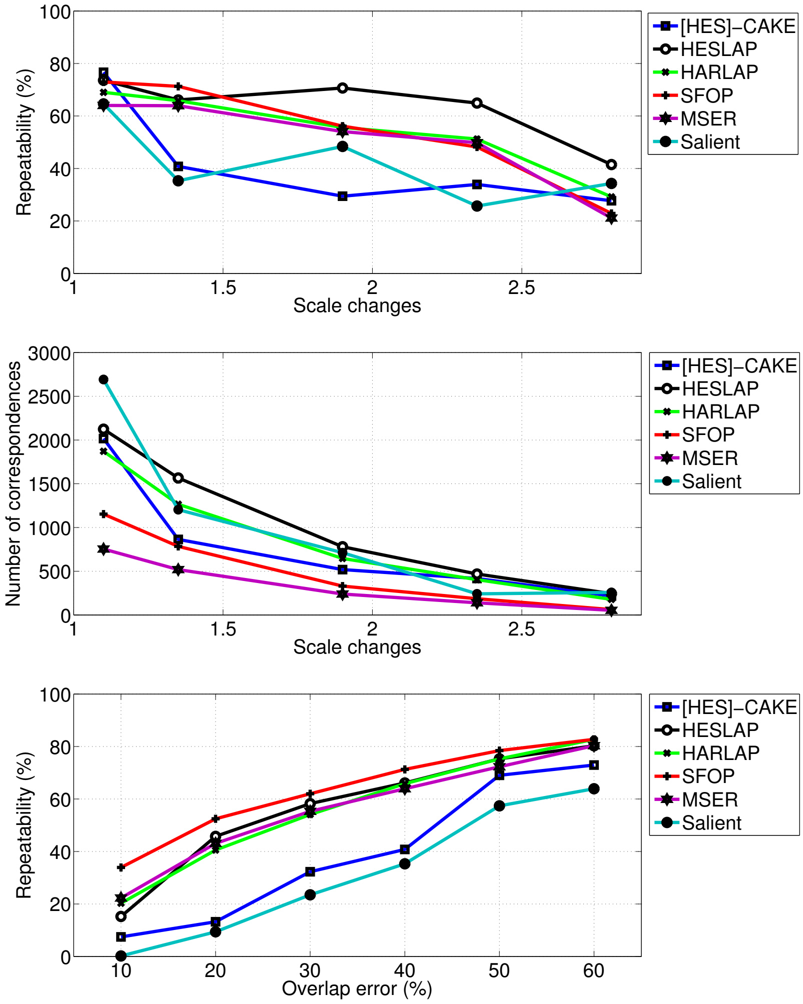


Figure 4.12: Repeatability results for the Boat sequence (scale change). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

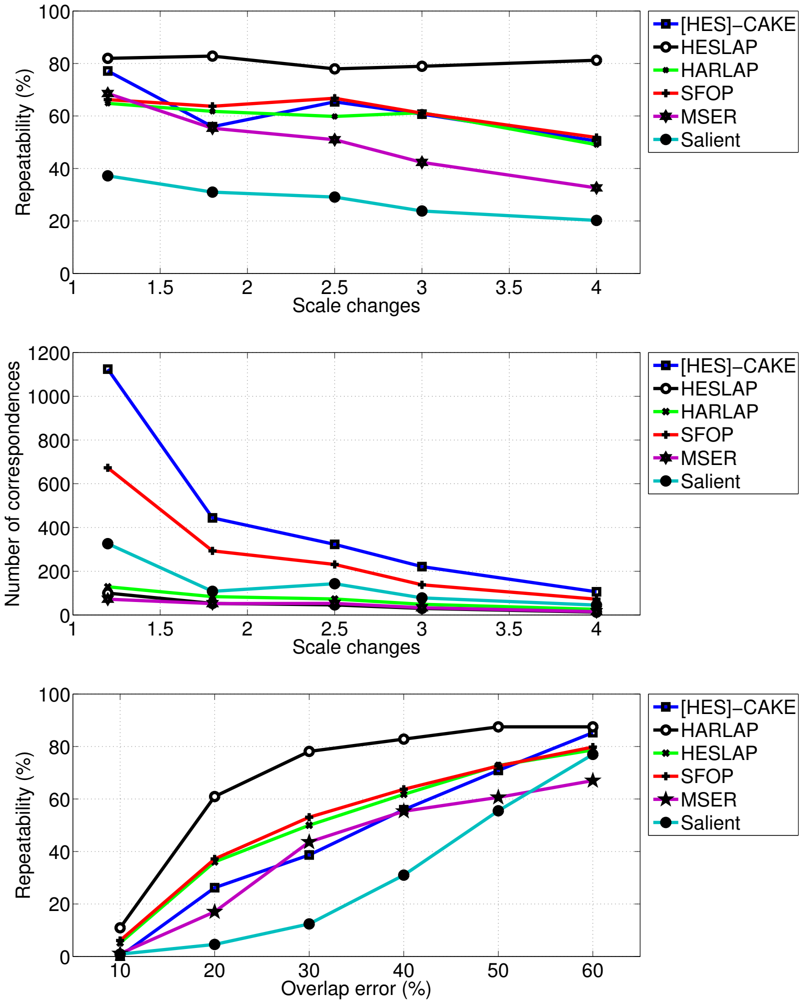


Figure 4.13: Repeatability results for the Bark sequence (scale change). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

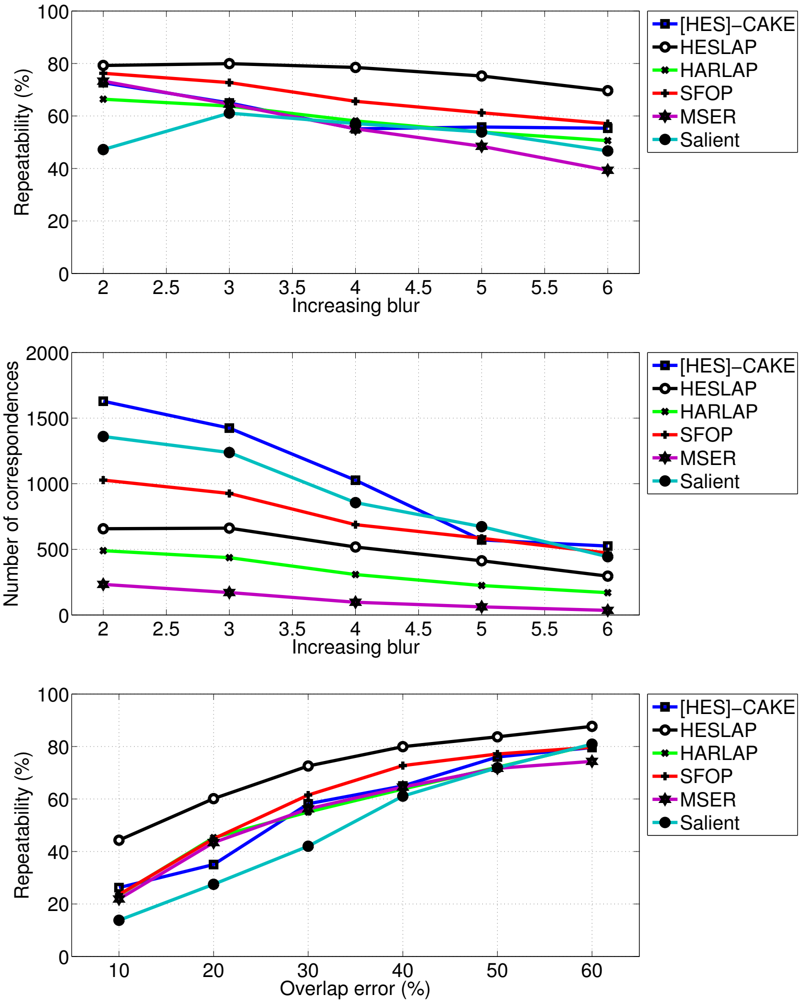


Figure 4.14: Repeatability results for the Bikes sequence (increasing blur). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

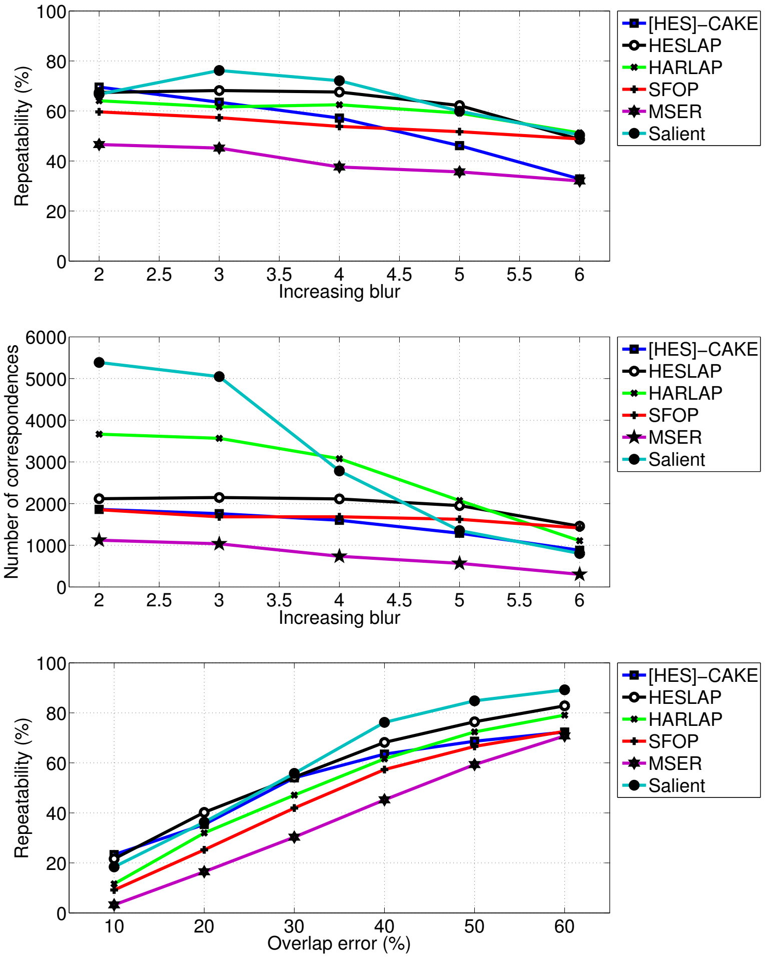


Figure 4.15: Repeatability results for the Trees sequence (increasing blur). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

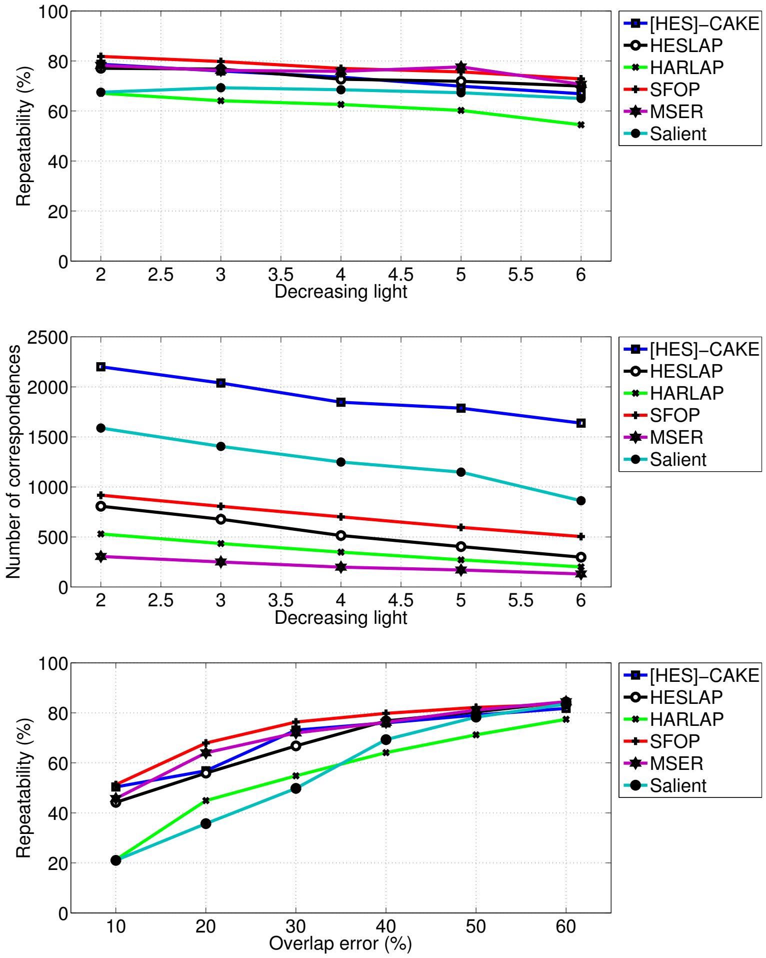


Figure 4.16: Repeatability results for the Leuven sequence (decreasing light). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

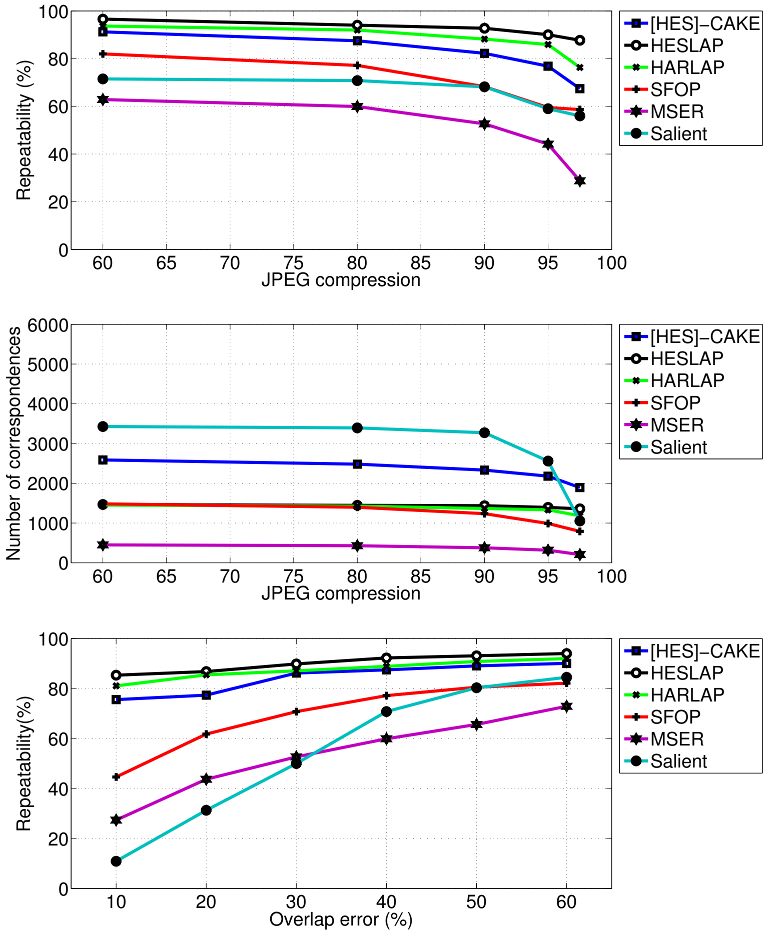


Figure 4.17: Repeatability results for the UBC sequence (JPEG compression). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

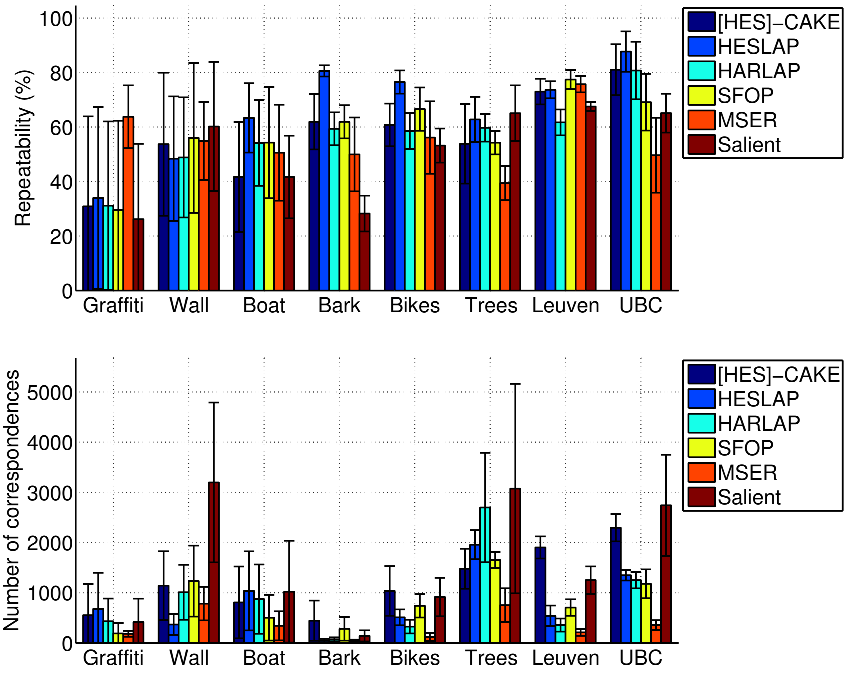


Figure 4.18: Repeatability score and number of correspondences with an overlap error of 40% for the Oxford dataset. Top row: average repeatability (error bars indicate the standard deviation). Bottom row: average number of correspondences (error bars indicate the standard deviation).

sets of [HES]-CAKE regions and Salient Regions extracted from an image I. From each set, we extract the n highest ranked features (both methods provide a well-defined hierarchy among features), where $n = \min\{|\mathcal{F}_{[HES]-CAKE}(I)|, |\mathcal{F}_{Salient}(I)|\}$.

The parameter settings for [HES]-CAKE are outlined in Table 4.7. We used only 3 scales to reduce the difference between the number of [HES]-CAKE features and the number of regions retrieved by its counterparts. Note that [HES]-CAKE already responds to complementary features, yielding feature sets with a higher cardinality. However, for a more insightful study of the complementarity between [HES]-CAKE features and the remaining ones, it is fundamental that the completeness of [HES]-CAKE features is not maximal. By using this reduced number of scales, we achieve such trade-off. For the same reason, we only considered 50 % of the regions detected by [eigSTM]-CAKE, whose parameter settings are outlined in Table 4.8.

