TECHNISCHE UNIVERSITÄT WIEN
FAKULTÄT FÜR INFORMATIK

D I S S E R T A T I O N

Scale and Rotation Invariant Shape Matching

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

a.o.Univ.Prof. Dipl.-Ing. Dr. techn. **Robert Sablatnig**
Dipl.-Ing. Dr. techn. **Allan Hanbury**
183/2
Institut für rechnergestützte Automation
Arbeitsgruppe Mustererkennung und Bildverarbeitung

eingereicht an der Technischen Universität Wien
Fakultät für Informatik

von

Dipl.-Ing. **Lech Szumilas**

Matrikelnummer 0527953

1100 Wien, Schröttergasse 34/7

Wien, am 22.02.2008

Lech Szumilas

Abstract

Recognition of objects from images is one of the central research topics of computer vision. The use of shape for recognizing objects has been actively studied since the beginning of object recognition in 1950s. Several authors suggest that object shape is more informative than its appearance – the object appearance properties such as texture and color vary between object instances more than the shape e.g. bottle, caps, cars, airplanes, cows, horses etc. Recent methods are concentrated on extracting shape features and learning the object models directly from images which impose such problems as object occlusion, incomplete and often fragmented object boundaries, varying camera view-points. While these approaches are designed to learn object models from fragmented and incomplete object boundaries, achieving invariance to rotation, scale and affine transformations has not been fully solved.

This thesis address the problem of learning object models that use shape properties with full rotational and scale invariance. A new approach is proposed where invariance to image transformations is obtained through invariant matching rather than typical invariant features. This philosophy is especially applicable to shape features, represented by edges detected in images which do not have a specific scale or specific orientation until assembled into an object. Our primary contributions are: a new shape-based image descriptor that encodes a spatial configuration of edge parts, a technique for matching descriptors that is rotation and scale invariant and shape clustering that can extract frequently appearing image structures from training images without a supervision.

This thesis also presents an overview of the object recognition field and our other contributions in the area of local appearance based methods, texture detection and image segmentation.

Keywords: object recognition, shape, image descriptors, interest points.

Acknowledgments

I would like to thank my wife for her help, an extraordinary patience and support during the research that led to completion of this thesis.

I am most grateful to my advisors Prof. Robert Sablatnig and Dr. Allan Hanbury for their support, advice and attention to details. This thesis would not exist without their help. Thanks to Dr. Allan Hanbury and Prof. Walter Kropatsch for believing in me and making my work in the Pattern Recognition and Image Processing Group possible.

I have spent endless hours discussing the topics described in this thesis with my colleagues. They helped me with their knowledge, interest and kindness, inspired me as well as pointed out my mistakes. I am especially thankful to Dr. Allan Hanbury, Prof. Robert Sablatnig, Prof. Walter Kropatsch, Dr. Branislav Mičušik, Dr. Georg Langs, Horst Wildenauer, René Donner, Adrian Ion and Julian Stöttinger.

I was lucky to have the opportunity of visiting the Intelligent Sensory Information Systems laboratory at the University of Amsterdam. I am grateful to Dr. Nicu Sebe and Prof. Theo Gevers for making it possible as well as for their advice and kindness during my stay in Amsterdam. I have gained invaluable experience and inspiration for my future research.

I thank all at the Pattern Recognition and Image Processing Group for providing a friendly environment to work in.

My research was sponsored by the Austrian Science Foundation (FWF) grant SESAME (P17189-N04) and the European Union Network of Excellence MUSCLE (FP6-507752).

The medical image data was provided by the Department of Radiology at the General Hospital Vienna, Medical University of Vienna.

Contents

Chapter 1

Introduction to Object Recognition

Object recognition is one of the central research topics in the computer vision field. It is also one of the most actively studied subjects with many real applications in such areas as security, surveilance, medicine, agriculture, document analysis, traffic and transport, image and video retrieval and others [14, 53]. The primary goal of object recognition is detection and localization of arbitrary objects in an image [53]. Although the research in this area started in 1950s, resulting in a large number of different approaches, the existing methods are far from matching the abilities of human vision. Many currently available methods are tailored toward solving a specific problem, such as face recognition [69], vehicle and pedestrian recognition [29], detection of anatomical structures in medical images [9] or achieve good performance only on a limited number of objects and images [53]. Finding a general solution would allow the employment of a single approach for all applications, although the analysis of methods presented in the following chapters indicates that such a solution would have to use several cues to recognize different objects e.g. selective use of local appearance, texture and shape features for discriminating between different object types.

This thesis concentrates on the use of shape features for learning and detecting object models from images. The applicability of shape properties has been studied since the introduction of early object recognition methods e.g. "Block World" [53] and is also a topic of current research (see Chapter 3). Recent methods [13, 33, 51] address the problem of learning object models from fragmented, cluttered and incomplete object boundaries extracted from images. These approaches are able to separate fragments of object bound-

aries from the background or random appearances and build an object model that encodes the spatial relationship between boundary fragments. However, the problem of invariance to geometric transformations such as object rotation, scale change or projective transformation has no general solution. This means that the learning of object models is limited to object instances that share scale and orientation [33] or orientation [13].

The primary contribution of this thesis is a shape-based image descriptor that is capable of rotational and scale invariant matching of structures in images (see Chapter 4). The proposed method allows to learn repeatable shape structures in images, even if they differ by scale and orientation (see Chapter 5). For example it is possible to extract models of bone contours in the human hand and other anatomical structures from a set of training images without supervision, as it is shown in Chapter 5. The important difference between this approach and other methods presented in Chapter 3 is that invariance to geometric transformations is achieved through invariant matching and not invariant features. This methodology is especially applicable to features such as edges detected in images which on their own do not represent a particular scale or orientation. Invariant matching attempts to find the transformation that produces the optimal fit between multiple corresponding boundary fragments in two matched image descriptors and thus does not require invariant features. The unsupervised learning of objects or object parts from training images can be applied in medical image analysis, replacing time consuming manual object annotation [67].

1.1 Overview of Object Recognition

The noun *object* is used to express a "thing that you can see or touch"* or "anything that is visible or tangible and is relatively stable in form"† or "something material that may be perceived by the senses"‡. The notion of *object* is related to a 3-dimensional material thing that can be *seen*, which is a crucial property from a computer vision perspective. However, none of the definitions cited precisely discriminates an object from a non-object, nor allows to precisely differentiate between different object types. How is it then possible to create an object recognition approach without knowing what an object really is? As will be explained in the following sections, the object recognition techniques used in computer vision offer guesses rather than precise answers as to what objects are visible in the scene.

*Cambridge Advanced Learner's Dictionary
†Dictionary.com
‡Merriam-Webster's Online Dictionary

The problem of object recognition in general is mathematically ill-posed i.e. it is impossible to provide a single solution that has 100% certainty, which can be attributed to several reasons:

1. **lack of precise definition** – the notion of an *object* corresponds to a human perception which is a result of individual sensing capabilities, knowledge and experience.

2. **object appearance varies** – the complete knowledge of all possible appearance variations of every object class that one desires to detect is not available in case of methods that learn from a finite set of 2D images. Humans typically do not draw hard boundaries between related object types (e.g. frying pan and pot) which introduces further uncertainty in case of object recognition methods that use human generated ground-truth. Additional complications are caused by deformable objects (such as animals) or possible variations in color and texture. The available solution to these problems is a probabilistic object recognition that estimates probabilities of object occurrence in the scene [13, 33, 49, 51, 57].

3. **3D → 2D projections** – the majority of object recognition approaches in computer vision operate on 2D still images or sequences of 2D images which are representations of the projected 3D world. A 2D appearance of a 3D object changes together with the camera view point with a possibility of multiple object occlusions occurring [47]. The projection of a 3D scene into a 2D image reduces the amount of information available and introduces additional uncertainty for inferring about object types present in the scene [27]. Figure 1.1 shows a set of "peep-hole" perception demonstrations constructed by Ames and his associates [27] in which human subjects viewed three arrangements of wire edges through a peephole. Despite all three 2D projections forming an image of the chair only one of them (left image) was produced by the 3D edge arrangement that form the real chair. In the middle arrangement edges were not parallel in 3D space and in the third arrangement edges even did not co-terminate but produced the same 2D projection in all cases.

In computer vision the characterization of individual object types is either provided manually, through a specification of object type model (e.g. Geometric Hashing, see Section 3.2.1), or as a set of examples e.g. a number of images containing object(s) of interest, from which the characteristic object type properties are extracted in the learning process (examples are provided in Chapter 3). Computer vision provides a family of object recognition approaches that differ by: the feature types used (e.g. local image

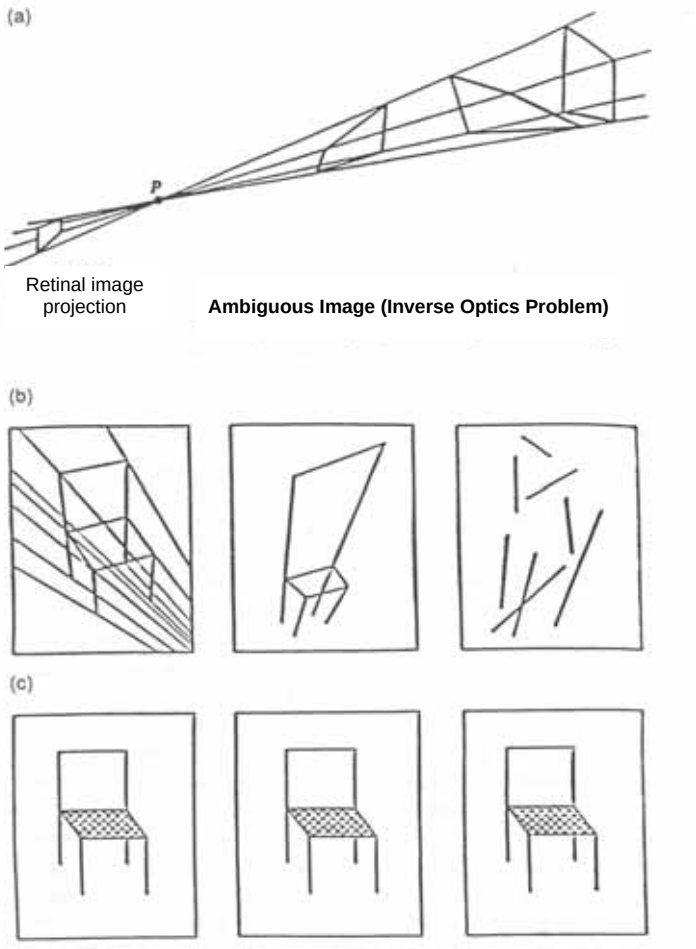


Figure 1.1: The Ames peephole perception demonstrations. The figure was obtained from [27]. (a) Illustration of the inverse optics problem: A single image can be produced by an infinity of possible real-world objects. [27] (b) Three 3D arrangements of wired edges constructed by the Ames group. (c) The retinal projection of the stimuli viewed through the peep hole.

patches or shape features from edges), detection of multiple vs. only single object in the image, ability to localize objects in the image using bounding boxes or more precise object boundaries, tolerance to changes in the camera view point (rotational, scale, affine or projective invariance) and occlusions, learning approach and other characteristics [53]. The performance of the object recognition methods depends also on the number and properties of objects to be learned and detected [74].

The origins of object recognition research goes back to 1950s and early 1960s where advances in signal processing and detection theory inspired first object recognition systems (e.g. character recognition or fingerprint analysis) [47]. The initial period was followed by the attempts to discover 3D properties of the objects in the 2D representation of the scene by a bottom-up object boundary analysis using a set of generic rules. Objects were assumed to be a combination of simple parts (e.g. polyhedra, cylinders), while no attention was given to difficulties occurring in real images, such as low contrast leading to poor edge detection, background clutter, occlusions etc. [47]. This in turn led to the realization that 2D images of real scenes cannot be used to obtain complete reconstruction of a real 3D scene containing complex and arbitrarily distributed objects [47]. The idea of a bottom-up scene segmentation has been replaced by the fragmentary feature segmentations in terms of 2D points, lines or curve segments, however this time the feature segmentation was based on a specific object model. The era of model based recognition began in the 1970s (manual models) and continues till today [28, 47], though present activities concentrate on automatic and semi-automatic model extraction from image examples as well as on the model deformation (see Chapters 3 and 5). The 1980s brought another class of methods, based on intensity and later color appearance [57]. Initial approaches concentrated on the analysis of the whole image (e.g. PCA based methods used for face recognition [69]) which later gave way to a local feature extraction (e.g. interest point detectors and local image descriptors described in Chapters 2 and 3).

1.2 Trends

Currently used object recognition methods can be divided based on the following trends: appearance based methods, correspondence of local features, part based models and shape based detectors [14, 53]. The list does not cover all possible branches of object recognition but rather focuses on those which are oriented at general solutions and are still actively developed.

- **Appearance based methods** – This class of methods operates on image intensity, color and/or features such as intensity gradient, detected corners, convolution filter responses that are computed using image intensity [21]. Traditionally these methods extract global features from the whole image, though subdivision of the image is also used [57]. The two dominant approach types are related to the representation of image features: histograms [57, 62] and eigenvectors of the feature covariance

matrix obtained by Principal Component Analysis (PCA) [11, 48].

The color histogram uses the distribution of quantized colors to describe objects. It maintains low sensitivity to object rotation, scale and even changes in the viewing angle [57]. However it is applicable only to objects that can be distinguished by color alone since information such as shape or texture is not used. A major difficulty is also associated with ensuring the color constancy under illumination changes [16].

PCA is a pattern recognition technique that is used to reduce the dimensionality of the feature space and models Gaussian distribution of feature values [11]. This approach can also be used to represent each image by a small number of coefficients (*eigenpictures*) that can be efficiently stored and searched. This approach however is sensitive to image transformations (translation, scaling, view point changes) and changes in scene illumination [48].

These approaches reached their peak popularity in 1980s and 1990s [53]. The color histogram and the PCA are techniques that are still used at present as subcomponents rather than the primary means of detecting objects in a scene [24, 49].

- **Correspondence of local features** – uses a set of local image appearances, typically extracted from multiple image patches around the *interest points* (see Chapter 2), to describe an object or a scene [53, 74]. The advantage of such approach is that the object boundaries or precise image segmentation are not needed. In theory the object can be recognized using unique local appearance (if such exists), however in practice the object recognition methods use multiple local appearances, also called a bag of keys, to increase accuracy of object classification [74]. As we show in Section 4.2.9 local features alone may not be sufficient to distinguish an object part from the background i.e. not all local features are discriminative.

  Local features are at present one of the dominant trends in the object recognition field. Chapters 2 and 3 provide a detailed description of several well known techniques used for the extraction and matching of local features.

- **Part based models** – correspond to spatial configuration of object parts that are either related to local image appearances or object shape fragments. Part based models attempt to close the gap between local and global object feature representations by building probabilistic global object models from local features, which allows the detection of objects even if not all model parts are present [30].

Sections 3.2.4, 3.2.5 and 3.2.6 provide examples of the shape-based approaches that use the part based representation of an object model. Examples of appearance based approaches can be found in [49].

- **Shape based detectors** – attempt to learn and detect shape properties of objects such as object contours. Recent methods [13, 33, 51] operate on edges extracted from images. Since edge extraction cannot provide complete data about object boundaries, due to occlusions, noise or low image contrast, these methods represent object models as configurations of spatially related boundary fragments which resemble also ideas from part based models. Chapter 3 provides an overview of shape based object detectors and examples where shape based methods outperform others, based on appearance. The major contribution of this thesis is a novel shape-based detector, that can learn frequently appearing structures in the training data without supervision.

Existing work [50, 74] related to feature fusion indicates that combinations of different types of features, such as local appearance and shape features, improves object classification accuracy when compared with methods using a single feature type.

1.3 Thesis Structure

This thesis provides an overview of existing object recognition techniques and introduces novel shape-based image descriptors that are invariant to rotation and scale transformations.

Chapter 2 presents interest point and region detectors that are used by methods based on local image descriptors and the shape based image descriptors introduced in this thesis. The discussion of existing local image descriptors and shape-based object detectors, along with their performance in object recognition tasks, is presented in Chapter 3. These chapters provide background information for a comparison with the local and semi-local descriptors introduced in this thesis.

Chapter 4 presents two novel shape-based image descriptors. Section 4.2 describes the local image descriptor *Orientation-invariant Radial Configuration* (ORC) and provides a comparison of the matching performance with the state of the art SIFT descriptor [38]. Section 4.3 describes the semi-local image descriptor *Radial Edge Configuration* (REC) and evaluates its matching characteristics.

The primary contribution of this thesis is an unsupervised shape-based learning of repeatable structures in the image and their detection, presented in Chapter 5. This

technique uses the REC descriptor and symmetry based interest points discussed in Section 2.5.

Chapter 2

The Role of Interest Points and Interest Regions

2.1 Overview

Interest points and region detectors are used in such areas of computer vision as object recognition [74], matching different views of the same scene [44], texture detection [64], image segmentation [65] and others [53]. Their primary purpose is to detect locations of characteristic structures such as blobs, corners or local image symmetry, independently of the changes in the view point. The differences between the presented interest point algorithms are associated with the type of image structures detected and can be divided based on several popular categories [44, 53, 74]:

- blob detectors – based on the space-scale theory introduced to computer vision by Witkin [73], Koenderink [25] and then extended by Lindeberg [35] based on differential methods such as Laplacian of Gaussians (LoG), difference of Gaussians (DoG) and Determinant of Hessian (DoH) [36]. The modification of these techniques has been also used for a ridge detector [61]. Another technique within the blob detector category but unrelated to scale-space theory is Maximally Stable Extremal Regions (MSER) introduced by Matas et al. [41].

- corner detectors – originate from work by Moravec [45] and Harris [18] and are used to extract local curvature maxima from the intensity gradient. Recent extensions to these methods are described in [43]. Corner detectors are used in motion detection, tracking, object recognition, 3D modeling and image mosaicing [72].

- symmetry detectors – attempt to find rotational and/or mirror symmetry of a 2D intensity distribution in the image regions, popularized by Resifeld et al. [55]. Section 2.5 provides an overview of existing methods [40, 55] and introduces a new symmetry detector called Radial Symmetry Transform.

- salient interest points – Saliency of an image feature can be defined to be inversely proportional to the probability of occurrence of that image feature [17, 23, 60]. Neuroscientists in turn attempt to model human attentional mechanisms which are considered to be the key for learning in the survival of organisms [20].

This chapter discusses approaches related to blob, corner and symmetry detection.

The interest point and region detectors allow to reduce computational complexity in scene matching and object recognition applications by selecting only a subset of image locations corresponding to specific and/or informative structures [53]. Interest region detectors estimate local isotropic and anisotropic scale [44], which is used for matching scenes undergoing affine transformations [56]. The relative position of the interest points can provide additional information which improves matching reliability [52, 56] or can be used to detect complex geometrical structures [10].

The existing comparisons of interest points and region detectors emphasize a repeatability criterium [44] and object classification accuracy [74] as a performance measure. The repeatability of interest points measures the accuracy of interest point localization and scale estimation relative to the detected image structures which influences the performance of the scene matching methods [44]. The accuracy of object classification depends on the consistency of interest point detection, type of local image descriptor chosen, and the learning and classification technique used. It is possible to compare interest point types using the same local image descriptor, learning/classification technique and data, as demonstrated in [74].

Section 2.6 provides a quantitative evaluation of the interest points discussed in Sections 2.3 and 2.4 using a test image introduced in Section 2.2 which contains a combination of structures such as blobs and corners. The purpose of this test is to estimate the consistency of detection of structures for which a particular method is designed (how many were missed), sensitivity to rotations, scale and illumination changes as well as sensitivity to noise.

2.2 Evaluation Methodology

The description of interest point and region detectors presented in this chapter contains examples computed from real images and computed from an artificially generated test image containing combinations of blob and corner structures that are targeted by blob, corner and symmetry detectors. The use of an artificially generated test image allows to quantitatively evaluate consistency and accuracy of interest point positioning and local scale estimation, since the location and scale of all blobs and corners is known (see Section 2.6). The image intensities shown in Figure 2.2 were obtained through the superposition of Gaussian-based functions, which allows the generation of isotropic and anisotropic Gaussian blobs, rectangular blobs and any combinations of the above. The Gaussian-based function has a following form:

$$f(x,y) = h \exp\left(\frac{(\widehat{x}(\alpha) - x_c)^{2f_1}}{2\sigma_1^2} + \frac{(\widehat{y}(\alpha) - y_c)^{2f_2}}{2\sigma_2^2}\right) \tag{2.1}$$

where h is the intensity in the center of the blob at position (x_c, y_c), $f_{\{1,2\}}$ decides about the shape of the blob and is related to the "cornerness" measure introduced in Section 2.4, $\sigma_{\{1,2\}}$ defines the anisotropic scale of the blob and $(\widehat{x}(\alpha), \widehat{x}(\alpha))$ are (x, y) coordinates rotated around the center of the blob by α degrees.

The test image is divided into test regions as shown in Figure 2.1, discussed below:

- A1 – contains isotropic blobs at scales $\sigma_1 = \sigma_2 = \{10, 20, 30, 40\}$ (pixels), $f_1 = f_2 = 1$ and intensities $h = \{0.25, 0.5, 1\}$. It is expected that blob and symmetry detectors precisely locate all blobs while corner detectors find no corners.
- A2 – contains isotropic blobs at identical scales and $f_{\{1,2\}}$ as in A1, intensity $h = 1$ and superimposed white noise of amplitude $0.25I(x, y)$, where $I(x, y)$ is the image intensity at pixel (x, y).
- A3 – contains isotropic square blobs at identical scales as in A1, intensities $h = \{0.5, 1\}$ and $f_1 = f_2 = 2$. It is expected that in addition to blobs, corners will be also detected.
- A4 – same as A3 except $f_1 = f_2 = 4$.
- B1 – contains anisotropic blobs at scales $\sigma_1 = \{10, 20, 30, 40\}$ (pixels), $\sigma_2 = 2\sigma_1$, $f_1 = f_2 = 1$, $h = \{0.5, 1\}$ and $\alpha = \{0, 45, 90, 135\}$ (degrees). It is expected that blob and symmetry detectors precisely locate all blobs while corner detectors find no corners.

- B2 – same as B1 except $f_1 = f_2 = 2$ (square). It is expected that in addition to blobs, corners will be also detected.
- B3 – same as B2 except $f_1 = f_2 = 4$ (square).
- B4 – contains anisotropic rectangular blobs at scales $\sigma_1 = 20$, $\sigma_2 = \{80, 150, 250\}$ (pixels) and $\alpha = \{11, -11, 17\}$ (degrees) respectively. Intensity $h = 1$ and $f_1 = f_2 = 4$. These structures are intended for the evaluation of scale estimation of elongated blobs.
- C1 – contains three structures that are obtained through a superposition of isotropic blobs. The distance between any two blob centers in each structure is smaller than the sum of the corresponding isotropic scales. The resulting structures therefore no longer fit the gaussian blob model used by blob detectors.
- C2 – contains three pairs of isotropic square blobs with $f_1 = f_2 = 4$ and with the distance between their centers smaller than the sum of the corresponding isotropic scales. This setup is intended for the measurement of the interest point drift (discussed in Section 2.3).
- C3 – Contains two identical structures - a result of a superposition of elongated, square blobs. The second structure has superimposed intensity noise of amplitude $0.25I(x, y)$. This setup is intended for the measurement of the interest point drift in the case of blob detectors (discussed in Section 2.3) and the sensitivity of corner detectors.
- C4 – represents a superposition of isotropic and anisotropic blobs. The distance of any two blob centers in each structure is smaller than the sum of the corresponding isotropic scales. This setup is intended for the measurement of the blob detector sensitivity.
- C5 – contains the same structure as in C4 with additional square blobs superimposed. This setup is intended for the measurement of the blob and corner detector sensitivity.

2.3 Blob detectors

Detection of blobs in images has been studied since the 1980s. There are two dominant trends based on differential methods and local intensity or color extrema methods [44].

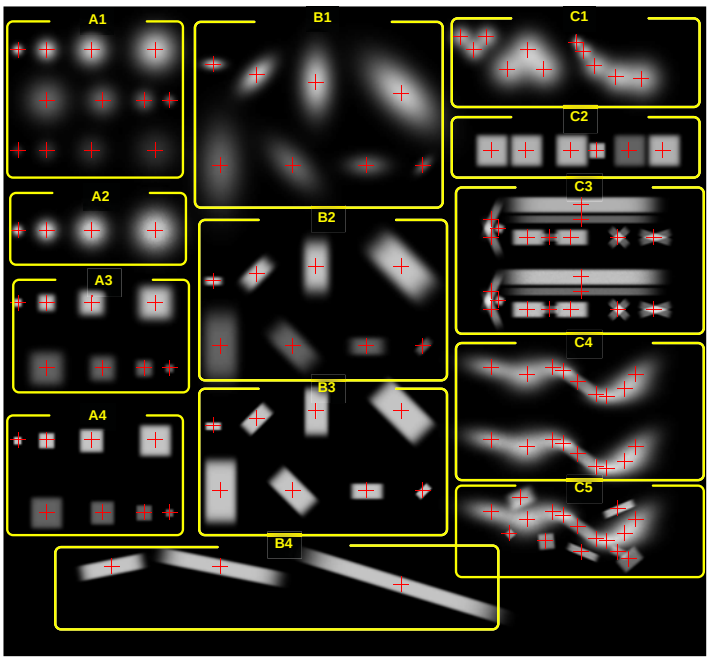


Figure 2.1: Test image used for the evaluation of interest point and region detectors. Red cross markers show centers of gaussian blobs used to generate image structures.

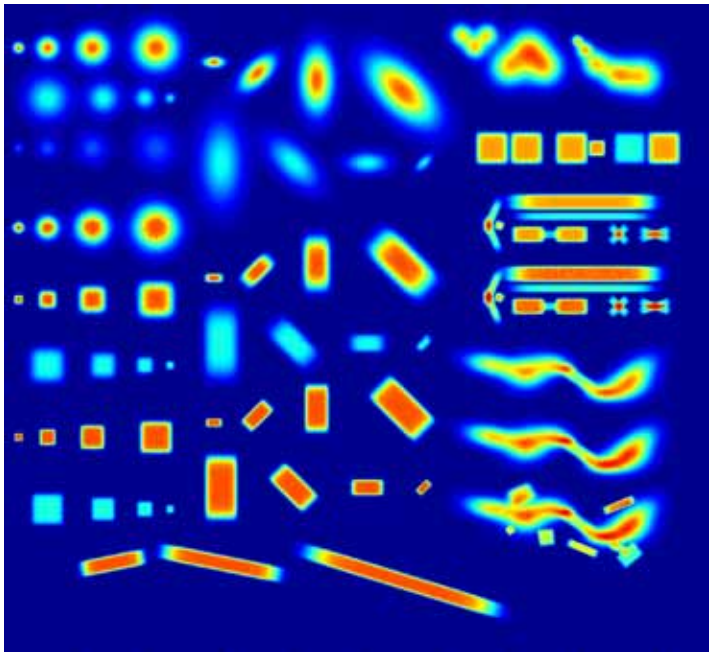


Figure 2.2: The intensity map of the image shown in Figure 2.1. Lower intensities are shown in blue and higher intensities in red.

The differential methods originate from the work of Witkin and Koenderink [25] and later Lindeberg [35]. The scale-space based interest point detectors estimate locations of interest points and the corresponding local scales at the same time. For this reason these methods are also referred to as interest region detectors. The interest region detectors are built on a similar idea of convolving the image multiple times with a blob modeling the kernel function (usually Gaussian) at a set of scales and finding the scale that maximizes the result of the convolution based operator across the space and scale [25, 35].

Lindeberg proposed the use of the Laplacian of Gaussian (LoG) filter for blob detection [35]. It uses an isotropic Gaussian $g(\boldsymbol{x}, \sigma_k)$ function as a blob model:

$$g(\boldsymbol{x}, \sigma_k) = \frac{1}{2\pi\sigma_k} e^{-(\sum_i x_i^2)/(2\sigma_k^2)} \tag{2.2}$$

where σ_k is a scale of the kernel function and i is the index of elements in the vector $\boldsymbol{x} = \{x, y\}$.

The scale-space representation $L(\boldsymbol{x}, \sigma_k)$ is obtained by convolving image I by the Gaussian kernel:

$$L(\boldsymbol{x}, \sigma_k) = g(\boldsymbol{x}, \sigma_k) * I(\boldsymbol{x}) \tag{2.3}$$

This operation is also an equivalent of Gaussian smoothing at scale σ_k.

Finally the Laplacian operator is applied to the convolution result:

$$\nabla^2 L = L_{xx} + L_{yy} \tag{2.4}$$

where L_{xx} and L_{yy} are second order derivatives of the scale space representation. The Laplacian operator produces positive maxima for dark blobs and negative minima for bright blobs of extent $\sqrt{\sigma_k}$. The formal proof of this statement can be found in [35, 36]. To illustrate it better let us apply the operation (2.4) to a Gaussian blob in the continuous 1D domain. The first step is to convolve a 1D infinite image containing Gaussian $g(x, \sigma)$ with a Gaussian kernel $g(x, \sigma)$. The use of a Gaussian blob allows one to obtain a closed form solution:

$$L(\mu) = \int_{-\infty}^{\infty} g(x, \sigma) g(x - \mu, \sigma) dx = e^{-\mu^2/(4\sigma)} \frac{\operatorname{erf}\left(\frac{2x - \mu}{2\sigma}\right)}{2\sqrt{2}} \Bigg|_{-\infty}^{\infty} = e^{-\mu^2/(4\sigma)} \frac{1}{\sqrt{2}} \tag{2.5}$$

Note that for simplicity we have assumed that the blob and Gaussian kernel have the same scale σ, however it is possible to obtain a similar closed form result when the scales differ.

The second step is to apply the Laplacian operator to the convolution result. In this case it is a second order derivative of the convolution result:

$$\nabla L(\mu) = L_{\mu\mu} = e^{-\mu^2/(4\sigma)} \left(-2\sigma + \mu^2\right) \frac{1}{4\sqrt{2}\sigma^4} \tag{2.6}$$

Function (2.6) reaches its minimum at $\mu = 0$ which coincides with the center of the blob, as is shown in Figure 2.3.

So far we have shown how the spatial location of a blob can be detected at a single scale. To obtain a multi-scale blob detector we use a scale normalized Laplacian operator:

$$\nabla_n^2 L = \sigma_k \left(L_{xx} + L_{yy}\right) \tag{2.7}$$

Figure 2.3 shows that scale normalized Laplacian response at $\mu = 0$ and varied kernel scale σ_k attains a minimum when the kernel scale σ_k equals the blob scale σ. Since the blob location corresponds to the extremum of the normalized Laplacian response along spatial coordinate and its scale corresponds to the extremum along the scale coordinate, the blob detection relies on a search in the three dimensional scale-space domain for a local extremum. This requires that the scale normalized Laplacian operator is applied to the convolved input image at a range of scales, which produces a scale-space volume $\nabla_n^2 L(\boldsymbol{x}, \boldsymbol{\sigma})$. The interest point locations $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$ and local scales $\hat{\boldsymbol{\sigma}}$ coincide then with the

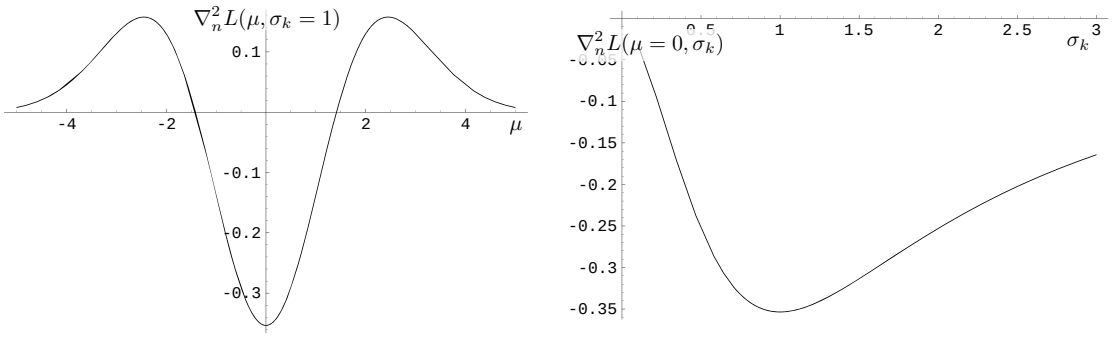


Figure 2.3: Response of the scale normalized version of the Laplacian operator (2.6) for $\sigma_k = 1$, μ varied (left) and for $\mu = 0$, σ_k varied (right). The scale of the blob $\sigma = 1$.

local extrema in the volume:

$$(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{\sigma}}) = \underset{x,y,\sigma}{\text{argminmax}} \left(\nabla_n^2 L\right) \tag{2.8}$$

where "argminmax" is a local extremum detector.

Figures 2.4 and 2.5 contain examples of the blob scale and location detection, using (2.8), for Gaussian and non-Gaussian blobs. Special attention should be given to the asymmetrical blob distribution case in Figure 2.5, where the detection of the larger blob is affected by a drift and the associated scale is underestimated. The drift is caused by the Gaussian smoothing of the asymmetrical features in the signal. The drift progressively increases with the scale as a scale related neighborhood influences the local scale-space representation. The scale underestimation is related to the drift itself, however even in the symmetrical case, scale estimation is biased by using a specific blob model (see Figures 2.6, 2.7 and 2.8). These problems are further discussed in [36] and solutions are provided in [61]. The results also show that not only blobs but also their boundaries produce local extrema in the Laplacian response. The interest points at blob edges exhibit lower scale (see Figure 2.5) and Laplacian response (see two symmetrical maxima in Figure 2.3) than interest points associated with the corresponding blob. In [36] the final set of interest points is selected from the K strongest Laplacian extrema, where K is a manually selected parameter.

As demonstrated in [38], the LoG result can be approximated with the Difference of Gaussians (DoG) at reduced computational complexity (see Section 3.1.2). In this case the Laplacian operator $\nabla^2 L(\boldsymbol{x}, \boldsymbol{\sigma})$ is approximated by the difference between two Gaussian smoothed images:

$$\nabla^2 L(\boldsymbol{x}, \boldsymbol{\sigma_k}) = \frac{1}{2(\xi - 1)\sigma_k} \left(L\left(\boldsymbol{x}, \xi\boldsymbol{\sigma_k}\right) - L\left(\boldsymbol{x}, \boldsymbol{\sigma_k}\right)\right) \tag{2.9}$$

The result of blob detection using either LoG or DoG methods depends on the choice of scale sampling rate which is analyzed in [38] using real images containing outdoor scenes, human faces, aerial photographs and industrial images and the optimal value $\xi = \sqrt{2}$ for this data set is selected.

Another extension of the LoG method is Determinant of Hessian (DoH) [36], which instead of the Laplacian operator uses the determinant of the Hessian matrix $H(\boldsymbol{x}, \sigma_k)$ at particular scale:

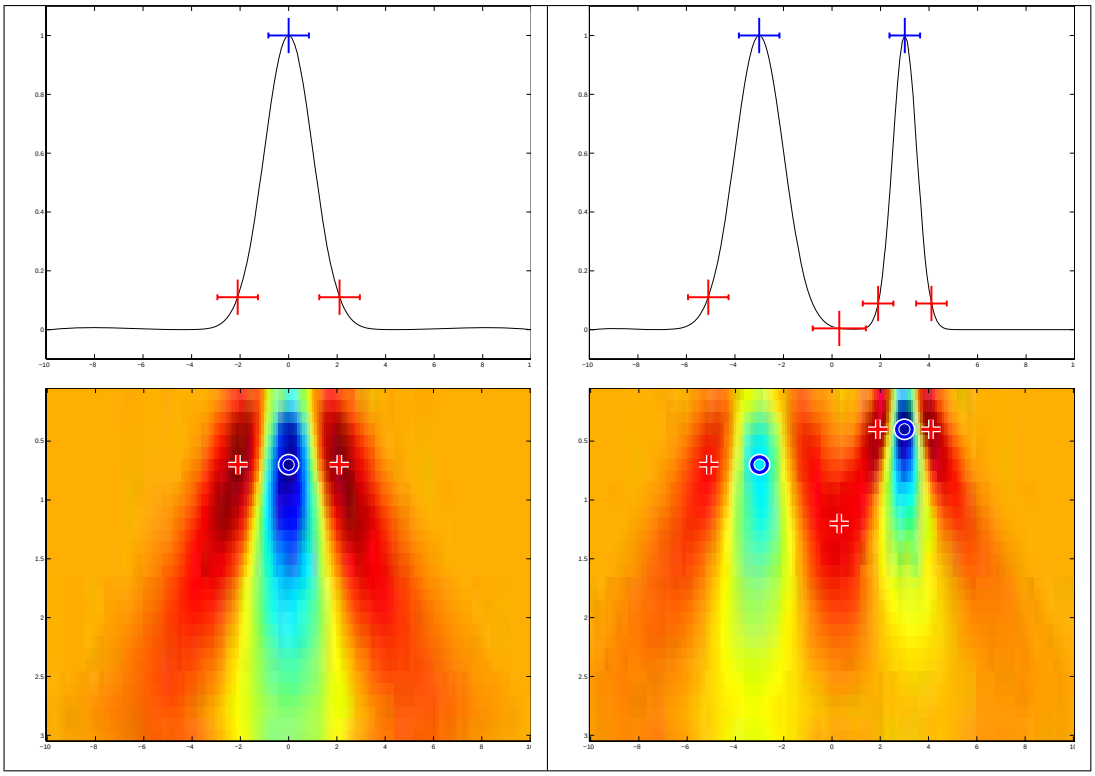


Figure 2.4: Examples of the interest point location and scale detection on Gaussian blobs using (2.8). The top row shows the blobs with the interest point locations (vertical bars) and the associated scale (horizontal bars). Red and blue colors indicate local maximum and minimum respectively of the $\nabla^2_n L$ transform. The bottom row shows the map $\nabla^2_n L$ with the horizontal axis corresponding to the spatial coordinate and vertical axis to the scale coordinate. The local maxima and minima are indicated with red crosses and blue circles respectively.

$$H(\boldsymbol{x}, \boldsymbol{\sigma_k}) = \begin{bmatrix} L_{xx}(\boldsymbol{x}, \boldsymbol{\sigma_k}) & L_{xy}(\boldsymbol{x}, \boldsymbol{\sigma_k}) \\ L_{xy}(\boldsymbol{x}, \boldsymbol{\sigma_k}) & L_{yy}(\boldsymbol{x}, \boldsymbol{\sigma_k}) \end{bmatrix} \tag{2.10}$$

$$\det\left(H(\boldsymbol{x}, \boldsymbol{\sigma_k})\right) = \sigma_k^2 \left(L_{xx}(\boldsymbol{x}, \boldsymbol{\sigma_k}) L_{yy}(\boldsymbol{x}, \boldsymbol{\sigma_k}) - L^2_{xy}(\boldsymbol{x}, \boldsymbol{\sigma_k})\right) \tag{2.11}$$

The results of the DoH method are similar to the LoG, except DoH penalizes elongated structures for which the second order derivative in a single orientation is particularly small i.e. $L_{xx}(\boldsymbol{x}, \boldsymbol{\sigma_k}) L_{yy}(\boldsymbol{x}, \boldsymbol{\sigma_k}) \approx L^2_{xy}(\boldsymbol{x}, \boldsymbol{\sigma_k})$.

An approach to detect blobs that is not based on space-scale theory has been proposed by Matas et al. [41] (2002). Their *Maximally Stable Extremal Regions* (MSER) is based

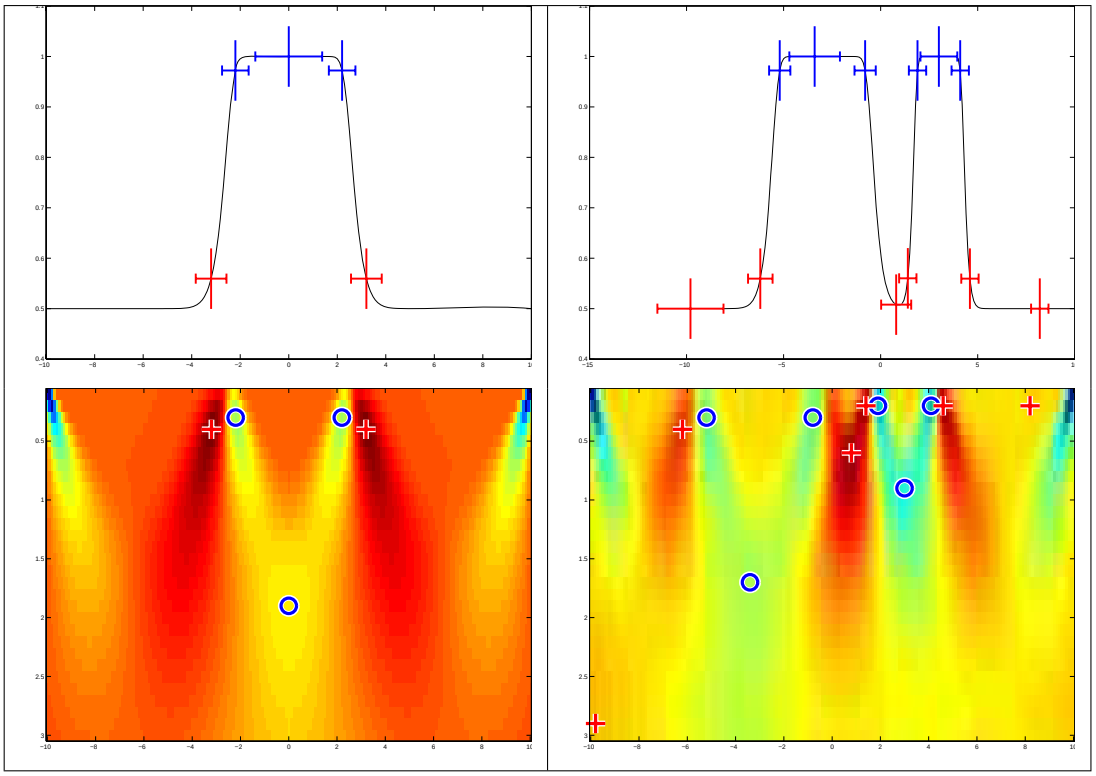


Figure 2.5: Examples of the interest point location and scale detection on non-Gaussian blobs. Note that the interest point location and scale for the larger blob on the right are affected by the Gaussian smoothing of asymmetrically distributed blobs.

on the analysis of local intensity extrema in the image. The idea originates from the observation that distinctive regions in the image correspond to the area that remains stable for a range of intensity thresholds. The paper [41] introduces the notion of *extremal regions* that are used for the extraction of *maximally stable extremal regions*. An image region Q is extremal if the intensity of all pixels $q \in Q$ is higher than the intensity of boundary pixels p (adjacent to Q) $I(q) > I(p)$ for maximum intensity regions or lower $I(q) < I(p)$ for minimum intensity regions. Region Q is a contiguous image patch i.e. there is a path S connecting any two pixels $q \in Q$ such that $S \in Q$. Figure 2.9 shows several extremal regions such that $I(Q_i + \Delta) > I(Q_i) > I(Q_i - \Delta)$. The *maximally stable extremal region* is the one for which variation of the area $q(i)$ has a local minimum at i:

$$q(i) = \frac{|Q_{i+\Delta}| - |Q_{i-\Delta}|}{|Q_i|} \tag{2.12}$$

In the original paper [41], MSERs were used for wide baseline stereo matching as a

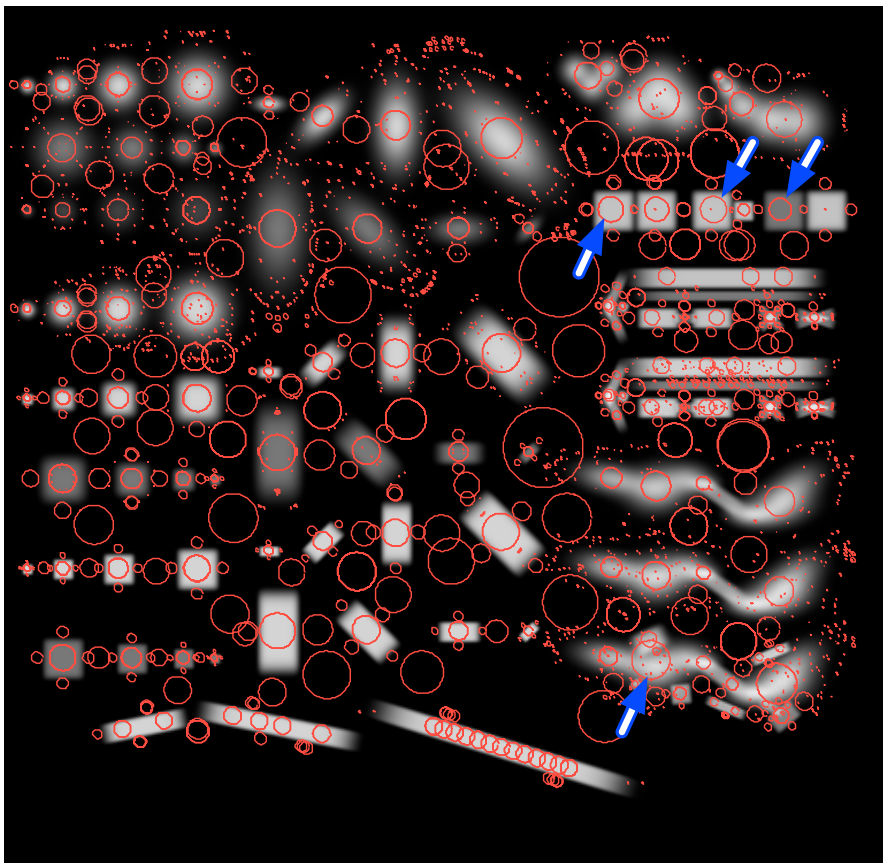


Figure 2.6: DoG interest regions detected in the test image from Figure 2.1. While the majority of blobs have been correctly detected, asymmetrically distributed blobs result in misplaced interest regions (drift) marked by the blue arrows. The method generates a significant number of tiny regions corresponding to the blob boundary, which in this case can be easily filtered out. The region detection along elongated structures is not consistent.

region detector which is invariant to affine transformation of image intensities. Since then MSERs were also successfully applied as an interest region detector for object recognition [31]. This approach however is sensitive to the choice of parameters as demonstrated in Figures 2.10 and 2.11*. The result in Figure 2.10 shows two properties of MSER algorithm:

- the stability measure (2.12) is correlated to the smoothness of the blob, i.e. the intensity gradient along the extremal region boundaries. The increase of parameter

*The results were obtained using an MSER implementation available at http://www.robots.ox.ac.uk/∼vgg/research/affine/detectors.html.

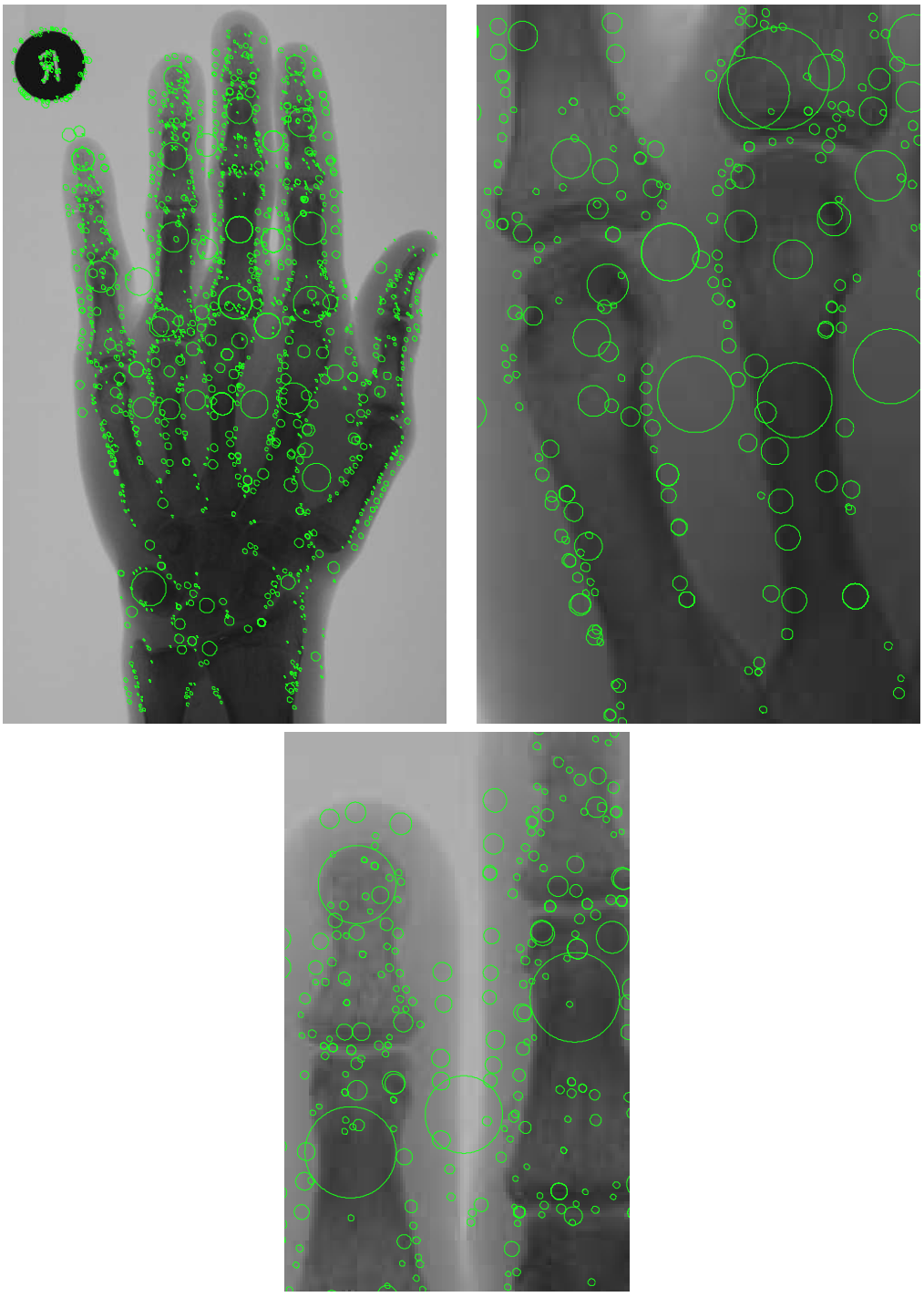


Figure 2.7: Example of interest region detection using DoG. Due to low contrast in the image and non-uniform distribution of larger scale structures e.g. bones, the precision of region scale and position detection is lower than in the case of the test image shown in Figure 2.6. Interest point drift is visible in some of the larger scale blobs and many small scale blobs are detected near the finger bone boundaries.

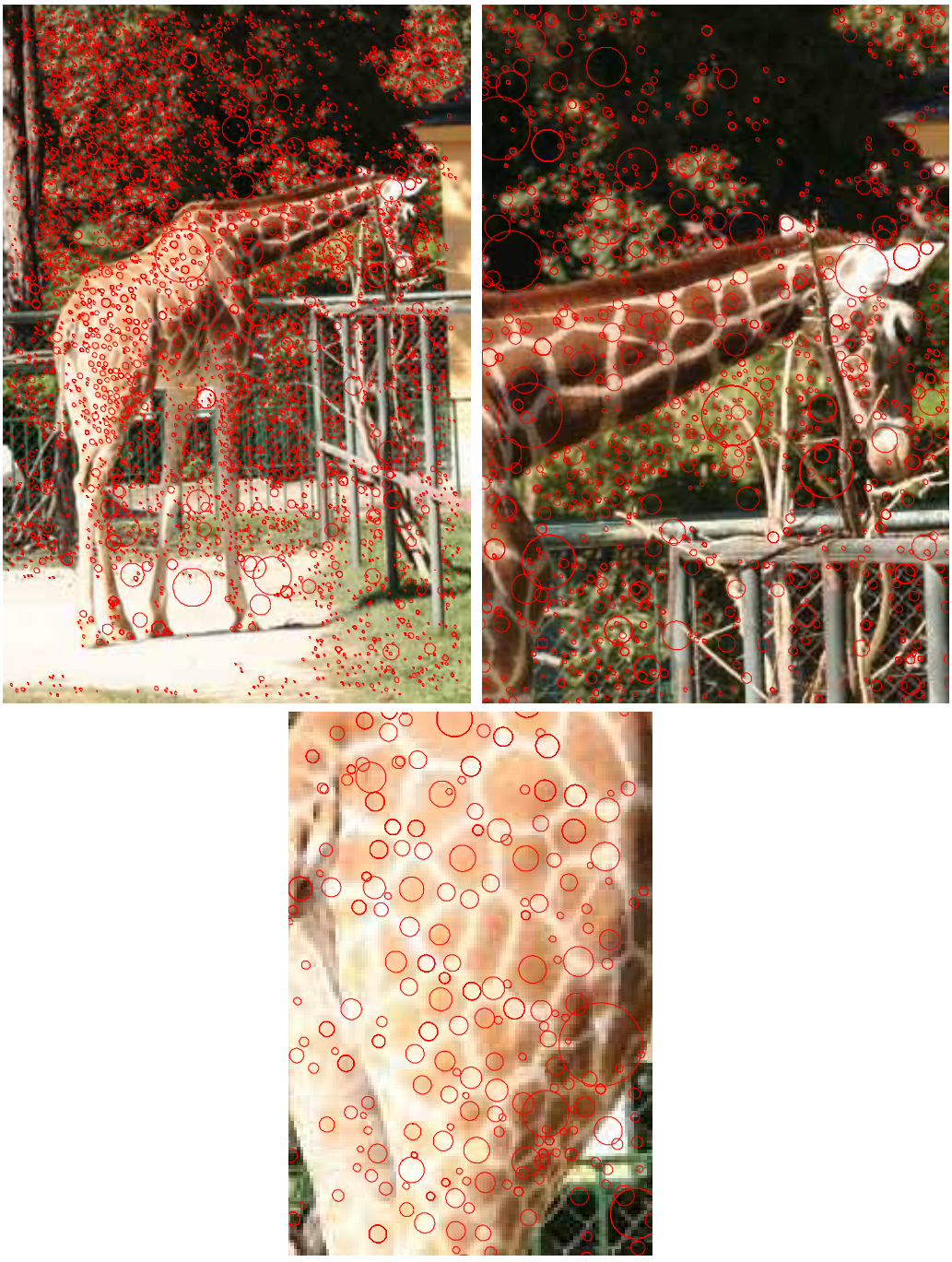


Figure 2.8: Example of interest region detection using DoG. Note that the scale estimation of smaller image structures is more consistent than in the case of larger structures.

M from 1 to 5 (the minimum number of intensity levels) results in almost all Gaussian blobs in regions A1 and B1 not being detected, independently of the intensity of the blobs. Further increase of $M = 10$ results in missing all smooth structures in regions A1, B1, C1 and C4. However smooth blobs with superimposed noise in regions A2 and C4 are still detected. At the same time all structures containing square blobs generated with intensity $h \approx 1$ are detected.

- MSER is sensitive to intensity – as in the case of smooth blobs, the detection of low intensity blobs $h = 0.25$ or $h = 0.5$ deteriorates for $M > 1$.

MSER is designed to capture homogenous intensity regions in the image and therefore it is not surprising that smooth structures such as Gaussian blobs in Figure 2.10 are detected only with the lowest value of parameter M. The results in Figure 2.11 demonstrate the use of MSER on a real scene which shows how the number of regions detected increases with the decreasing value of parameter M.

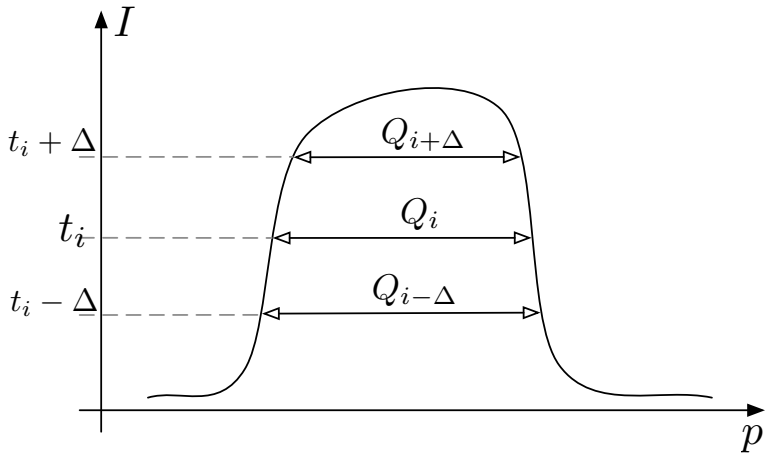


Figure 2.9: Illustration of extremal regions in 1D curve.

2.4 Corner Detectors

This section describes the Harris corner detector [18] as well as extensions of this method that add scale and affine invariance [43].

The Harris interest point detector [18] is built on earlier work of Moravec [45], which defines a corner as a point with a low self-similarity within an image region. The corner

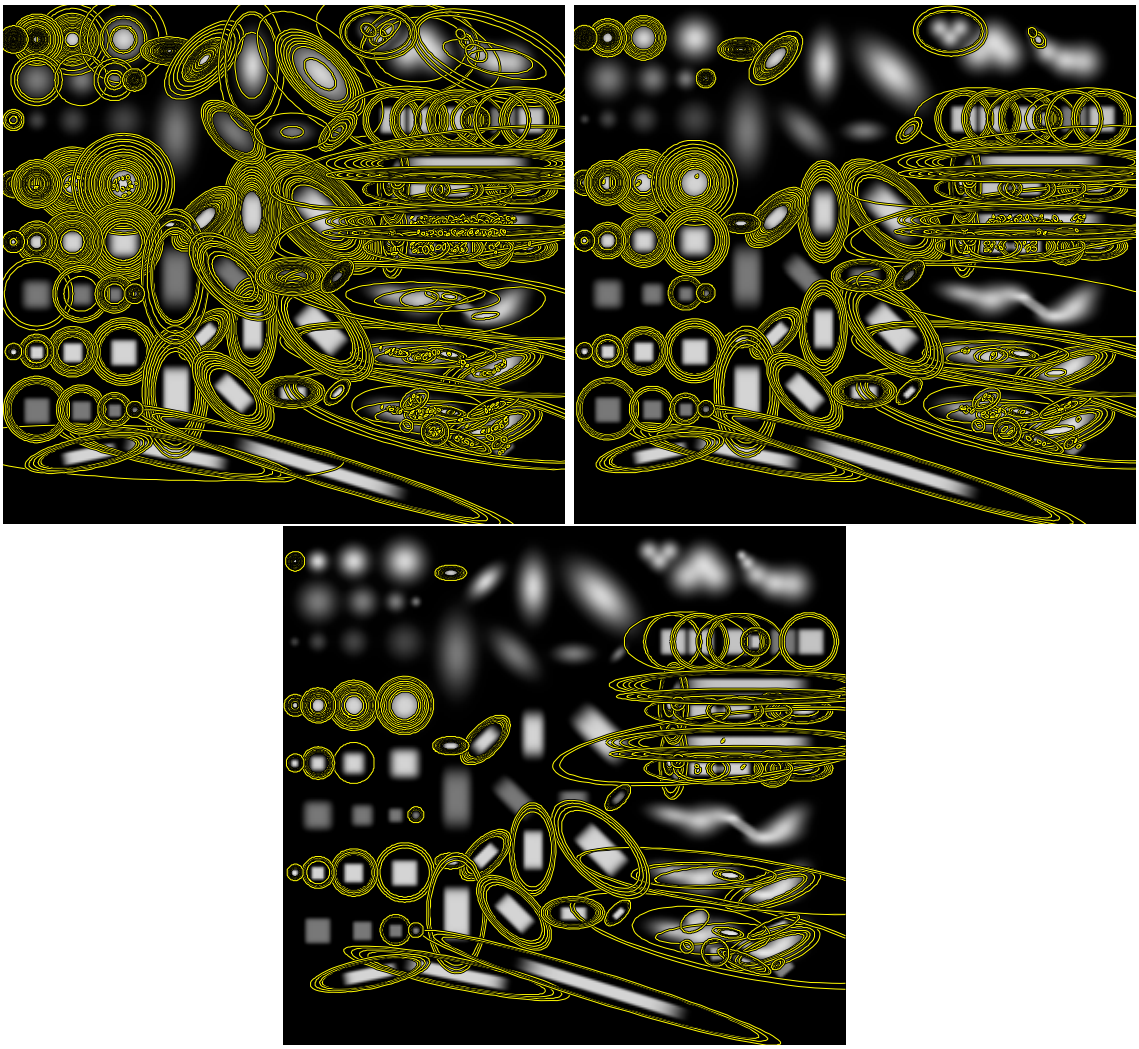


Figure 2.10: The result of MSER region detection for test image in Figure 2.1 with varying parameter M - the smallest number of intensity levels that each region needs to be considered as 'stable'. Top left $M = 1$, top right $M = 5$, bottom $M = 10$. MSER regions are represented by fitted ellipses.

strength is the smallest squared difference (SSD) between a patch centered at a particular pixel (u, v) and one of the horizontal, vertical or diagonal neighbors separated by $\triangle \boldsymbol{x} = [\triangle x, \triangle y]$.

$$SSD(\boldsymbol{x}, \triangle \boldsymbol{x}) = \sum_{\boldsymbol{x}} \big(I(x, y) - I(x - \triangle x, y - \triangle y)\big)^2 \tag{2.13}$$

The Moravec corner detector however fails when the edges around pixel (u, v) are not

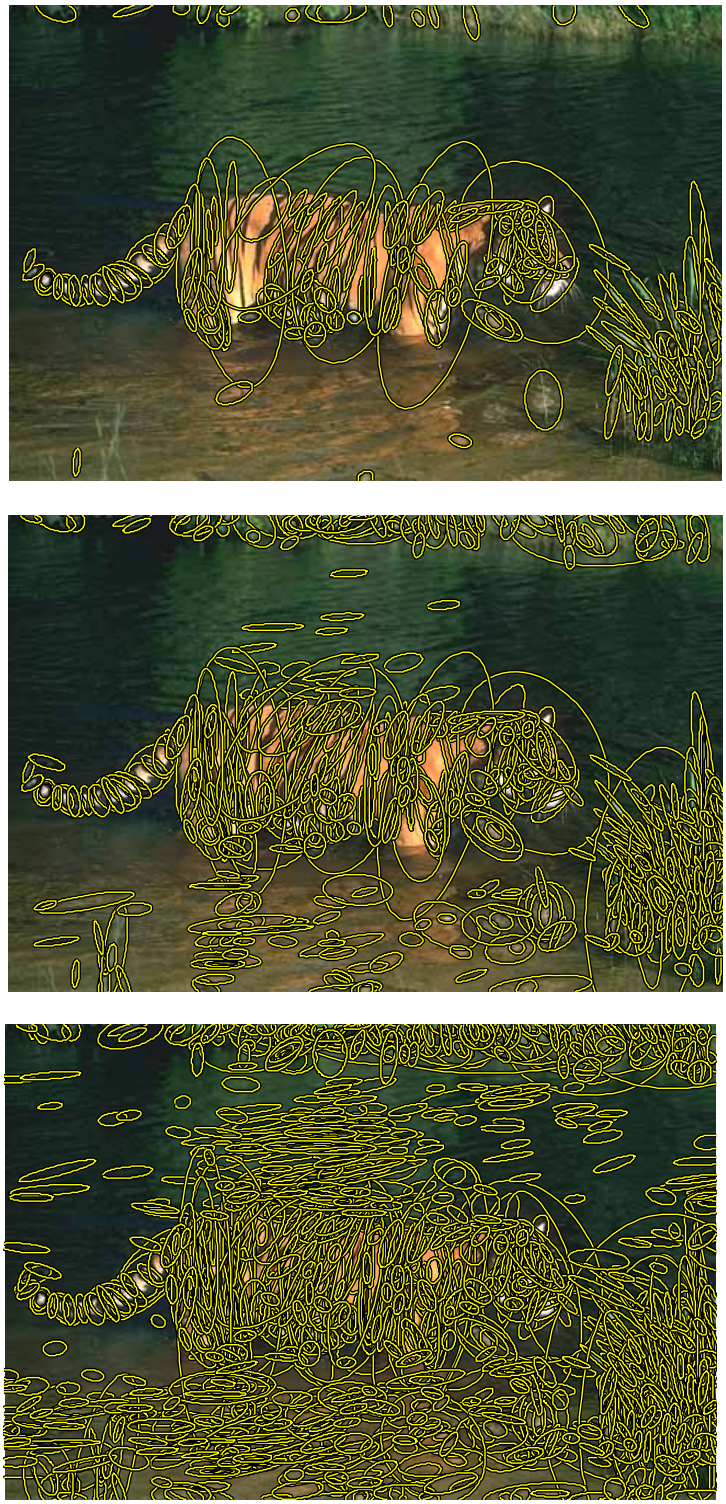


Figure 2.11: The result of MSER region detection with varying parameter M. From top to bottom $M = 20, 10, 5$. MSER regions are represented by fitted ellipses.

oriented in the horizontal, vertical or diagonal direction. This problem has been solved by Harris and Stephens [18] by using the second derivative of the SSD measure in the form of the Harris matrix, also referred to as the second moment matrix:

$$\mu(\boldsymbol{x}) = \begin{bmatrix} \sum_{\boldsymbol{x}} (I_x(\boldsymbol{x}))^2 & \sum_{\boldsymbol{x}} I_x(x,y) I_y(\boldsymbol{x}) \\ \sum_{\boldsymbol{x}} I_x(\boldsymbol{x}) I_y(\boldsymbol{x}) & \sum_{\boldsymbol{x}} (I_y(\boldsymbol{x}))^2 \end{bmatrix} \tag{2.14}$$

where $I_x(\boldsymbol{x})$ and $I_y(\boldsymbol{x})$ are horizontal and vertical derivatives of the image intensity at location $\boldsymbol{x}$.