Table 4.7: Parameter settings for [HES]-CAKE.

[HES]-CAKE	
Number of scales	3
t_{i+1}/t_i (ratio between successive scale levels)	1.19
t_0 (initial scale)	1.4
Non-maximal suppression window	3×3
T (threshold)	none
σ_k	optimal
N_R (number of samples)	200

Table 4.8: Parameter settings for [eigSTM]-CAKE.

[eigSTM]-CAKE	
σ_D (derivation scale)	1.5
σ_I (integration scale)	3
Non-maximal suppression window	3×3
T (threshold)	(50% of points)
σ_k	optimal
N_R (number of samples)	200

Figure 4.19 is a summary of the main completeness evaluation. Results are shown for each image category, in terms of the distance $d_H(p_H, p_c)$. The plot includes the line $y = \sqrt{\frac{1}{2}}$, which corresponds to an angle of 90 degrees between $\sqrt{p_H}$ and $\sqrt{p_c}$. For a better interpretation, the average number of features per category is also shown.

Regardless of the image collection, [HES]-CAKE retrieves more features than the other algorithms, which contributes to achieve the best completeness scores. The exception is the Brodatz category, which essentially contains highly textured images. For this category, Salient Regions achieve a better completeness score despite the lower number of regions. We also observe that the performance of [eigSTM]-CAKE is comparable to other methods with higher sparseness. This is mainly due to the fact that this instance performs a single-scale extraction.

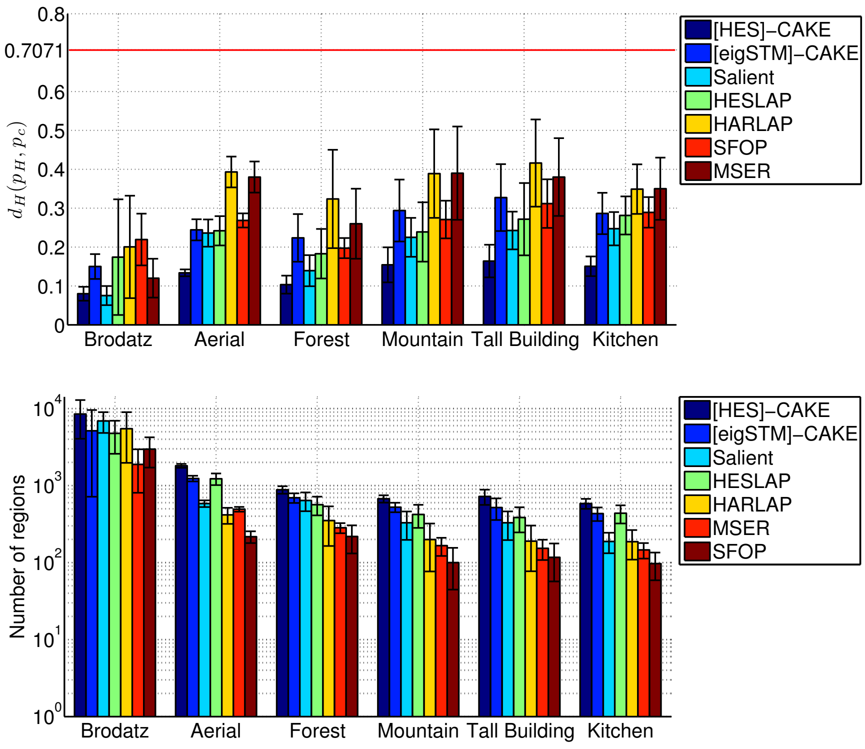


Figure 4.19: Completeness results. Top row: average dissimilarity measure $d_H(p_H, p_c)$ for the different sets of features extracted over the categories of the dataset (error bars indicate the standard deviation). Bottom row: average number of extracted features per image category (error bars indicate the standard deviation).

The additional test computes the completeness scores of context-aware regions and Salient Regions for the first 20 images in each category using the same number of features. The results are summarized in Fig. 4.20. Here, Salient Regions achieve better results. However, the difference between scores is not significant. Aerial and Kitchen are the categories where context-aware features exhibit the lowest scores. This is explained by the strong presence of homogeneous regions, which might be part of salient objects within the image context, such as roads, rooftops (Aerial category), home appliances, and furniture

(Kitchen category).

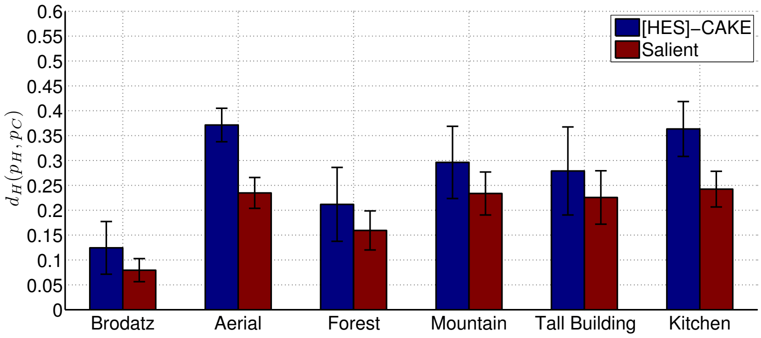


Figure 4.20: Average dissimilarity measure $d_H(p_H, p_c)$ for the different sets of features extracted over the categories of the dataset (20 images per category).

Complementarity was also evaluated on the first 20 images of each category by considering combinations of two different feature types. The results are summarized in Table 4.9. As expected, any combination that includes [HES]-CAKE regions achieves the best completeness scores. We give particular emphasis to the complementarity between HESLAP and [HES]-CAKE: both methods are Hessian-based and yet they produce complementary regions. The combination of [HES]-CAKE and Salient Regions is also advantageous: the latter provides a good coverage of "busy" parts composed of repetitive patterns.

4.4.3 *Context-awareness evaluation*

For a qualitative evaluation of the context-awareness of [HES]-CAKE regions, we used three images typically used in the validation of algorithms for visual saliency detection (e. g., Goferman et al., 2012). Each one of the test images shows a salient object over a background containing partially salient elements. Figures 4.21 and 4.22 depict the test images, the corresponding information maps given by the CAKE instance, as well as the coverage provided by context-aware regions when 100 and 250 points are used. These results were obtained using the parameter settings outlined in Tables 4.7 and 4.8. In all cases, our algorithm succeeds in covering distinctive elements of the salient objects. However, [eigSTM]-CAKE regions cover the fine details of these elements, while [HES]-CAKE features not only cover these elements. In fact, with 250 [HES]-CAKE regions, the coverage becomes a relatively robust image representation for all cases.

Table 4.9: Average dissimilarity measure $d_H(p_H, p_c)$ for different sets of complementary features (20 images per category).

[HES]-CAKE	[eigSTM]-CAKE	HESLAP	HARLAP	SFOP	MSER	SALIENT	Brodatz	Aerial	Forest	Mountain	Tall building	Kitchen	Overall
■	■						0.078	0.123	0.11	0.189	0.161	0.17	0.138
■		■					0.062	0.113	0.085	0.127	0.116	0.114	0.103
■			■				0.078	0.122	0.106	0.156	0.145	0.139	0.124
■				■			0.074	0.12	0.105	0.157	0.147	0.145	0.125
■					■		0.068	0.122	0.102	0.164	0.149	0.147	0.125
■						■	0.063	0.123	0.101	0.15	0.142	0.149	0.121
	■	■					0.096	0.181	0.169	0.23	0.196	0.19	0.177
	■		■				0.116	0.22	0.228	0.29	0.26	0.263	0.229
	■			■			0.124	0.189	0.172	0.259	0.248	0.255	0.208
	■				■		0.102	0.222	0.213	0.309	0.262	0.276	0.231
	■					■	0.078	0.168	0.137	0.224	0.183	0.21	0.167
		■	■				0.117	0.245	0.209	0.252	0.259	0.23	0.219
		■		■			0.113	0.194	0.147	0.187	0.202	0.193	0.173
		■			■		0.093	0.239	0.185	0.24	0.223	0.21	0.198
		■				■	0.062	0.168	0.115	0.168	0.155	0.158	0.138
			■	■			0.134	0.251	0.195	0.247	0.275	0.263	0.227
			■		■		0.113	0.348	0.278	0.357	0.323	0.317	0.29
			■			■	0.077	0.214	0.147	0.215	0.195	0.211	0.177
				■	■		0.105	0.232	0.17	0.235	0.24	0.249	0.205
				■		■	0.076	0.174	0.13	0.187	0.181	0.181	0.16
					■	■	0.069	0.208	0.148	0.219	0.209	0.22	0.179

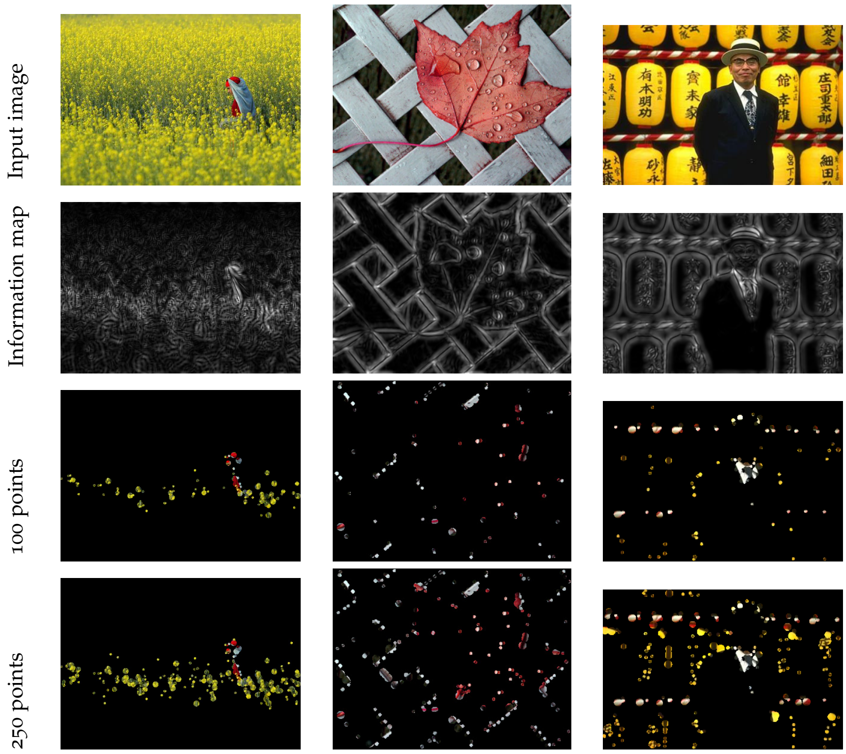


Figure 4.21: [HES]-CAKE information maps and extraction results in terms of coverage.

4.5 OBJECT CLASSIFICATION: A PLAUSIBLE APPLICATION

Context-aware features represent a viable solution to the problem of providing a robust image representation through the use of local features. As shown in §4.4, our context-aware method provides complete sets of features. In addition, the repeatability and accuracy of these features is comparable with the ones retrieved by state-of-the art algorithms. Hence, these results corroborate the suitability of context-aware features for tasks mainly requiring the robust and compact image representation, such as recognition/classification tasks.

We illustrate the application of context-aware features with an object classification problem. The goal is to classify objects (or scenes) into one of the different categories by training and testing on a fixed number of images from each category of a given dataset. In our problem, we opted for the widely-used Caltech-101 dataset (Fei-Fei et al., 2004). It contains 101 image categories of objects and a background category. Each one has 40 to 800 images with a resolution of approximately 300×200 pixels. Each category shows variations in appearance, shape, scale, and color. In our experiments, only 10 categories were used. Figure 4.23 depicts examples of images of the categories

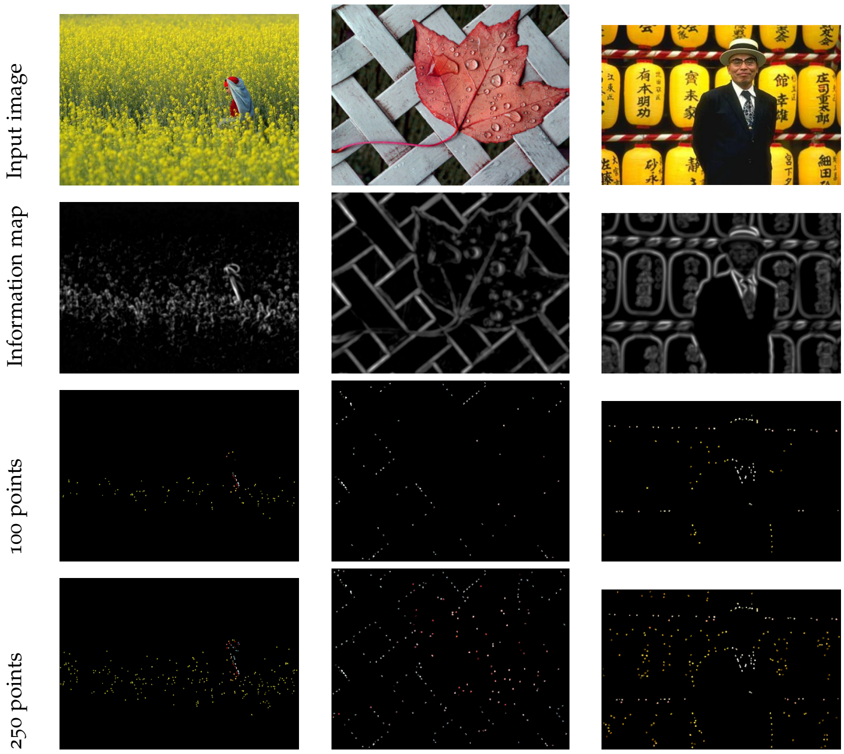


Figure 4.22: [eigSTM]-CAKE information maps and extraction results in terms of coverage.

used in the experiments.

To perform the required object classification, a bag of words model was used (Lazebnik et al., 2006). Bag of words models are a widely-used and popular technique for object recognition/classification. The concept of bag of words is borrowed from natural language processing: the idea is to treat local features as (visual) words. The vector (or histogram) that stores the occurrences of the different visual words corresponds to the bag of words. In this approach, the spatial layout of features is explicitly neglected, whereas the frequency of features is the most relevant factor.

The first step in the construction of a bag of words is feature detection. The subsequent step is to compute feature descriptors over the previously detected image patches. The collection of descriptors that represents the image is then clustered into an image vocabulary. In the following step, the histograms are fed to a Support Vector Machine (SVM) for classification. In the final step, the trained SVM is used to classify the test images.

In our classification, the number of randomly selected images per category was 15. The number of test images was also 15. As for feature detectors and descriptors, we combined a [HES]-CAKE feature detection with a standard SIFT description. The use of a descriptor (namely SIFT) densely sampled on a regular grid has been shown to outperform the classification based on sparsely located local features. As a consequence, we directly compare the performance of our [HES]-CAKE-based classification with one based on densely sampled descriptors. Our choice was the Pyramid of Histograms of Visual Words (PHOW) descriptor (Bosch et al., 2007), which succinctly corresponds to dense SIFT descriptors computed at multiple scales. The reason for this choice was based on the excellent results yielded by this kind of approach (Lazebnik et al., 2006).

To create the two versions of the bag of words model, we used and adapted the model implemented by Vedaldi & Fulkerson (2008). In the original version, this implementation uses PHOW descriptors, Elkan's k-means (Elkan, 2003) for a fast clustering, and a homogeneous kernel map that transforms a χ^2 SVM into a linear one. To create a second version, we replaced PHOW description with a standard SIFT description computed over [HES]-CAKE features. Important parameter settings of the model are summarized in Table 4.10.

Table 4.10: Parameter settings for the bag of words model

Bag of words model (versions 1 and 2)	
Vocabulary size (number of words)	300
Number of training images	15
Number of test images	15
PHOW scales (version 1)	{4,6,8,10}
PHOW grid step (version 1)	3
PHOW window size (version 1)	1.5
[HES]-CAKE settings (version 2)	see Table 4.6

We use the confusion matrix as an evaluation metric. Figure 4.24 shows the resulting confusion matrix for each version of the model. For these categories and parameter settings, both versions produce similar results either in terms of accuracy or in terms of misclassifications. Due to the heterogeneity of objects in Background_Google category, several objects from this category were misclassified by both versions of the model. Nevertheless, the second version performs a more accurate classification. In fact, [HES]-CAKE features combined with SIFT descriptors provide a slightly more accurate classification for most categories, with the exception of Faces_easy and Faces categories, which are redundant and prone to misclassifications.

Despite the fewer number of features, the second version of the model produced comparable results with a version based on dense description. Although dense descriptors have been shown to be an improvement over the traditional local feature detection combined with description, these results support the idea that context-aware features can provide the so-called robust image representation and compete with dense descriptors.

4.6 CONCLUDING REMARKS

In this chapter, we presented a context-aware feature extractor, which represents a new paradigm in local feature extraction. The idea is to retrieve salient locations within the image context, which means no assumption is made on the type of structure to be extracted. Such scheme was designed to provide a robust image representation, with or without the contribution of other local features. The algorithm follows an information theoretic approach to extract salient locations. The possible shortcomings of such approach were analyzed, namely the difficulties in defining sufficiently discriminating descriptors and estimating the information of the inherent distributions in an efficient way.

The experimental evaluation showed that relying on image statistics to extract keypoints is a winning strategy. A robust image representation can be easily achieved with context-aware features. Furthermore, the complementarity between context-aware features and strictly local ones can be exploited to produce an even more robust representation.

The use of different descriptors (codewords) allows us to construct different instances of the keypoint extractor. Two instances were suggested: [eigSTM]-CAKE, which is based on the eigenvalues of the structure tensor matrix, and [HES]-CAKE, which is based on the components of the Hessian matrix computed at multiple scales. The former represents a straightforward instance based on a codeword with a reduced number of dimensions. Despite its simplicity, [eigSTM]-CAKE features showed an efficient capture of informative content, namely in terms of the image details. The latter was designed to provide a more complete coverage of informative content; the use of second order derivatives promoted the capture of structures that carry most image information, such as blobs, and structures where the fine details of image can be found.