The Harris matrix represents the distribution of the gradient in a local patch. The two eigenvalues of the Harris matrix represent the average intensity gradients in the maximum gradient direction λ_1 and orthogonal to it λ_2. The relative value of the two eigenvalues can be used to detect local image patches containing: corner $\lambda_1 \approx \lambda_2$ and edge $\lambda_1 \gg \lambda_2$ or $\lambda_2 \gg \lambda_1$, formulated as a "cornerness" measure:

$$M_\mu(\boldsymbol{x}) = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 = \det(\mu(\boldsymbol{x})) - \kappa(\mathrm{trace}(\mu(\boldsymbol{x})))^2 \tag{2.15}$$

where κ is a tunable parameter that decreases the "cornerness" measure proportionally to the difference between the two eigenvalues. The second part of the equation (2.15) is the formulation proposed by Harris and Stephens for increased computational efficiency [18]. The locations of corners correspond to the local maxima of the "cornerness" measure computed for every pixel of the image.

It is possible to show that the result is independent of the choice of $\triangle\boldsymbol{x}$. However the final result depends on the size of the patch and the parameter κ. Figure 2.12 shows Harris points detected in the test image. The number of points detected without thresholding M_μ reaches over 6000, however less than 10% represent perceptible corners. Applying a threshold to M_μ allows the removal of points with lower "cornerness" measure than the chosen threshold.

The Harris corner detector is not invariant to scale and affine transformations. The solution to this problem has been proposed by Mikolajczyk and Schmid with the scale invariant Harris-Laplace and later affine invariant Harris-Affine interest point detector which combines the Harris corner detector with a Laplacian-based scale selection [43].

In their approach the second moment matrix (2.14) is replaced by the scale adapted version:

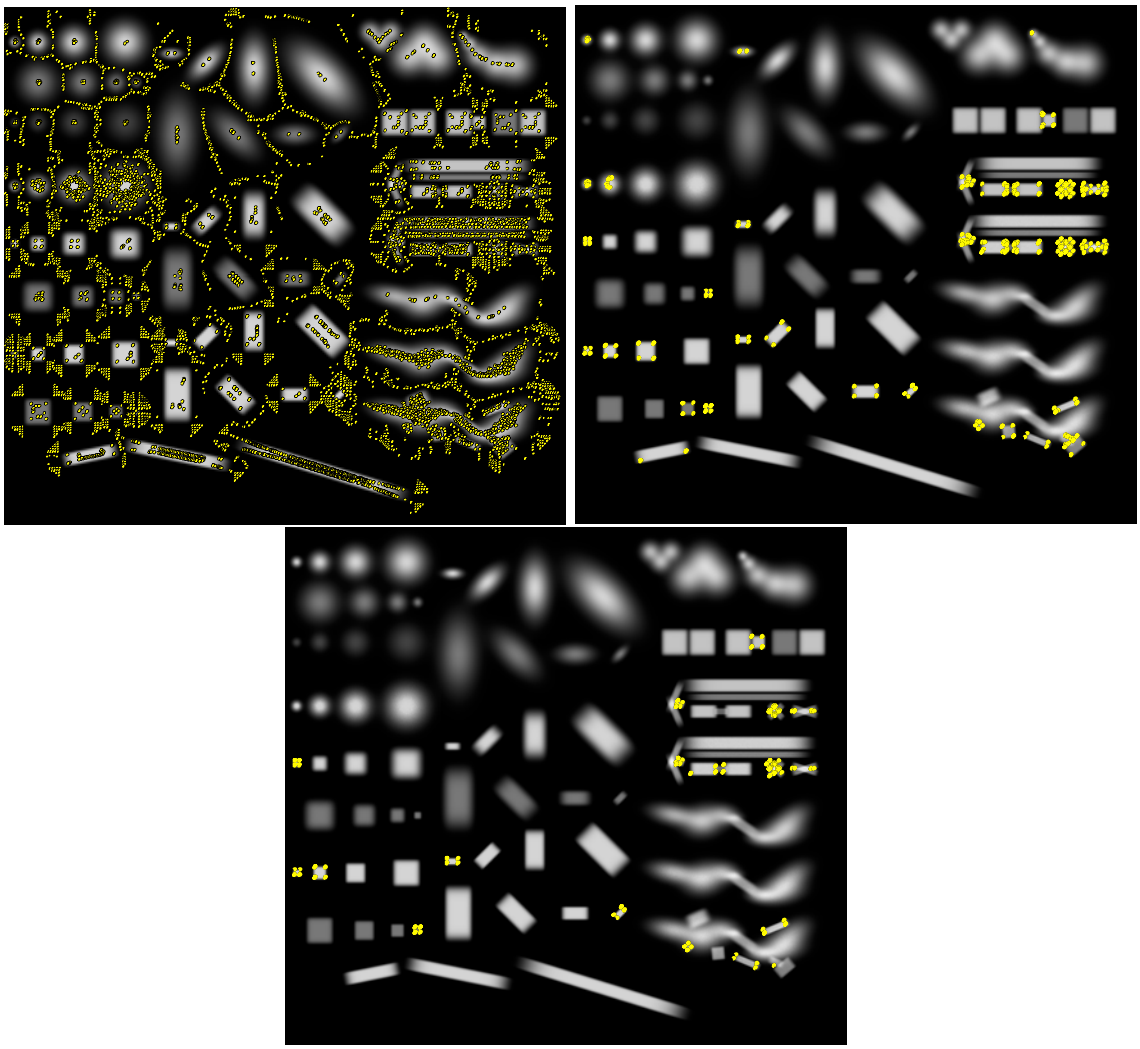


Figure 2.12: The result of Harris interest point detection for the test image in Figure 2.1 with varying threshold of M_μ. Top left: no threshold applied (over 6000 points), top right: 250 points corresponding to the highest "cornerness" score, bottom row: 100 points corresponding to the highest "cornerness" score.

$$\mu(\boldsymbol{x}, \sigma_k, \sigma_I) = \sigma_k^2 g(\sigma_I) * \begin{bmatrix} (L_x(\boldsymbol{x}, \sigma_k))^2 & L_x(\boldsymbol{x}, \sigma_k) L_y(\boldsymbol{x}, \sigma_k) \\ L_x(\boldsymbol{x}, \sigma_k) L_y(\boldsymbol{x}, \sigma_k) & (L_y(\boldsymbol{x}, \sigma_k))^2 \end{bmatrix} \tag{2.16}$$

where σ_k is also called the differentiation scale and σ_I is the integration scale, corresponding to the patch size around each pixel used for calculation of the Harris matrix. To avoid extensive computational complexity both scales are related to each other by a constant

factor $\sigma_k = s\sigma_I$, with $s = 0.7$ as in [43].

The "cornerness" measure is then calculated similarly to (2.15):

$$M_\mu(\boldsymbol{x}, \sigma_I) = \det\left(\mu(\boldsymbol{x}, \sigma_I)\right) - \kappa(\mathrm{trace}(\mu(\boldsymbol{x}, \sigma_I)))^2 \tag{2.17}$$

Note that parameter σ_k was removed since it is now coupled to σ_I.

The locations of interest points are aligned with the spatial extrema of the "cornerness" measure $M_\mu(\boldsymbol{x}, \sigma_I)$, while the related scale is estimated from the scale extrema of the LoG at the detected locations:

$$\begin{aligned} \hat{\boldsymbol{x}} &= \underset{\boldsymbol{x}}{\mathrm{argminmax}}\left(M_\mu(\boldsymbol{x}, \sigma_I)\right) \\ \hat{\boldsymbol{\sigma}} &= \underset{\boldsymbol{\sigma_k}}{\mathrm{argminmax}}\left(\nabla^2_n L(\hat{\boldsymbol{x}}, s\boldsymbol{\sigma_I})\right) \end{aligned} \tag{2.18}$$

where $\boldsymbol{\sigma}_I$ corresponds to a vector of scale values.

However, the "cornerness" measure is computed at a range of scales $\sigma_i = \sigma_0\xi^i$ for $i = 0..N$ and locations of extrema vary between them due to scale differences of Gaussian smoothing. It means that the maximum of $\mu(\boldsymbol{x}, \sigma_I)$ calculated for the corner of extent σ_I is not located in the same place as the maximum of measure $\mu(\boldsymbol{x}, 0.5\sigma_I)$ because the same corner was smoothed with two different scales σ_I and $0.5\sigma_I$. The goal of the algorithm is however to provide a single interest point per detected corner. Mikolajczyk and Schmid [43] propose an iterative search to find points at which both "cornerness" and LoG response attain local extrema. The method consists of several steps summarized below:

1. Compute $M_\mu(\boldsymbol{x}, \sigma_i)$ at a range of scales $\sigma_i = \sigma_0\xi^i$, $i = 0..N$.

2. Compute $\nabla^2_n L(\boldsymbol{x}, s\sigma_i)$ at a range of scales. Note that some computation results can be re-used from the previous step.

3. Obtain $\hat{\boldsymbol{x}}_\mathbf{0}$ from the $M_\mu(\boldsymbol{x}, \sigma_0)$ ($\hat{\boldsymbol{x}}_\mathbf{0}$ is the initial estimation of corner locations).

4. Estimate $\hat{\boldsymbol{\sigma}}_\boldsymbol{l}$ for each $\hat{\boldsymbol{x}}_\boldsymbol{l}$ using (2.18), where index l corresponds to the $l-th$ repetition of the steps 4–7 (initially $l = 0$ and is incremented with each iteration).

5. Reject all points for which the LoG response does not attain an extremum or the response is below a threshold.

6. Detect the spatial location $\hat{\boldsymbol{x}}_{\boldsymbol{l}+\mathbf{1}}$ closest to $\hat{\boldsymbol{x}}_\boldsymbol{l}$ from the $M_\mu(\boldsymbol{x}, \hat{\boldsymbol{\sigma}}_l)$. Note that $\hat{\boldsymbol{\sigma}}_\boldsymbol{l}$ is independent for each location $\hat{\boldsymbol{x}}_\boldsymbol{l}$.

7. Return to step 4 if $\hat{\boldsymbol{\sigma}}_{\boldsymbol{l+1}} \neq \hat{\boldsymbol{\sigma}}_{\boldsymbol{l}}$ or $\hat{\boldsymbol{x}}_{\boldsymbol{l+1}} \neq \hat{\boldsymbol{x}}_{\boldsymbol{l}}$.

The Harris-Laplace detector is invariant to rotation and scaling operations. Figure 2.13 demonstrates region detection using the Harris-Laplace detector. The first difference to the Harris detector is that the scale of corners is estimated. The second difference is that the number of regions detected without thresholding of the "cornerness" measure M_μ is approximately 7 times smaller than in the case of the Harris detector (over 800 points in comparison to over 6000), since any points that do not attain a LoG extremum are rejected. Finally, the corners in square blobs of extent $\sigma > 20$, previously missed by the Harris detector (except for a single instance in region A4), are now detected. However not all corners were consistently detected. The low intensity structures $h \leqslant 0.5$ are at a particular disadvantage.

Mikolajczyk and Schmid have also shown that it is possible to achieve invariance to affine transformation within a limited range [43]. The primary difference between the Harris-Laplace and Harris-Affine detector is that in the former one the regions are described by ellipses for which the isotropy measure reaches its maximum. The local isotropy measure is defined as the ratio between the eigenvalues of the second moment matrix (2.16):

$$\mathcal{Q} = \frac{\lambda_{min}(\mu)}{\lambda_{max}(\mu)} \tag{2.19}$$

and varies in the range [0..1] with 1 corresponding to a perfect isotropic structure.

This approach has been applied to matching different views of the same scene, where image patches undergo affine transformations. However, it is impossible to asses whether a local image structure is a 3D projection of the isotropic 2D appearance or non-projected anisotropic structure without a priori knowledge e.g. an elliptical shape can be a projected circle or an unprojected ellipse. The conclusion of evaluation of local feature detectors and descriptors [74] is that "for most datasets in our evaluation, we show that local features with the highest possible level of invariance (affine) do not yield the best performance".

The affine transformation of the image region can be represented using anisotropic Gaussians and correspondingly the second moment matrix:

$$\mu(\boldsymbol{x}, \Sigma_k, \Sigma_I) = \det(\Sigma_k)\, g(\Sigma_I) * \left(\nabla L(\boldsymbol{x}, \Sigma_k) \nabla L(\boldsymbol{x}, \Sigma_k)^T\right) \tag{2.20}$$

where Σ_I and Σ_k are the covariance matrices which determine the integration and differentiation Gaussian kernels.

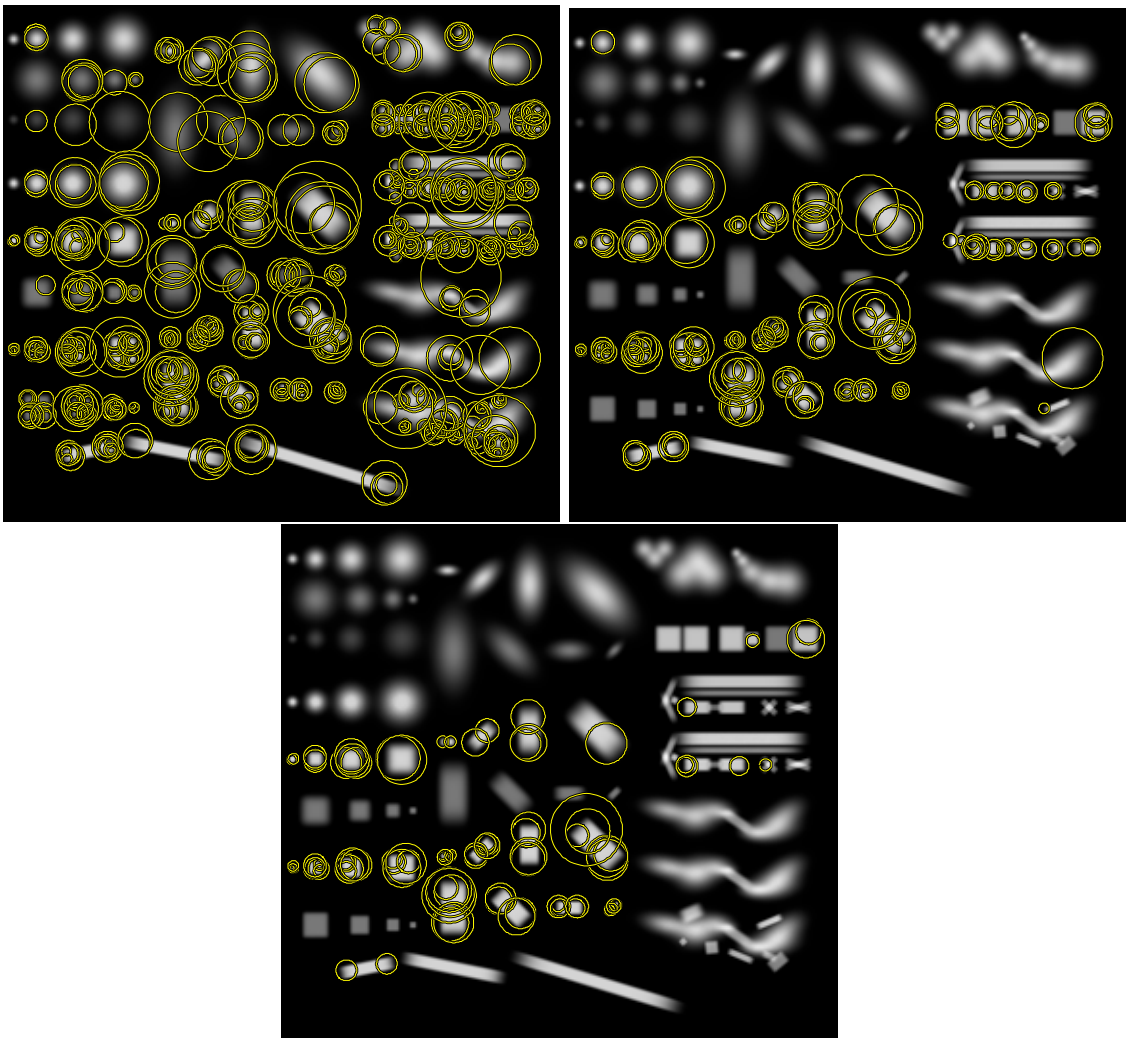


Figure 2.13: The result of the Harris-Laplace region detection in the test image in Figure 2.1 with varying threshold of M_μ. Top left: no threshold applied (over 800 regions), top right: 250 regions left after M_μ thresholding, bottom row: 100 regions left after M_μ thresholding.

The direct computation of (2.20) is not feasible as the quantization of the Σ_I would yield a large number of possible values and transfer the solution into 5-dimensional domain (2 spatial coordinates and 3 independent covariance matrix elements). In [43] Mikolajczyk and Schmid derive an iterative algorithm which attempts to find affine regions in the image that maximize the isotropy of a back-projected image patch at a given location and scale. Instead of using anisotropic kernels and the corresponding second moment matrix (2.20) the proposed method iteratively estimates affine regions, transforms the underlying image

patches into circular regions and then uses the isotropic second moment matrix (2.16) in combination with the isotropy measure (2.19) to further refine affine regions until the isotropy measure attains a value close to 1 for each region. The affine region is updated for each detected point independently together with the corresponding local measure of isotropy in the course of the iteration process. The initial set of points and scales are obtained from the scale adapted Harris-Laplace detector. The iteration stops for a given point when the isotropy measure is close to 1 (above a user defined threshold). A detailed description and discussion of the method is presented in [43]. The results of Harris-Affine corner detector shown in Figure 2.14 closely correspond to those of the Harris-Laplace detector.

The Harris-Laplace region detector has been used in the object recognition and stereo matching fields, due to their rotation, scale and affine invariance (see Section 2.6). However the choice of the "cornerness" threshold parameter that would maximize the the number of "true" (perceptible) corners detected and minimize detection of other structures is not trivial. Figures 2.13 and 2.14 show that even without the thresholding not all perceptible corners have been detected. The Harris-Laplace and Harris-Affine regions detected from the hand x-ray image shown in Figure 2.7 produced less than 50 regions, which was caused by the low contrast in the image. All experiments were performed using constant parameter κ[†].

2.5 Symmetry Based Interest Points

The symmetry based interest points originate from the work of Resifeld et al. [55] which introduced the *Generalized Symmetry Transform* (GST) – a multi-scale approach that can be used to detect rotational and reflectional symmetries in images. The method produces a symmetry map which for each pixel of the image provides the magnitude and direction of symmetry estimated from all other points in the image. The final result is essentially a sum of symmetry contributions from all possible pixel pair sets in the image. Two pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ define a set $\Gamma_{ij}(\boldsymbol{p})$ (see Figure 2.5) as:

$$\Gamma_{ij}(\boldsymbol{p}) = \left\{ (i,j) \middle| \frac{\boldsymbol{p}_i + \boldsymbol{p}_j}{2} = \boldsymbol{p} \right\} \tag{2.21}$$

[†]Corner detection results have been obtained using code available at http://www.robots.ox.ac.uk/∼vgg/research/affine/detectors.html

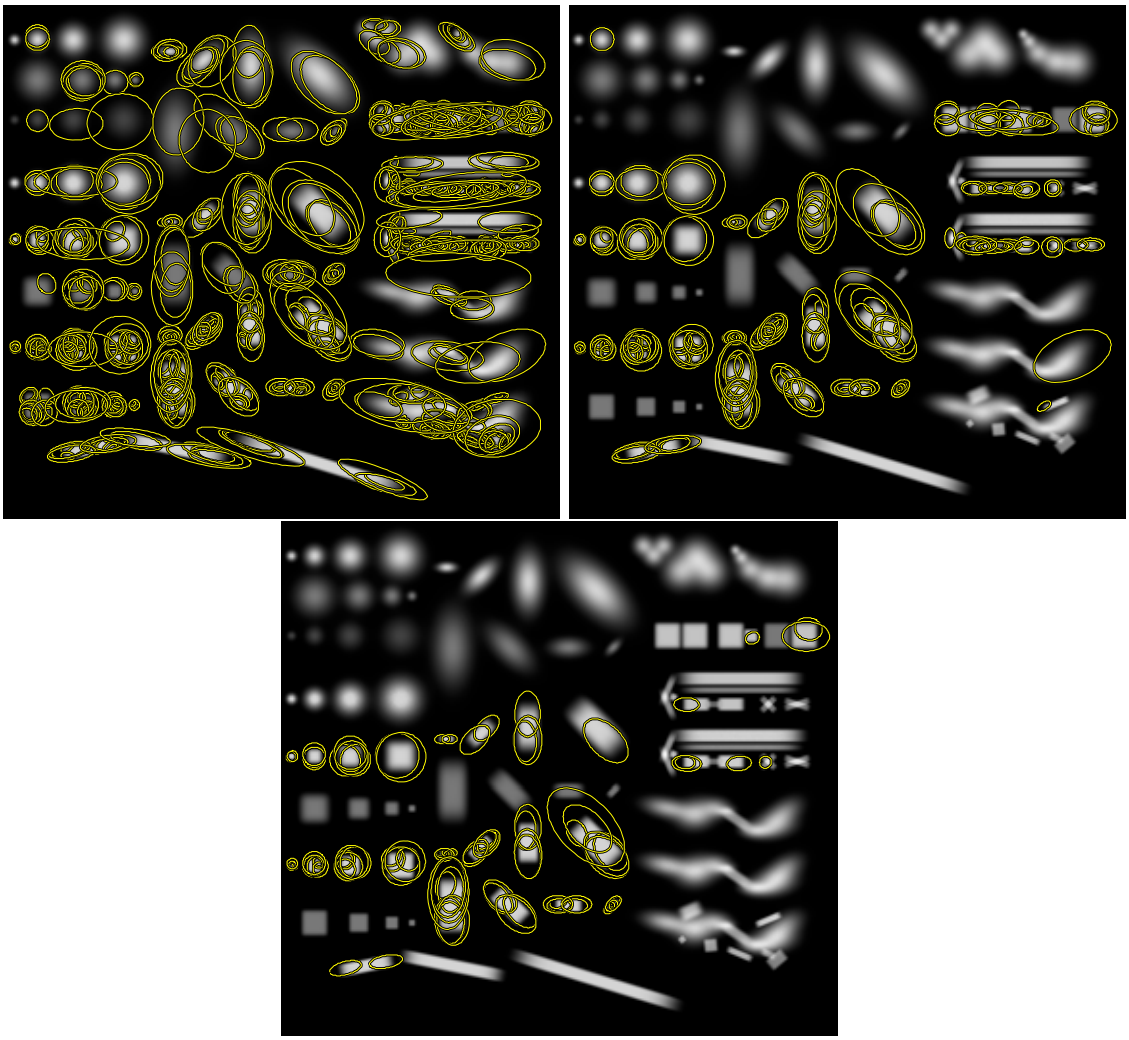


Figure 2.14: The result of Harris-Affine region detection in the test image in Figure 2.1 with varying threshold of M_μ. Top left: no threshold applied (over 800 regions), top right: 250 regions left after M_μ thresholding, bottom row: 100 regions left after M_μ thresholding.

where pixel $\boldsymbol{p}$ corresponds to a central location between pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. The similarity of gradient magnitude and relative gradient orientation at points i and j is used to calculate the symmetry measure $C(i, j)$ contributing to the symmetry transform at pixel $\boldsymbol{p}$.

The contribution $C(i, j)$ is a product of logarithmic gradients r_i and r_j at the corresponding pixels weighted by the distance $D_\sigma(i, j)$ and phase $P(i, j)$ weight functions:

$$D_\sigma(i,j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|}{2\sigma}} \tag{2.22}$$

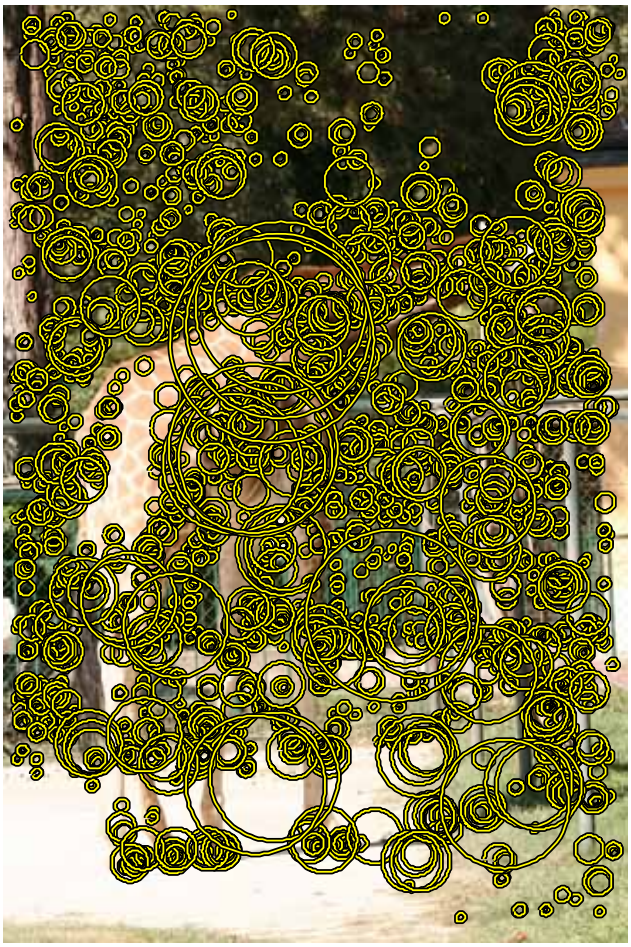


Figure 2.15: Example of Harris-Laplace region detection without thresholding M_μ (over 800 regions).

$$P(i,j) = (1 - \cos(\theta_i + \theta_j - 2\alpha_{ij}))\,(1 - \cos(\theta_i - \theta_j)) \tag{2.23}$$

$$C(i,j) = D_\sigma(i,j)P(i,j)r_i r_j \tag{2.24}$$

where $r_k = \log(1 + \|\nabla p_k\|)$, $\theta_k = \arctan(\partial_y p_k / \partial_x p_k)$ for $k = \{i, j\}$ and α_{ij} can be seen in Figure 2.5.

The distance weight $D_\sigma(i,j)$ defines the scale at which the symmetry contributions have the highest influence in the final result, thereby allowing for a multi-scale symmetry representation. The first term in the phase weight $P(i,j)$ is maximized if the gradient

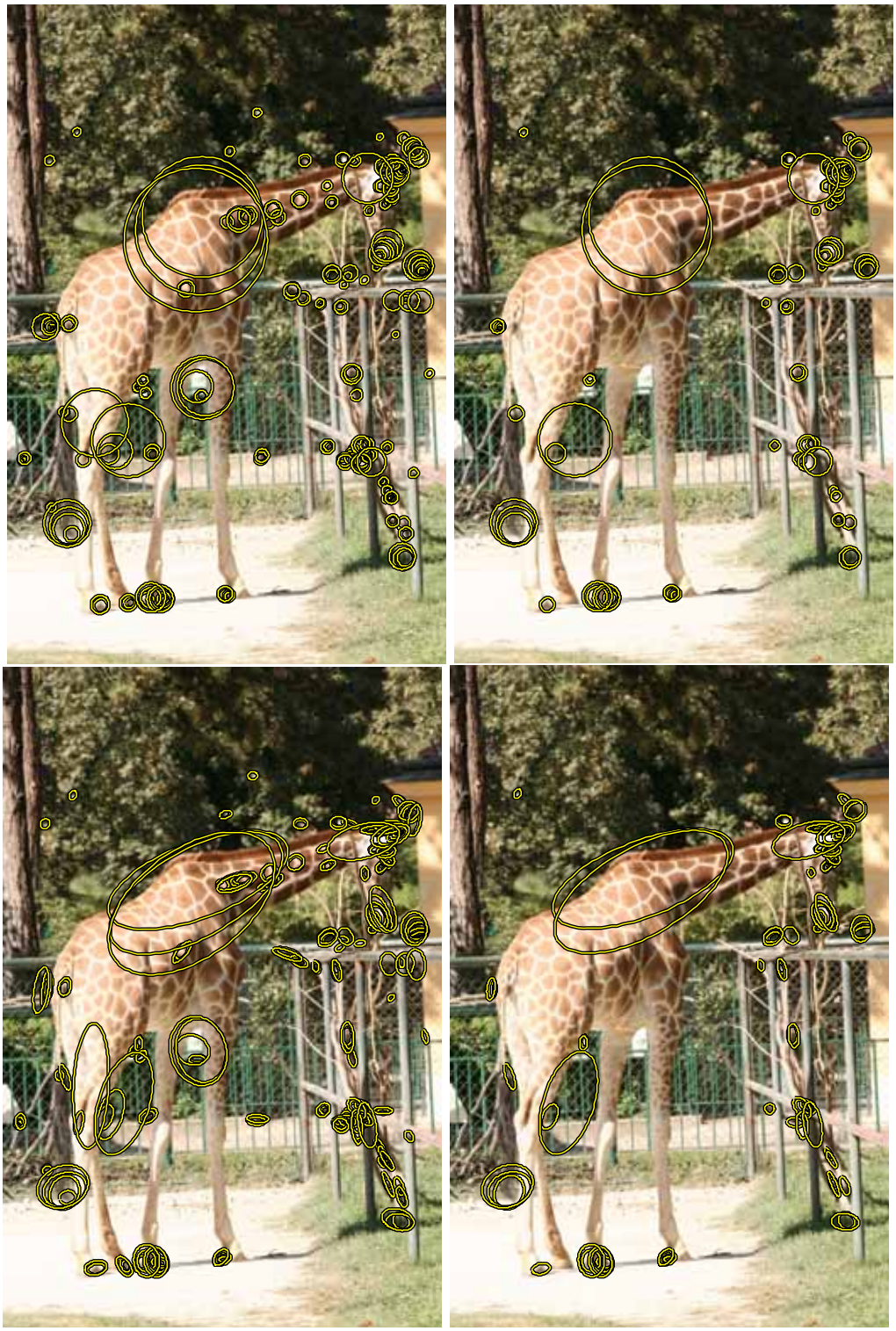


Figure 2.16: Examples of Harris-Laplace (top row) and Harris-Affine (bottom row) region detection with thresholding of M_μ. The columns show 250 (left column) and 100 (right column) regions, remaining after M_μ thresholding.

orientations at points i and j are symmetrically oriented, while the second term prevents matching of pixel pairs with similar gradient orientations, e.g. pixels along a straight edge.

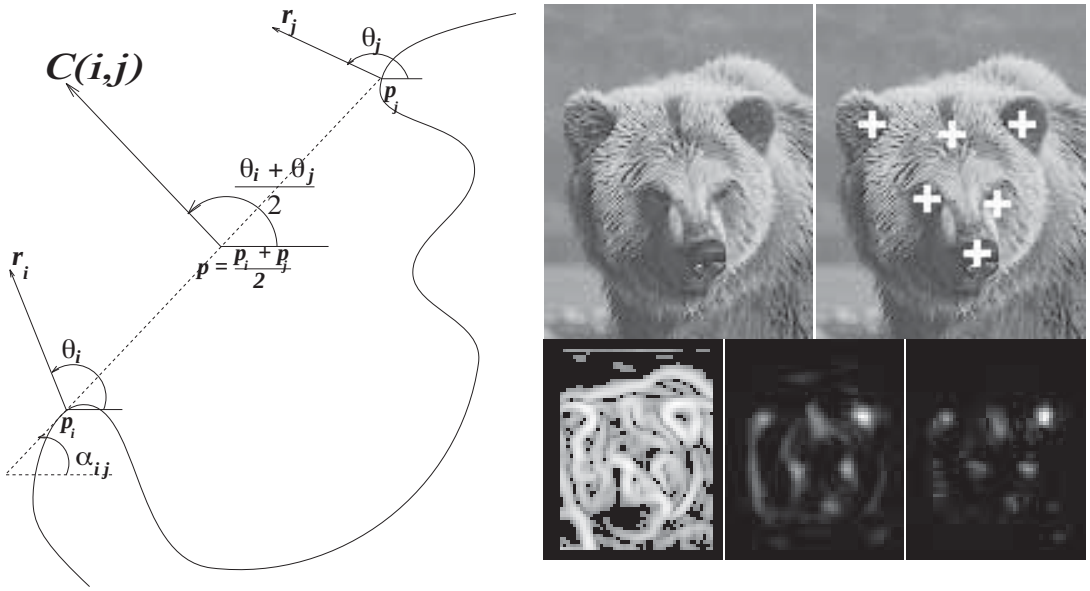


Figure 2.17: Left: Illustration of point pair configuration used for estimation of a symmetry map. Right: Eexample of the symmetry map and the detected interest points. Bottom row (left to right): edge detection, isotropic symmetry, radial symmetry. The images were obtained from [55].

The isotropic symmetry measure or symmetry magnitude of each point $\boldsymbol{p}$ can be calculated as a sum of contributions of all corresponding pixel pairs which averages symmetry contributions in all orientations:

$$M_\sigma(p) = \sum_{(i,j)\in\Gamma(\boldsymbol{p})} C(i,j) \tag{2.25}$$

The symmetry direction is defined as $\phi(\boldsymbol{p}) = \frac{\theta_i+\theta_j}{2}$ where i and j corresponds to the maximum $C(i,j)$ for all $(i,j) \in \Gamma(p)$. The symmetry of the point $S_\sigma(\boldsymbol{p})$ consists then of symmetry magnitude and direction:

$$S_\sigma(\boldsymbol{p}) = (M_\sigma(\boldsymbol{p}), \phi(\boldsymbol{p})) \tag{2.26}$$

The Generalized Symmetry Transform has inspired other methods e.g. [34, 40]. The "Fast Radial Symmetry Transform" (FRST) by Loy and Zelinsky (2002) [39, 40] aims at

efficient detection of radial symmetries. The computational complexity of the FRST is $O(KN)$ versus $O(KN^2)$ in case of GST, where K is the number of pixels in the image and N is the size of the neighborhood (radius) used for the calculation of the symmetry measure at a given location. In their method, each pixel of the image contributes to a symmetry measure at two locations called negatively and positively affected pixels shown in Figure 2.5. The coordinates of negatively affected $\boldsymbol{p}_{-ve}$ and positively affected $\boldsymbol{p}_{+ve}$ pixels are defined by the gradient orientation at pixel $\boldsymbol{p}$ and a distance n (called in [39] range) as follows:

$$\boldsymbol{p}_{+ve} = \boldsymbol{p} + \texttt{round}\left(\frac{\boldsymbol{g}(\boldsymbol{p})}{\|\boldsymbol{g}(\boldsymbol{p})\|}n\right) \tag{2.27}$$

$$\boldsymbol{p}_{-ve} = \boldsymbol{p} - \texttt{round}\left(\frac{\boldsymbol{g}(\boldsymbol{p})}{\|\boldsymbol{g}(\boldsymbol{p})\|}n\right) \tag{2.28}$$

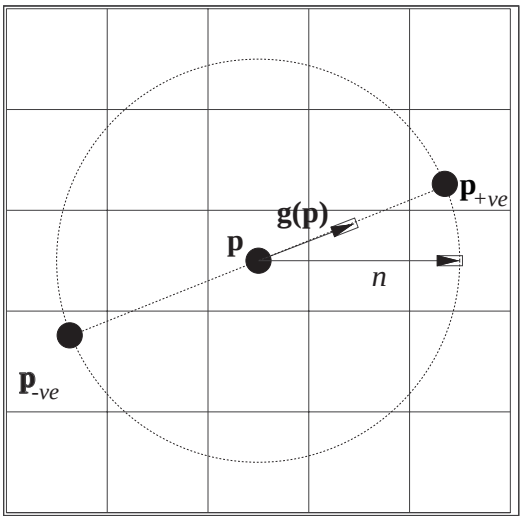


Figure 2.18: The pixel $\boldsymbol{p}$ contributes to the symmetry measure at locations $\boldsymbol{p}_{+ve}$ and $\boldsymbol{p}_{-ve}$ both in the positive and negative direction of the maximum gradient $g(\boldsymbol{p})$ respectively. Image obtained from [39].

The symmetry measure is a combination of orientation projection O_n and magnitude projection M_n maps, which are obtained through agglomeration of positively and negatively affected pixel contributions. Each positively affected pixel increments the corresponding element of the orientation projection map by 1 and magnitude projection map by $\|\boldsymbol{g}(\boldsymbol{p})\|$ while the negatively affected pixel decrements the map by these values:

$$O_n\left(\boldsymbol{p}_{+ve}(\boldsymbol{p})\right) = O_n\left(\boldsymbol{p}_{+ve}(\boldsymbol{p})\right) + 1 \tag{2.29}$$

$$O_n\left(\boldsymbol{p}_{-ve}(\boldsymbol{p})\right) = O_n\left(\boldsymbol{p}_{-ve}(\boldsymbol{p})\right) - 1 \tag{2.30}$$

$$M_n\left(\boldsymbol{p}_{+ve}(\boldsymbol{p})\right) = M_n\left(\boldsymbol{p}_{+ve}(\boldsymbol{p})\right) + \|\boldsymbol{g}(\boldsymbol{p})\| \tag{2.31}$$

$$M_n\left(\boldsymbol{p}_{-ve}(\boldsymbol{p})\right) = M_n\left(\boldsymbol{p}_{-ve}(\boldsymbol{p})\right) - \|\boldsymbol{g}(\boldsymbol{p})\| \tag{2.32}$$

The radial symmetry measure at range n is a combination of normalized orientation and magnitude projection maps, additionally smoothed by a Gaussian kernel:

$$S_n = g(\sigma_n) * \left(\frac{M_n}{k_n}\right)\left(\frac{|O_n|}{k_n}\right)^{\alpha} \tag{2.33}$$

where k_n is the scale normalization factor and α is the radial strictness parameter which allows to attenuate the symmetry response from ridges. The orientation projection map used for final calculations is thresholded using k_n. The detailed discussion of the parameters is provided in [40].

The symmetry measure can be also calculated over a set of ranges $N = \{n_1, ...n_K\}$ which allows to achieve a partial scale invariance:

$$S = \frac{1}{K}\sum_{n \in N} S_n \tag{2.34}$$

The authors of the method claim that it attains comparable or superior results to GST while at lower computation cost ($O(KN)$ vs $O(KN^2)$). It is argued that a sparse set of ranges N is a sufficient approximation of a symmetry measure, which allows for real-time calculation of the result. However the optimal choice of a set of ranges N for a given image is not trivial. The ranges N should correspond to the scales of features present in the analyzed image but these are not known a priori. The set of ranges N define the scale sampling – increasing sampling density increases accuracy of the symmetry measure but also proportionally increases computational complexity. The result of scale under-estimation (the maximum range is lower than the maximum scale of structures in the image) is generation of more than one interest point for radially symmetrical structures with under-estimated scale. Examples of symmetry detection using a range of scales is

provided in Figures 2.19 and 2.20. The result in Figure 2.20 also shows the presence of interest point drift (asymmetrically distributed blobs in region C2 and C3) that appears for similar reasons as those discussed in Section 2.3.

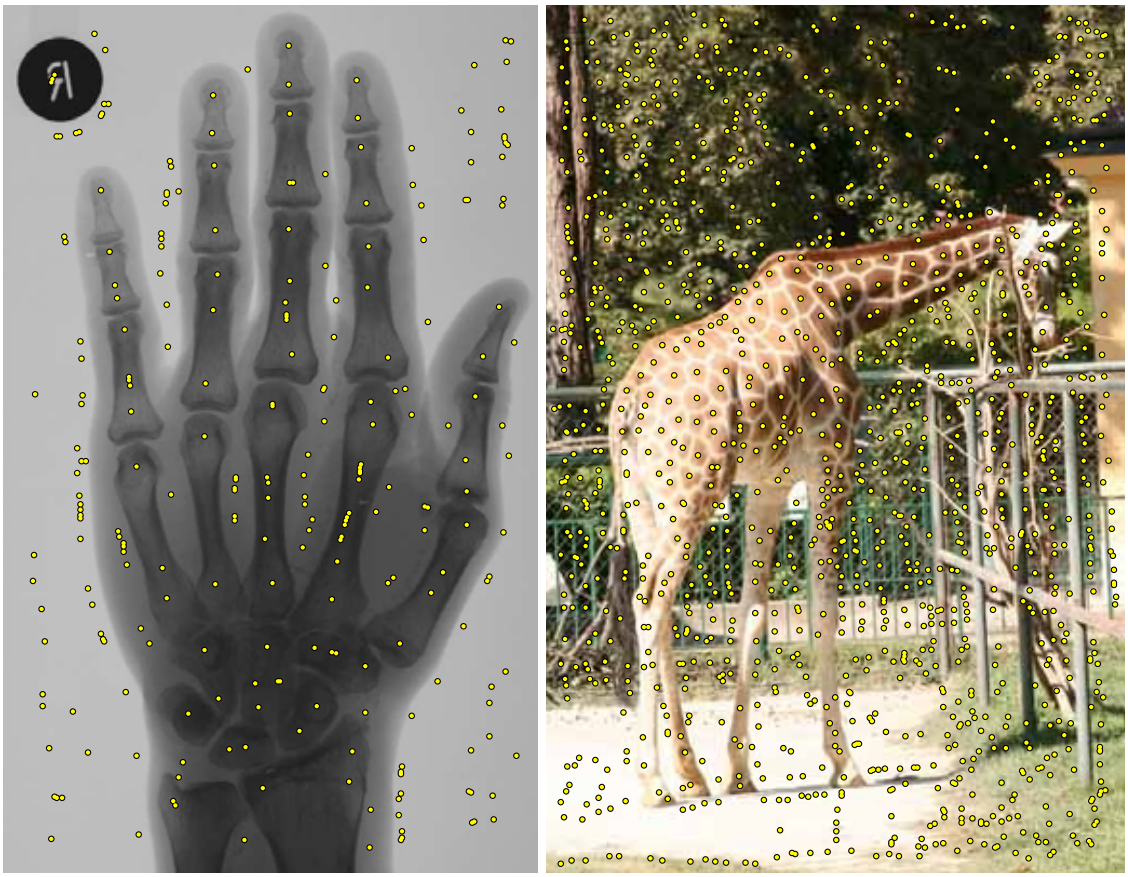


Figure 2.19: Examples of FRST interest point detection. The set of ranges used: hand x-ray $n = \{10, 15, 20, 30\}$, giraffe $n = \{5, 10, 15\}$. The results show a high level of invariance to intensity changes.

The primary intention behind the symmetry detectors described so far is the localization of symmetrical regions which can serve as an attention mechanism for object recognition. Another class of symmetry based detectors has been introduced which attempts to find the majority of locally symmetrical features in the image [10, 63]. These approaches aim at detecting individual structures in textures or any image patches containing symmetrical distribution of gradients or intensity.

The *Radial Symmetry Transform* (RST) attempts to find locations in the image where the intensity distribution attains locally maximum radial symmetry. In the case of ho-

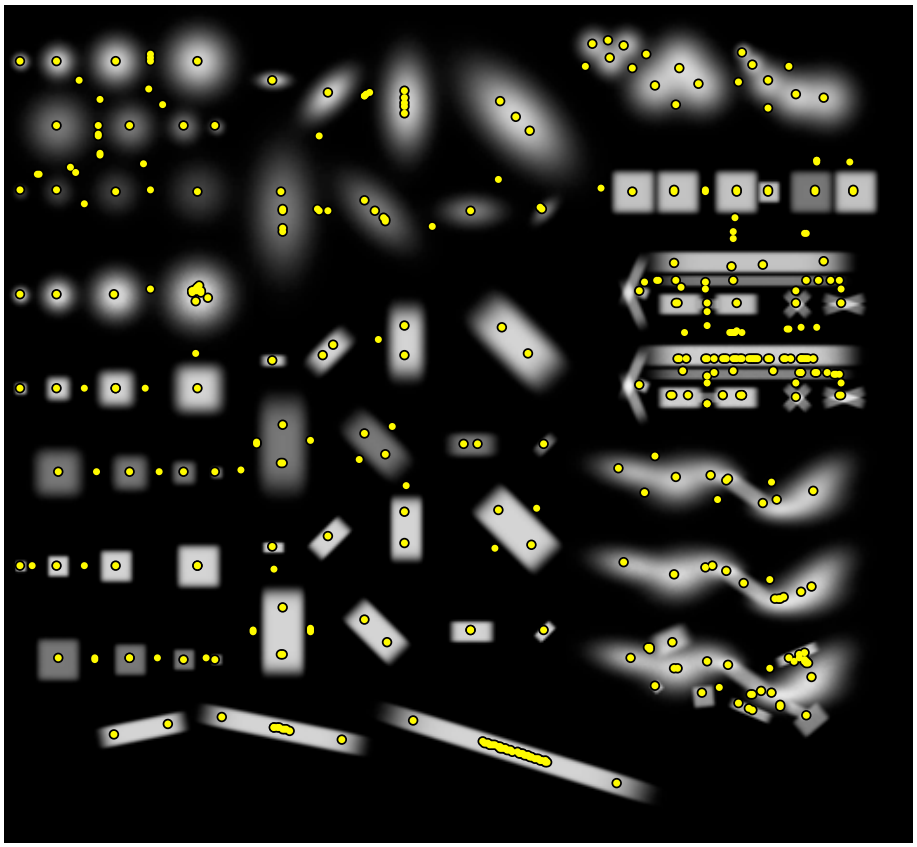


Figure 2.20: Examples of FRST interest point detection in test image ($n = \{10, 20, 30, 40\}$). The interest point drift is visible in regions C2 and C3.

mogenous intensity image patches the method locates interest points approximately at the centers of round/isotropic structures or along the symmetry axis of elongated shapes. The symmetry measure $S_r(x, y)$ is calculated for each pixel (x, y) of the image separately and the interest points are aligned with local symmetry maxima.

$$S_r(x,y) = -\sum_{i=-r}^{r}\sum_{j=0}^{r} g(\sqrt{i^2+j^2}, \sigma_r = 0.5r)\|\boldsymbol{I}(x+i, y+j) - \boldsymbol{I}(x-i, y-j)\| \qquad (2.35)$$

where $\boldsymbol{I}(x+i, y+j)$ is an image pixel intensity or color at coordinates $(x+i, y+j)$ and r defines the image window size used for the symmetry measure calculation to be a $(2r+1) \times (2r+1)$ rectangle. Each contribution of the pixel pair at $(x+i, y+j)$ and $(x-i, y-j)$ is weighted by the Gaussian $g(\sqrt{i^2+j^2}, r)$ which decreases the influence of

pixel pairs increasing distance from (x, y) and normalizes the transform with respect to the chosen scale R.

In the basic version, the interest point locations $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ correspond to the maxima of the S_r transform:

$$(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \underset{x,y}{\texttt{argmax}} \left(S_r\right) \tag{2.36}$$

The symmetry measure S_r reaches a maximum (equal to 0) if all corresponding pixel pairs $(x_c + i, y_c + j)$ and $(x_c - i, y_c - j)$ are identical. It reaches a local maximum at the center of radially symmetric shapes (like a filled circle, star, etc.) or along the symmetry axis of elongated shapes. At the same time it reaches a local minimum along the edges.

Although the symmetry measure S_r is tuned to a particular scale of $0.5r$ it consistently detects symmetrical structures in the image up to to the extent of r. It is also possible to obtain a scale adapted set of interest points using a similar iterative approach as for the scale adapted Harris detector described in Section 2.4. In this case the interest point locations are detected using the symmetry transform and the related scale is detected using the Laplacian operator (2.8). Alternatively, an approximation of the scale adapted symmetry measure is a sum of S_r over a sparse set of radii R:

$$S = \sum_{r \in R} S_r \tag{2.37}$$

Examples of interest point detection is presented in Figures 2.21, 2.22 and 2.23.

This method has been used for the detection of interest points in natural scenes and medical images generating consistent and repeatable results [63, 66] regarding the detection of symmetrical structures. Figures 2.21, 2.22 and 2.23 show that isotropic as well as elongated structures can be captured. The numerical evaluation of interest point in Section 2.6 shows that RST and FRST are invariant to intensity changes as opposed to the tested blob detectors. However, as in case of scale-space based blob detectors they are affected by interest point drift. This issue could be addressed by using scale detection for refining of the local symmetry estimation.

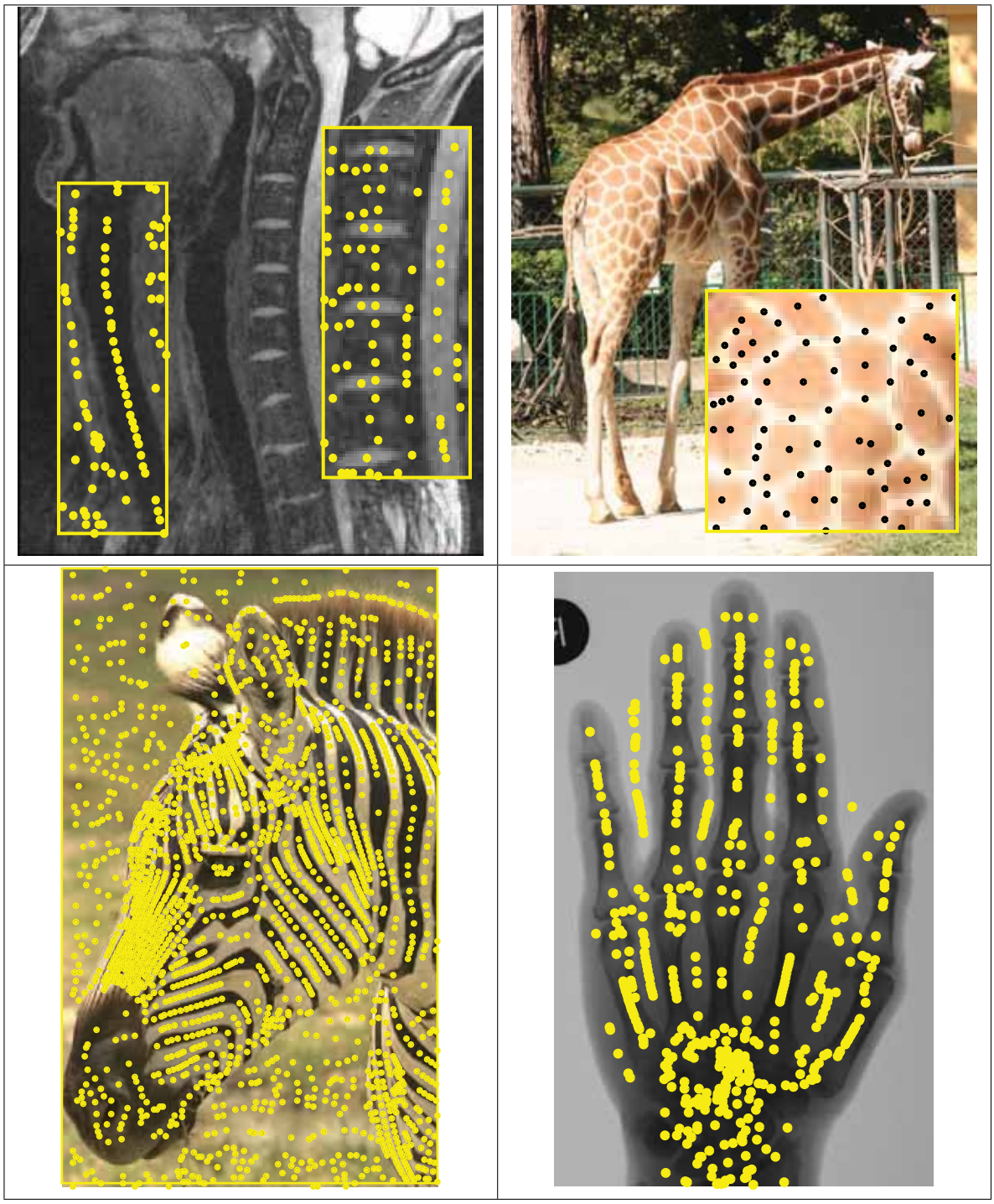


Figure 2.21: Examples of RST based interest points computed at a single scale ($r = \varsigma/50$, where ς is a lower value out of horizontal and vertical image size in pixels).

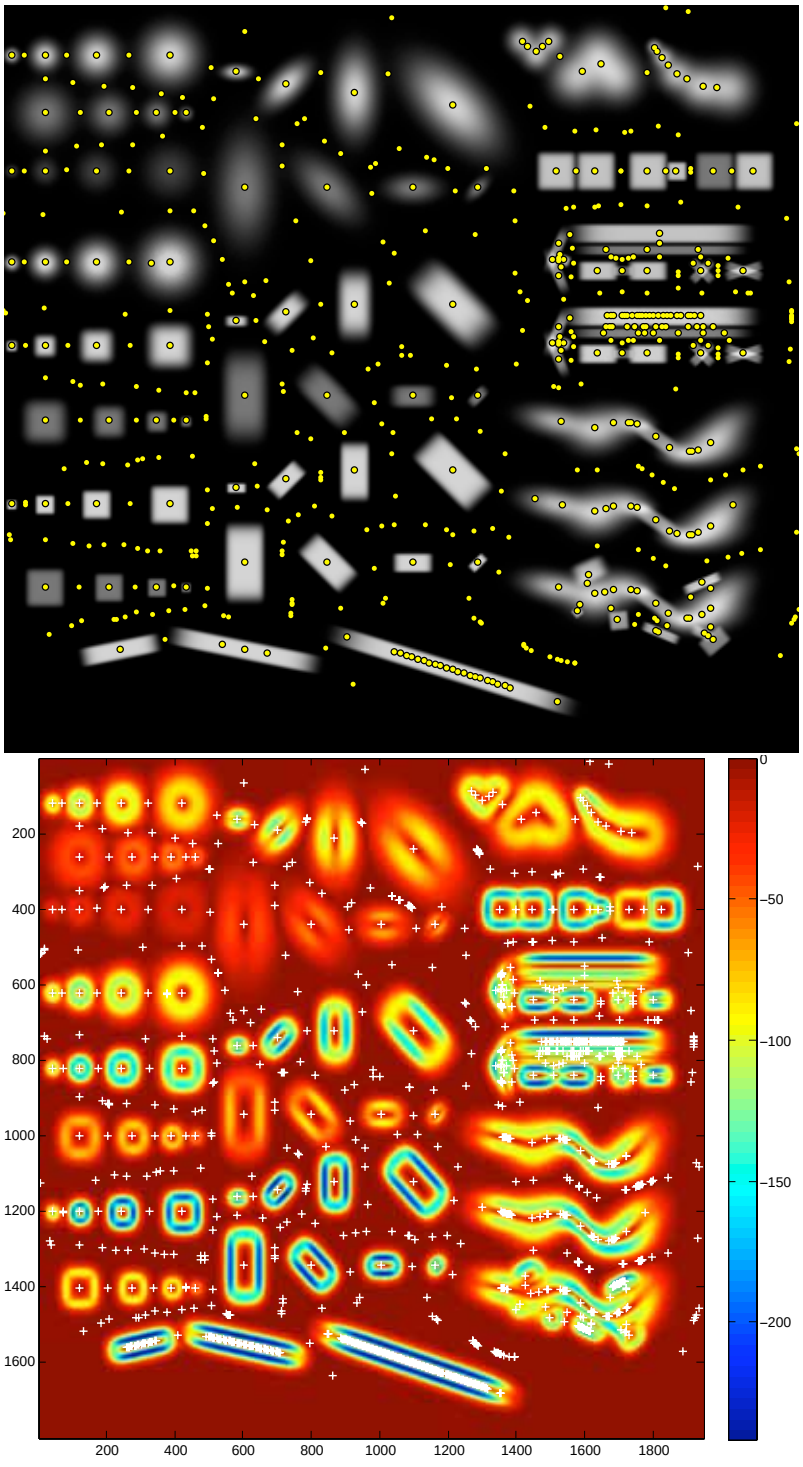


Figure 2.22: Examples of RST interest point detection on the test image ($\sigma_r = \{10, 15, 20\}$). The test image (top) is accompanied by the symmetry measure S (below). The results show a high level of invariance to intensity changes.

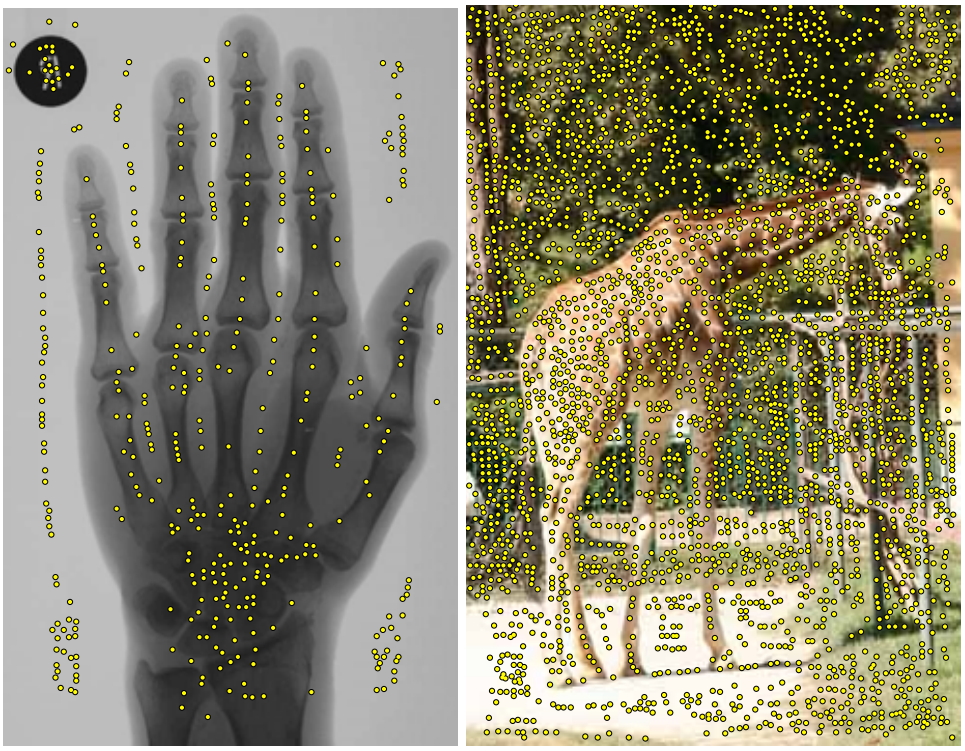


Figure 2.23: Examples of RST interest point detection. The set of σ_r used: hand x-ray $\sigma_r = \{8, 12, 16, 20\}$, giraffe $\sigma_r = \{3, 5, 10\}$, test image $\sigma_r = \{10, 15, 20\}$. The results show a high level of invariance to intensity. Note that the use of a finer scale produces more interest points e.g. white stripes separating giraffe patches are also detected.

2.6 Interest Point Performance

Interest points are the basis of scene matching methods [44, 52] and object recognition approaches [42, 53, 63, 74] that use local image descriptors. Their consistency in locating image structures influences the accuracy of scene matching and object recognition as shown in [44, 74].