As for the applicability of the method, we believe that most of the tasks requiring a robust image representation will benefit from the use of context-aware features. In this category, we include tasks such as image retrieval, object (class) recognition, and image compression.

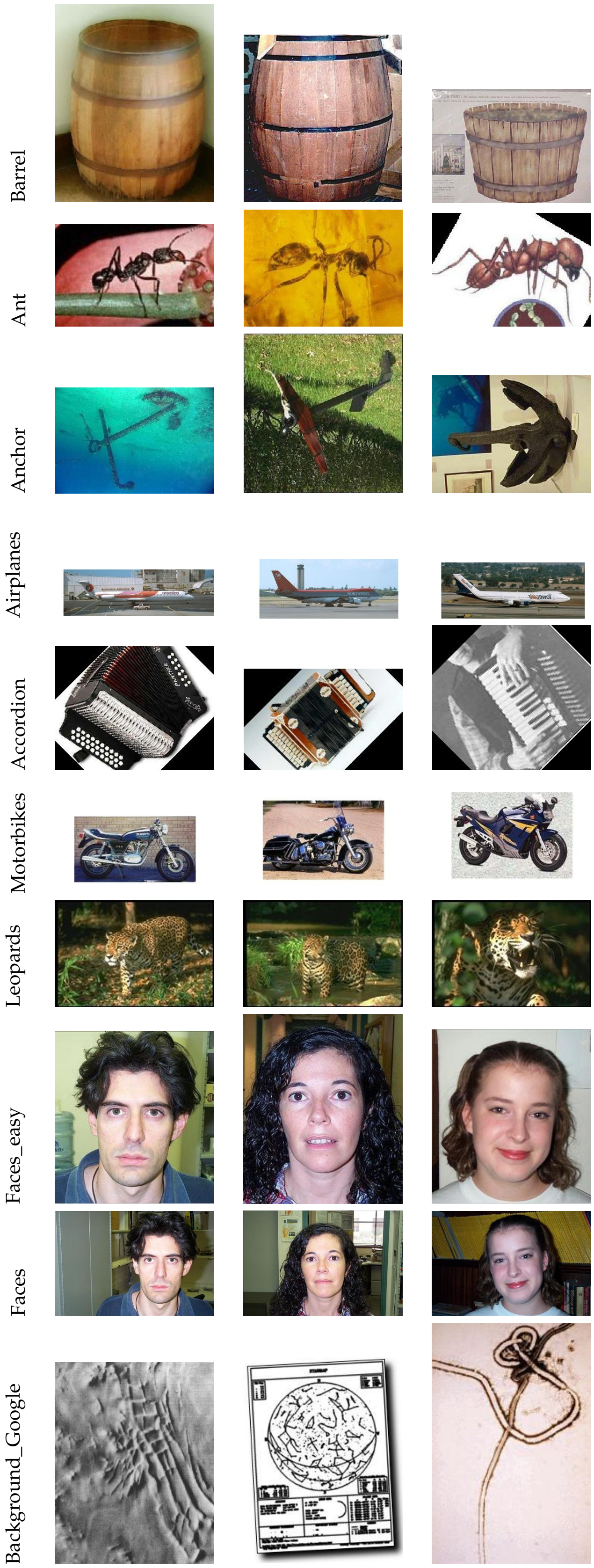


Figure 4.23: Example of images from the 10 categories used in the experiments.

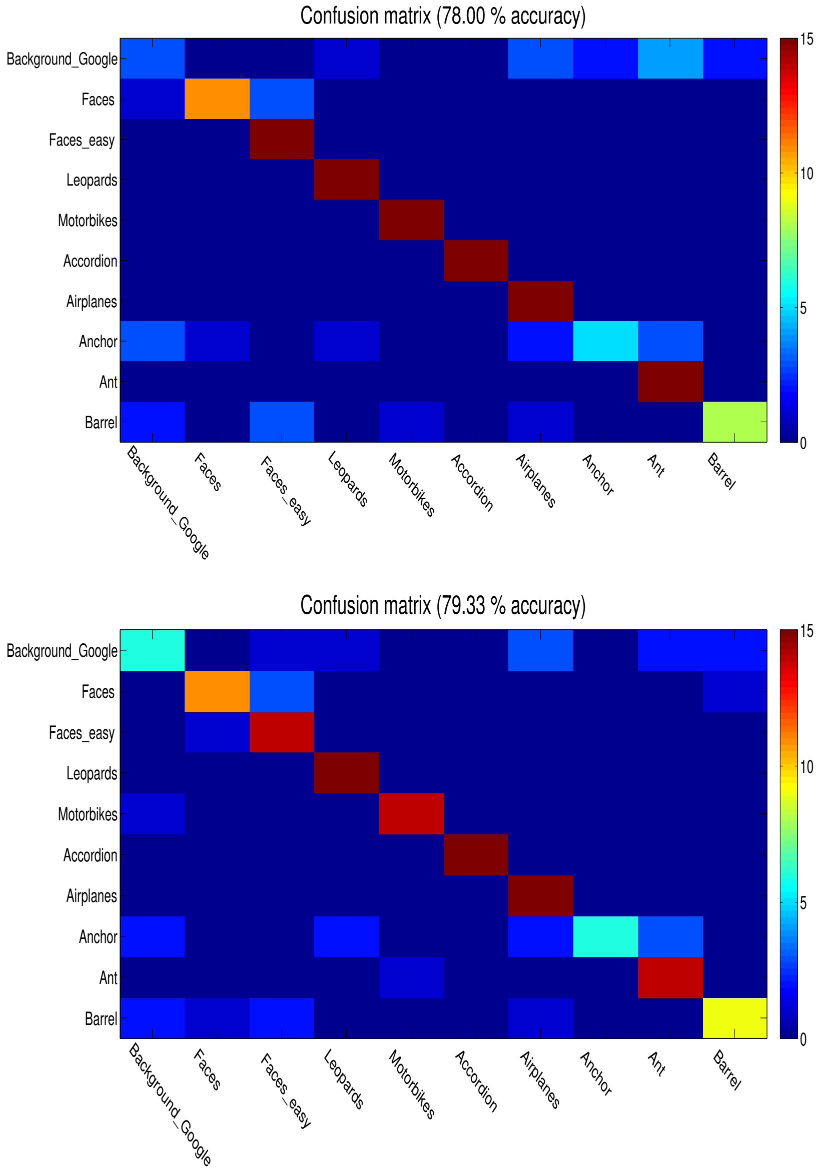


Figure 4.24: Confusion matrices for the 10 categories used in the experiments. Top row: version 1 (PHOW descriptor); bottom row: version 2 ([HES]-CAKE regions + standard SIFT descriptor).

5 STABLE (SALIENT) SHAPES: FEATURE-DRIVEN MAXIMALLY STABLE EXTREMAL REGIONS

In this chapter, we introduce Stable Salient Shapes, a novel type of features, which are the result of performing a specific feature-driven detection of Maximally Stable Extremal Regions. The main motivation for our contribution comes mainly from the well-known advantages in obtaining affine covariant features from extremal regions as well as the shortcomings that such strategy entails. In comparison with MSER, the new features appear in higher number and are more robust to blur. In addition, Stable Salient Shapes are designed to provide a better coverage of informative image parts.

5.1 MOTIVATION

Compared with keypoints, semi-local structures, such as edges as well as curvilinear shapes, tend to be more robust to intensity, color, and pose variations (Deng et al., 2007). There are only a few local feature detectors that explicitly or implicitly take advantage of this robustness by detecting stable regions from semi-local structures. Two well-known examples are the algorithm for the detection of Principal Curvature-Based Regions (PCBR) (Deng et al., 2007) and the Maximally Stable Extremal Regions (MSER) detector (Matas et al., 2002), which were described in Chapter 2. In a direct comparison of both methods, the latter shows several advantages over its counterpart, namely in terms of computational efficiency and the accuracy of regions. On the other hand, PCBR features tend to produce a better coverage of relevant objects within the scene, which is due to the exclusive use of robust structural information to construct stable regions.

The MSER detector does not always show the desired performance. In the large-scale comparative study on affine covariant regions performed by Mikolajczyk et al. (2005), MSER and Hessian-Affine features showed higher repeatability scores. However, the MSER detector showed an inconsistent performance: blurred sequences of images as well as textured sequences produced less repeatable features (see Figs. 5.1 and 5.2 as introductory examples). The low repeatability scores in the above-mentioned conditions is a well known downside of MSER detection. The sensitiveness to image blur can be explained by the undermining effect that blur has on the stability criterion, which is illustrated in Fig. 5.3: by applying different levels of blur, we change the area of extremal regions. Additionally, as the blurring effect increases, the number of extremal regions decreases. As for textured scenes, they are a not a suitable domain for MSER detection since intensity perturbations cause an irregular area variation of extremal regions in busy parts of the image.

Another downside of MSER detection is related to the number of regions that the detector retrieves. The number of MSER tends to be lower than the number of regions retrieved by detectors such as the Hessian-Affine or the Harris-Affine. A reduced number of features may not provide the best coverage of the content, which impairs the robustness of the method against object occlusions and the suitability for tasks requiring a robust image representation (e.g., object class recognition). Furthermore, the detector prefers homogeneous regions over heterogeneous ones, which might discard relevant image content.

Kimmel et al. (2011) observe that the affine covariance of MSER is verified if and only if objects possess smooth boundaries. As the affine covariance of these features is an immediate consequence of the covariance of the image level sets with affine transformations of the coordinates, it is required that the point-spread function of camera lenses is small compared to the natural blurring of objects. The authors also note that the stability criterion as defined in Eq. (2.36) prefers regular (round) shapes to irregular ones. This bias for regular shapes was demonstrated by showing that if two regions have the same area and the same intensity along the boundaries, the one with a shorter boundary will yield a lower value of ρ. We emphasize the importance of such property as most scenes contain irregular shapes and we surely cannot claim that regular shapes are always more distinctive than irregular ones.

5.2 stable (salient) shapes

At a first glance, the ideal image for the MSER detector is the one that is well structured, with uniform regions separated by strong intensity changes (Tuytelaars & Mikolajczyk, 2008). However, the affine covariance holds for MSER if and only if the boundaries of the objects in the scene are smooth. These are the principles in which our detector is based on. We can succinctly describe the construction of our features, which we will refer to as Stable Salient Shapes, as a feature-driven MSER detection, where such features correspond to structures related to objects boundaries or even symmetry axes.

The first step of the proposed algorithm, coined as feature highlighting, consists in building a saliency map for each one of the fea-

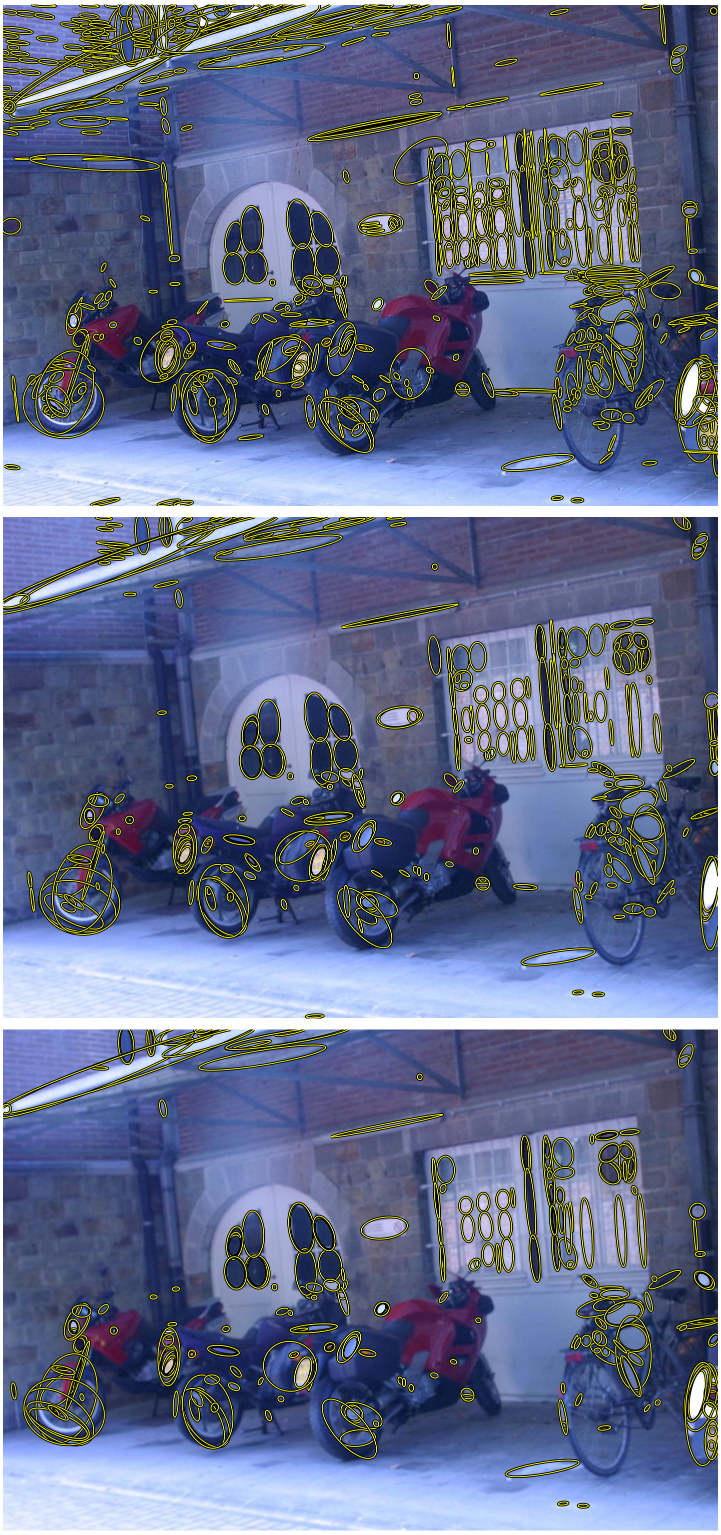


Figure 5.1: MSER detection on the first three images of the Bikes sequence.

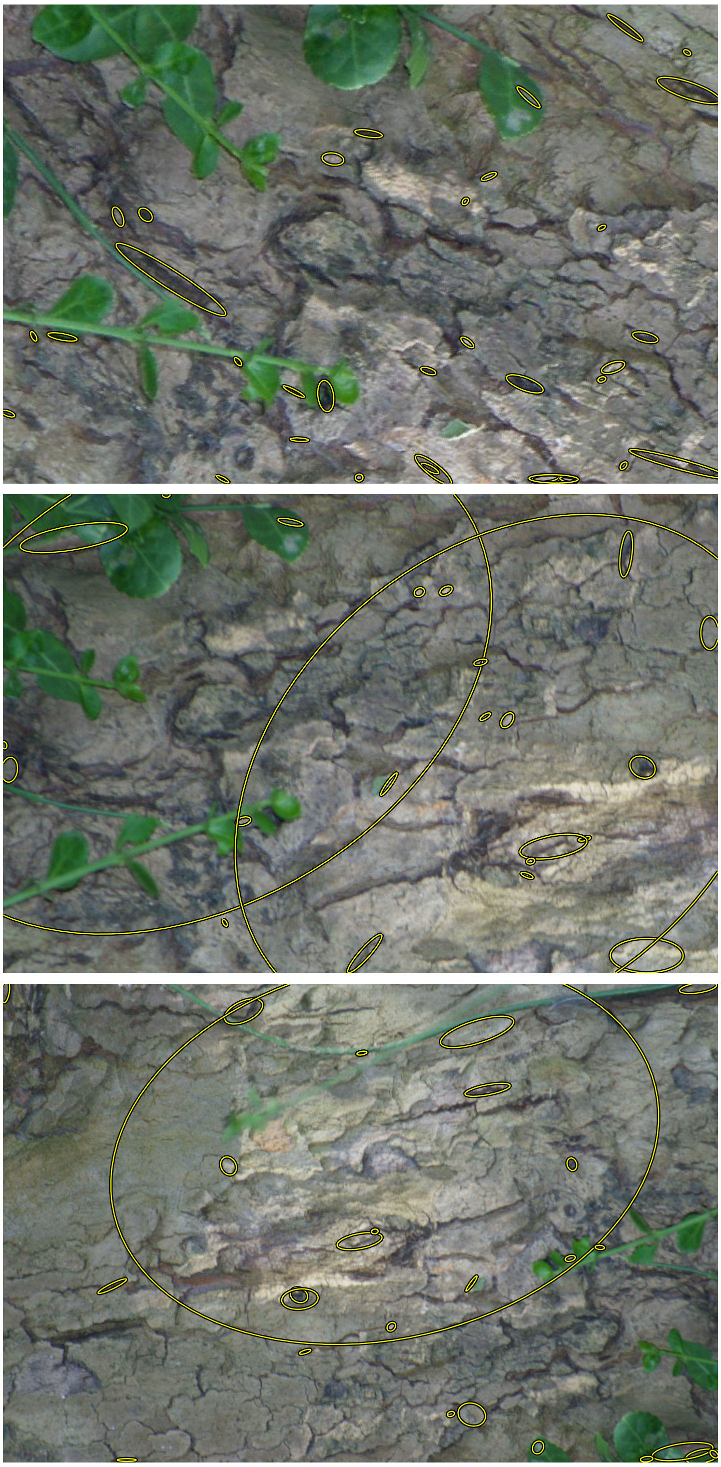


Figure 5.2: MSER detection on the first three images of the Bark sequence.

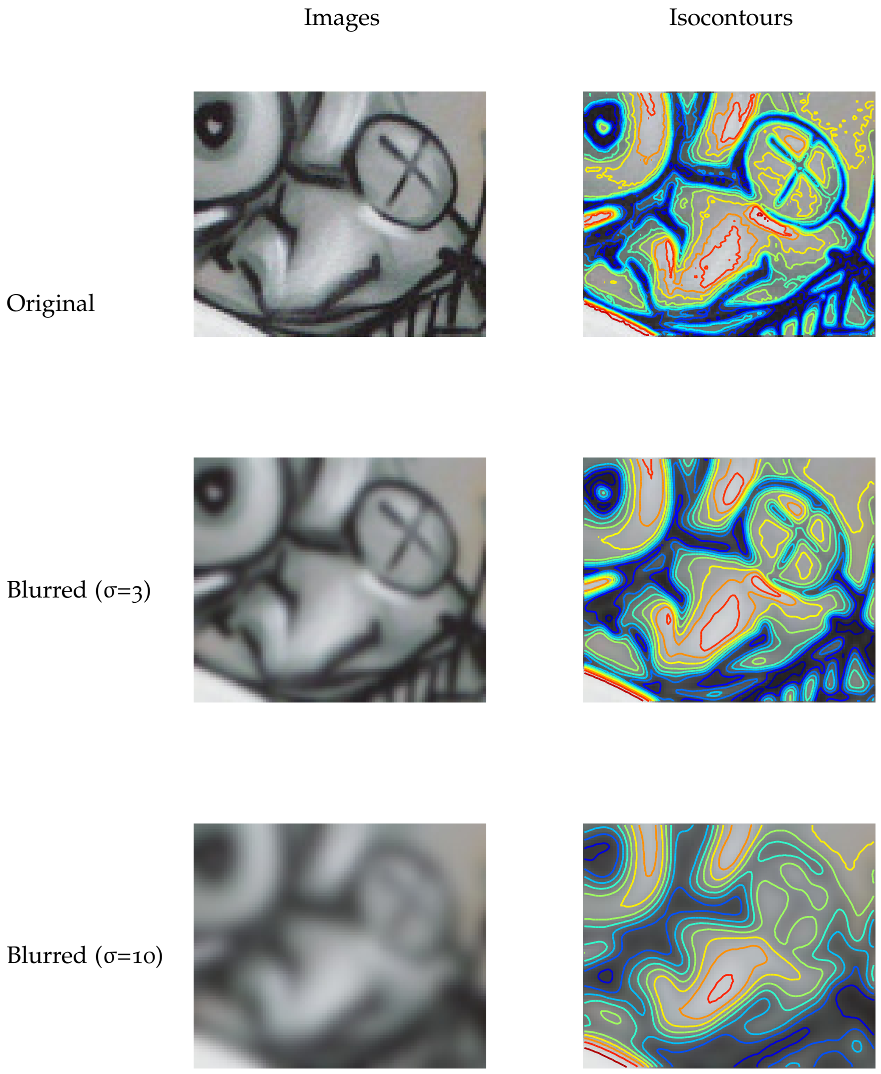


Figure 5.3: Isocontours and blur: the application of different levels of (Gaussian) blur produces different extremal regions.

tures to be highlighted. These maps are intended to be suitable domains for MSER detection. Note that the boundaries of MSER often correspond to objects boundaries (see Fig. 5.4 as an example). As the stability of an extremal region with respect to intensity changes is measured at its boundary, objects boundaries will be reshaped to yield more stable extremal regions.

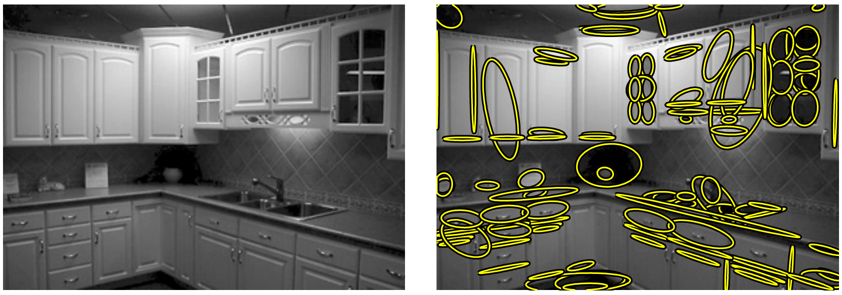


Figure 5.4: An example of MSER detection where most regions are anchored at objects boundaries. Left: input image; right: MSER detection

The subsequent step of the algorithm consists in detecting MSER on the saliency maps. If we use more than one map, we cannot expect a full complementarity among the extremal regions detected along the different maps, since they are related to the same structures. In this case, there is a third step of the algorithm that takes into consideration the potential overlapping or even the duplication of regions and performs a region pruning. Figure 5.5 depicts the main steps of SSS detection.

We can see our method as a hybrid between the MSER algorithm and the PCBR detector, since we use structural information (shapes) to define suitable domains for MSER detection. The idea is to combine the advantages of PCBR detection (good coverage) with the advantages of MSER detection (computational efficiency, repeatability, and accuracy) and simultaneously overcome some of the major limitations of the latter, namely the lack of robustness to blurring and the biased preference for round shapes.

5.2.1 *Feature highlighting*

We propose to highlight edges and ridges by means of two differential-based measures, which will produce two different saliency maps. Edges reflect the presence and the shape of objects in a scene. Ridges are related to symmetry axes as they often correspond to the major axis of symmetry of elongated objects.

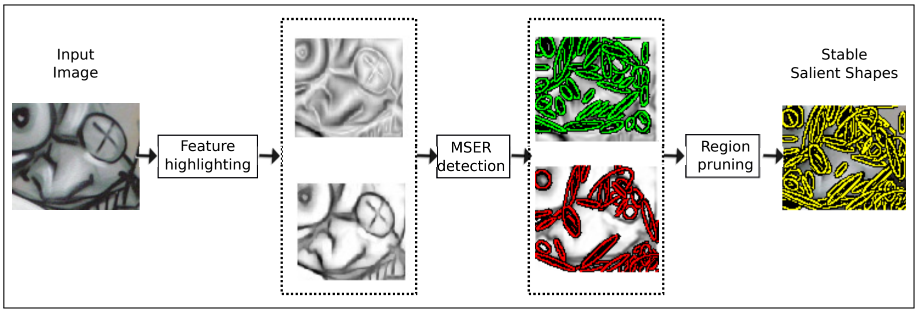


Figure 5.5: Algorithm for the detection of Stable Salient Shapes (feature-driven MSER). The first step of the algorithm produces saliency maps that will be used in the next step as input images for MSER detection. The maps emphasize features that are related to semantically meaningful structures, such as boundaries and symmetry axes.

5.2.1.1 *Edge highlighting*

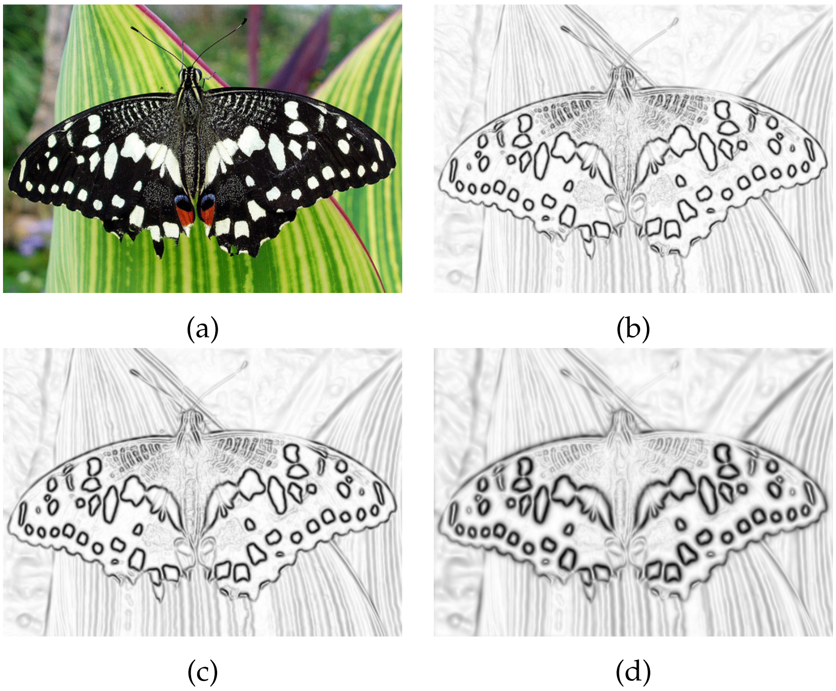
Our first measure highlights edges and simultaneously delineates smooth transitions at the boundaries. The detection of structures at different scales will help us to define smooth transitions. The process of averaging information over scales is the key component to obtain the desired smoothness. We start by computing the gradient magnitude by means of Gaussian derivatives at several scales. Let $L(:, \sigma)$ be a smoothed version of image I by means of a Gaussian kernel G at the scale σ, i. e., $L(\mathbf{x}, \sigma) = G(\sigma) * I(\mathbf{x})$. The edge strength can be found by measuring the gradient magnitude,

$$|\nabla L(\mathbf{x}, \sigma)| = \sqrt{L_x^2(\mathbf{x}, \sigma) + L_y^2(\mathbf{x}, \sigma)}, \tag{5.1}$$

where L_x and L_y denote the first order partial derivatives of L in the x and y directions, respectively. From (5.1), we obtain our measure for edge highlighting:

$$F_1(\mathbf{x}) = \sum_{i=1}^{N} \sigma_i \, |\nabla L(\mathbf{x}, \sigma_i)| \, , \tag{5.2}$$

where the standard deviation σ_i varies in a geometric sequence $\sigma_i = \sigma_0 \xi^{i-1}$, with $\sigma_0 \in \mathbb{R}^+$, $\xi > 1$, and N denotes the number of scales. The final image is, therefore, the result of averaging gradient magnitude computed at different scales. By doing this, with a reasonable number of scales, smooth transitions at the edges will be obtained. Note that a larger number of scales leads to smoother boundaries and an increasing loss of image details. To illustrate this observation, we depict the proposed edge highlighting in Fig. 5.6, using 4 and 12 scales.



(a)

(b)

(c)

(d)

Figure 5.6: Proposed edge highlighting: (a) input image; (b) edge strength (gradient magnitude); (c) edge highlighting (4 scales); (d) edge highlighting (12 scales). Darker structures in the saliency maps are the most salient ones. The parameters σ_0 and ξ were set to 1 and $\sqrt[4]{2}$, respectively.

5.2.1.2 *Ridge highlighting*

The measure for ridge highlighting derives from the Hessian matrix,

$$\mathcal{H}(\mathbf{x},\sigma) = \begin{bmatrix} L_{xx}(\mathbf{x},\sigma) & L_{xy}(\mathbf{x},\sigma) \\ L_{yx}(\mathbf{x},\sigma) & L_{yy}(\mathbf{x},\sigma) \end{bmatrix}, \tag{5.3}$$

where L_{xx}, L_{xy} and L_{yy} are the second order partial derivatives of L, a Gaussian smoothed version of image I. The principal curvature (Deng et al., 2007), which highlights curvilinear structures is either given by

$$P_{max}(\mathbf{x},\sigma) = \max(0, \lambda_2(\mathcal{H}(\mathbf{x},\sigma))), \tag{5.4}$$

or

$$P_{min}(\mathbf{x},\sigma) = \min(0, \lambda_1(\mathcal{H}(\mathbf{x},\sigma))), \tag{5.5}$$

where λ_1 and λ_2 denote the minimum and maximum eigenvalues, respectively. Note that (5.4) and (5.5) respond to complementary structures: the former responds to dark lines on a brighter background, whereas the latter detects brighter lines on a dark background. From the principal curvature, we obtain the measure for ridge highlighting:

$$F_2(\mathbf{x}) = \sum_{i=1}^{N} \sigma_i^2 P_{max}(\mathbf{x}, \sigma_i), \tag{5.6}$$

where $\sigma_i = \sigma_0 \xi^{i-1}$, with $\sigma_0 \in \mathbb{R}^+$, $\xi > 1$, and N denotes the number of scales.

Our ridge highlighting measure uses the principal curvature measure to detect darker lines on a bright background. However, the measures defined in (5.4) and (5.5) can be used interchangeably. In Fig. 5.7, we depict the proposed ridge highlighting, using 4 and 12 scales. To further illustrate our feature highlighting, we depict in Fig. 5.8 both saliency maps, using a license plate as the input image. It is readily seen that both saliency maps preserve the structural information of the image and add some smoothness to the scene. While the edge highlighting mainly captures and accentuates the objects boundaries, the ridge highlighting provides a clearer structural sketch of the scene (Deng et al., 2007). For the purpose of MSER detection, we can regard the map that emphasizes edges as a more suitable domain, as it generates more uniform regions separated by heavy intensity changes and is less sensitive to noise. However, the second map provides us complementary regions, whose detection is important to improve the coverage of the content.

At a glance, our proposed strategy for feature highlighting entails two major shortcomings, which may impair the validity of the method. Integrating either the edge strength or the principal curvature over an isotropic scale-space will hinder the affine covariance of the resulting features. Further, the use of a fixed number of scales does not ensure a full scale covariance. However, it is important to note that these drawbacks become relatively minor if our method detects a substantially higher number of features than the MSER algorithm without a significant drop in the repeatability score when scale changes or affine transformations occur.

5.2.2 *MSER detection and region pruning*

Apart from the input image, there are no differences between the feature-driven MSER detection and the original one, i.e., we assess the stability of extremal regions using the original stability criterion. However, in our method, the biased preference of the MSER detector towards regular shapes is not neglected; the Gaussian smoothing has a regularization effect on the shapes.

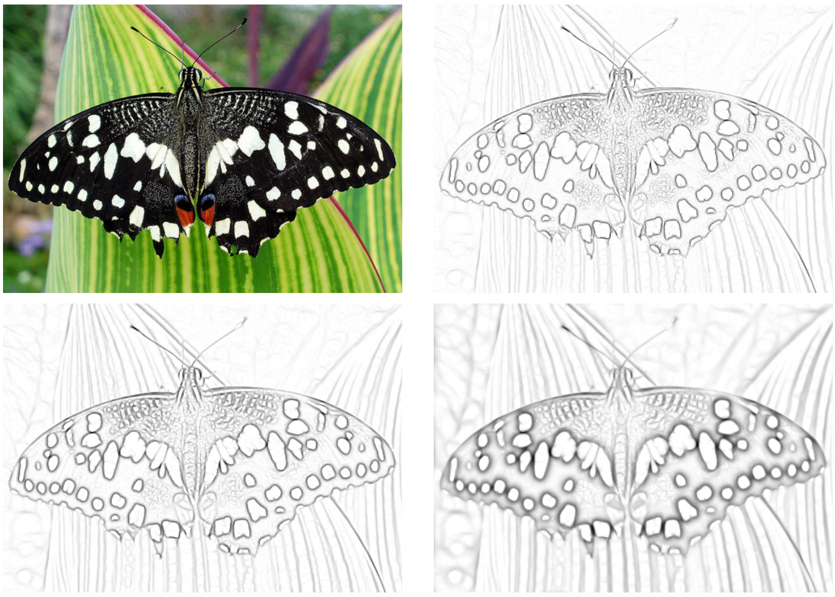


Figure 5.7: Proposed ridge highlighting: (a) input image; (b) principal curvature response; (c) ridge highlighting (4 scales); (d) ridge highlighting (12 scales). Darker structures in the saliency maps are the most salient ones. The parameters σ_0 and ξ were set to 1 and $\sqrt[4]{2}$, respectively.

Figures 5.9 and 5.10 help to illustrate the advantages of our feature-driven MSER detection over the standard one. In Fig. 5.9, we depict two well-structured scenes and the corresponding isophotes on the luminance channel and on both of the saliency maps. Both maps show a higher number of extremal regions and due to the Gaussian smoothing, the irregularity of extremal regions is attenuated, which will compensate for the preference towards regular shapes.

Figure 5.10 compares standard MSER detection with SSS detection in the presence of Gaussian blur. As blur increases, both types of regions decrease in number. However, this reduction is more significant for MSER. For $\sigma = 3$, the number of MSER decreases by 19%, whereas the number of SSS decreases by 11%. For $\sigma = 10$, the number of MSER decreases 93%, while the number of feature-driven MSER decreases by 53%. The example also shows that SSS are in higher number and cover the most informative parts of the scene, regardless of the amount of blur.

The two saliency maps do not provide fully complementary regions. Thus, we eliminate regions that are duplicated. To find duplicates, we compare the centroid distance. If this distance is lower than 0.1, we compute the overlap error between the corresponding fitted ellipses. If this error is less than 10%, we discard the region

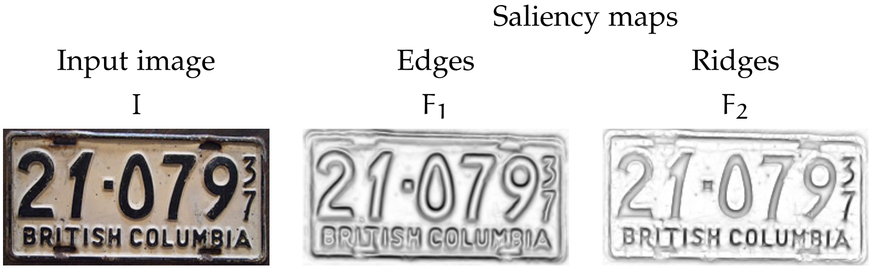


Figure 5.8: An example of the proposed feature highlighting. Darker structures in the saliency maps are the most salient ones. To obtain the final saliency maps – F_1 and F_2 –,12 scales were used. The parameters σ_0 and ξ were set to 1 and $\sqrt[4]{2}$, respectively.