The existing comparisons of interest points [42, 44, 58, 74] concentrate on the applicability of interest points for scene matching and object recognition. For example Mikolajczyk et al. [44] proposed a method to verify the repeatability of interest point and region detectors under affine transformations of the same scene. The detectors compared were Harris and Hessian affine points, MSERs, and the edge based region detector from Tuytelaars and Van Gool [70] (see Chapter 3). The authors of this comparison concluded that there is no clear winner outperforming other detectors for all scene types and all types of transformations. Zhang and Marszalek [74] evaluate object classification accuracy using

blob and corner detectors as well as combinations of these. The authors of this work show that the best recognition results are achieved using a combination of blob and corner detectors when compared to the use of only blob or only corner detectors. The results of object classification indicate that the use of affine invariant features does not guarantee better classification accuracy than features invariant to rotation and scale only, which is in agreement with the discussion of the Harris-Affine interest region detector in Section 2.4. The primary focus of this comparison is on local image descriptors, though a comparison of Harris-Laplace, Harris-Affine and LoG combined with SIFT is presented. The overall classification results are similar and no clear winner can be selected.

2.6.1 BLOB DECTION PERFORMANCE

The discussed interest point comparisons operate on real images and provide an application specific performance. A complementary comparison that evaluates blob detection accuracy using the test image in Figure 2.1 is now presented. The compared methods are: Difference of Gaussians (DoG), Determinant of Hessian DoH, Affine Determinant of Hessian (DoHA), MSER, Fast Radial Symmetry Transform (FRST) and Radial Symmetry Transform (RST). The evaluation procedure measures detection precision as the ratio between the number of blobs detected by at least one interest point to the total number of all blobs in the test image:

$$\text{precision} = \frac{\#\text{blobs detected}}{\#\text{test blobs}} \tag{2.38}$$

The blob is detected if the distance between an interest point and the center of the blob is smaller than $\xi\sigma$, where $\sigma = min(\sigma_1, \sigma_2)$ and ξ is the interest point position accuracy threshold. Tests are executed for $\xi = 0.25$ and 0.5.

The other property measured is the ratio between the number of interest points which do not match any of the test blobs (false positives) and the total number of interest points generated:

$$1 - \text{recall} = \frac{\#\text{false positives}}{\#\text{interest points}} \tag{2.39}$$

The total number of points generated is regulated by thresholding the scale-space representation in the case of scale-space based methods, thresholding the symmetry transform in the case of symmetry based methods and varying M in the case of MSER (see Section 2.3). The detection accuracy is evaluated for the number of interest points

$K = \{N, 2N, 4N, 8N\}$, where $N = 134$ is the total number of test blobs, and presented in Figures 2.24 and 2.25.

The results contain the overall performance (for all test regions) and partial results calculated for regions A1-4∪B1-4 (standalone blobs) and regions C1-5 (blob combinations). The partial results were obtaining by ignoring all test blobs and interest points in unrelated regions.

The results presented in Figures 2.24 and 2.25 show the following:

- MSER is not designed for detection of smooth blobs, like those in Region A1. It has the worst detection accuracy in regions A1-4∪B1-4, but is also a top performer in regions C1-5.
- Apart from MSER, all other blob detectors show higher detection accuracy for regions A1-4∪B1-4 than for regions C1-5. This is expected since the features of individual blobs merge together in the dense combinations of blobs (regions C1, C4 and C5).
- At the lowest number of interest points $K = 134$ RST does not detect any blobs in regions C1-C5 for which the symmetry measure maxima are approx. 3 times lower than in case of standalone blobs and fall below the threshold.
- MSER, DoH and DoHA produced between 3 and 8 interest points per blob while DoG and symmetry points produced less than 2. This is associated with the fact that the scale-space representation of the image can contain several scale local maxima for a single spatial location.
- FRST always produced more interest points per blob than RST.
- RST is the only method which was able to detect all blobs in regions A1-4∪B1-4 at 500 interest points generated. The next top performer were DoH and DoHA, that detected 90% of blobs but at more than 2000 interest point which corresponds to almost 7 interest points per blob.
- Both symmetry detectors provide locations that correspond to the local maxima of the symmetry measure. This means that the blob and the background between blobs maximize symmetry. This is also true for scale-space methods (see Figure 2.6) – while the blobs produce local minima of the scale-space representation (negative values), the background produces local maxima (positive values). The strength of

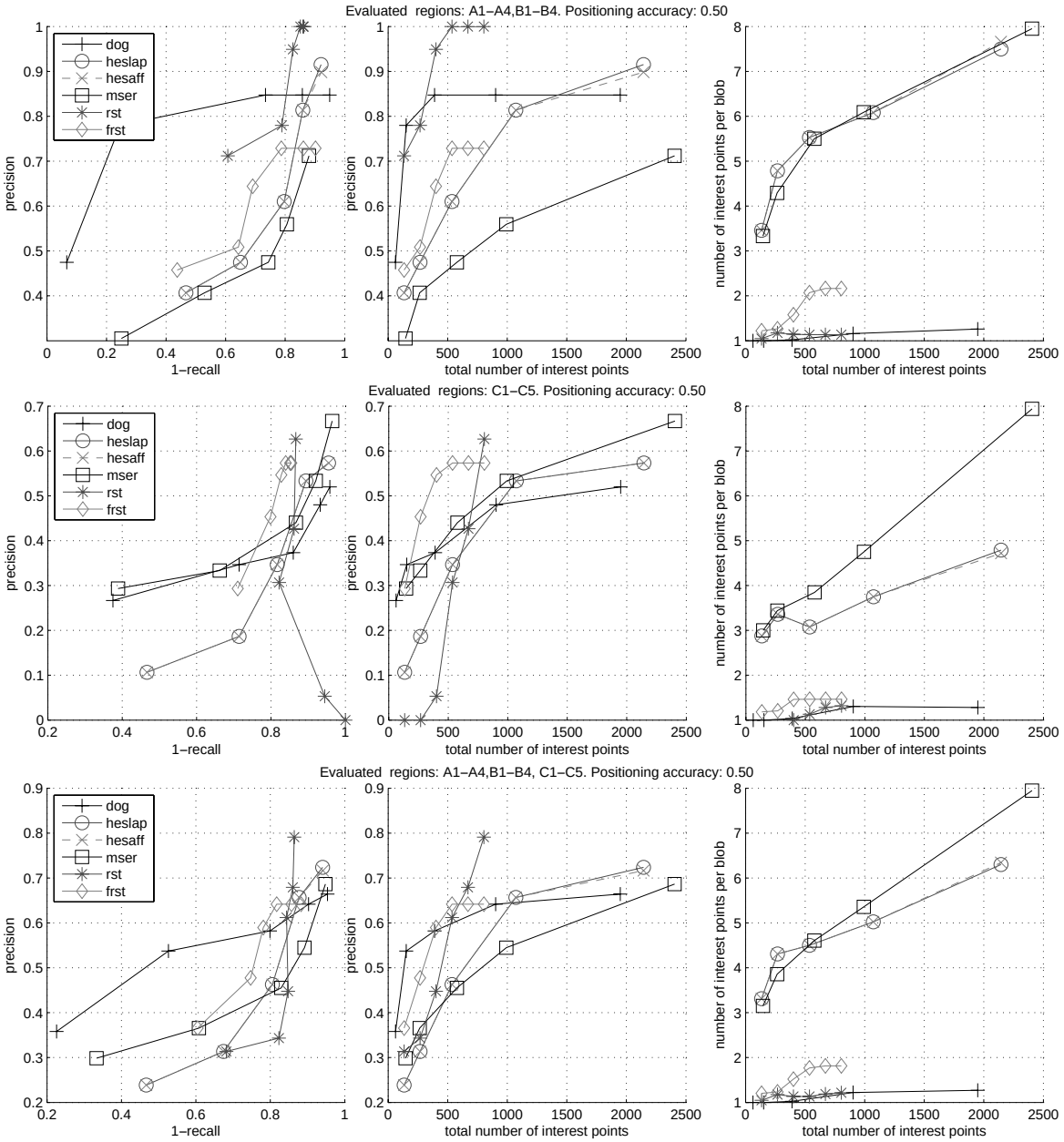


Figure 2.24: The results of blob detection accuracy for $\xi = 0.5$. Rows contain results for regions: A1-4∪B1-4, C1-5, A1-4∪B1-4∪C1-5 respectively from top to bottom. Labels "heslap" to DoH method and "hesaff" to DoHA

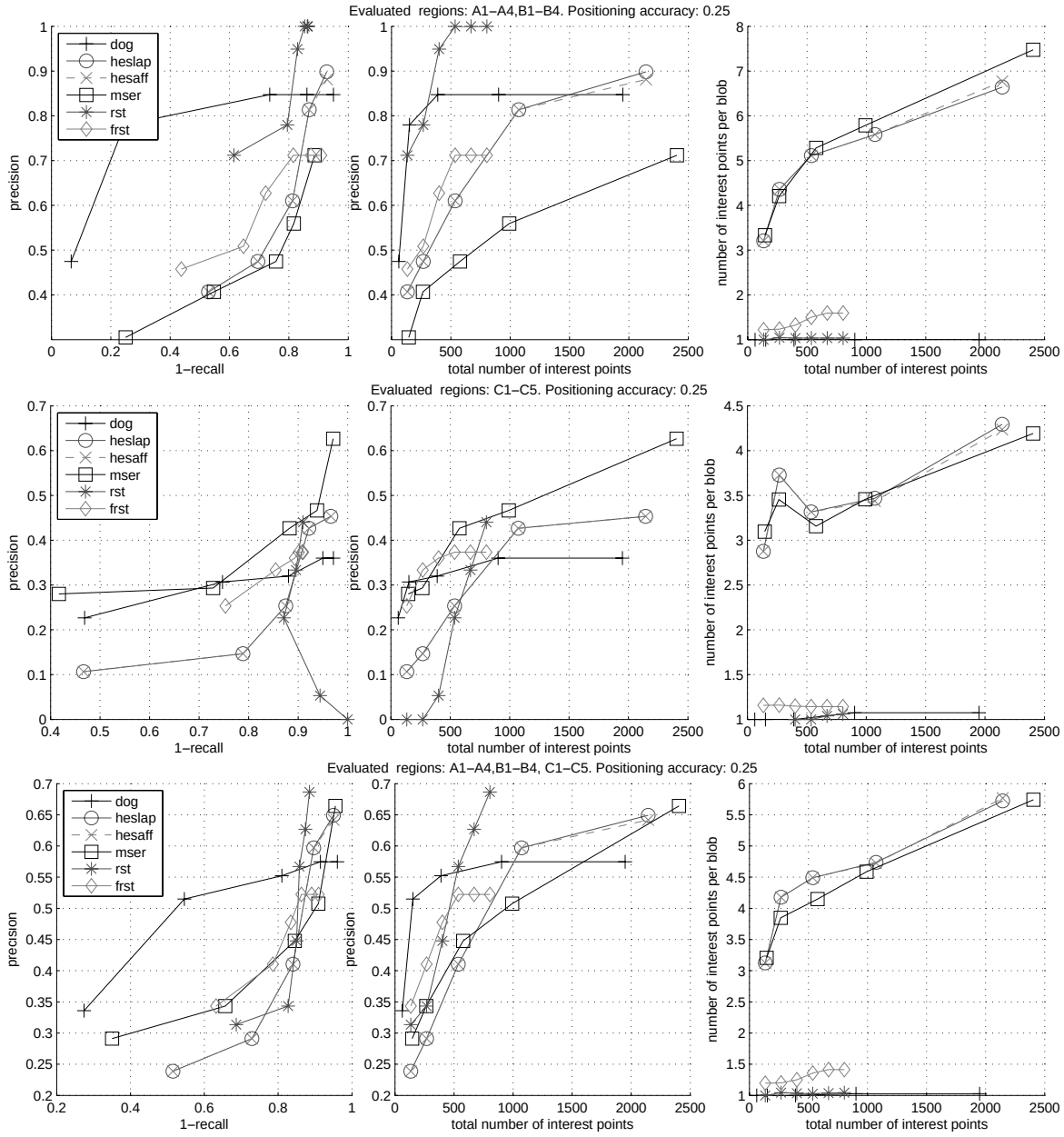


Figure 2.25: The results of blob detection accuracy for $\xi = 0.25$. Rows contain results for regions: A1-4∪B1-4, C1-5, A1-4∪B1-4∪C1-5 respectively from top to bottom.

these extrema depends on the similarity of corresponding structures to the Gaussian blob.

2.6.2 Corner Detection Performance

Corner detectors are evaluated using the test image in Figure 2.26. The image is a composite of 9 regions, each containing 12 squares at 4 scales and 3 orientations. Regions differ by intensity (0.25, 0.5, 1) and the size of Gaussian filter applied (0, 2, 5). The purpose of this evaluation is to show the sensitivity of the corner detectors to scale, rotation, intensity and blur variations.

The evaluation is carried out on the Harris and Harris-Laplace corner detectors described in Section 2.4. The results of Harris-Laplace apply also to the Harris-Affine detector since the difference between them relate to the scale estimation (not evaluated) and not corner location detection.

The test procedure measures the number of interest points produced per corner in each region, obtained for a set of "cornerness" thresholds t that correspond to the total number of interest points extracted from the test image $K = 50, 75, 100, 125, 150, ..., 3000$. The results of evaluation are presented in the form of tables in Figure 2.26 that show the average number of interest points per single corner in each region at a set of "cornerness" thresholds. Instead of absolute values, the relative threshold $\tau = t/t_{max}$ is shown, where t_{max} corresponds to 50 interest points extracted from the test image.

The evaluation results show the following:

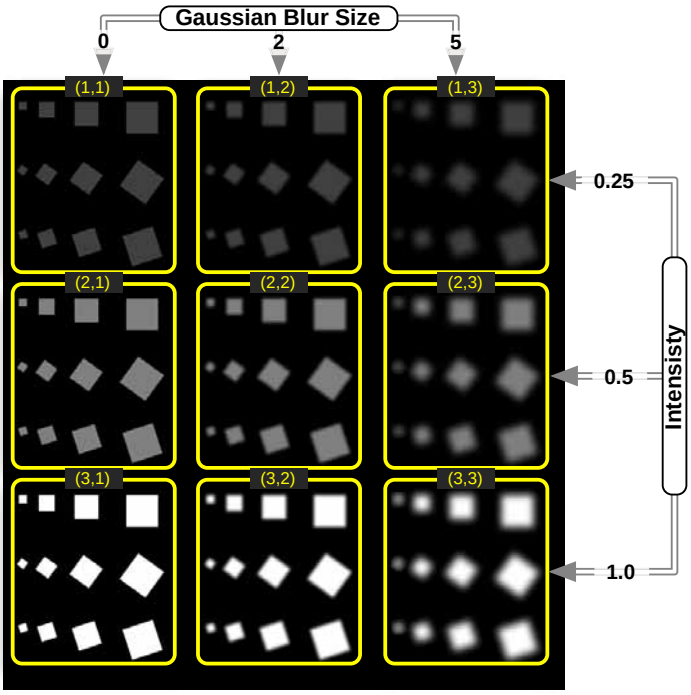
- The "cornerness" measure is sensitive to intensity changes. The ratio between threshold t_1 that allows to detect all corners in regions (3,1) (intensity=1, no blur) is 16.4 times higher than the threshold t_2 allowing detection of all corners in the region (2,1) (intensity=0.5, no blur). This means that the "cornerness" maxima in region (2,1) are approximately 16 times smaller than in the region (3,1) in case of the Harris detector. The same ratio for the Harris-Laplace detector is equal to 13.4. The t_1/t_3 ratio, where t_3 is the threshold that allows to detect all corners in regions (1,1) (intensity=0.25, no blur) equals to 227 in case of Harris and 263 in case of Harris-Laplace.

- The scale adapted Harris-Laplace detector is less sensitive to blur than the Harris detector. The results of corner detection in regions that differ only by the amount of blur are almost identical for the Harris-Laplace detector. In the case of the Harris

detector, the ratio between threshold t_1 and the threshold that allows to detect approximately half of the corners in the region (3,3) (intensity=1, blur $\sigma = 5$ pixels) is 227 which means that the corner maxima in the blurred region are over 200 times smaller than in the non-blurred, same intensity region.

- Both detecors are insensitive to rotation – the measured differences between "cornerness" maxima at different square orientations are negligible.

- The locations of interest points obtained from the Harris-Laplace detector at threshold t_1 ($\tau = 1$) correspond to the centers of squares and not individual corners as shown in the Figure 2.28. The detected scales indicate that the "cornerness" maxima correspond to the whole square. This is because the gradient covariance (and subsequently the eigenvalues of the second moment matrix (2.16)) calculated over the patch containing the square is higher than in case of the patch containing only the corner which occupies 25% of the patch area. The Harris detector operates at a constant scale and therefore cannot adapt the patch size to the whole square which would maximize the "cornerness" measure. However, the scale invariance of the Harris-Laplace detector causes that both square and individual corners are detected at lower thresholds $\tau < 0.6$ shown in Figure 2.28.



Harris

$\tau = t/t_{max}$	(3,1)	(2,1)	(1,1)	(3,2)	(2,2)	(1,2)	(3,3)	(2,3)	(1,3)
0	12.1667	13.1042	12.5417	9.5208	10.5833	9.5833	6.0000	6.0625	6.0833
0.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0208	0	1.1458	1.1667
0.0005	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	0.1250	1.1250
0.0034	1.0000	1.0000	1.0000	0.5208	1.0000	1.0000	0	0	1.0417
0.0044	1.0000	1.0000	1.0000	0	1.0000	1.0000	0	0	0.5417
0.0384	0.6875	1.0000	1.0000	0	1.0000	1.0000	0	0	0
0.0475	0.2083	1.0000	1.0000	0	1.0000	1.0000	0	0	0
0.0541	0	1.0000	1.0000	0	0.6250	1.0000	0	0	0
0.0609	0	1.0000	1.0000	0	0.0833	1.0000	0	0	0
0.6162	0	0.6042	1.0000	0	0	1.0000	0	0	0
0.7618	0	0.0833	1.0000	0	0	1.0000	0	0	0
0.8700	0	0	1.0000	0	0	0.5417	0	0	0
1.0000	0	0	1.0000	0	0	0.0625	0	0	0

$1/\tau = 2000$, $1/\tau = 227.3$, $1/\tau = 16.4$

Harris-Laplace (scale adopted)

$\tau = t/t_{max}$	(3,1)	(2,1)	(1,1)	(3,2)	(2,2)	(1,2)	(3,3)	(2,3)	(1,3)
0	12.1667	13.1042	12.5417	9.5208	10.5833	9.5833	6.0000	6.0625	6.0833
0.0000	7.6250	7.7292	7.7500	7.6250	7.7083	7.7292	5.4375	7.4167	7.6458
0.0015	6.6667	7.6875	7.7500	3.0417	7.6667	7.7292	0.7083	4.4792	7.0625
0.0038	0.7917	7.6667	7.7292	0.4583	7.6458	7.7083	0	3.3542	6.2708
0.0242	0	6.7292	7.7083	0	3.0208	7.6875	0	0.7083	4.4792
0.0342	0	4.2708	7.7083	0	1.7500	7.6875	0	0.6042	3.9375
0.0545	0	1.0417	7.6875	0	0.6667	7.6667	0	0.1250	3.6458
0.0744	0	0.4792	7.6875	0	0.1875	7.3750	0	0	3.1875
0.1189	0	0	7.6875	0	0	6.4583	0	0	2.5208
0.3177	0	0	7.5000	0	0	3.7708	0	0	1.2500
0.5223	0	0	4.7500	0	0	1.8542	0	0	0.5833
0.6414	0	0	2.2708	0	0	1.0625	0	0	0.3333
0.8088	0	0	1.2083	0	0	0.7500	0	0	0.1250
0.8971	0	0	0.8958	0	0	0.5625	0	0	0.1250
1.0000	0	0	0.6667	0	0	0.3750	0	0	0

$1/\tau = 263.2$, $1/\tau = 13.4$

Figure 2.26: The top row contains the test image used for evaluation of corner detectors, divided into 9 regions that differ by intensity and amount of blur. The tables below show evaluation results for Harris and Harris-Laplace detectors. The first column in both tables contains the relati ve threshold of the "cornerness" measure. Subsequent columns contain average numbers of interest points per corner in test regions. The red lines in the tables show the maximum relative thresholds τ required to detect all corners in corresponding regions.

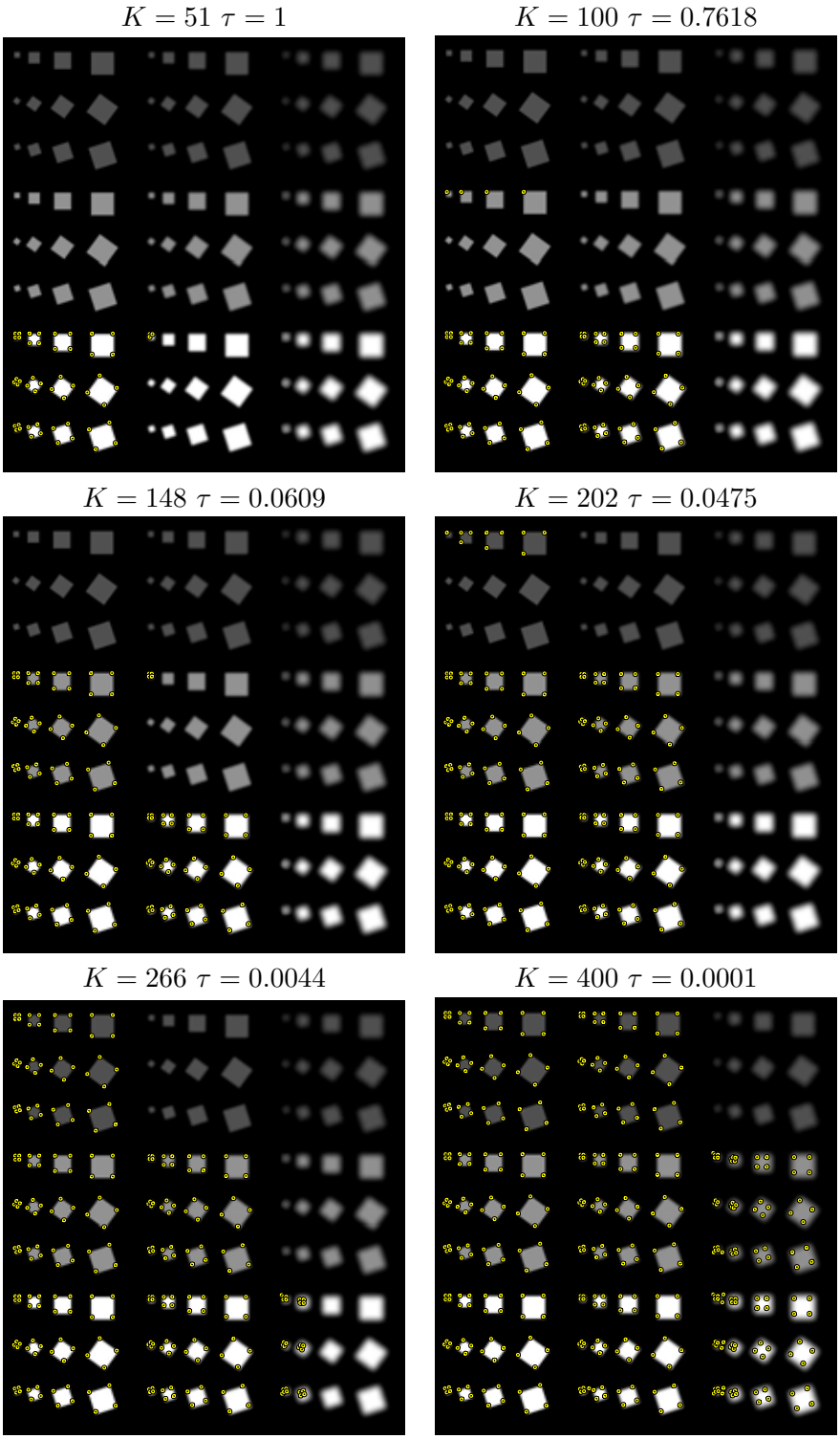


Figure 2.27: The Harris interest points obtained by thresholding the "cornerness" measure.

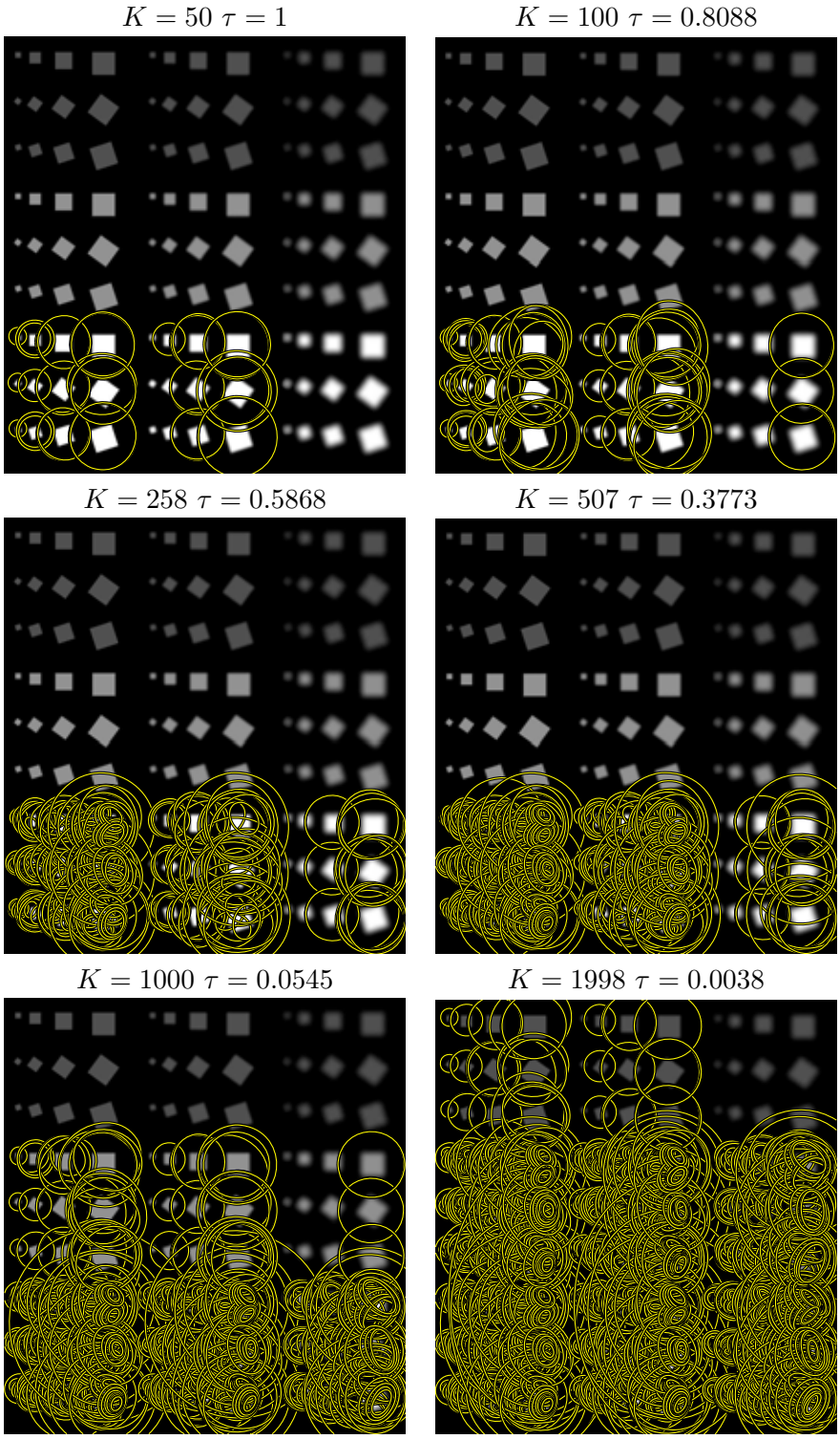


Figure 2.28: The Harris-Laplace interest points obtained by thresholding the "cornerness" measure.

Chapter 3

Image Description

An overview of current trends in object recognition was given in Section 1.2. The primary difference between them is a type of features used for recognizing objects. This chapter discusses the state of the art approaches related to two feature classes: local image descriptors and shape detectors which are related to the two novel image descriptors introduced in this thesis (see Chapter 4).

3.1 Local Image Descriptors.

This section provides an overview of existing local image descriptors used for patch based object recognition and scene matching methods. Their development started in 1980s with differential invariants by Koenderink et al. [26], steerable filters by Freeman and Adelson [15], moment invariants by Van Gool et al. [71], complex filters by Shaffalitzky and Zisserman [56], shape context by Belongie et al. [2], SIFT by Lowe [38] and its variations (e.g. PCA-SIFT, RIFT, GLOH).

Section 3.1.1 gives a brief discussion of each descriptor mentioned in this section which is followed by a detailed description of the SIFT and shape context approaches that were found to outperform other methods in existing comparisons [42, 74].

3.1.1 Overview of Local Image Descriptor

Differential invariants – were introduced by Koenderink and van Doom [26] in late 1980s and are closely associated with the creation of scale-space theory. The local image appearance at a given point is described as a set of derivatives up to the k-th order of the image convolved with a Gaussian at various scales. The length of the

feature vector depends on the maximum derivative order $(0, ..., k)$ and the number of scales used. Further extension of differential invariants are steerable filters discussed below.

Steerable filters – are a linear combination of basis filters, which allow to adaptively "steer" a filter to any orientation and measure the filter response as a function of the orientation. In practice the orientation domain is quantized and filter responses at arbitrary orientations are interpolated. The set of filters used in [15, 42] consists of 2D Gaussian derivatives up to fourth order as well as their rotated and scaled versions. It is shown in [15] that a steerable pyramid can be used for multi-scale image decomposition, much like a wavelet transform. Description of an image patch can be obtained by combining the responses of each filter into a feature vector. The particular set of filters is application dependent. According to the local descriptor evaluation in [42], the steerable filters (14 filters used, two orientations) were outperformed by SIFT and shape context descriptors in all scene matching tests. For example, matching 400 regions in the "grafitti" scenes undergoing view point change with steerable filters produced approximately half of the true positives scored by SIFT and approximately 25% more false positives. However it remains to be verified how the performance changes when the number of filters is increased, specifically when more orientations and scales are available.

Moment invariants – are based on shape and intensity moments up to the second order, calculated over a region Ω:

$$MS_{Cpq} = \int\int_{\Omega} x^p y^q dxdy \quad \text{and} \quad MI_{Cpq} = \int\int_{\Omega} i(x,y) x^p y^q dxdy$$

where (p, q) depict the order of the moment. It is shown in [71] that the combinations of these moments provide quantities that remain invariant under affine transformations. According to the local descriptor evaluation in [42] moment invariants outperformed steerable filters but were also inferior to the SIFT based descriptors.

Complex filters – are of the form $K_{mn}(x, y) = (x + \imath y)^m (x - \imath y)^n g(x, y)$, where g is a Gaussian function [56]. The image patch is then represented by the feature vector consisting of filters computed with different combinations of m and n (typically $m + n \leqslant 6$ and $m \geqslant n$ which gives 16 complex filter responses per patch). The advantage of these filters over steerable filters is that the rotation changes the phase but not the magnitude of the filter response which produces intrinsic rotational

invariance. Their performance however did not match the one produced by steerable filters, according to [42].

Shape context – is a 2D histogram of boundary point positions relative to the center of a circular interest region [2]. A detailed description is given in Section 3.1.3.

The Scale Invariant Feature Transform (SIFT) – encodes weighted histograms of gradient orientations computed from a scale-space representation of the local image patch [38]. A detailed description is given in Section 3.1.2. The existing evaluations of local image descriptors [42, 74] show the superiority of this method over the other descriptors mentioned in this section.