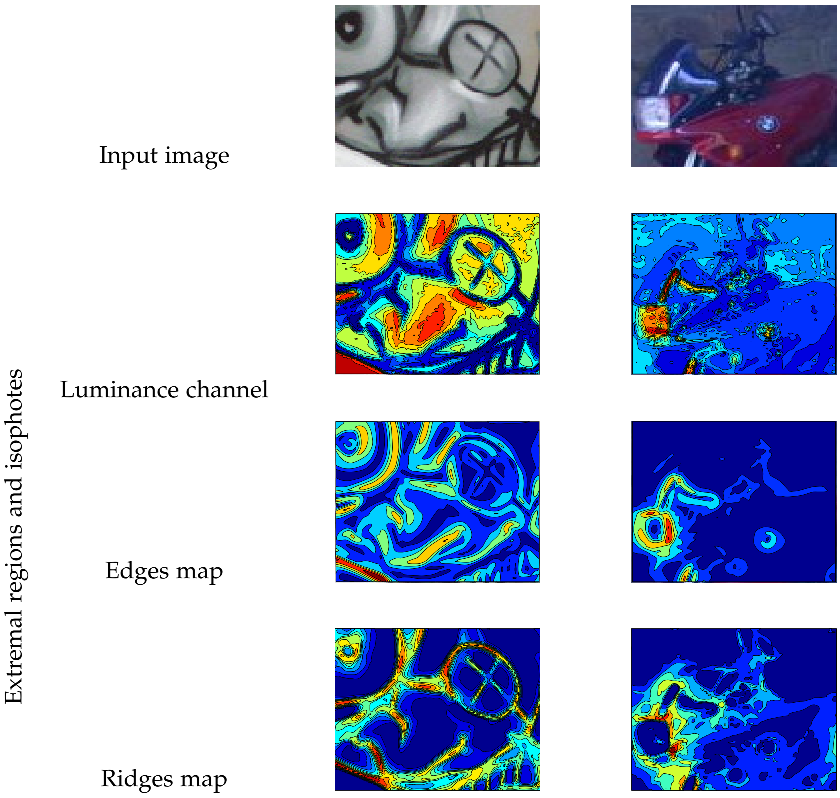
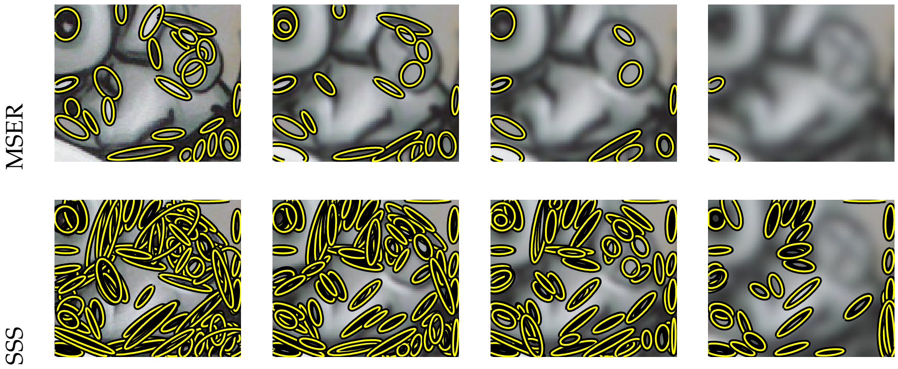


Figure 5.9: Regions delineated by isophotes on the different domains.



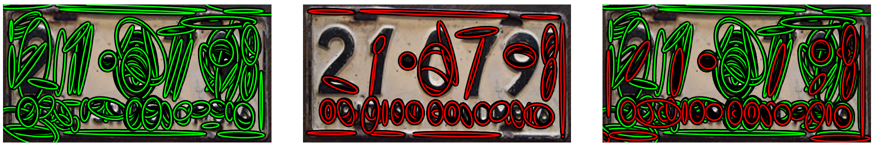
Original image

Blurred image ($\sigma = 3$)

Blurred image ($\sigma = 5$)

Blurred image ($\sigma = 10$)

Figure 5.10: MSER and SSS detection on images with increasing Gaussian blur (original image, $\sigma = 3$, $\sigma = 5$, and $\sigma = 10$). Top row: MSER detection; bottom row: SSS detection.



SSS (edges map)

SSS (ridges map)

SSS (final)

Figure 5.11: An example of the proposed region pruning.

with higher ρ. Figure 5.11 depicts the proposed region pruning. A previous pruning, which removes regions based on the area or the stability measure ρ, is performed on each map. We describe it in the upcoming section.

To conclude the section, we present the results of different detectors on a Siemens star (see Fig. 5.12). The detection includes the scale covariant SFOP regions (Förstner et al., 2009a) and affine covariant features, such as MSER, PCBR, SSS, Harris-Affine, and Hessian-Affine regions. In this example, SSS cover the most relevant image content without the presence of redundant regions. Moreover, most of the content covered by other regions, is also covered by the proposed features.

5.3 EXPERIMENTAL RESULTS

To validate our method, we performed a comparative evaluation on the Oxford dataset. Repeatability and completeness were the main criteria for assessing the performance of the SSS detector. We compared the repeatability scores of our method with the ones of the fol-

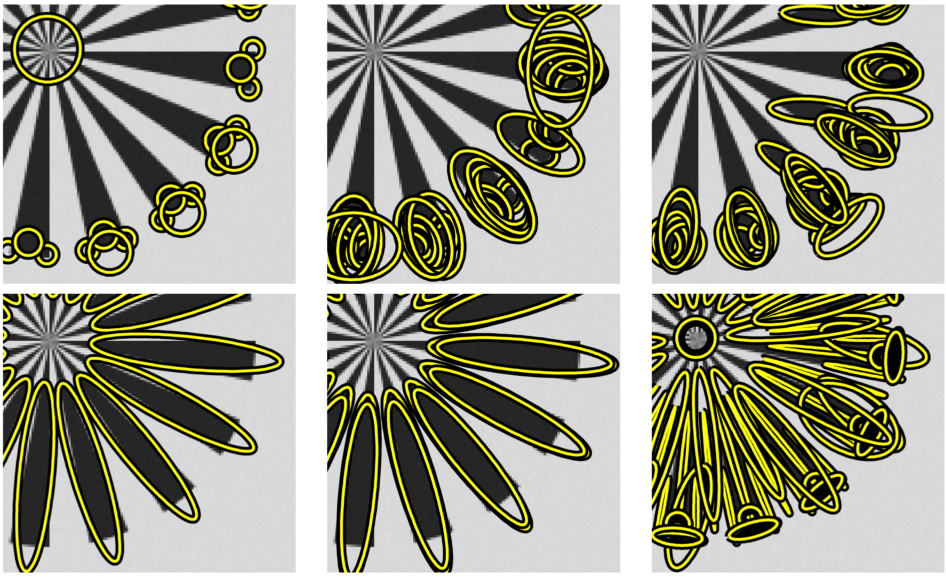


Figure 5.12: Local feature detection on the beams of a "Siemens star" (Förstner et al., 2009a). Top left to bottom right: SFOP, Harris-Affine, Hessian-Affine, MSER, PCBR, and SSS. The proposed detection responds to most of the structures detected by the remaining detectors. Only SFOP and SSS detect the center of the star.

lowing affine covariant detectors: MSER, Hessian-Affine (HESAFF), Harris-Affine (HARAFF), IBR, EBR, and PCBR. The completeness test provides a detailed comparison between the completeness values of SSS and MSER. Apart from the MSER detector, all the implementations correspond to the ones provided and maintained by the authors. For the SSS and MSER detectors, we made use of the code provided by Vedaldi & Fulkerson (2008). In the case of the SSS detector, this code was modified to deal with images whose intensity values vary in a range different from {0,...,255}, since the saliency maps intensity values might be greater than 255.

We built the saliency maps with $\sigma_0 = 1$, $\xi = \sqrt[4]{2}$, and $N = 12$. The stability threshold Δ was set to 20. For the MSER detector, this parameter was set to 10. The minimum and maximum region area were set to 30 and 1% of the image area, respectively, for both of the detectors. For repeatability evaluation, we only considered MSER and SSS whose ρ was lower than 0.7. Figure 5.13 depicts SSS detection using these parameters. For completeness evaluation, we discarded the threshold defined for ρ and the maximum allowed region area was increased up to 50% in order to increase the coverage provided by these two types of features.

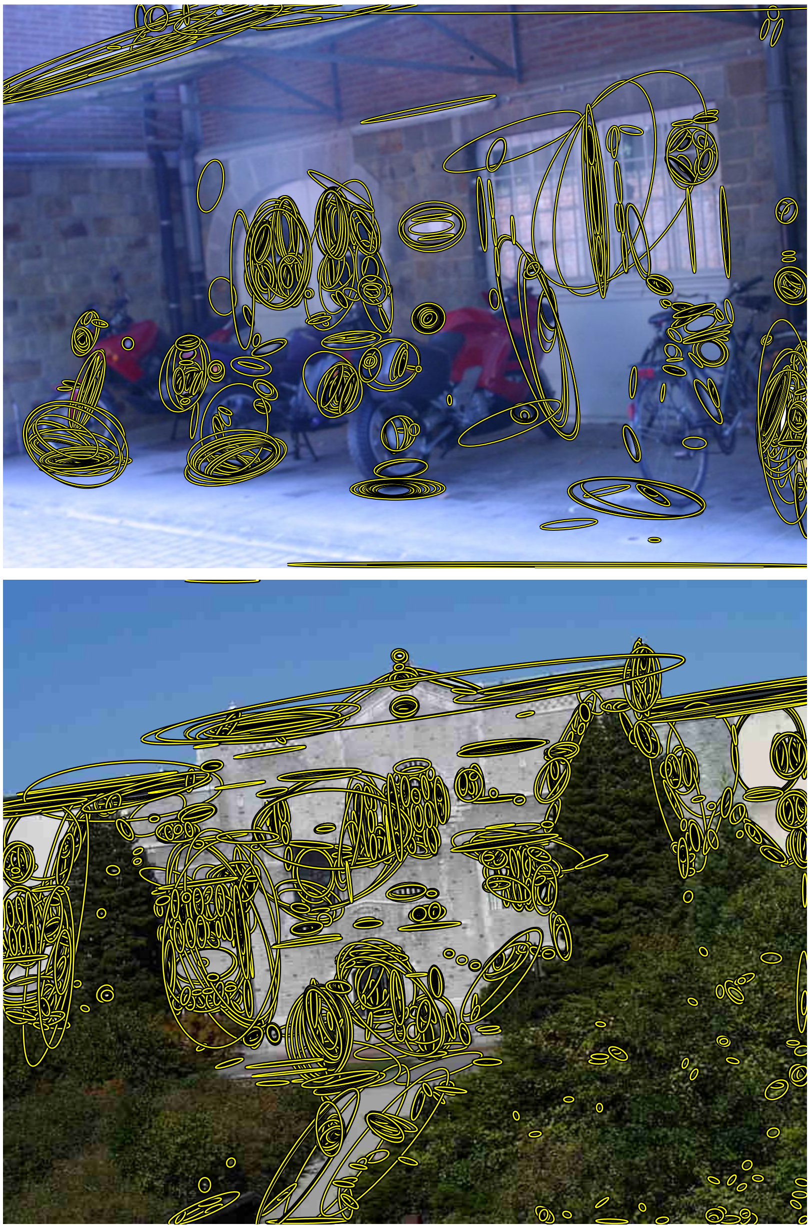


Figure 5.13: SSS detection. Top: Bikes sequence (third image); bottom: UBC sequence (third image)

5.3.1 *Repeatability evaluation*

Figures 5.14 to 5.21 depict the repeatability scores and the number of correspondences for the different sequences with an overlap error of 40%. The plot in the bottom row of each figure shows the repeatability score as function of the overlap error, which gives us an idea of the accuracy of the detectors.

The SSS detector yields the highest number of correspondences for most of the sequences (the exception is the Wall sequence), while its repeatability score is comparable to those of its counterparts. In comparison with MSER, SSS regions are more robust to blur and JPEG compression. As expected, MSER exhibit a slightly higher repeatability score for viewpoint changes in the Graffiti sequence as well as for the zoom and rotation variations (Boat and Bark sequences). However, the number of correspondences between MSER is considerably lower. Note that a substantially higher number of correspondences accompanied by a slight decrease of the repeatability rate is often preferable to a minor increase of the repeatability with less regions, since in the former the absolute number of repeated regions is considerably higher, which might provide a better coverage of the content with a similar repeatability score.

PCBR and EBR features show the worst performances in terms of repeatability score, although the PCBR detector provides highly repeatable and accurate features for the Graffiti sequence. This sequence is particularly suitable for this type of detector, as the objects in the scene are delineated by well defined boundaries.

A higher overlap error yields more correspondences and a higher repeatability score. We verify that SSS is an accurate detector as well as MSER, IBR, EBR, and PCBR detectors. HARAFF and HESAFF detectors tend to improve their ranking as the overlap increases, which means that the regions retrieved by these detectors are the less accurate among the different types of affine covariant regions.

We complemented the main repeatability evaluation with the analysis of the trade-off between the number of correspondences and the repeatability score for different stability thresholds as suggested by Perdóch et al. (2007). We extended this analysis to assess the trade-off between the matching score and the number of matches using the SIFT descriptor. For these secondary experiments, we considered three sequences: Bikes, UBC, and Wall. Results are reported in Figs. 5.22–5.24. With the SIFT descriptor, SSS and MSER tend to exhibit similar matching scores. However, as one expected, the former provides a considerably higher number of correct matches.

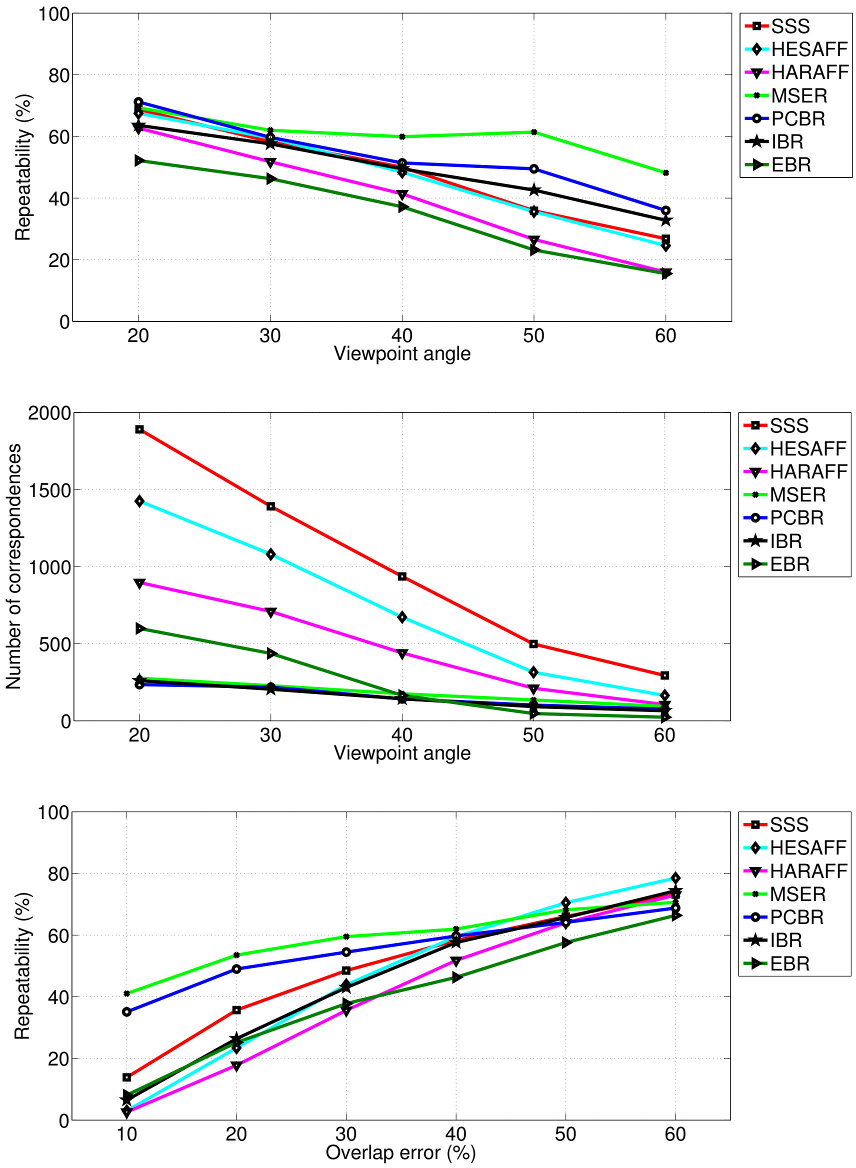


Figure 5.14: Repeatability results for the Graffiti sequence (increasing blur). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

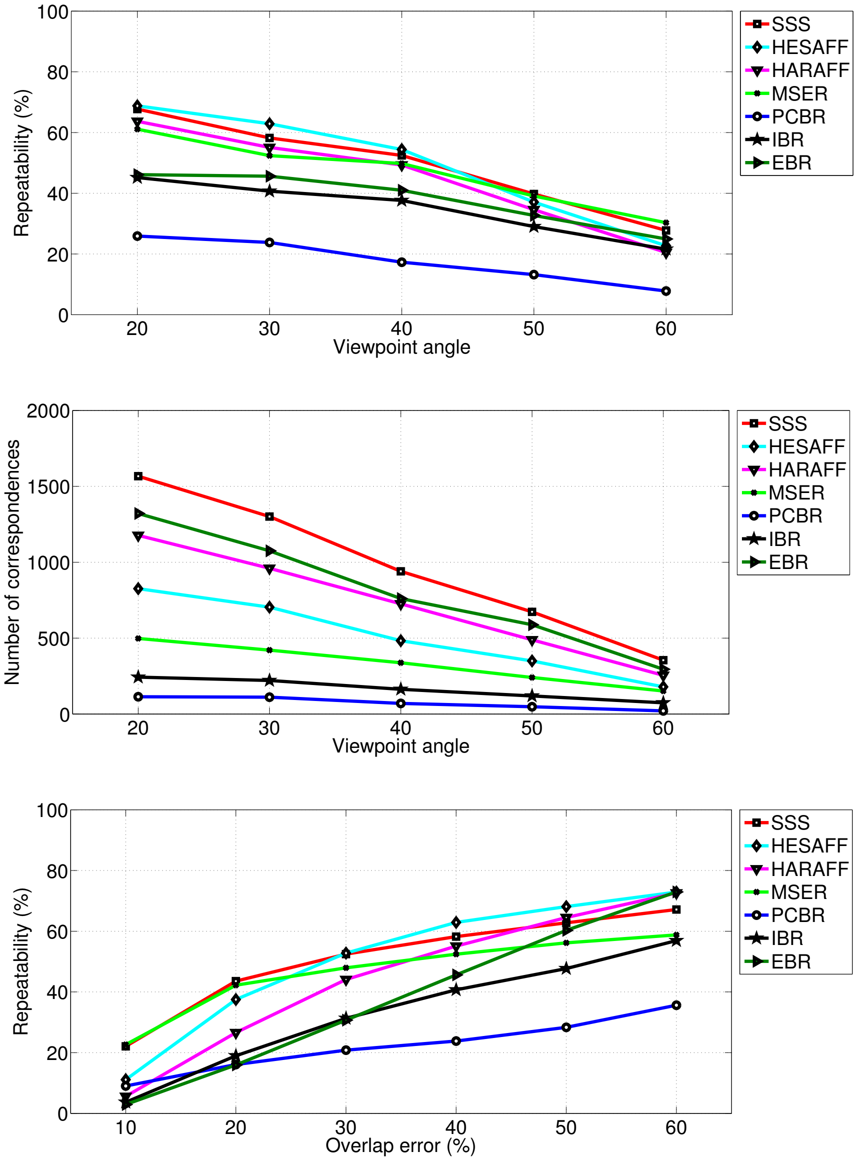


Figure 5.15: Repeatability results for the Bikes sequence (viewpoint change). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