3.1.2 SIFT

The Scale Invariant Feature Transform by Lowe [38] is one of the top performing local image descriptors currently used [42, 52], with at least six implementations available* and 1611 citations†. It is used for object recognition [52] and scene matching [42, 52].

The SIFT descriptor is intended as a tool to extract and encode local image features that are scale and illumination invariant. The descriptor features are extracted from a region of the image and therefore the invariance to rotation and scale or affine transformation depends on the particular region detector applied e.g. Harris-Affine (see Section 2.4). In a typical case a single image is represented by a number of descriptors associated with the detected interest regions.

In the original paper [37] SIFT adopts a DoG blob detector (see Section 2.3) to obtain interest point locations and scale. This approach is later enhanced in [38] to refine the interest point positions and allow the elimination of edge responses. The scale-space is divided into octaves and in every octave the Gaussian smoothed image is down-sampled by a factor of 2 as is shown in Figure 3.1. The down-sampling of the image reduces the computational complexity of the feature extraction since it is more efficient than convolving the image with a kernel of twice the size. It also allows the extraction of features using the same size (typically 16×16 pixel) window in each octave while preserving scale invariance. The scale resolution is defined by the multiplicative constant factor k thus the difference of gaussian D is defined as follows:

$$D(\boldsymbol{x}, \sigma) = L(\boldsymbol{x}, k\sigma) - L(\boldsymbol{x}, \sigma) \tag{3.1}$$

*http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT
†by 12 Dec 2007 - http://scholar.google.at/scholar?q=david+lowe&hl=en&lr=&btnG=Search

Note that the formula does not contain the scale normalization factor. It is shown however in [38] that for values of k close to 1 (typically $k = \sqrt{2}$) this approach can be used to approximate LoG.

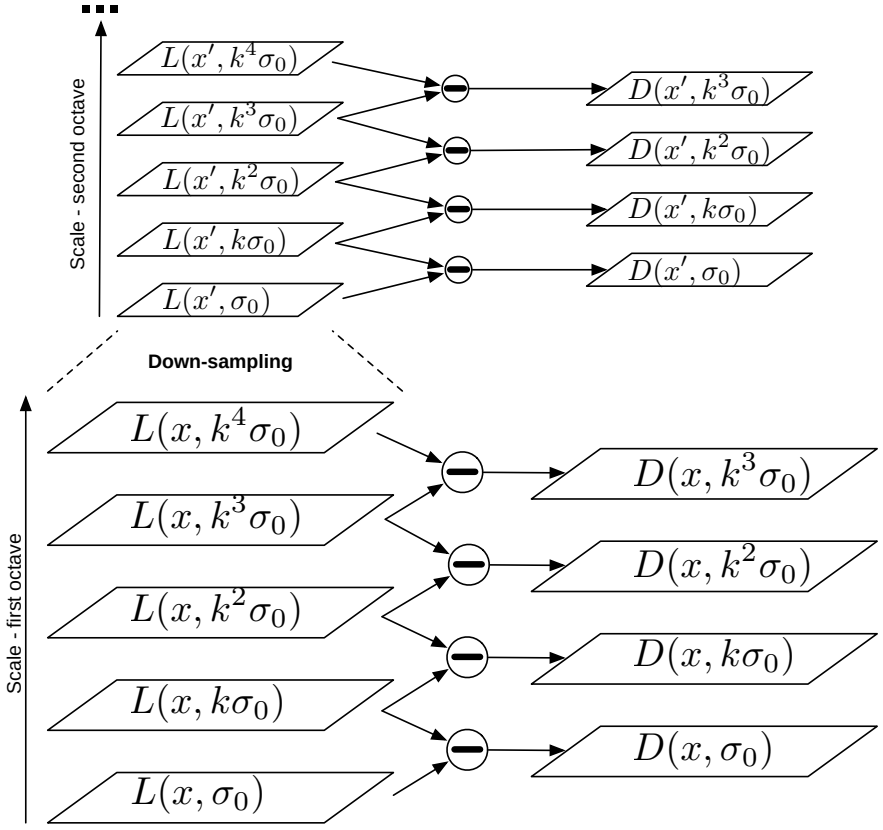


Figure 3.1: A scale-space volume of the image used by the SIFT, based on the DoG approach. The symbol x' represents the Gaussian image down-sampling by a factor of two.

Rotational invariance is achieved through a gradient orientation estimation or orientation obtained from a region detector, depending on the chosen strategy. In [38] the local patch orientation is computed from a histogram of gradient orientations of sample points within a region around the interest point. The histogram contains 36 bins covering a 360 degree range of orientations. Sample orientations $\theta(x, y)$ added to the histogram are weighted with the gradient magnitude $m(x, y)$:

$$m(x,y) = \sqrt{(L_{\sigma_{max}}(x+1,y) - L_{\sigma_{max}}(x-1,y))^2 + (L_{\sigma_{max}}(x,y+1) - L_{\sigma_{max}}(x,y-1))^2} \tag{3.2}$$

$$\theta(x,y) = \arctan \frac{L_{\sigma_{max}}(x,y+1) - L_{\sigma_{max}}(x,y-1)}{L_{\sigma_{max}}(x+1,y) - L_{\sigma_{max}}(x-1,y)} \tag{3.3}$$

where $L_{\sigma_{max}}(\boldsymbol{x} = \{x,y\}) = L(\boldsymbol{x}, \sigma_{max})$ and σ_{max} is a detected scale.

Histogram peaks which are within 80% of the strongest peak value are then used as dominant orientations of the region and for each dominant orientation a separate SIFT descriptor is extracted. This means that a single region can be represented by multiple descriptors which share position and scale but differ in their orientation. The number of descriptor instances per detected region depends on the particular method for orientation estimation.

The SIFT descriptor is extracted from the rectangular region of the position, extent and orientation corresponding to the interest point. The region contains gradient magnitude and orientation computed from the scale-space representation $L(\boldsymbol{x}, \sigma)$ (possibly down-sampled) corresponding to the detected scale σ at the interest point. These are additionally weighted by a Gaussian function to prioritize pixels closer to the interest point. The rectangular region is divided into windows of size 4×4 pixels, as shown in Figure 3.2. Typically 4×4 windows are used which correspond to 16×16 pixel areas covering the whole region, however the number of windows as well as the number of pixels can be adjusted if needed. The histogram of pixel orientations is computed for each window, containing 8 bins covering a 360 degree range of orientations. The contribution of each pixel to the histograms is distributed over a range of bins using tri-linear interpolation to avoid all boundary effects (see "Descriptor representation" in [38]). The SIFT descriptor contains 16 histograms, which in total produce a 128 element feature vector. The vector is further normalized to a unit length.

The parameters used in [38], that is 8 orientation bins and $4 \times 4 = 16$ windows, have been experimentally verified by the authors and found to consistently performed better than lower dimensionality descriptors, while increasing the dimensionality did not improve results significantly or even caused some performance degradation since descriptors became more sensitive to shape distortions and occlusions [38].

Let us now discuss two extensions of the SIFT approach: PCA-SIFT [24] and GLOH [42] that were shown by their authors to improve performance of scene matching

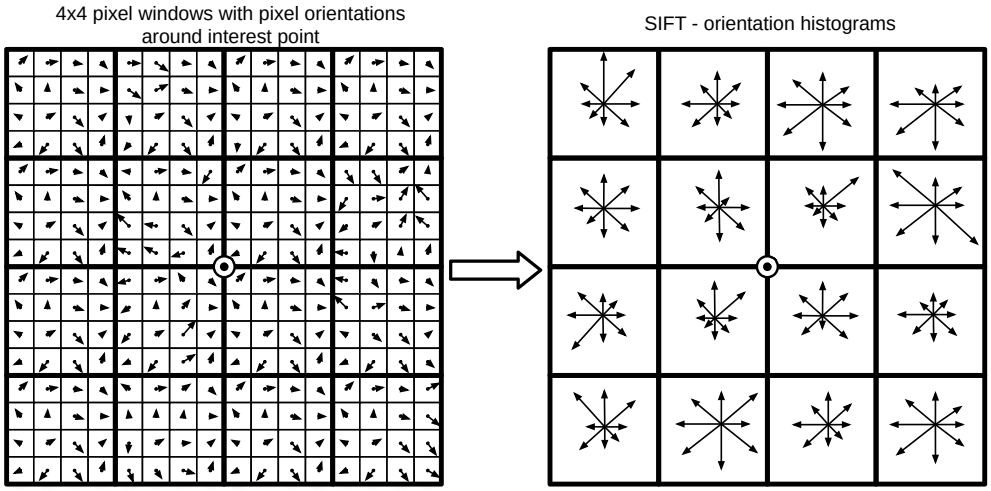


Figure 3.2: The SIFT descriptor typically consists of 16 (4×4) orientation histograms, extracted from 4×4 pixel windows each around the interest point (marked with a double circle). Each histogram contain 8 bins covering a 360 degree orientation range.

and object recognition.

The PCA-SIFT [24] employs the same approach for the extraction of image regions and image gradient representation as the SIFT method. The difference is that the extracted data (vertical and horizontal gradients) from a region of size 41x41 are projected into a pre-computed eigenspace which reduces the dimensionality of the descriptor and allows to obtain a compact feature vector (in [24], 20 elements as opposed to 128 elements in SIFT). The eigenspace is obtained by applying PCA [11] to 21000 feature vectors (concatenated vertical and horizontal gradients in the region) corresponding to regions extracted from a diverse set of images. The authors claim that their method reduces the computational complexity and improves distinctiveness of the descriptor which leads to increased matching accuracy. However, the performance evaluation of local descriptors in [42] does not confirm a superior matching accuracy of PCA-SIFT over SIFT, where in majority of cases PCA-SIFT has been outperformed by SIFT, GLOH and shape context (see Section 3.1.3).

The primary difference of the *Gradient Location-Orientation Histogram* GLOH descriptor [42] to SIFT is that the feature vector is extracted from a log-polar location grid with 3 radial bins (at the radius 6, 11 and 15) and 8 bins in the angular direction which corresponds to 17 windows. The gradient orientations within each window are quantized

into 16 bins which produces a 272 element feature vector. The size of the feature vector is further reduced using PCA as in case of the PCA-SIFT method, except that the covariance matrix used for PCA is computed from 47000 image patches. The 128 largest eigenvectors are used. GLOH was shown to outperform or match the performance of the SIFT and other tested descriptors in the task of matching different views of the scene [42].

3.1.3 Shape Context

Shape context is a local shape and image descriptor, however it can also be used for recognition of complex shapes which undergo distortions and affine transformations [2]. In its basic form shape context is a histogram of point coordinates, corresponding to the object boundary, relative to the central location at which the shape context is extracted. Figure 3.3 shows the example of boundary points sampled along the edges of the letter A, the histogram bins that cover a log-polar space around the descriptor center and the histogram itself. The log-polar spatial distribution of the histogram bins makes the descriptor more sensitive to the positions of points closer to the center than further away. The boundary points are uniformly sampled from edges obtained from an edge detector e.g. Canny [4], and then assigned to the histogram bins depending on their relative position with respect to the descriptor center (see Figure 3.3).

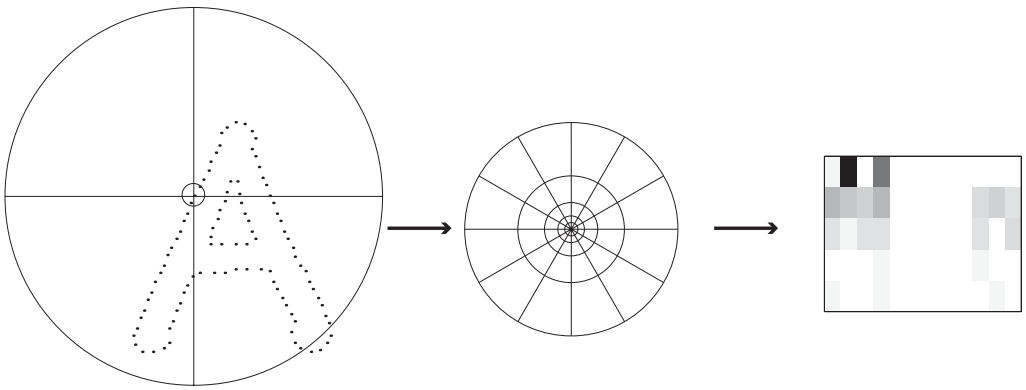


Figure 3.3: Illustration of the shape context descriptor extraction. Left: the circle with cross inside mark the center and extent of the region from which the descriptor is extracted. Center: the histogram bins occupy uniform log-polar space (though exact proportions are implementation dependent). Right: two dimensional histogram representing the number of points within each bin.

The number of histogram bins is application dependent, but in a typical case it varies

from 3 to 5 bins in the radial direction and 9 to 16 bins in the angular direction. Since the single bin is a counter of boundary points in the corresponding image region, the local shape deformations that do not cause boundary points to leave regions occupied by their initial bins have no effect on the final histogram. Because the image area occupied by a single bin grows with the radial distance from the descriptor center the overall tolerance to shape deformations increases but at a cost of the ability to discriminate details at a scale lower.

Let us consider two similar shapes, represented by points $P = \{p_1, ..., p_N\}$ and $Q = \{q_1, ..., q_M\}$. Shape context descriptors extracted from the regions around each point will differ, since the values of histogram bins depend on the location of the region center. Therefore for each point p_i the corresponding point q_j can be found for which the dissimilarity measure C_{ij} between associated shape context descriptors is minimized. As shape context is an approximation of boundary point distribution it is natural to use the χ^2 measure:

$$C_{ij} = \frac{1}{2} \sum_{k=1}^{K} \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)} \tag{3.4}$$

where $h_i(k)$ and $h_j(k)$ are k-th bins of K-bin histograms extracted around points p_i and q_j respectively.

Similarity between two shapes can be estimated as the total cost of matching all points p_i with their most similar corresponding points:

$$H = \sum_i \min_j (C_{ij}) \tag{3.5}$$

This measure can be calculated in an efficient way as described in [2].

It is possible to obtain rotational invariance either by using an edge tangent vector at the region center as an orientation estimation or by using an interest region detector for orientation and scale estimation in which case scale invariance is also possible.

A slightly modified form of the shape context, encoding both edge point positions and orientations, has been compared with other local descriptors in [42] outperforming all other methods but SIFT in the task of matching different views of the same scene. Another performance evaluation related to matching shapes has been reported in [2]. The authors described a method to model shape deformations and transformations using a thin plate spline (TPS) model. This approach allows one to obtain a similarity measure that resembles human perception e.g. the deformation between shapes such as letter Z

and digit 2 or letter S and digit 5 is lower than between letters Z and S or digits 2 and 5. The evaluation of the shape similarity measure is provided in [2], however dependence of the measure to scale variations remains to be investigated.

Generalized Correlograms (GC) [1] are an extension of the shape context descriptor, allowing one to encode in the histogram multiple features related to the object boundary points. The GC approach uses the same log-polar spatial distribution of bins as the shape context, however each bin contains a histogram of multiple properties of points that are spatially covered by the bin. In [1] the boundary points are sampled from from the contours of the segmented image. The result is a 3D histogram of $n_\alpha \times n_r \times n_L$ bins, where n_α is the number of bins in the angular direction, n_r is the number of radial bins and n_L is the number of feature histogram bins associated with each boundary point. The feature histogram is a combination of the local contour orientation angle histogram, quantized into n_θ bins, and the color histogram quantized into n_c bins. Note that this approach can accommodate any number and type of features, however special consideration should be given to histogram normalization if the number of histogram bins dedicated to different features varies. The final dimensionality of the feature vector obtained from the 3D histogram depends on the number of local features used and their quantization. It is possible to reduce this dimensionality by adaptively using only a subset of the local features e.g. use only contour orientation angles or local colors, depending on which feature gives a more discriminative description of a particular part of the object. In order to obtain scale invariance GC is extracted at a number of predefined scales (in [1] 7 scales of radius in range 130..280 pixels have been used) while the radial distances are normalized.

In [1] the GC descriptors have been used for object-class recognition, where a weakly supervised learning architecture allows multiple object models to be obtained. A matching algorithm was employed for detecting the presence of learned objects in the scene. The object models were represented as a constellation of GC descriptors and learned using the *Joint Boosting* based algorithm [68]. The evaluation of the approach on a classification of objects in the CALTECH image database [12] have shown that GC consistently outperformed shape context and local histograms.

3.1.4 Discussion

The performance of local descriptors in applications such as object recognition and scene matching depends on three factors: interest region detection, feature extraction (descriptors) and the pattern recognition approach used (in case of object recognition). The local

image descriptors are used either to find correspondences between image regions (scene matching) or estimate similarity to learned local image representations (object recognition). The result of similarity estimation depends on the descriptiveness of the descriptor but also on the consistency of interest region detection which provides both the location and scale selection relative to the compared image structures (see Section 2.6). The performance dependency on the choice of local descriptors and region detectors is shown in [42] (scene matching) and [74] (object recognition). In [42] the SIFT descriptor, its extension GLOH, PCA-SIFT and shape context outperform all other descriptors (see Section 3.1.1) in every test scenario. However, the object recognition performance evaluation in [74] (using image databases such as CALTECH and PASCAL) shows that the combination of two or more different descriptor types yields better classification accuracy (by up to 7% in their test scenario) than in case of a single descriptor.

The applicability of local descriptors to scene matching or object recognition depends also on the contents of the analyzed images i.e. how informative the local image patches are. In [52] the SIFT descriptor is used for finding the same urban scenes from images taken at different light conditions and view points with average accuracy higher than 0.6 (average precision score) on a 5000 image dataset. However, the SIFT descriptor applied to matching animals in natural scenes [63] exhibited an average animal matching accuracy of approximately 30% (180 images).

3.2 Image Shape Detectors.

Since the primary contribution of this thesis is a novel shape detector we will now discuss several other approaches to shape detection and classification in still images and relate them to our work.

Our primary focus is on the methods that can learn shape-based object models from training images presented in Sections 3.2.4, 3.2.5 and 3.2.6 which are related to our contributions presented in Chapters 4 and 5. We also discuss the older *Geometric Hashing* method in Section 3.2.1, which introduces general shape detection techniques used also at present.

3.2.1 Geometric Hashing

Geometric Hashing is a model based object detection approach proposed by Schwartz and Sharir in 1986 [59], extended later by Schwartz, Wolfson and Lamdan [28]. Their approach allows to obtain invariance to view point transformations (including translation, rotation, scaling, affine and projective transformations), is robust to partial occlusions and able to operate on arbitrarily large databases.

In the original work geometric hashing operates on objects that are represented as clouds of point coordinates (e.g. extracted from detected edges). The object model contains scene independent information encoded in a hash table that describes geometric relations between these points and possibly other types of features. Let us consider an object from Figure 3.4 (left) represented by 12 points (order and linking of points is not used). We also need to define a frame of reference in order to make the object detection invariant to view-point changes, e.g. associating a frame of reference with any two points belonging to the object allows to achieve invariance to translation, rotation and scaling. We have chosen an ordered pair of points $\overrightarrow{\boldsymbol{p}_7\boldsymbol{p}_1}$ in Figure 3.4 (right) to be the basis for such a frame of reference. The object points are then transformed (translated, rotated and scaled) such that the $|\overrightarrow{\boldsymbol{p}_1\boldsymbol{p}_7}| = 1$ and points p_7 and p_1 are located at coordinates $(-1/2, 0)$ and $(1/2, 0)$ respectively. Thus if we know the location of p_7 and p_1 of the related object we can predict the locations of other points in the reference frame. Since our goal is to detect objects under partial occlusion there is no guarantee that both basis points will always appear in each instance of the object. The object model therefore must encode all possible combinations of point pairs taken as the basis of the reference frame.

Figure 3.4 shows an example of the object represented by a set of points along the object contour (left) and the object aligned to the reference frame defined by the pair of

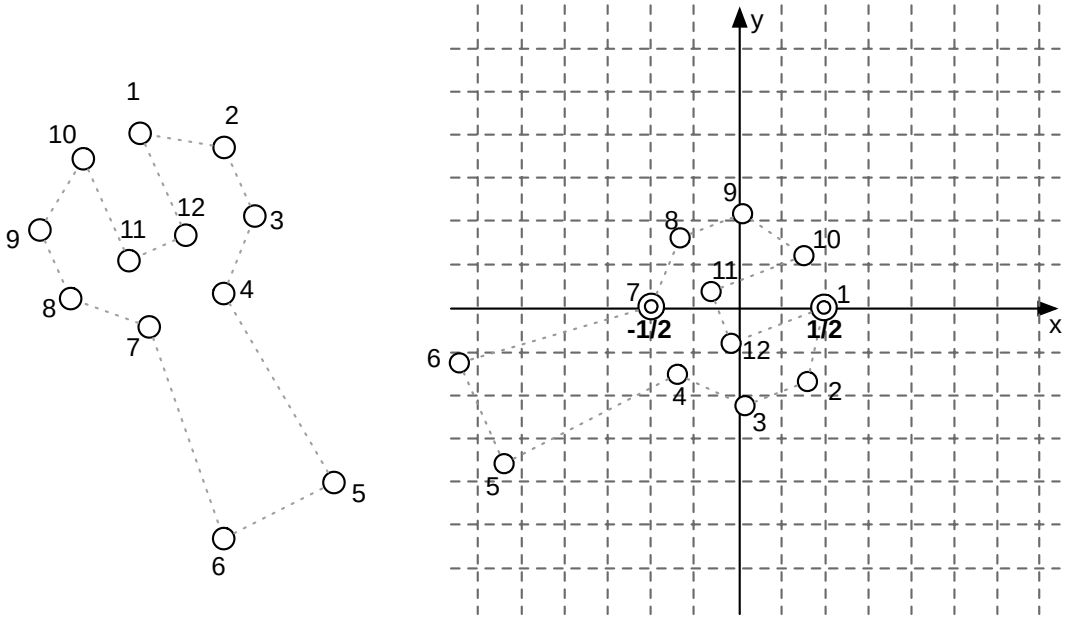


Figure 3.4: Left: example of object represented by 12 points. Right: the object points are translated, rotated and scaled such that points p_7 and p_1 are placed at coordinates $(-1/2, 0)$ and $(1/2, 0)$ respectively and become the basis of the reference frame. The grid visualized with dashed lines represents hash table cells associated with coordinates of the transformed object points.

object points (right). The hash table is an M-dimensional array of cells, where each cell spans a range of M feature values (visualized as a grid in Figure 3.4) in the feature space, such as (x, y) coordinates of the transformed points in Figure 3.4 (right). Each object point in Figure 3.4 (right) is stored in a hash table cell, corresponding to the transformed coordinates, as an entry containing the object category and the basis point pair (i, j). A hash table cell can contain multiple entries that correspond to different objects, different basis points or same object and basis points, depending on the quantization settings of the feature space. The creation of the hash table requires that all the feature vectors (such as point coordinates) representing only objects are provided, which is a drawback in comparison to other methods presented in this chapter that can extract object related data during the training process.

The frame of reference basis defined as a pair of object points suffices to achieve the invariance to a similarity transform. However geometric hashing is a unified approach that applies also to other transformations such as the affine transformation which requires

a three point basis or the projective transformation with a four point basis. However the increase in the number of basis points also increases the computational complexity as the the number of a single object's transformed points encoded in the hash table is $(N - K)N^K$, where N is a number of the initial object points and K is the number of basis points.

Assuming we have a hash table that encodes the desired object models, the object detection can be summarized in few simple steps:

1. Extract a set of key points S from the image, such as points along the detected edges.
2. Choose an arbitrary basis-set of K points.
3. Transform the coordinates of points in S into the coordinate system defined by the selected basis.
4. For each point in S: find the hash table cell that corresponds to the point coordinates and for every entry belonging to this cell cast a vote for the object category and the basis. This process effectively creates a voting histogram where each histogram bin corresponds to a unique combination of object category and basis.
5. Select histogram bins with a number of votes that exceed some threshold that represent potential object matches.
6. For each potential object match find the transformation T that results in the best least-squares match between all corresponding feature pairs.
7. Transform the object model according to the transformation T and verify the object model features against the image features. If the verification fails return to step 2.

Two primary advantages of the presented method are the ability to handle arbitrary image transformations and robustness against partial occlusions. The computational complexity depends on the type of invariance required – the overall number of different reference frames is N^K, where N is a number of key points and for each frame of reference (can be randomly selected) the object detection has to be repeated. In its basic form this approach requires object model related features to be precisely specified, which in reality requires a precise manual segmentation of the objects or their boundaries. Another potential problem is associated with the sensitivity of this approach to the object deformations, which is tightly related to the feature space quantization and cannot be regulated for each object separately – all object models are stored in the same hash table.

3.2.2 Scale Invariant Shape Features

Jurie and Schmid [22] proposed scale-invariant shape regions and applied them to object detection using a descriptor similar to the shape context (see Section 3.1.3).

The scale-invariant shape regions are aligned with the maxima of saliency measures in the scale-space domain, corresponding to a local shape convexity. This measure is calculated at several scales and for each pixel in the smoothed and down-sampled image at each scale. The measure at pixel c is obtained from contributions of individual pixels $\boldsymbol{p}_i$ near the circle centered at $\boldsymbol{c}$ of radius σ as in Figure 3.5 which shows the point p_i along the extracted edge, the corresponding intensity gradient g_i as well as the angle between the gradient g_i and the line connecting pixel c with the pixel p_i. The pixel contributions reflect the closeness to the circle represented by the weight $w_i^d(\boldsymbol{c}, \sigma)$ and alignment with its local tangent represented by the weight $w_i^\alpha(\boldsymbol{c}, \sigma)$:

$$w_i^d(\boldsymbol{c}, \sigma) = \exp\left(-\frac{(\|\boldsymbol{p}_i - \boldsymbol{c}\| - \sigma)^2}{2(s\sigma)^2}\right) \tag{3.6}$$

$$w_i^\alpha(\boldsymbol{c}, \sigma) = \|\boldsymbol{g}_i\| \cos \angle(\boldsymbol{g}_i, \boldsymbol{p}_i - \boldsymbol{c}) \tag{3.7}$$

where s defines the scale of detection corresponding to the distance of the point p_i from the circle ($s = 0.2$ in [22]).

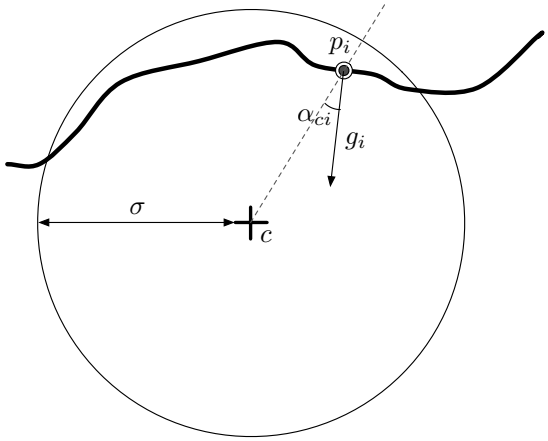


Figure 3.5: The region extraction is based on finding local saliency maxima across scale and space. The saliency measure at given point $\boldsymbol{c}$ and scale σ is calculated using points p_i near the circle along with their gradient g_i and the relative gradient orientation α_{ci}.

The final weight of the point p_i is a combination of both weights:

$$w_i(\boldsymbol{c}, \sigma) = w_i^{\alpha}(\boldsymbol{c}, \sigma) w_i^{\alpha}(\boldsymbol{c}, \sigma) \tag{3.8}$$

The saliency measure $C(\boldsymbol{c}, \sigma)$ is a product of *tangent edge energy* and *contour orientation entropy*. The *tangent edge energy* $E(\boldsymbol{c}, \sigma)$ measures the strength and alignment of the edges with the circle:

$$E(\boldsymbol{c}, \sigma) = \sum_{i=1}^{N} w_i(\boldsymbol{c}, \sigma)^2 \tag{3.9}$$

where N is a number of edge pixels around point $\boldsymbol{c}$.

The *contour orientation entropy* $H(\boldsymbol{c}, \sigma)$ measures the support from the distribution of points from around the circle[‡]:

$$H(\boldsymbol{c}, \sigma) = -\sum_{k=1}^{M} h(k, \boldsymbol{c}, \sigma) \log \Big(h(k, \boldsymbol{c}, \sigma) \Big) \tag{3.10}$$

where $h(k, \boldsymbol{c}, \sigma)$ is the k-th bin of the gradient orientation histogram:

$$h(k, \boldsymbol{c}, \sigma) = \frac{1}{\sum w_i(\boldsymbol{c}, \sigma)} \sum_{i=1}^{N} w_i(\boldsymbol{c}, \sigma) K \left(k - \frac{M}{2\pi} o_i \right) \tag{3.11}$$

where o_i is the angular orientation in radians of the contour gradient vector g_i and $K(x) = exp(-x^2/2)$.

The saliency measure is then:

$$C(\boldsymbol{c}, \sigma) = E(\boldsymbol{c}, \sigma) H(\boldsymbol{c}, \sigma) \tag{3.12}$$

In the experiments the authors used 30 scale levels spaced by a factor of 1.1 (parameter that regulates the quantization of scale domain) for the region detection. At each scale the input image is down-sampled which allows to use a constant circle radius of 5 pixels and also reduces computational complexity. The saliency measure maxima are suppressed within 9×9 window spatially and 3 in scale. Figure 3.6 shows an example of regions extracted at different scales marked with white circles where the detection scale corresponds to the circle radius.

The scale-invariant shape regions are applied to object detection in images from the

[‡]This measure should be relatively low if only few points on one side of the circle provided exhibit strong weights w_i.