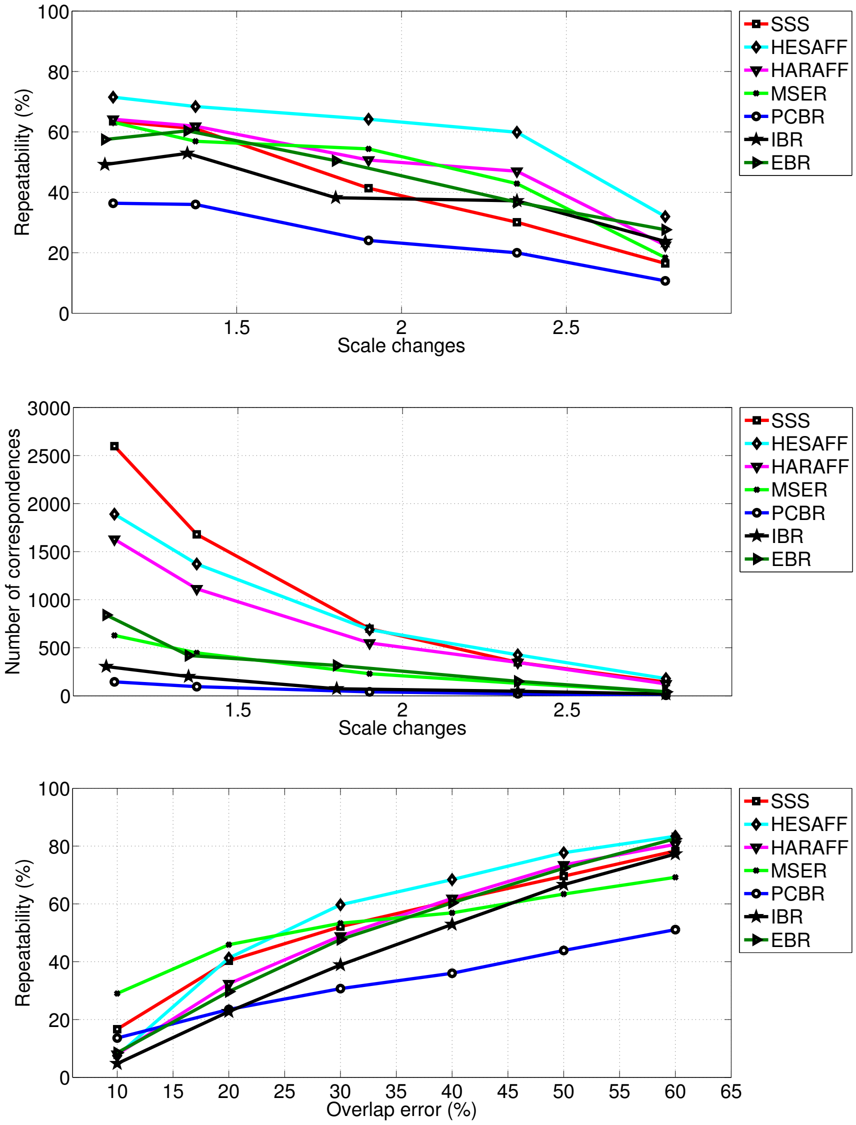


Figure 5.16: Repeatability results for the Boat sequence (scale change). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

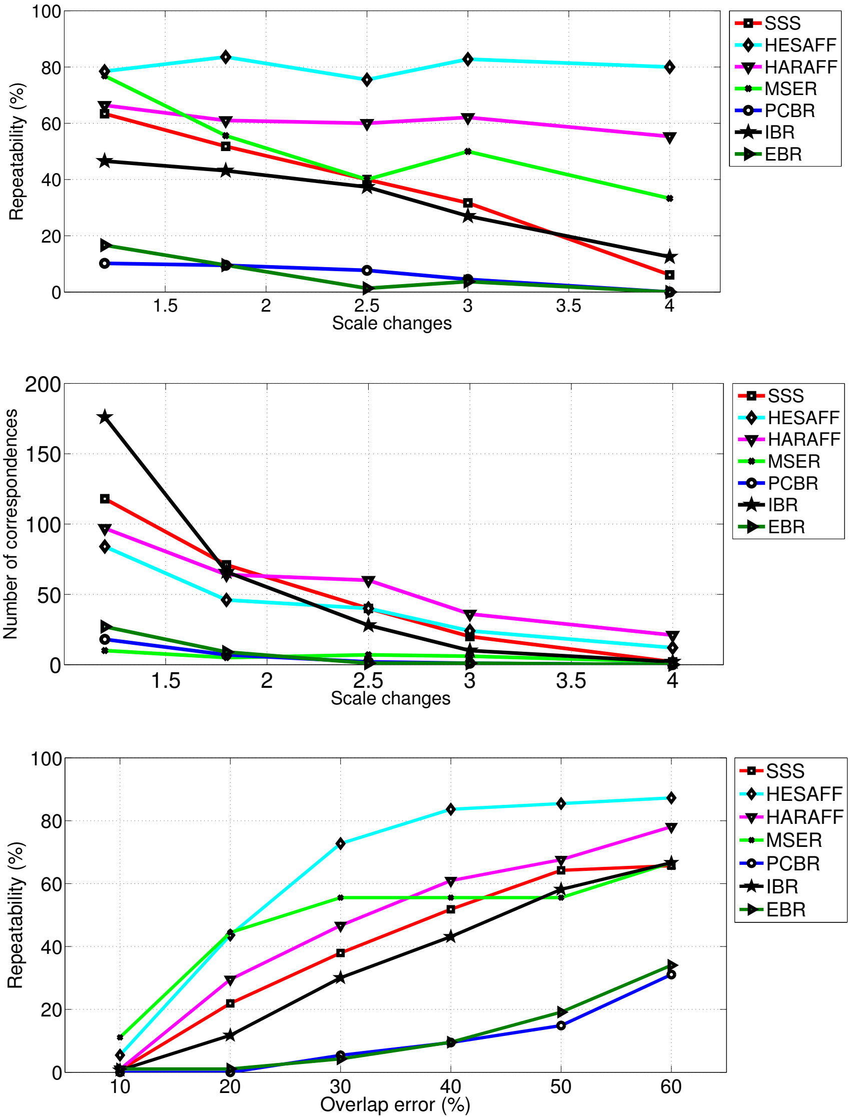


Figure 5.17: Repeatability results for the Bark sequence (scale change). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

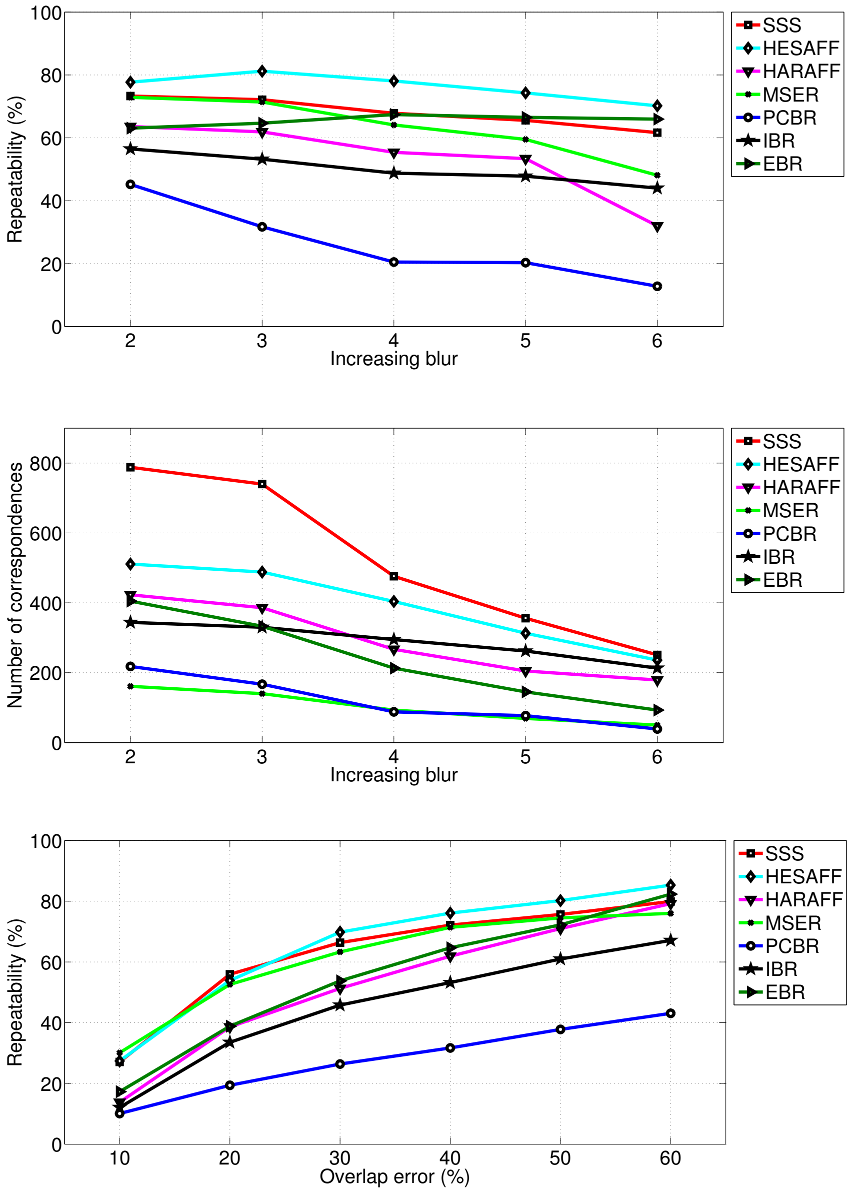


Figure 5.18: Repeatability results for the Bikes sequence (increasing blur). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

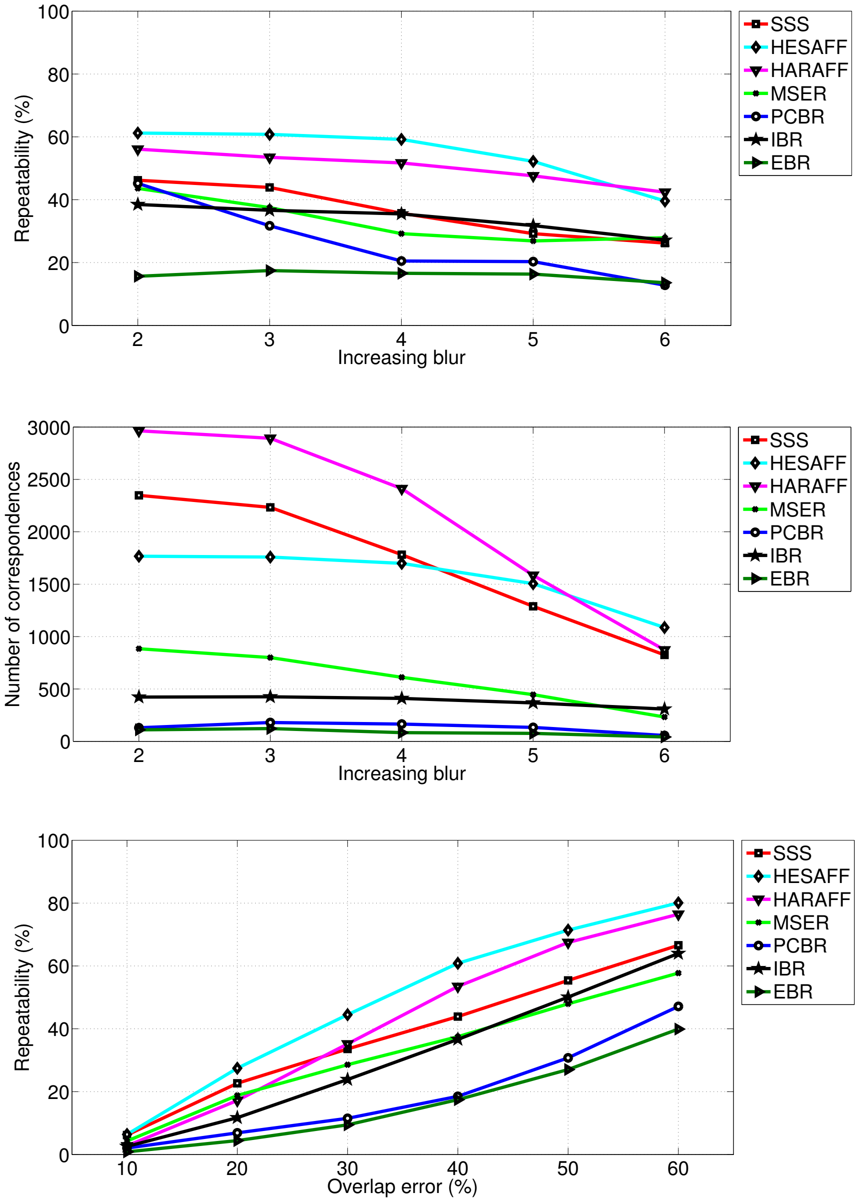


Figure 5.19: Repeatability results for the Trees sequence (increasing blur). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

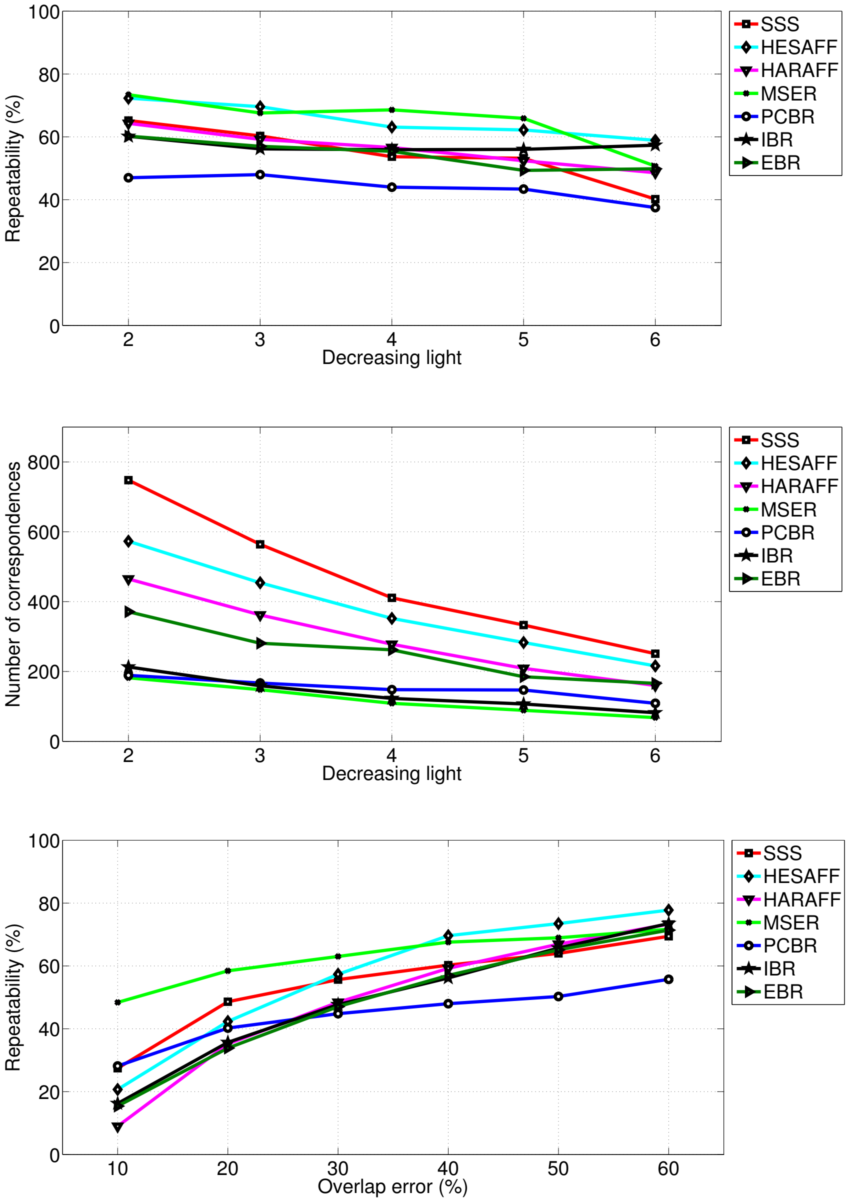


Figure 5.20: Repeatability results for the Leuven sequence (decreasing light). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

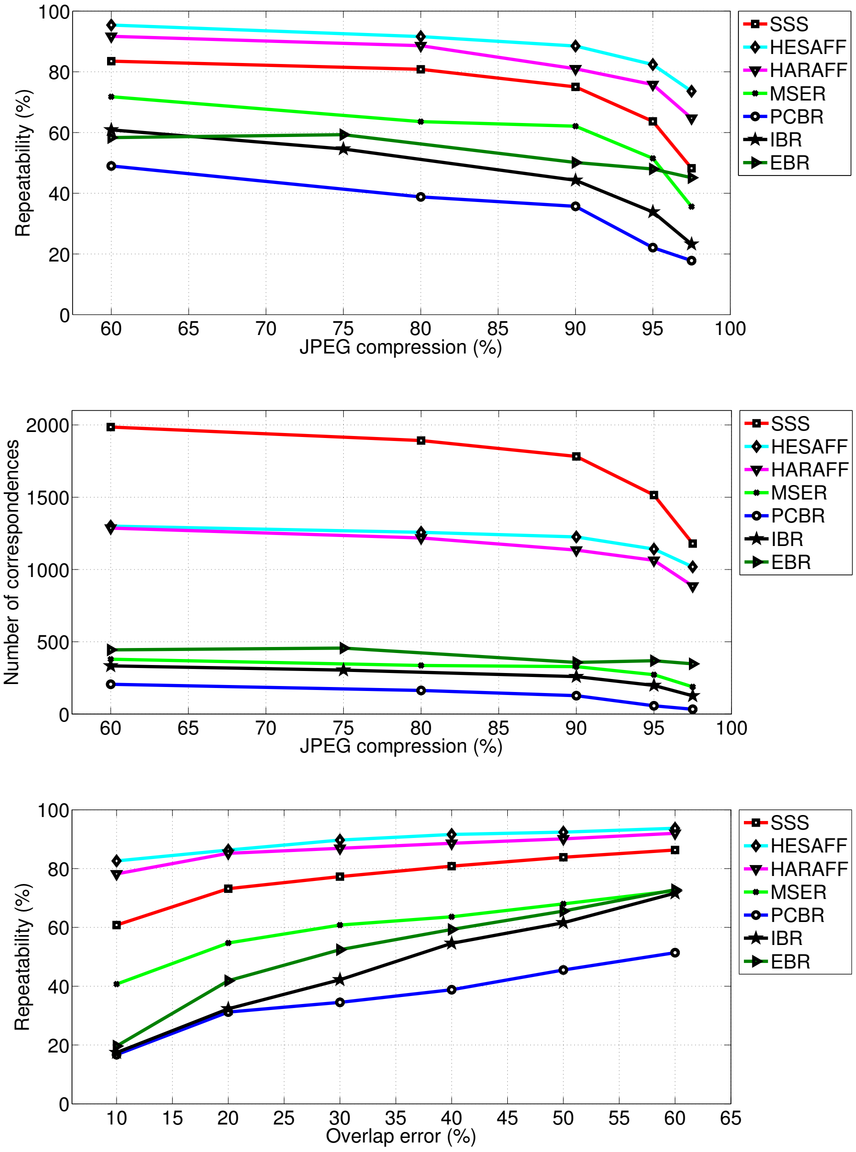


Figure 5.21: Repeatability results for the UBC sequence (JPEG compression). Top row: repeatability (overlap error of 40%); middle row: number of corresponding regions (overlap error of 40%); bottom row: repeatability for the third image w.r.t. to the first one.

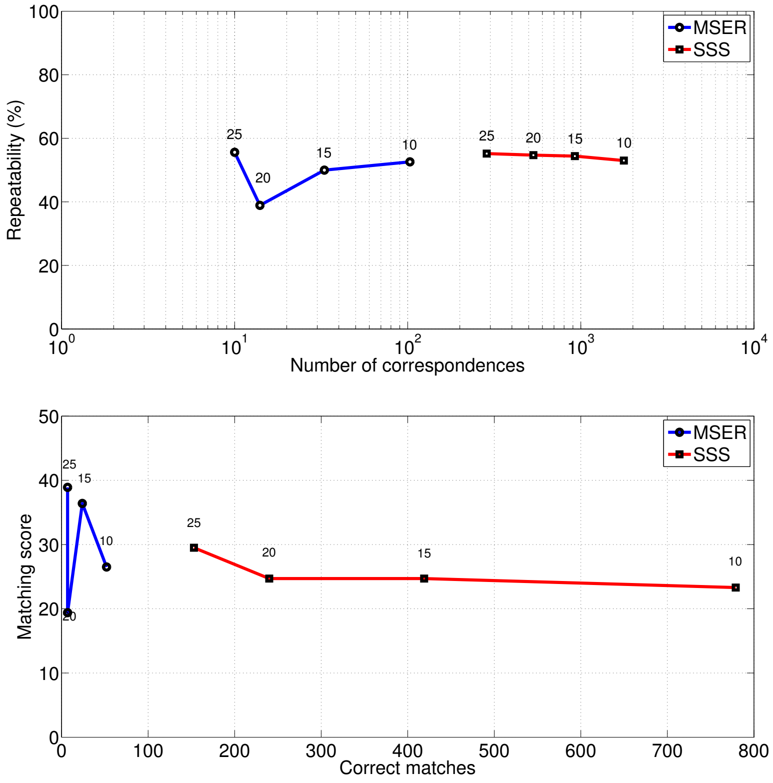


Figure 5.22: Repeatability and matching results for the Bikes sequence (third image), with an overlap error of 20%. Top row: number of correspondences vs. repeatability score for different stability thresholds (10, 15, 20, 25), bottom row: number of matches vs. matching score for different stability thresholds (10, 15, 20, 25).

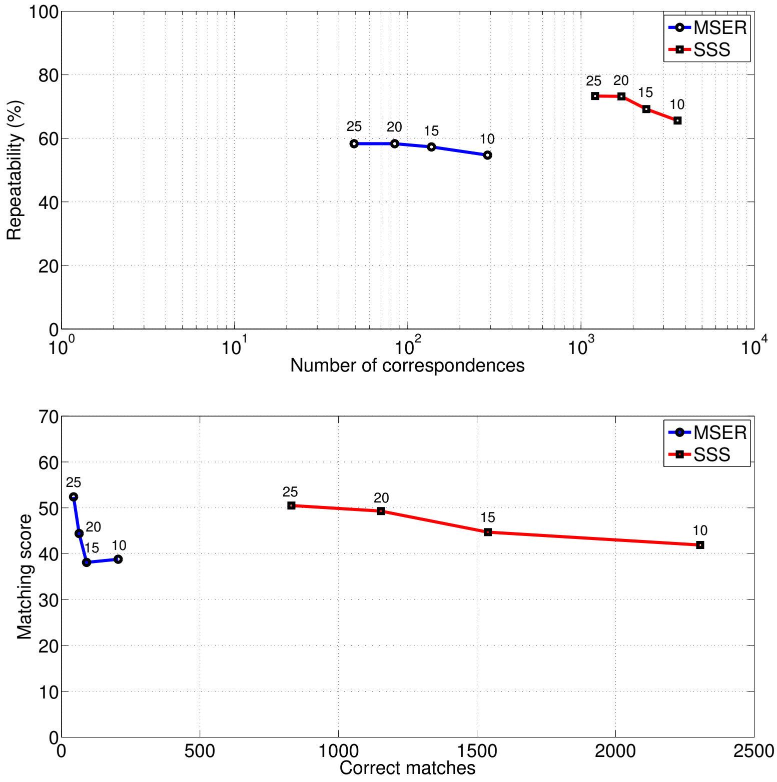


Figure 5.23: Repeatability and matching results for the UBC sequence (third image), with an overlap error of 20%. Top row: number of correspondences vs. repeatability score for different stability thresholds (10, 15, 20, 25), bottom row: number of matches vs. matching score for different stability thresholds (10, 15, 20, 25).

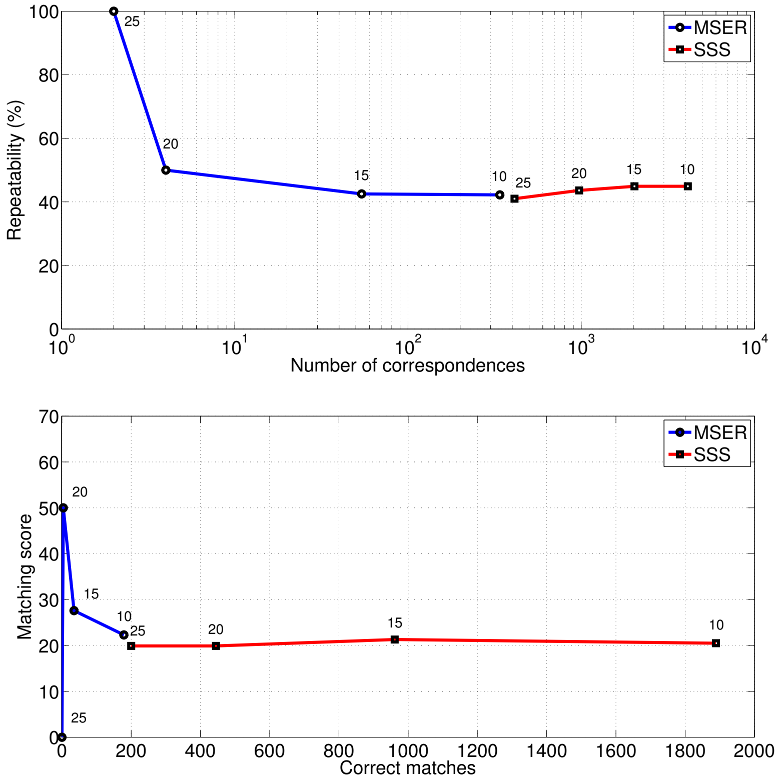


Figure 5.24: Repeatability and matching results for the Wall sequence (third image), with an overlap error of 20%. Top row: number of correspondences vs. repeatability score for different stability thresholds (10, 15, 20, 25), bottom row: number of matches vs. matching score for different stability thresholds (10, 15, 20, 25).

5.3.2 *Completeness evaluation*

The results reported in the previous subsection suggest that in comparison with MSER, SSS regions provide a better coverage of the content. However, a better coverage of the content does not always mean a better coverage of relevant image content. We compared the completeness values of MSER and SSS using the third image of each sequence of the Oxford dataset as the input. For a more detailed analysis, we included several subsets of SSS features, namely edge-SSS (MSER on the edge-map), edge-SSS+ (MSER+ on the edge map), ridge-SSS (MSER on the ridge map), ridge-SSS+ (MSER+ on the ridge map). Note that edge-SSS+ features and standard MSER are expected to coincide in some cases, since the former correspond to darker regions in the edge map. We also computed the completeness values for the combined SSS+MSER detector to analyze the complementarity between these two types of features.

One important conclusion to be drawn from the results in Table 5.25 is that either edge-SSS detection or ridge-SSS detection will provide us a more complete set of features than the one comprised of standard MSER. When the number of regions is similar, edge-SSS and ridge-SSS are more complete than MSER. The combination of MSER and SSS features gives us the most complete feature sets for each sequence. However, the complementary between both feature sets is practically non-existent; the completeness values for SSS $+$ MSER are comparable to the ones for SSS sets, i. e., the relevant content preserved by standard MSER is also preserved by SSS.

5.4 PLAUSIBLE APPLICATIONS

SSS detection represents an improvement over MSER detection in several aspects, namely in terms of robustness against blurring and in terms of the completeness of features. Despite some differences in terms of performance, the results presented in §5.3 suggest that both methods share similar application domains.

Matching tasks are sometimes based in MSER detection (e. g., Matas et al., 2002; Forssén & Lowe, 2007). SSS could also support this task. If SSS were used, the main advantage would be a higher number of correspondences. On the downside, SSS detection would require a slightly higher computational load. Note that the relatively lower repeatability rate of the SSS detector in the presence of more severe geometric distortions would not be a significant shortcoming, as the number of matches provided by this method tends to be substantially

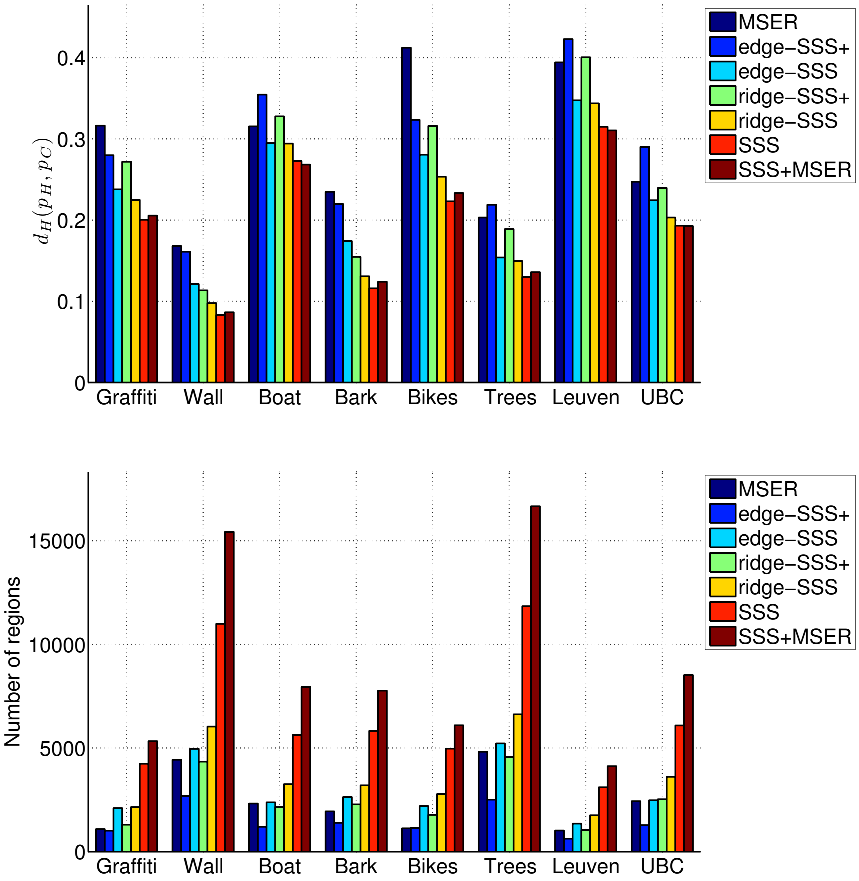


Figure 5.25: Completeness results. Top row: Average dissimilarity measure $d_H(p_H, p_c)$ for the different sets of features extracted over the categories of the dataset. Bottom row: Average number of extracted features per image category.

higher, which compensates for the lower repeatability rate.

For recognition/classification tasks, both methods would also be suitable. MSER has been successfully used in object recognition (e. g., Nistér & Stewénius, 2006). However, as already shown by Mikolajczyk et al. (2005), MSER detection has a poor performance in object class recognition. The denser and more complete sets provided by the SSS detector make it more suitable for object class recognition. A denser and more complete SSS feature set usually contains overlapping regions exhibiting some variation. This variation provides multiple descriptions of a same pattern, which is particularly useful to deal with the large intraclass-variations in the same area (Deng et al., 2007).

5.5 CONCLUDING REMARKS

In this chapter, we addressed the major shortcomings of MSER detection as well as the desired properties of an image in order to provide a reliable MSER detection. As result, we introduced a novel type of features, Stable Salient Shapes (SSS), which are the result of performing a feature-driven MSER detection. At the first stage, we construct saliency maps that will be used as domains for MSER detection. These maps are characterized by the highlighting of features related to semantically meaningful structures, e. g., boundaries, and the simultaneous presence of smooth transitions at the boundaries. Our algorithm overcomes significant limitations of a standard MSER detection, namely the sensitivity to image blur, the presence of a reduced number of regions, and the biased preference towards regular shapes.

The experimental validation on the Oxford repeatability benchmark showed that SSS are comparable to the most prominent affine covariant regions in terms of repeatability score. Concerning the absolute repeatability, our algorithm compares favorably to state-of-the art solutions. Moreover, our solution is efficient; it combines an already efficient MSER detection with a computationally inexpensive image filtering.

The new set of features is more complete as it preserves more of the relevant image content. The high level of completeness shown by the feature-driven MSER is a major improvement over MSER. In addition, SSS detection tends to preserve most of the regions retrieved by the MSER detector as evidenced by the complementarity results of our evaluation.

As future work, we intend to improve the accuracy and the repeatability of the SSS detector. A possible solution is to replace the linear

Gaussian scale-space with a non-linear one. The Gaussian scale-space suffers from the drawback of indiscriminately smoothing the boundaries and noise. While our method requires smoothness at the boundaries, the blurring effect should not be too excessive. With a non-linear scale space, we can perform a locally adaptive smoothing that preserves boundaries (Alcantarilla et al., 2012).

6 CONCLUSIONS AND PERSPECTIVES

6.1 CONCLUSIONS

Local feature detection has been a central and extremely active research topic in the fields of computer vision and image analysis. The role played by local features in a number of tasks is undeniably fundamental. Well-known and reliable solutions to prominent problems such as wide-baseline stereo matching, content-based image retrieval, object (class) recognition, just to name a few, are based on local features.

The more than three decades of research and the indisputable success of local features on the resolution of prominent problems explain the tendency to regard this topic as a mature one with a small room for improvement. However, we can only draw these conclusions if we perform a superficial analysis of the subject by disregarding relevant aspects such as the existence of a reductive and biased evaluation, the relative covariance of local features with respect to geometric transformations, or the less-studied and -discussed complementarity between features. Some of these aspects were the main motivation for this dissertation.

We started by conducting a literature overview on local feature detection. Along with this overview, we brought into discussion fundamental open issues in current local feature detection research. A more careful analysis of local feature detection shows that evolution of local feature detectors has been mainly based on the improvement of repeatability, especially by adding new types of covariance or invariance. Despite the importance of having repeatable features in the presence of a large class of image transformations, there are other requirements that are often overlooked. For example, various applications make use of local features in order to obtain a robust image representation. The so-called robust image representation is only achieved if features cover the most informative parts of the image. Local features are, by definition, informative parts of an image. However, algorithms are not explicitly designed to extract features that cover the most informative parts. Combining complementary features is a relatively viable solution to achieve a better coverage of the informative parts. Nonetheless, the complementarity between features has been barely exploited and the studies on complementarity are rare. To our knowledge, the work of Dickscheid et al. (2011) is the only study that makes a more comprehensive analysis of it.