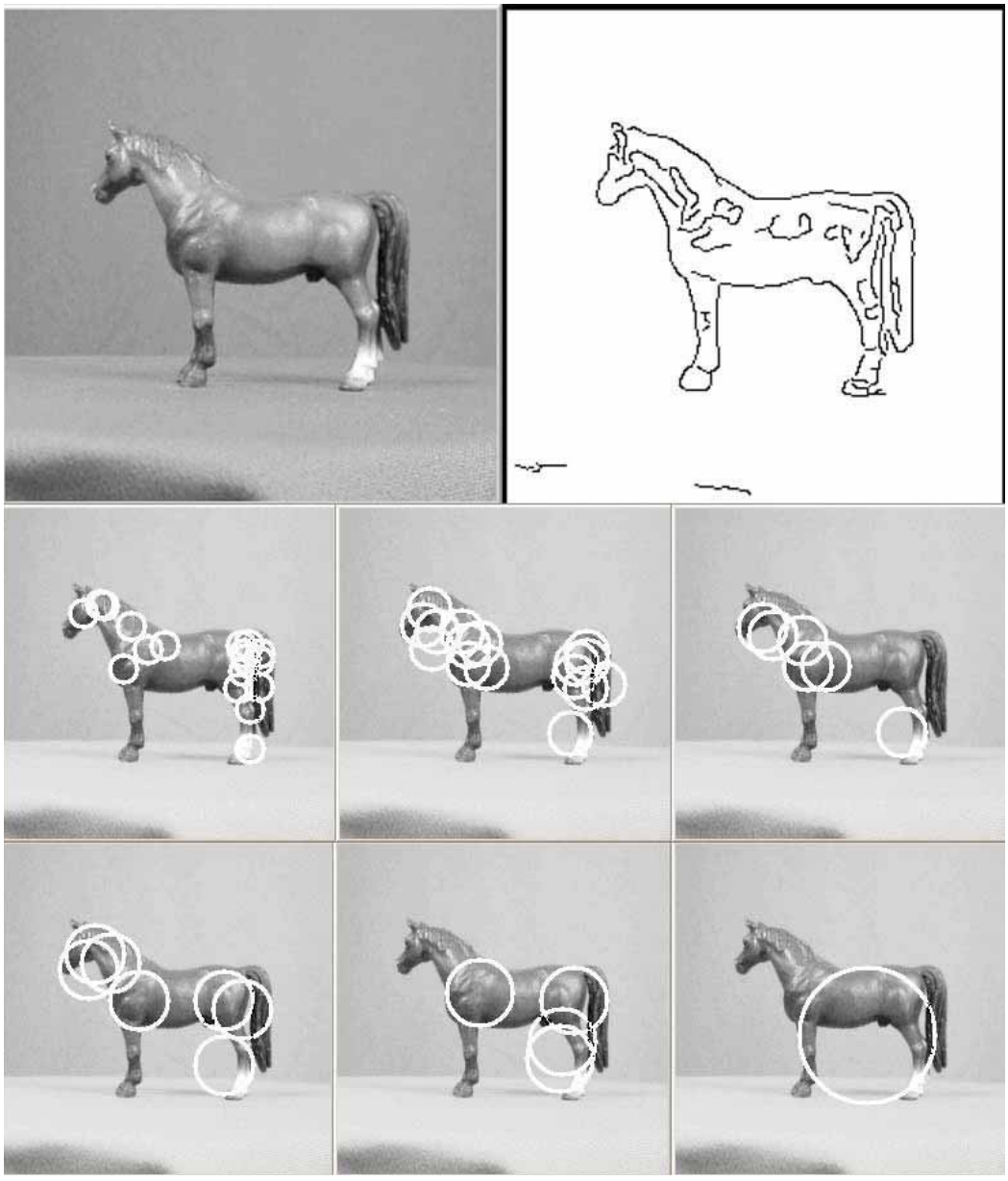


Figure 3.6: Example of scale-invariant shape regions detected at various scales. The image was obtained from [22].

ETH-80 database[§] using descriptors similar to the shape context discussed in Section 3.1.3. The primary difference is that the log-polar histogram of spatial distribution of edge pixels is computed near the circle of the detected region with 32 angular bins and 4 radial bins. Object models consist of descriptors along with the relative object frame (relative position and scale) learned from few training images. The final object detection is based on finding correspondences between model descriptors and the descriptors from the analyzed image. Each match votes for the position and size of the object frame. The object candidates are obtained using Mean-Shift Mode estimation [8]. For each mode the object frame is aligned and the number of matches calculated. The object is recognized if this number is above the threshold.

The method is primarily aimed at the detection of scale-invariant regions that exhibit a shape saliency and shows promising results in the classification of different objects from the ETH-80 database (cars, bicycles, horses etc.). In addition, the location of recognized objects within the image is detected in the form of the rectangular bounding box.

3.2.3 Recognition of Wiry Objects

Carmichael and Hebert introduced a method for shape based recognition of wiry objects [5]. Their approach is built upon the analysis of local edge distribution, which allows one to classify individual edge pixels with high accuracy as those which belong to the object or otherwise background. The edge pixels corresponding to objects are then aggregated into a rectangular window representing the region occupied by the object. The method does not allow the detection of the type of object, however further extensions are possible. The classification process of the edge pixels relies on two components: a multi-scale edge distribution descriptor and a *Decision Tree* [54] based classifier. We will now focus on the novel feature extraction method and then briefly discuss the classifier itself.

The basis of the edge classification approach into "object" and "background" categories is a local edge configuration descriptor called *aperture*. The aperture consists of *edge probes* distributed over a regular grid around the aperture center as shown in Figure 3.7-a. Each edge probe $ep(\boldsymbol{p}, \Omega)$ measures the edge pixel Gaussian density around its center $\boldsymbol{p}$:

$$ep(\boldsymbol{p}, \Omega) = \sum_{\boldsymbol{t} \in \Omega} \exp\left(-\frac{\|\boldsymbol{p} - \boldsymbol{t}\|^2}{\sigma^2}\right) \tag{3.13}$$

[§]http://www.mis.informatik.tu-darmstadt.de/Research/Projects/categorization/eth80-db.html

where Ω is a circular region around the center of the edge probe and σ defines the extent of the gaussian function.

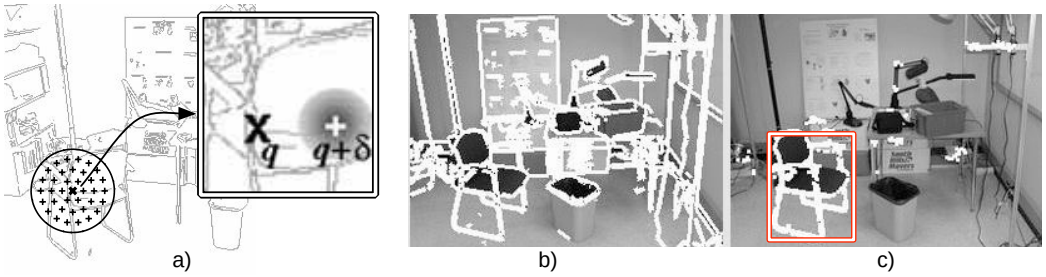


Figure 3.7: An example of edge point classification into object and background categories: a) distribution of edge probes (crosses) inside the aperture (circle), b) detected edges c) "object" edges after removal of "background" edges, object enclosed by aggregation window. (From [5])

The classification of edge pixels into "object" and "background" categories is performed using an iterative approach in the following way:

1. The initial set of pixels classified as "object" is obtained using the user defined minimum aperture size r_0

2. The aperture size is increased (user defined parameter) and the pixels previously assigned to the "object" category are classified again. Pixels re-classified as "background" are removed.

3. Step 2 is continued until the aperture size reaches a user defined value.

This approach allows to capture object specific features at multiple scales, however it is expected that the descriptiveness of the features increases together with the scale and therefore by gradual increasing of the aperture size the previously misclassified "background" pixels can be pruned.

The classification of aperture based features is performed using the Decision Tree method [54] for its ability to classify based on a sparse set of features and efficiency.

The classifiers were trained to detect chairs, carts and ladders in cluttered scenes. The ground truth was obtained from images of chairs and carts taken against a blue screen at various angles. Then a random background was added and the sampled features were used for training. The image containing ladders were manually segmented and the same

learning technique applied. The final classification performance exhibits a high average rate of correct pixel classification exceeding 70% in the majority of cases, while the amount of false positives oscillated around 10% of all edge pixels. This result allows the reliable location of learned objects within the scene.

3.2.4 Shape Alphabet

Opelt at al. proposed a shape based object recognition method which introduced a *Boundary Fragment Model* (BFM) for description of object shape and incremental category learning which allows the addition of new category models without the necessity of restarting the training procedure [51]. They have reported outstanding classification and detection results in a database containing 17 objects such as cars, airplanes, cows, horses, persons, bicycles, cups, etc.

Their method addresses the problem of learning object shape from a set of fragmented edges produced by edge detectors. In contrast to a complete object contour the individual edge carries only partial information about the corresponding object shape which depends on the length and curvature of the edge. However, even if the full object contour is not available, the combination of edge fragments and their spatial relationship can be used to increase the probability of correct object classification which is the fundamental idea behind BFM [51]. The spatial relationship between edge fragments is obtained through a positioning of a centroid point associated with each edge fragment. An object can be described as a set of edge fragments with centroids which are close to each other as shown in Figure 3.8.

The learning of boundary fragments requires that objects in the training images are delineated by bounding boxes and that object centroids are specified in a set of validation images. Note that the manually provided centroid position must be consistent across similar objects. The candidate boundary fragments are extracted from the training images and then optimized using the validation images. The informativeness of a candidate boundary fragment corresponding to a given object depends on its uniqueness (the similarity to related fragments in the same object class versus other object classes) and the overall accuracy of centroid positioning across instances of the corresponding object.

The candidate boundary fragments are obtained from randomly distributed seeds on the boundary of objects in the training images and then optimized using validation images which results in a BFM codebook containing many boundary fragments. The optimization process estimates the length of each boundary fragment γ_i that minimizes the matching

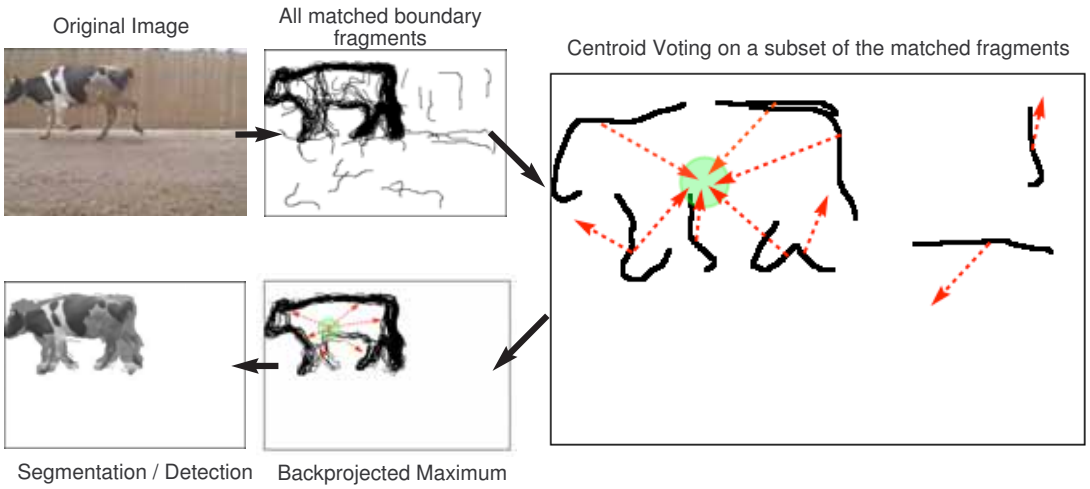


Figure 3.8: Several stages of object detection using BFM. The rightmost picture shows a set of edge fragments that have their centroids positioned close to each other. The diagram has been obtained from [51].

cost $C(\gamma_i)$ between this fragment and the most similar edge fragment (in terms of the edge matching distance) in the validation images. The matching cost $C(\gamma_i)$ is a combination of the cost related to similarity between γ_i and edges in validation images using the Chamfer distance [3] and the distance (in pixels) between the object centroid (provided as part of validation data set) and the candidate boundary fragment centroid $c_{loc}(\gamma_i)$:

$$C(\gamma_i) = c_{match}(\gamma_i) c_{loc}(\gamma_i) \tag{3.14}$$

$$c_{match}(\gamma_i) = \frac{\sum_{j=1}^{L+} \text{distance}\left(\gamma_i, P_{v_j}\right) / L^+}{\sum_{j=1}^{L-} \text{distance}\left(\gamma_i, N_{v_j}\right) / L^-} \tag{3.15}$$

where the L^+ and L^- are the numbers of positive and negative validation examples respectively in positive validation images P_{v_j} and negative validation images N_{v_j}. The distance $\left(\gamma_i, I_{v_j}\right)$ is the similarity measure between the boundary fragment candidate and the best matching edge in I_{v_j}:

$$\text{distance}\left(\gamma_i, I_{v_j}\right) = \frac{1}{|\gamma_i|} \min_{\gamma_i \in I_{v_j}} \left(\sum_{t \in \gamma_i} DT_{I_{v_j}}(t) \right) \tag{3.16}$$

where $DT_{I_{v_j}}$ is the distance transform.

The authors use 8 orientation planes to obtain invariance to rotation and multi-scale representation of the BFM codebook to obtain scale invariance. Since cluttered or highly textured image areas are the source of spurious edge matches, the method extracts the 10 (value experimentally adjusted by the authors) best matches (with lowest distance $\left(\gamma_i, P_{v_j}\right)$) and chooses the one with the best centroid prediction.

The candidate boundary fragment optimization starts with small (20 pixel long experimentally adjusted by the authors) edge fragments extracted from the random seeds in the training images which are grown in steps of 30 pixels up to 520 pixels. At each step the cost $C(\gamma_i)$ is computed and the minimum chosen for a boundary fragment stored in the BFM codebook. The selected boundary fragments are further clustered to reduce redundancy of the codebook entries.

The BFM codebook is used to build weak detectors composed of k (typically 2 or 3) optimized boundary fragments. The weak detectors are chosen such that their k boundary fragments exhibit higher similarity to the edges in the corresponding objects than to the edges in other object classes, and that their centroid estimates concur and agree with the object centroid provided in validation images. The distance $D(h_i, I)$ of a weak detector h_i applied to image I is a sum of k boundary fragment distances matched to the edges in an image:

$$D(h_i, I) = \frac{1}{m_s^2} \sum_{j=1}^{k} \text{distance}\left(\gamma_j, I\right) \tag{3.17}$$

where $m_s = k$ if the boundary fragment centroids are closer to the object centroid than a predefined threshold d_c (typically 15 pixels) and otherwise $m_s = 0.5k$.

The weak detector h_i fires in the image I if its distance is lower than the learned threshold th_{h_i}:

$$h_i(I) = \begin{cases} 1 & D(h_i, I) < th_{h_i} \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

The weak detectors are then used to build a strong object detector based on a weighted sum of weak detectors:

$$H(I) = \text{sign}\left(\sum_{i=1}^{T} h_i(I) w_{h_i}\right) \tag{3.19}$$

where T is a number of weak detectors.

The process of object detection can be described in the following steps:

1. Detect edges in the unseen image.

2. Apply weak detectors - match boundary parts with edges in the image.

3. Each weak detector h_i votes with the corresponding weight w_{h_i} and the votes are accumulated within a circular window of radius d_c around candidate centroid points represented by Mean-Shift modes [8].

4. The Mean-Shift modes that are above the threshold t_{det} correspond to object instances. A confidence of the detected object is estimated using Bayesian probability as in [51].

3.2.5 Object Detection with Deformable Shape Models

Ferrari et al. [13] introduced an approach that allows to learn shape models from configurations of pairs of approximately straight edge fragments that repeatedly appear in the training images. The learning phase selects object contours that appear across same object instances and discards edges related to the background or random appearances e.g. labels on bottles and cups. The object contour models are learned from training images where each object of interest is contained by the bounding box. However in the detection phase the object contours are estimated from the learned model, instead of detecting only a bounding box (as is the case in methods presented in Sections 3.2.2 and 3.2.3).

The shape is represented by the combination of *pairs of adjacent, approximately straight segments* (PAS) shown in Figure 3.9. The Figure shows examples of PAS features appearing in images containing mugs as well as examples of co-occurring PAS pairs. Each PAS feature encodes the central position (mean over the two segment centers), a scale (distance between the segment centers), a strength (mean edge detector confidence) as well as segment orientations, segment lengths normalized by the scale and their relative position, which together provide a scale and translation invariant description of the object contour parts. The PAS features are extracted from inside the bounding boxes in training images.

The PAS features are used to build a code-book related to repeatable object parts. The code-book is constructed by clustering PAS features extracted from the training images, which reduces their number and therefore increases matching efficiency.

The learning of an object contour model is based on an assumption that the PAS part belonging to the model will appear across training instances of the related object

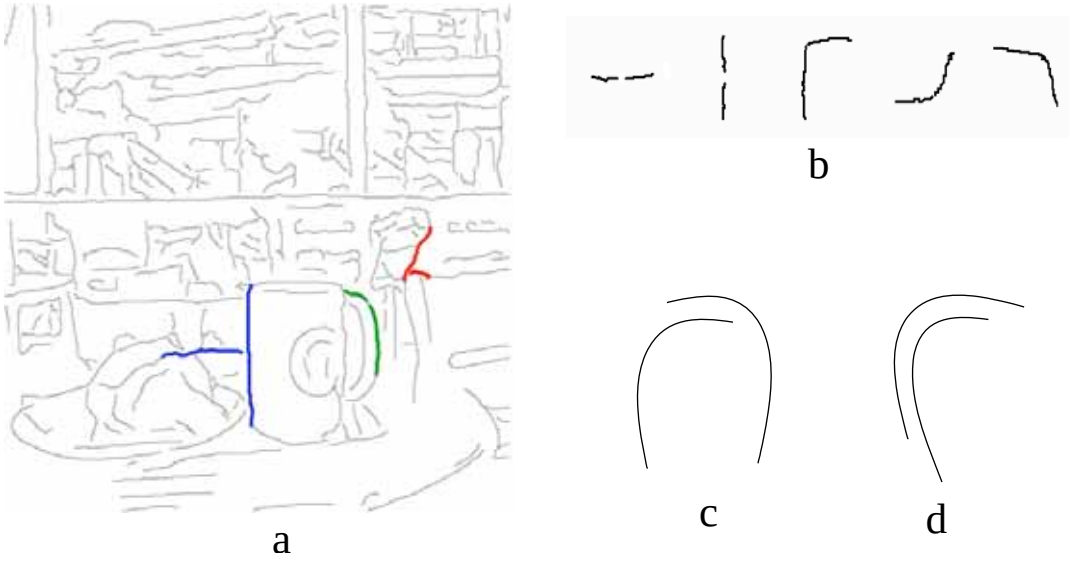


Figure 3.9: a) Examples of PAS features. b) Some frequently occurring PAS. c) and d) two model parts with connectedness. The picture has been obtained from [13].

at repeatable locations and scales. This process performs a selection of PAS parts that constitute the final object model parts using a confidence measure based on the accuracy of part localization and scale across object instances (explained later). Note that both the part location and scale are related to the bounding box. The bounding boxes are transformed into zero-centered rectangles with unit height and width equal to the geometric mean of rectangle aspect ratios (width over height) over the object training examples. This operation produces reference frames for object instances that minimize scale differences, translation and cancels out shape variations due to different aspect-ratios.

The object contour model is represented by a set of PAS parts from the code-book. PAS features extracted from the training examples are used to vote for each of the code-book parts with shape, location and size like its own. A single PAS feature votes for each of the code-book PAS parts within a dissimilarity threshold, however votes are proportionally weighted by the edge strength and inversely proportionally weighted by the dissimilarity between the PAS feature and the code-book part. The votes are accumulated for each code-book part in a location and scale accumulator space. The local location-scale maxima yield a model part that has a specific location and scale relative to the bounding box as well as a specific shape corresponding to the code-book part. The value of the local maximum

is a measure of model part strength (confidence). This procedure results in a set of model parts, which produces the initial approximation of the object shape. However, since there is no information about the connectivity of the model parts, fragments of the object contour can be represented by multiple parts or short discontinuous lines. In [13] examples are provided of the initial model part detection and a technique that finds the connectivity between initial parts through the analysis of training PAS segments that correspond to two different parts. The shape model is further refined by matching it back to the training images, estimating the Thin-Plate Spline (TPS) [7] non-rigid transformation between the model and each training example and finding the mean shape from all the TPS transformed models. Note that TPS provides point to point correspondences between shape model segments and the segments extracted from the training images which also allow to learn the intra-class shape variations.

The object detection is achieved through voting of PAS features from the test image using their shape properties. Each match between a PAS feature from the test image and the model part induces a translation and scale transformation which is translated into a vote for the presence of an object instance at a particular image location (object center) and scale. All votes are weighted accordingly to the shape similarity between matched PAS feature and model part. This procedure produces an estimate of the location and scale of object instances represented by the local maxima in voting space. The estimate of object location and scale allows to project a shape model into the test image and use the Thin-Plate Spline Robust Point Matching algorithm (TPS-RPM) [7] to find point to point correspondences and the non-rigid transformation between the object model and its instance in the test image. Note that in this case points sampled along edges (test image) and PAS segments (shape model) are used by TPS-RPM algorithm. The TPS-RPM allows for non-rigid transformation which may result in significant model shape deformations. The matching is therefore constrained by the shape deformation model learned from the training examples.

The authors evaluated their method using six object classes (such as bottles, giraffes, mugs, swans, Apple Inc. logo) found in two datasets: *ETHZ Shape Classes*[¶] and *INRIA Horses*[‖]. The detection performance is reported at 0.4 false-positives per image, exhibiting object detection of over 80%, except for Giraffes over 70%.

[¶]http://www.robots.ox.ac.uk/ ferrari/datasets.html
[‖]http://ralyx.inria.fr/2006/Raweb/lear/uid34.html

3.2.6 Category Recognition from Pairwise Interactions of Simple Features

Leordeanu at al. [33] proposed an idea of using pairwise geometric relationships between boundary fragments as the basic building blocks of the object shape model. The shape model is represented as a graph of interconnected object boundary parts, where graph edges model pairwise geometrical relationships between the boundary parts. Each graph node represents an object boundary part which can be connected to another boundary part if both parts co-occurred in at least one training image.

Each object boundary part can be seen as a point and its associated normal. Figure 3.10 shows the pairwise relationship between two object boundary parts i and j that is represented by the distance between these points d_{ij} and their relative orientations α_{ij} and σ_{ij} which form a 7-dimensional feature vector $\boldsymbol{e}_{ij} = \{\theta_i, \theta_j, \sigma_{ij}, \sigma_{ji}, \alpha_{ij}, \beta_{ij}, d_{ij}\}$, where β_{ij} is the angle between the two normals.

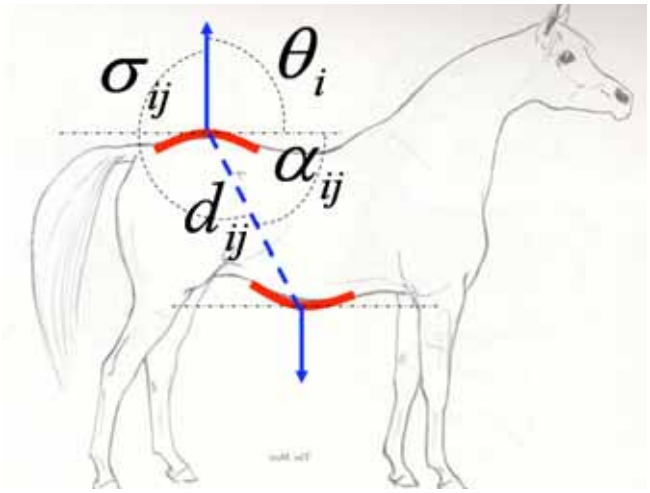


Figure 3.10: The illustration of parameters that define pairwise geometric relation between two edge fragments. Image obtained from [33].

The object localization is defined as finding the features in the image that best match each model part. The matching score E_x combines the matching of individual model parts with their pairwise relationships and is written as:

$$E_x = \sum_{ia;jb} x_{ia} x_{jb} G_{ia;jb} \tag{3.20}$$

where i and j represent model parts, a and b image features matched to model parts, $x_{ia} = 1$ if model part i is matched to image feature a (0 otherwise) and $G_{ia;jb}$ is a *pairwise potential* that reflects the accuracy with which the parts i and j preserve their geometric relationship, defined as:

$$G_{ia;jb} = \frac{1}{1 + \exp\left(-\boldsymbol{w}^T \boldsymbol{g}_{ij}(a,b)\right)} \tag{3.21}$$

where $\boldsymbol{w}$ is a learned vector of sensitivity to deformations (see [33]), $\boldsymbol{g}_{ij}(a,b) = [1, \epsilon_1^2, ..., \epsilon_7^2]$ is a vector describing the geometric deformations between the parts (i,j) and their matched features (a,b) and $\boldsymbol{\epsilon} = \boldsymbol{e}_{ij} - \boldsymbol{e}_{ab}$.

The object localization is achieved through finding the assignment vector x^* that maximizes the matching score E:

$$x^* = \text{argmax}\left(\boldsymbol{x}^T \boldsymbol{G} \boldsymbol{x}\right) \tag{3.22}$$

The vector $\boldsymbol{x}$ represents a mapping between model parts and image features constrained to one-to-one correspondences (one model part can match only to one image feature and vice versa). This combinatorial optimization is known as the quadratic assignment problem (QAP), which is approximated using an efficient spectral matching algorithm [32].

The final recognition of a potential object localized in the image is achieved through an approximation of the posterior $P(C|\boldsymbol{x}^*, D)$, where $C = \{0,1\}$ describes an occurrence of a particular object and D represents the data. The posterior is approximated with the following logistic classifier:

$$S\left(\boldsymbol{G}_0, \boldsymbol{r}\right) = \frac{1}{1 + \exp\left(-q_0 - q_1 \sigma(\boldsymbol{r})^T \boldsymbol{G}_0 \sigma(\boldsymbol{r})\right)} \tag{3.23}$$

where $\boldsymbol{r}$ is a vector of relevance parameters where high values correspond to individual parts that are discriminative against the background, $q_{\{0,1\}}$ are function parameters that are learned and $\sigma(r_i) = 1/(1 + e^{-r_i})$ (sigmoid function), where r_i is the i-th element of $\boldsymbol{r}$.

The object models are obtained from the training images using weakly supervised learning. The only information provided with each of the training images is the object type present in the image – no bounding boxes or manual segmentation is needed. The training procedure learns a set of model parts for a given object and the model parameters: pairwise geometric relationships $\boldsymbol{e}_{ij}$, the relevance parameters $\boldsymbol{r}$, q_0 and q_1 and the sensitivity to deformations $\boldsymbol{w}$. The initial set of model parts and geometric relationships $\boldsymbol{e}_{ij}$ is taken from the first training image, which includes both object and background related features.

The relevance parameters $\boldsymbol{r}$ are set to 0. The model parameters are updated through sequential analysis of training images and minimizing the objective function J that is a familiar sum of error squares:

$$J = \sum_{n=1}^{N} b_n \left(S(\boldsymbol{G}_0^{(n)}, \boldsymbol{r}) - t^{(n)} \right)^2 \tag{3.24}$$

where $t^{(n)} = 1$ if the n-th image contains the learned object and $t^{(n)} = 0$ otherwise. The weights b_n are set to N/P if $t^{(n)} = 1$ and 1 otherwise (N and P are the number of negative and positive images respectively).

The parameters are updated using a sequential gradient descent, iterating over all images for a fixed number of times. The update of any given model parameter λ has the general form of:

$$\lambda \leftarrow \lambda - \rho b_n \left(S(\boldsymbol{G}_0^{(n)}, \boldsymbol{r}) - t^{(n)} \right) \frac{\partial S(\boldsymbol{G}_0^{(n)}, \boldsymbol{r})}{\partial \lambda} \tag{3.25}$$

where ρ denotes the learning rate.

The less relevant model parts, for which the $\sigma(r_i) \approx 0$, are discarded in the course of the iteration process while new previously "unseen" parts are added to the model from the sequence of training images.

The sensitivity to deformation parameters $\boldsymbol{w}$ is learned from a set of manually selected correct and incorrect correspondences, as described in [33].

The evaluation of object recognition accuracy is performed using a Pascal challenge training dataset** (587 images) and compared with another method that uses the local appearance (local texture information) for object recognition [33]. The texture based classifier is outperformed in this comparison by 10% on average, when the training features are extracted from bounding boxes surrounding objects and by 5% when the bounding boxes are not used for training of the presented method (they are always used for the texture based classifier). Overall object recognition accuracy reaches over 80%, though no error rates are specified. The authors conclude that the shape is a stronger cue than local appearance for the analyzed class of objects.

The authors provide another performance comparison with the method of Opelt et al. (see Section 3.2.4) where their method performs better by approximately 5% in the majority of object categories.

The approach presented in this section is similar to methods presented in Sections 3.2.4

**http://www.pascal-network.org/challenges/VOC/voc2005/

and 3.2.5 as all of them represent and learn the object model as a spatial configuration of edge fragments. However, other methods do not capture pairwise geometrical relationships between edge fragments which increase the discriminative power of individual parts and allow for learning object models without specifying the object location in the training images through bounding boxes, as is required by other methods.

3.2.7 Discussion

The shape-based object recognition methods in this chapter are presented in approximately chronological order. We can see that older approaches such as Geometric Hashing (Section 3.2.1) do not have capabilities to learn object models directly from the training images. In this particular case the object model is constructed from complete object boundaries e.g. CAD generated wireframe or manual segmentation. This gap is filled by methods presented in Sections 3.2.4, 3.2.5 and 3.2.6, which are designed to learn object models from partial shape features extracted directly from the training images and to separate object related features from the background features. These methods share an idea of representing objects as a spatial configuration of edge fragments. The approach described in Section 3.2.6 extends this idea further and uses spatial configuration of edge fragment pairs, which increases the informativeness of individual parts used. The common difficulty associated with these methods is achieving invariance to rotation, scale and possibly affine or projective transformations. To do this these methods require invariant features however individual edge fragments do not carry information about object scale or orientation unless an image is specifically annotated (as in Section 3.2.5). The technique for learning the object model described in Section 3.2.6 is not invariant to any transformations and other methods also do not implement full rotation and scale invariance simultaneously.

Our solution to the problem of invariant learning of object models is presented in Chapter 4 and 5. Instead extracting invariant features, it is better to apply invariant matching. We find sub-sets of features, such as edges, in the two compared images, which preserve similarity and spatial relationships under rotation and scaling. The difference is that we use an assembly of edge parts from the start and find relative rotation and scale between sets of edge parts in different images. Matching multiple images can extract and refine frequently occurring image structures, even when rotation and scale of individual instances of these structures vary. Another advantage of invariant matching is the ability to extract multiple repeating structures appearing simultaneously in the training images.

Chapter 4

Local and Semi-local Shape Detectors and Matching

4.1 Introduction

This chapter presents two novel shape based image descriptors introduced by the thesis author in [63] and [66]. The first we classify as a local descriptor and the second as a semi-local descriptor – the primary difference between local and semi-local descriptors is that the later typically operate on larger scale structures which are not bounded by an interest region as it is the case in local descriptors.

The local image descriptors (see Chapter 3) are useful for detecting image regions that are part of a texture or descriptive details of an object. Section 3.1.3 discusses an example of a shape based local image descriptor called shape context. Here we present a novel method, the *Orientation-invariant Radial Configuration* (ORC), that combines a complex region detector and an image descriptor to capture the shape of homogenous color regions around interest points (see Section 4.2). The method differs from other local descriptors in two aspects. First it explicitly describes the shape of the region and allows to encode more than one concentric region of homogenous color. Second the scale and orientation of the region is not estimated a priori, instead the relative scale and orientation that minimizes the descriptor distance is computed during the descriptor matching. This approach is compared with a state of the art SIFT descriptor which uses invariant features (region orientation and scale are estimated) in Section 4.2.9.

The semi-local image descriptor *Radial Edge Configuration* (REC) presented in Section 4.3 allows to capture arbitrarily large image structures represented as spatial con-

figurations of edge fragments. The descriptors are extracted around the interest points and the edge fragments are represented as a set of ordered boundary points encoded in polar coordinates. The use of polar coordinates allows for the construction of a descriptor distance that is rotation and scale invariant and robust to edge fragmentation which is the primary difference to other shape detectors discussed in Chapter 3. However, due to angular coordinate quantization, the accuracy of edge fragment description and matching is inversely proportional to the distance from the interest point (center of the polar coordinate system), thus we categorize it as a semi-local image descriptor. Section 4.3.2 presents examples of the descriptor matching and Chapter 5 shows the descriptor applicability to object recognition in medical images.

4.2 Detection of Local Image Structures with Orientation-invariant Radial Configuration

Interest point and region detectors are fundamental parts of appearance based computer vision approaches as discussed in Chapter 2. The consistency of scale selection, whether it describes isotropic blobs or anisotropic elliptical regions directly influences the performance of local descriptors in applications such as object recognition or matching different views of the same scene. Here, we present a novel region detector and descriptor *Orientation-invariant Radial Configuration* (ORC) which extracts shape properties of local image patches and their boundaries at the same time (see Figure 4.1).