The evaluation of local features is another topic that we covered in this dissertation. Thus far, evaluation has been mainly based in the repeatability criterion. which is not sufficient to reflect the usefulness of features. While the goal of our research is not define a viable benchmark for local feature evaluation, our experimental validation relies on a wider range of criteria, including completeness and complementarity, in order to formulate more conclusive ideas about the usefulness of local features. This gives us an idea on the suitability of a given type of features for different categories of applications. A more conclusive evaluation would be achieved if the local features were used in different applications. However, this approach would be considerably more exhaustive and it would require various parameter tunings.

Our major contributions came in the form of two algorithms for local feature extraction. While being considerably different, both algorithms extract features which are expected to provide a robust image representation by capturing most image information. Requirements such as repeatability and accuracy were equally taken into consideration.

The first algorithm, coined as Context-Aware Keypoint Extractor (or CAKE), represents a new paradigm in local feature extraction. The algorithm is formulated under an information theoretic framework and it retrieves salient (highly informative) locations within the image context, which means no assumption is made on the type of structure to be detected. This scheme is explicitly designed to provide a robust image representation, with or without the contribution of other local features. Various instances of the method can be created, as different local representations (local descriptors) can be used to generate different context-aware feature extractors. In addition, the computational cost of estimating the probability density function is considerably reduced in our method (information theoretic-based methods are known for their high computational complexity).

We proposed two initial instances of the context-aware detector: [eigSTM]-CAKE, an instance based on the eigenvalues of the image structure tensor matrix, and [HES]-CAKE, which is based on the components of the Hessian matrix computed at multiple scales. Both instances retrieve features that cover informative parts, either at a local level or at a global level. The former is a simple instance based on a codeword with only two components. Despite the simplicity, [eigSTM]-CAKE features can efficiently capture informative image parts, especially in terms of the image details. The use of second order derivatives allows the Hessian-based instance to provide a more complete coverage of informative content. It captures structures that carry most image information, such as blobs, and also the structures where the fine details of the image can be found.

We compared the repeatability and accuracy of [eigSTM]-CAKE against the ones of Shi-Tomasi, Rohr, and Noble keypoint detectors. For the second instance, the direct counterparts were Hessian-Laplace, Harris-Laplace, MSER, SFOP, and Salient Regions. [eigSTM]-CAKE and its counterparts showed a comparable performance. As for [HES]-CAKE, it showed a similar performance to SFOP. However, the former provides a substantially higher number of features.

The second algorithm extracts Stable Salient Shapes (or SSS), a novel type of features which are obtained through a feature-driven Maximally Stable Extremal Regions detection. The advantages as well as the disadvantages of MSER detection were the main motivation for the design of the feature-driven MSER detector. The major advantages in MSER detection are the efficiency of the method as well as the repeatable and accurate features retrieved by the method when dealing with well-structured scenes. However, MSER detection suffers from several drawbacks, namely the lack of robustness against blur, the preference for round shapes, and the reduced number of features. In the particular case of this type of features, the reduced number of features does not mean significantly low completeness values. In fact, the MSER algorithm tends to cover informative image parts, which explains its moderate completeness values and the success in performing specific object recognition. Our algorithm tries to overcome the aforementioned shortcomings and simultaneously retain the advantages for which MSER detection is known for.

The idea behind the feature-driven approach is to provide suitable domains for MSER detection. These domains are saliency maps in which features related to semantically meaningful structures, e.g., boundaries and symmetry axes, are highlighted and simultaneously delineated under smooth transitions. In comparison with MSER, Stable Salient Shapes are more robust to blur and show substantially higher completeness values. Another relevant aspect of this feature-driven MSER detection is the regularization effect that the Gaussian smoothing has on the shapes, which reduces the preference of MSER towards round shapes.

6.2 PERSPECTIVES

The recent advances in local feature description have strongly supported the idea of using local feature detection as a tool to provide robust and compact image representations. Many applications have benefited from these representations (e. g., object recognition) and certainly many more will.

Our research work was mainly focused on the problem of providing robust image representations. With this goal in view, we discussed the completeness and complementarity of local features and proposed algorithms aimed at providing a robust image representation, without neglecting other requirements.

We can see the introduction of context-aware features as our major contribution. For tasks requiring a robust image representation, the use of context-aware features is undoubtedly a viable option. In the particular case of recognition tasks, context-aware features appear as valid alternative to densely sampled descriptors.

As future work, it is our intention to exploit optimized techniques based on context-aware feature extraction for object (class) recognition. We believe that this is an application domain that will strongly benefit from the use of context-aware local features.

As for the SSS detector, a new version based on a non-linear scale space will be studied. The idea is to find a computationally efficient representation that preserves details (the boundaries) and simultaneously reduces noise in order to replace the original Gaussian scale-space representation that indiscriminately smoothes details and noise at a similar level. The motivation for this approach is to improve the accuracy and the repeatability of our feature-driven detector.

Symmetry detection is an application domain in which we intend to exploit the use of our SSS detector. Using structural information to delineate features is a suitable technique for accurately finding symmetrical regions. The promising results obtained with the similar PCBR detector (Deng et al., 2007) are also a motivation.

BIBLIOGRAPHY

Aanæs, H., Lindbjerg Dahl, A., & Pedersen, K. S. (2012). Interesting Interest Points: A Comparative Study of Interest Point Performance on a Unique Data Set. *International Journal of Computer Vision*, *97*(1), 18–35.

Alcantarilla, P. F., Bartoli, A., & Davison, A. J. (2012). KAZE Features. In *Proceedings of the 12th European Conference on Computer Vision (ECCV'12)*, (pp. 214–227).

Attneave, F. (1954). Some Informational Aspects of Visual Perception. *Psychological Review*, *61*(3), 183–193.

Baumberg, A. (2000). Reliable Feature Matching across Widely Separated Views. In *Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, volume 1, (pp. 774–781).

Beaudet, P. R. (1978). Rotationally Invariant Image Operators. In *Proceedings of the 4th International Joint Conference on Pattern Recognition (ICPR'78)*, (pp. 579–583).

Bigün, J. (1990). A Structure Feature for Some Image Processing Applications Based on Spiral Functions. *Computer Vision, Graphics and Image Processing*, *51*(2), 166–194.

Bosch, A., Zisserman, A., & oz, X. M. (2007). Image Classification Using Random Forests and Ferns. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV'07)*, (pp. 1–8).

Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers*. New York, NY, USA: Dover.

Bruce, N. (2005). Features that Draw Visual Attention: An Information Theoretic Perspective. *Neurocomputing*, *65–66*, 125–133.

Canny, J. (1986). A Computational Approach to Edge Detetction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–698.

Cordes, K., Rosenhahn, B., & Ostermann, J. (2011). Increasing the Accuracy of Feature Evaluation Benchmarks Using Differential Evolution. In *Proceedings of the 2011 IEEE Symposium on Differential Evolution (SDE'11)*, (pp. 1–8).

Deng, H., Zhang, W., Mortensen, E., Dietterich, T., & Shapiro, L. (2007). Principal Curvature-Based Region Detector for Object

Recognition. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'07)*, (pp. 1–8).

Dickscheid, T. (2011). *Robust Wide-Baseline Stereo Matching for Sparsely Textured Scenes*. PhD thesis, University of Bonn.

Dickscheid, T., Schindler, F., & Förstner, W. (2011). Coding Images with Local Features. *International Journal of Computer Vision*, *94*(2), 154–174.

Donoser, M. & Bischof, H. (2006). 3d Segmentation by Maximally Stable Volumes (MSVs). In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, (pp. 63–66).

Dorkó, G. & Schmid, C. (2003). Selection of Scale-Invariant Parts for Object Class Recognition. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV'03)*, (pp. 634–640).

Elkan, C. (2003). Using the Triangle Inequality to Accelerate k-means. In *Proceedings of the 20th International Conference on Machine Learning (ICML'03)*, (pp. 147–153).

Fei-Fei, L., Fergus, R., & Perona., P. (2004). Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In *Workshop on Generative-Model Vision*, (pp. 178–178).

Fend, H. Y. & Pavlidis, T. (1973). Finding "Vertices" in a Picture. *Computer Graphics and Image Processing*, *2*(2), 103–117.

Forssén, P.-E. (2007). Maximally Stable Colour Regions for Recognition and Matching. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'07)*, (pp. 1–8).

Forssén, P.-E. & Lowe, D. (2007). Shape Descriptors for Maximally Stable Extremal Regions. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV'07)*, (pp. 1–8).

Förstner, W. (1986). A Feature Based Correspondence Algorithm for Image Matching. In *International Archives of Photogrammetry and Remote Sensing*, volume 26, (pp. 150—166).

Förstner, W. (1994). A Framework for Low Level Feature Extraction. In *Proceedings of the European Conference on Computer Vision (ECCV'94)*, volume 3, (pp. 383–394).

Förstner, W., Dickscheid, T., & Schindler, F. (2009a). Detecting Interpretable and Accurate Scale-Invariant Keypoints. In *Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV'09)*, (pp. 2256–2263).

Förstner, W., Dickscheid, T., & Schindler, F. (2009b). On the Completeness of Coding with Image Features. In *Proceedings of the British Machine Vision Conference 2009 (BMVC'09)*.

Gilles, S. (1998). *Robust Description and Matching of Images*. PhD thesis, University of Oxford.

Goferman, S., Zelnik-Manor, L., & Tal, A. (2012). Context-Aware Saliency Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(10), 1915–1926.

Grauman, K. & Leibe, B. (2011). *Visual Object Recognition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Harris, C. & Stephens, M. (1988). A Combined Corner and Edge Detector. In *Proceedings of the 4th ALVEY Vision Conference*, (pp. 147–151).

Julész, B. & Bergen, J. R. (1983). Textons, the Fundamental Elements in Preattentive Vision and the Perception of Textures. *Bell System Technical Journal*, *62*(6), 1619–1645.

Kadir, T. & Brady, M. (2001). Saliency, Scale and Image Description. *International Journal of Computer Vision*, *45*(2), 83–105.

Kadir, T., Zisserman, A., & Brady, M. (2004). An Affine Invariant Salient Region Detector. In *Proceedings of the 8th European Conference on Computer Vision (ECCV'04)*, (pp. 228–241).

Kenney, C. S., Manjunath, B. S., Zuliani, M., Hewer, G., & Nevel, A. V. (2003). A Condition Number for Point Matching with Application to Registration and Post-Registration Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(11), 1437–1454.

Kenney, C. S., Zuliani, M., & Manjunath, B. (2005). An Axiomatic Approach to Corner Detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, (pp. 191–197).

Kimmel, R., Zhang, C., Bronstein, A., & Bronstein, M. (2011). Are MSER Features Really Interesting? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(11), 2316–2320.

Koffman, E. B. & Wolfgang, P. A. T. (2007). *Objects, Abstraction, Data Structures and Design: Using C++*, chapter 12. New York, NY, USA: John Wiley & Sons, Inc.

Kokkinos, I., Maragos, P., & Yuille, A. (2006). Bottom-Up & Top-down Object Detection Using Primal Sketch Features and Graphical Models. In *Proceedings of the 2006 IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition (CVPR'06), volume 2, (pp. 1893–1900).

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, (pp. 2169–2178).

Li, F. & Perona, P. (2005). A Bayesian Hierarchial Model for Learning Natural Scene Categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, (pp. 524–531).

Lillholm, M., Nielsen, M., & Griffin, L. D. (2003). Feature-based Image Analysis. *International Journal of Computer Vision*, *52*(2/3), 73–95.

Lindeberg, T. (1993). Detecting Salient Blob-like Image Structures and Their Scales with a Scale-Space Primal Sketch: A Method for Focus-of-Attention. *International Journal of Computer Vision*, *11*(3), 283–318.

Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers.

Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, *30*, 79–116.

Lindeberg, T. & Gårding, J. (1997). Shape-Adapted Smoothing in Estimation of 3-D Depth Cues from Affine Distortions of Local 2-d Structures. *Image and Vision Computing*, *15*.

Liu, C., Yuen, J., & Torralba, A. (2011). SIFT Flow: Dense Correspondence across Scenes and Its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 978–994.

Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV'99)*, (pp. 1150–1157).

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*, 91–110.

Loy, G. & Eklundh, J.-O. (2006). Detecting Symmetry and Symmetric Constellations of Features. In *Proceedings of the 9th European Conference on Computer Vision (ECCV'06)*, (pp. 508–521).

Marr, D. (1982). *Vision*. San Francisco: W. H. Freeman and Company.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2002). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *Proceedings of the British Machine Vision Conference 2002 (BMVC'02)*, (pp. 384–393).

Mikolajczyk, K., Leibe, B., & Schiele, B. (2005). Local Features for Object Class Recognition. In *Proceedings of the 10th IEEE International Conference on Computer Vison (ICCV'05)*, (pp. 1792–1799).

Mikolajczyk, K., Leibe, B., & Schiele, B. (2006). Multiple Object Class Detection with a Generative Model. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, (pp. 26–36).

Mikolajczyk, K. & Schmid, C. (2001). Indexing Based on Scale Invariant Interest Points. In *Proceedings of the 8th IEEE International Conference on Computer Vison (ICCV'01)*, volume 1, (pp. 525–531).

Mikolajczyk, K. & Schmid, C. (2002). An Affine Invariant Interest Point Detector. In *Proceedings of the 7th European Conference on Computer Vision (ECCV'02)*, volume I, (pp. 128–142).

Mikolajczyk, K. & Schmid, C. (2004). Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, *60*(1), 63–86.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Gool, L. V. (2005). A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, *65*(1/2), 43–72.

Mirmehdi, M. & Periasamy, R. (2001). CBIR with Perceptual Region Features. In *Proceedings of the British Machine Vision Conference 2001 (BMVC'01)*, (pp. 511–520).

Moravec, H. (1980). Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Technical Report CMU-RI-TR-80-03, Carnegie Mellon University.

Moravec, H. P. (1977). Towards Automatic Visual Obstacle Avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI'77)*, (pp. 584–584).

Moreels, P. & Perona, P. (2007). Evaluation of Features Detectors and Descriptors Based on 3D Objects. *International Journal of Computer Vision*, *73*(3), 263–284.

Mühlich, M. & Aach, T. (2007). High Accuracy Feature Detection for Camera Calibration: A Multi-Steerable Approach. In *Proceedings of the 29th Annual Symposium of the German Association for Parttern Recognition (DAGM'07)*.

Nistér, D. & Stewénius, H. (2006). Scalable Recognition with a Vocabulary Tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, (pp. 2161 – 2168).

Noble, A. (1989). *Descriptions of Image Surfaces*. PhD thesis, Department of Engineering Science, University of Oxford.

Parida, L., Geiger, D., & Hummel, R. (1998). Junctions: Detection, Classification and Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(7), 687–698.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, *33*(3), 1065–1076.

Perdóch, M., Matas, J., & Obdržálek, S. (2007). Stable Affine Frames on Isophotes. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV'07)*, (pp. 1–8).

Pritchett, P. & Zisserman, A. (1998). Wide Baseline Stereo Matching. In *Proceedings of the 6th IEEE International Conference on Computer Vison (ICCV'98)*, (pp. 754–760).

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81–106.

Rohr, K. (1997). On 3D Differential Operators for Detecting Point Landmarks. *Image Vision Computing*, *15*(3), 219–233.

Rosenfeld, A. & Thurston, M. (1971). Edge and curve detection for digital scene analysis. *IEEE Transactions on Computers*, *C–20*(5), 562–569.

Rosten, E. & Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. In *Proceedings of the 9th European Conference on Computer Vision (ECCV'06)*, (pp. 430–443).

Rosten, E., Porter, R., & Drummond, T. (2010). Faster and Better: A Machine Learning Approach to Corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(1), 105–118.

Schmid, C. & Mohr, R. (1997). Local Grayvalue Invariants for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(5), 530–535.

Schmid, C., Mohr, R., & Bauckhage, C. (2000). Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, *37*(2), 151–172.

Schnitzspan, P., Roth, S., & Schiele, B. (2010). Automatic Discovery of Meaninful Objects Parts with Latent CRFs. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'10)*, (pp. 121–128).

Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, *27*, 379–423.

Shi, J. & Tomasi, C. (1994). Good Features to Track. In *Proceedings of the 1994 IEEE Computer Socitey Conference on Computer Vision and Pattern Recognition (CVPR'94)*, (pp. 593–600).

Smith, S. M. (1992). A New Class of Corner Finder. In *Proceedings of the 1992 British Machine Vision Conference (BMVC'92)*, (pp. 139–148).

Smith, S. M. (1996). Flexible Filter Neighbourhood Designation. In *Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96)*, volume 1, (pp. 206–212).

Smith, S. M. & Brady, J. M. (1997). SUSAN–A New Approach to Low Level Image Processing. *International Journal of Computer Vision*, *23*(1), 45–78.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.

Triggs, B. (2004). Detecting Keypoints with Stable position, Orientation and Scale under Illumination Changes. In *Proceedings of the 8th European Conference on Computer Vision (ECCV'04)*, (pp. 100–113).

Tuytelaars, T. & Gool, L. V. (1999). Content-based Image Retrieval based on Local Affinely Invariant Regions. In *Proceedings of the International Conference on Visual Information Systems*, (pp. 493–500).

Tuytelaars, T. & Gool, L. V. (2000). Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions. In *Proceedings of the British Machine Vision Conference 2000 (BMVC'00)*, (pp. 412–425).

Tuytelaars, T. & Gool, L. V. (2004). Matching Widely Separated Views based on Affine Invariant Regions. *International Journal of Computer Vision*, *59*(1), 61–85.

Tuytelaars, T., Gool, L. V., D'haene, L., & Koch, R. (1999). Matching of Affinely Invariant Regions for Visual Servoing. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA'99)*, (pp. 1601–1606).

Tuytelaars, T. & Mikolajczyk, K. (2008). Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*, *3*(3), 177–280.

Vedaldi, A. & Fulkerson, B. (2008). VLFeat: An Open and Portable Library of Computer Vision Algorithms. http://www.vlfeat.org/.

Witkin, A. P. (1983). Scale-Space Filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI'83)*, volume 2, (pp. 1019–1022).

Zuliani, M., Kenney, C., & Manjunath, B. S. (2004). A Mathematical Comparison of Point Detectors. In *Second IEEE Image and Video Registration Workshop (IVR)*, (pp. 172–172).