The primary difference between methods discussed in Chapter 2 and ORC is that ORC is capable of describing regions containing arbitrarily convex boundaries, while approaches such as blob and corner detectors discussed in Chapter 2 describe region shape as a circle or oriented ellipse. In this respect our approach is related to MSER (see Section 2.3), however the final results and methodology are different (note that in scene matching and object recognition MSER is also approximated by ellipse [44]). Furthermore the method is capable of detecting multiple concentric boundaries of multiple concentric homogenous color regions if present around the interest point (the exact number depends on the analyzed image patch and chosen parameters). The advantage of such an approach is the possibility to use the shape information as a feature for local image region discrimination. We show in Section 4.2.9 that the detected regions and shape features outperform the state of the art SIFT descriptor in the task of matching local image structures in an animal image database.

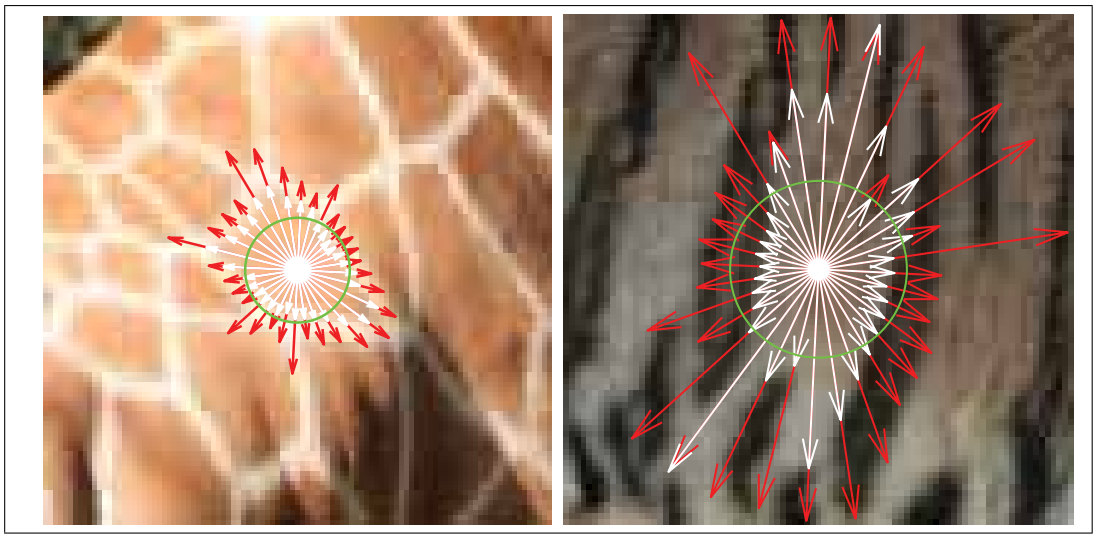


Figure 4.1: Example of boundary point detection: giraffe skin on the left and tiger skin on the right. The inner arrows (white) represent a geometrical description of the local structure interior (q_i^1), outer arrows (red) correspond to the local structure exterior (q_i^2).

4.2.1 Related Work

The ORC is related to *Shape Context* proposed by Belongie and Malik (see Section 3.1.3) which captures shape information from local image patches. Their approach however uses edge detection to extract local object boundaries, while ORC is intended as a detector of homogenous image patches which is also the goal of the MSER method described in Section 2.3. Furthermore shape context uses a log-polar histogram for shape description while ORC uses an explicit shape description in the form of sampled points encoded in polar coordinates.

Our detector is also related to the *Intensity-Based Method* of Tuytelaars and van Gool [70], which defines interest regions by detecting luminance transitions on rays emanating from local intensity extrema. Our approach differs in several ways. The main difference is that instead of fitting an ellipse to a detected region of irregular shape, the ORC descriptor encodes the shape. Furthermore, instead of detecting interest points at local intensity extrema, we use local symmetry extrema — the interest points then tend to appear in the center of salient image structures in the image. We also introduce an approach for detecting multiple pixel value transitions on the rays based on clustering. Finally, the proposed descriptor is able to encode multiple concentric structures in a single descriptor.

4.2.2 Interest Points

Since ORC is designed to capture the shape of basic local structures, which correspond to patches of homogenous color pixels, we locate interest points in the centers of round/isotropic structures or along the symmetry axis of elongated shapes. Such positioning not only helps to detect local shapes but also increases descriptor matching precision as will be shown in Section 4.2.8. For these reasons the detection of interest points utilized by the ORC descriptor is based on the Radial Symmetry Transform described in Section 2.5.

4.2.3 Orientation-invariant Radial Configuration

The ORC descriptor produces a description of the local structure's boundary in the form of distances between the central interest point and N boundary points distributed at equal angular intervals of $2\pi/N$ radians around the interest point. Figure 4.1 shows two examples of the boundary detection on the fragments of images containing a tiger and a giraffe. This approach allows the full description of convex shapes, both round and elongated.

Descriptors are extracted from constant scale r_{max} (adjustable parameter), circular regions surrounding the interest points. The scale r_{max} corresponds to the maximum extent of the local structure whose boundaries can be completely captured. The initial size of the region r_{max} is estimated from the interest point adjacency. The circular region is divided into N equal sectors and luminance profiles are extracted along the radius of each sector. The method can be extended to use other features such as color information. The profile in each sector is then clustered into coherent and spatially contiguous regions while boundary points are associated with the region boundaries. Figure 4.2 is an illustration of boundary point detection that corresponds to the boundaries of homogenous intensity regions. The boundary point configurations are then constructed of N points each, one from each sector.

4.2.4 Boundary Point Detection

Boundary points correspond to the edges or transitions between homogenous intensity regions of pixels along the sector radius. The operation of boundary point detection is repeated for each sector separately.

The pixel grouping is based on an agglomerative hierarchical clustering of pixel related features (luminance), except that only spatially adjacent clusters can be joined into a node

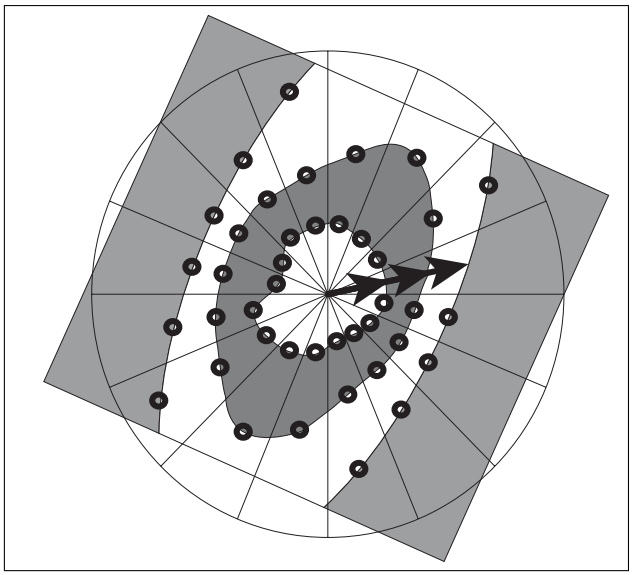


Figure 4.2: Multiple boundary points are detected along the rays. Arrows show three possible boundary point configurations corresponding to coherent image patches.

at the next clustering level. The mean feature value of cluster t at level l, containing the pixels at radial distances $r^l_{t,min} \leq r \leq r^l_{t,max}$ is denoted as C^l_t. At clustering level 1 each cluster contains exactly one pixel. The spatially adjacent clusters t and $t+1$ at level l are joined only if the following condition is fulfilled:

$$\|C^l_t - C^l_{t-1}\| > \|C^l_t - C^l_{t+1}\| < \|C^l_{t+1} - C^l_{t+2}\| \tag{4.1}$$

This way the clusters C^2_t at clustering level 2 always represent adjacent pixel pairs or single pixels (if (4.1) was not fulfilled) and clusters at higher levels represent continuous sections of pixels along the radius:

$$C^l_t = \frac{1}{r^l_{t,max} - r^l_{t,min} + 1} \sum_{r=r^l_{t,min}}^{r^l_{t,max}} I_r \tag{4.2}$$

where I_r is an image luminance value at radius r along the radial ray.

The operation of joining clusters is repeated until a complete clustering tree is built, containing two clusters at the top (see Figure 4.3)

A boundary point represents a transition between two homogenous intensity regions. In both cases we want that these regions to be represented by a single cluster each. This

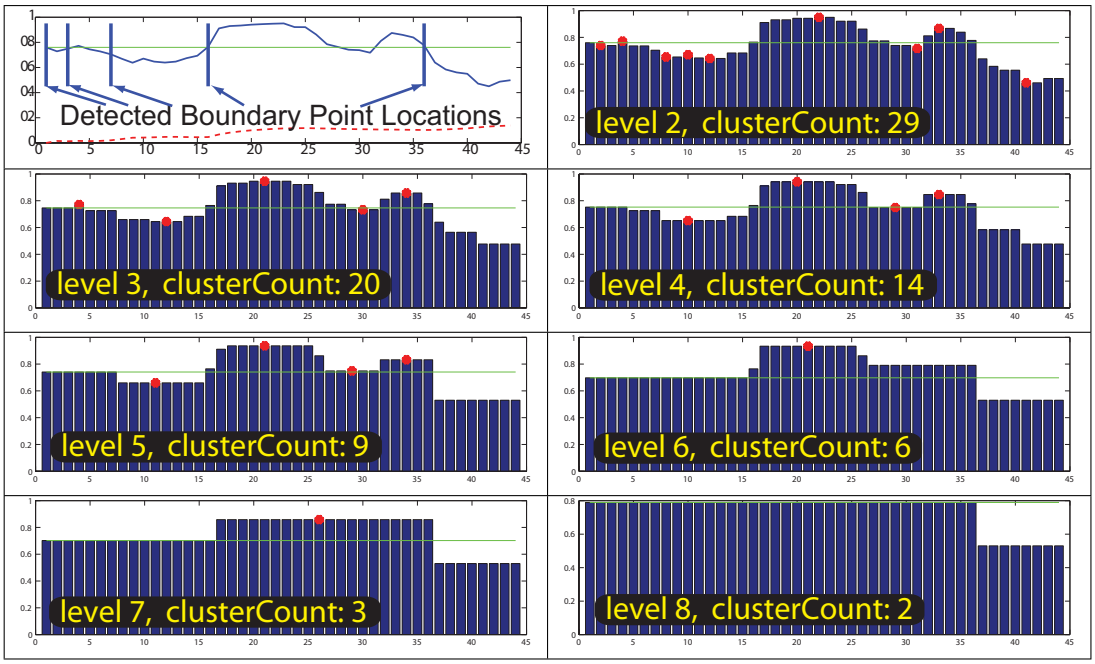


Figure 4.3: Example of pixel grouping along the radius in one of the sectors. The graph in the top-left corner represents the luminance profile along the sector radius. Other graphs represent grouping of the adjacent pixels at different clustering levels. Each bar represents one pixel, the bar heights correspond to the luminance. The red points indicate locations of extremal intensity regions along the ray occupied by the corresponding cluster.

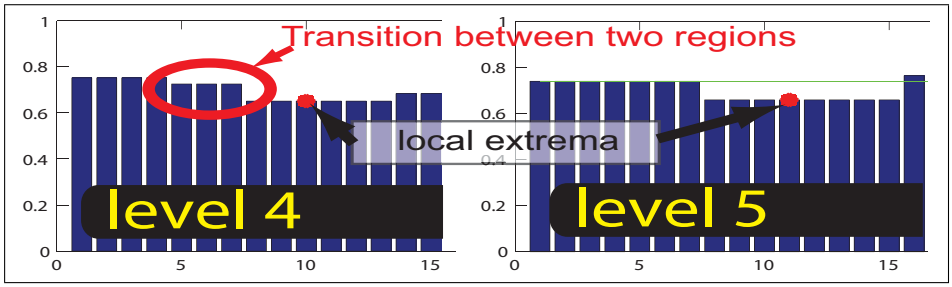


Figure 4.4: Example of the transition between different regions.

implies that the positions of the boundary points correspond to the spatial boundaries of the clusters. However, clustering starts from individual pixels and progressively reduces the number of clusters. Figure 4.3 shows an example of clustering and the reduction of clusters across 8 clustering levels. For example at clustering level 2, the clusters correspond to two pixels each while at clustering level 8 all pixels are divided between two clusters. The local minima and maxima serve as landmarks of extremal regions (as defined in

the MSER method in Section 2.3) while remaining clusters correspond to the intensity transition regions. Note that the number of extrema decrease together with the number of clusters along the clustering levels. The boundary points are extracted from extremal regions however the number of boundary points is limited to K region pairs exhibiting highest cluster transitions. The cluster transition $\Delta C^l_{s,t}$ between two clusters s and t at clustering level l is measured as an absolute difference between their intensity values:

$$\Delta C^l_{s,t} = \|C^l_s - C^l_t\| \tag{4.3}$$

The cluster transitions are extracted only between clusters related to local extrema (see Figure 4.3), i.e. a minimum followed by a maximum pair or vice-versa. Such clusters do not have to be adjacent (see Figure 4.4), but no other extremum can appear between them. This method produces up to K boundary points i.e. if the radial intensity profile contains two regions of constant intensity then only one boundary point will be detected.

The process of boundary point detection on each ray can be described by the following steps:

1. Perform hierarchical clustering of pixel feature values along the ray.
2. Find local cluster extrema at each clustering level.
3. Create a list of cluster transitions $\Delta C^l_{s,t}$ (using detected extrema) from all clustering levels.
4. Select up to K strongest cluster transitions and locate boundary points at the spatial boundary of related cluster pairs.

One of the advantages of the boundary point detection through hierarchical clustering is that each boundary point is associated with two clusters, which correspond to two regions separated by the luminance transition. The values of both clusters represent the mean luminance of these regions and are used for boundary point grouping in Section 4.2.5. Therefore each boundary point i is associated with two radii: q^1_i, which is the distance from the central interest point to the boundary of the local structure, and q^2_i, which covers also the second cluster ($q^2_i > q^1_i$) (see Figure 4.1).

The accuracy of boundary point detection depends on the orientation of the boundary versus the orientation of the ray. The highest accuracy is achieved when the ray is normal to the boundary while infinite error occurs when the ray is parallel to the boundary (no single solution exists) as shown in Figure 4.5.

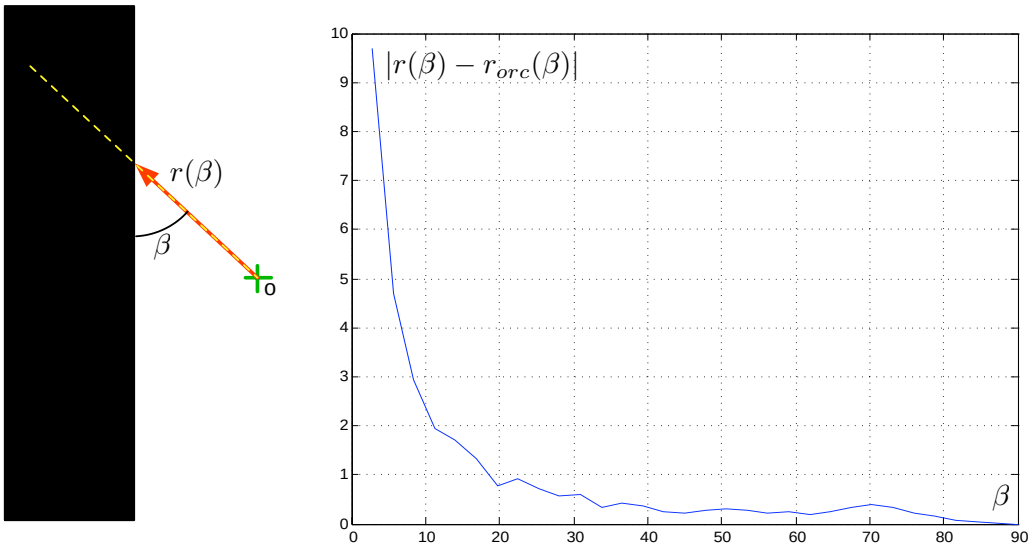


Figure 4.5: The accuracy of boundary point detection depending on the angle β between the ray and the boundary. The theoretical distance between o and the boundary $r(\beta)$ is compared with the detected value $r_{orc}(\beta)$ and the difference is shown in the graph (right). Note that as the angle β approaches 0 the difference $\|r(\beta) - r_{orc}(\beta)\| \rightarrow \infty$ which is an equivalent of looking for boundary point along the edge. The inaccuracy is a result of image quantization as well as quantization errors introduced during extraction of profiles along rays.

The boundary point detection is related to edge detection in the sense that the region boundaries associated with the intensity transitions are found. The primary difference with respect to typical edge detectors ([4] and [19]) is that we consider only a local patch of the image and therefore estimate edge strength relatively to local conditions. It is possible to detect boundary points by using one of the edge detectors, but our experiments have shown that edge detection parameters have to be adjusted individually for each image to obtain consistent results, insensitive to noise but preserving perceptible edges across the whole image. Our approach requires only two parameters r_{max} and K, which in all experiments (see evaluation in Section 4.2.9) were set to $r_{max} = 50$ and $K = 8$ and produced consistent results.

4.2.5 Boundary Point Grouping

The boundary point detection provides information about the localization of intensity transitions between homogenous intensity regions along the rays. Detection of a continuous

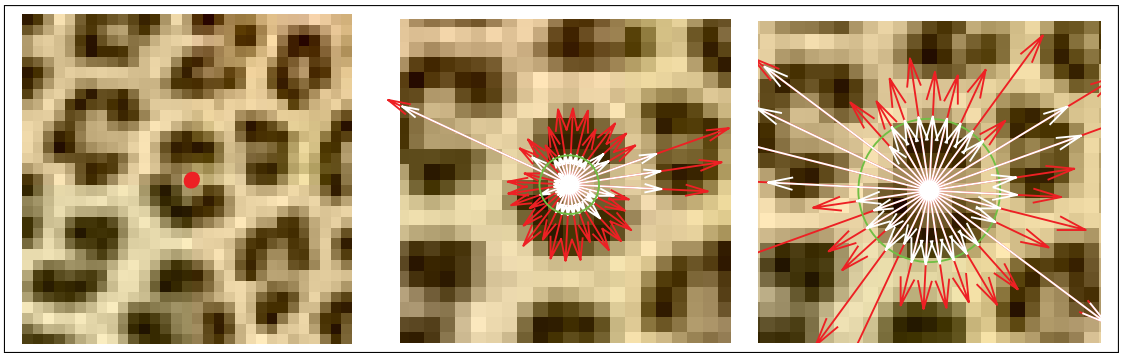


Figure 4.6: Example of multiple boundary point configurations for a patch on a leopard skin.

boundary of the homogenous 2D region requires grouping of boundary points between adjacent rays. Figure 4.2 shows an example of multiple local structures present around a single interest point. It is possible to detect them by grouping boundary points according to their similarity. Their correct detection depends on the choice of a similarity measure such as color distance.

We propose a similarity measure that is a combination of boundary luminance transition and inner luminance spread i.e. luminance standard deviation along the radius between interest point and boundary point. The boundary luminance transition is the cluster transition (Equation 4.3) associated with the boundary point. The inner luminance spread helps to capture multiple structures (see Figure 4.6) or when weak repetitive patterns/textures are present between interest point and boundary points. Both values, which constitute the *boundary point features*, are normalized by the average of luminance transitions and spreads of all boundary points.

The process of boundary point configuration detection can be summarized as follows:

1. Extract features for all boundary points in N sectors.
2. Perform hierarchical clustering of boundary point features in adjacent sectors (see Figure 4.7).
    - A single cluster cannot contain multiple boundary points from the same sector. This implies that the maximum number of boundary points within any cluster cannot be higher than the number of sectors.
    - The clusters cannot contain more than N boundary points, which is equivalent of a closed curve.

3. Extract all clusters B_k containing at least N_{min} boundary points. All other clusters are rejected.

    - The parameter N_{min} is typically set to $0.25N$ in order to discard small structures, which are usually less distinctive.

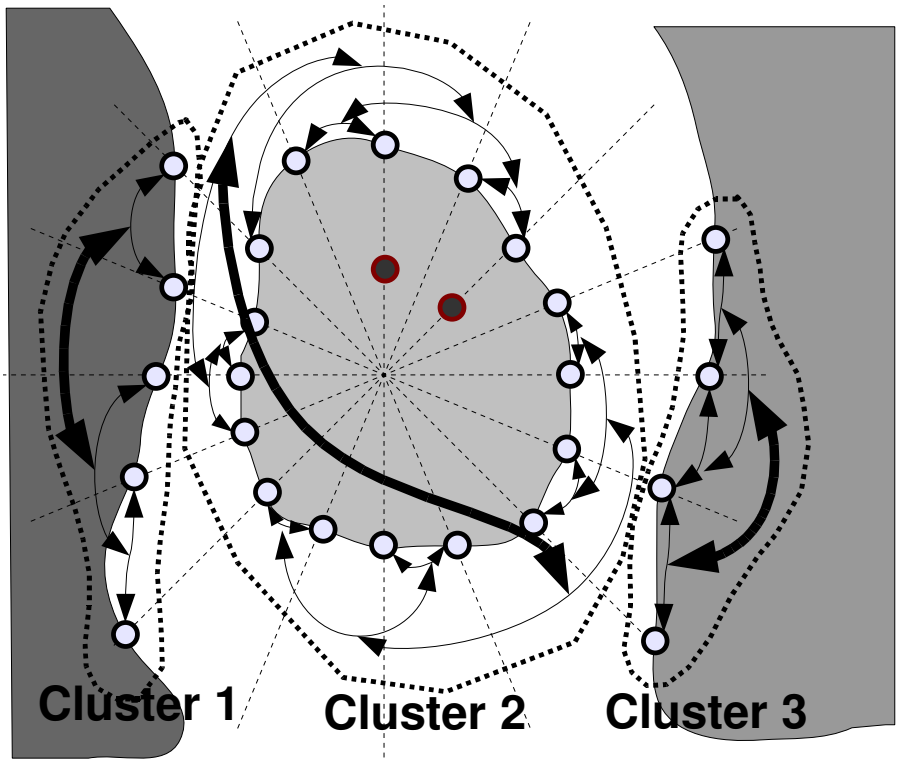


Figure 4.7: Example of boundary point grouping. Thin arrows show grouping of boundary points and clusters in adjacent sectors. Thick arrows show the top level clusters of the clustering tree (no more cluster merging possible).

Figure 4.6 shows an example of of two boundary point configurations extracted around the same interest point. The number of boundary point configurations in general depends on the analyzed image patch and the parameters r_{max} and K. Each configuration is separately used during descriptor matching.

The performance of this strategy is experimentally verified in Section 4.2.7.

4.2.6 Refining Scales in Low Gradient Sectors

The ORC descriptor is extracted from a finite circular image region. The size of the region is regulated by the parameter r_{max} which means that only boundaries at a relative

distance (from the interest point) smaller than r_{max} can be captured. For example it is possible that part of an elongated structure boundary exceeds the r_{max} range which makes estimation of boundary points sensitive to noise i.e. they are associated with weak edges or transitions and not with the real structure boundary. However, the luminance spread ν_i^k calculated over the range of radii $0..q_i^2$ (covers structure interior and exterior) is lower for "weak" boundary points than for the points associated with the real structure boundary. This measure can be used to both detect and refine "weak" boundary points. The i-th boundary point, belonging to the k-th configuration is classified as a weak one if:

$$\nu_i^k < 0.5\overline{\nu_{j\in B_k}^k}. \tag{4.4}$$

where the term $\overline{\nu_{j\in B_k}^k}$ corresponds to the luminance spread average over all boundary points within the k-th configuration. The consistent detection of "weak" boundary points using condition (4.4) is possible if the majority of boundary points correspond to real structure boundary, as in the example in Figure 4.1. The robustness of this condition is verified experimentally in 4.2.7.

The "weak" boundary points are removed if other boundary points exist in the corresponding sector, or otherwise q_i^1 of the point is increased until condition (4.4) is met or r_{max} is reached.

4.2.7 Evaluation of Boundary Point Detection

This section presents an evaluation of the region boundary detection in the presence of noise, blur and asymmetrical intensity distribution which may occur in real images. The first two scenarios operate on a test image in Figure 4.8 that contains two concentric circular structures of homogenous intensity. The inner circle of radius $r_1 = 30$ pixels is filled with intensity h that is varied during the tests. The outer circle of radius $r_2 = 40$ is filled with intensity 1 (image intensity is normalized to the range 0..1) and the background is filled with the intensity 0. The boundary detection carried on this image produces two boundary point configurations – one with $q_i^1 = r_1$ and other with $q_i^2 = r_2$ $(i = 1..N)$ for any value of $0 < h < 1$. In this case the result is independent of intensity h as long as h differs from the background and outer circle intensities.

The first test scenario measures the accuracy of boundary detection depending on the amount of blur applied to the test image. The test image is convolved with a Gaussian at a set of scales $\sigma = \{1, 2, ..., 20\}$ and the boundary point configurations are extracted at each scale. The overage difference between the estimated boundary points $q_i^{1,2}$ and theoretical

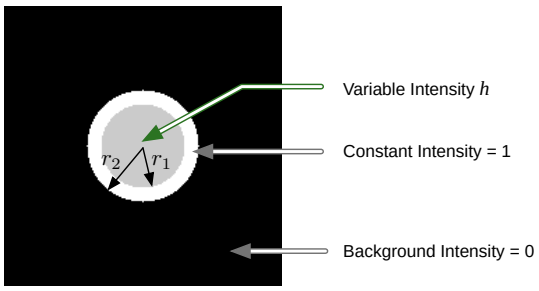


Figure 4.8: The test image used for evaluation of region boundary detection.

boundaries $r_{1,2} - \epsilon_c = \overline{|q_i^c - r_c|}$ is shown in Figures 4.9 and 4.10. Note that the maximum smoothing scale at which both boundaries are detected is with $\sigma = 15$ which is more than the difference between r_2 and r_1. Above $\sigma = 15$ only a single boundary is detected as both regions become indistinguishable due to smoothing. Due to smoothing the q_i^1 gradually decreases with the increase of σ. The fluctuations of ϵ_c curves are caused by instability of boundary detection in the presence of smoothed boundaries combined with quantization errors occurring during the sampling of radial profiles.

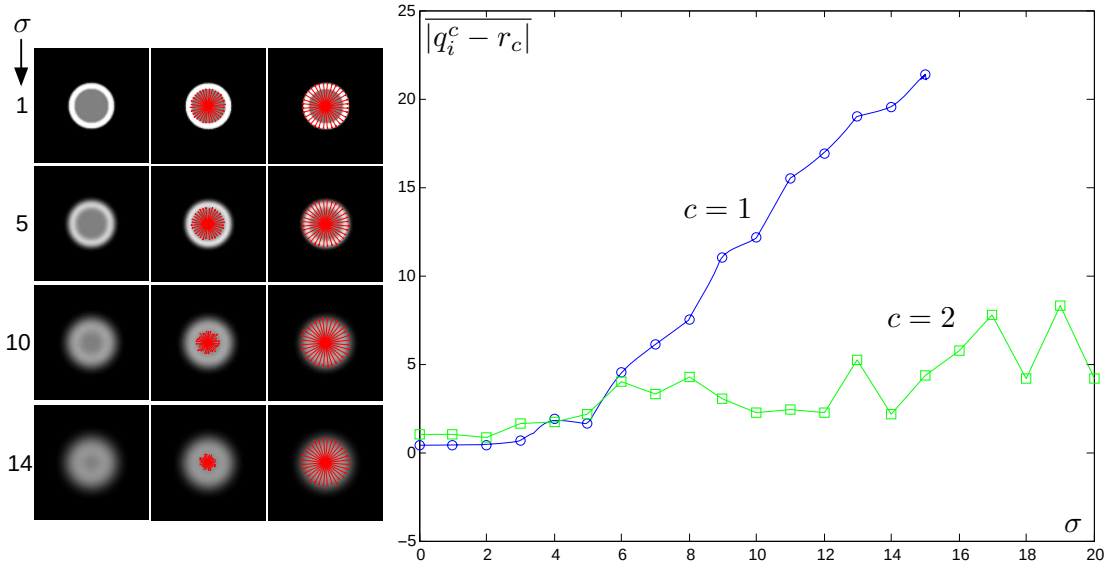


Figure 4.9: The result of boundary detection on the test image with $h = 0.5$ and Gaussian smoothing of scale σ applied. For each σ shown, the ORC produces two radial configurations corresponding to the inner and outer circular structures. The inner circular structure is detected for $\sigma \leqslant 15$.

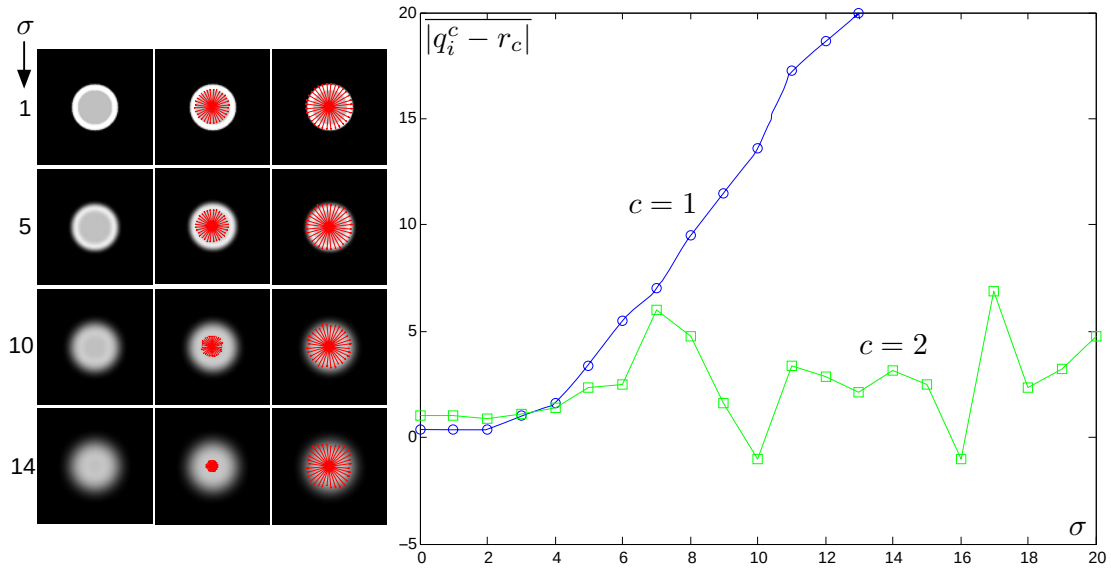


Figure 4.10: The result of boundary detection on the test image with $h = 0.75$ and Gaussian smoothing of scale σ applied. For each σ shown, the ORC produces two radial configurations corresponding to the inner and outer circular structures. The inner circular structure is detected for $\sigma \leqslant 13$.

The second test scenario measures the accuracy of boundary detection depending on the amount of white Gaussian noise applied to the test image. The noise is added to the test image at the set of amplitudes $\eta = \{0.05, 0.1, 0.15, ..., 1\}$ and the boundary point configurations are extracted at each level. The dependence of ϵ_c on the noise amplitude η is shown in Figures 4.11 and 4.12. Note that the boundary of the first region (r_1) is correctly estimated until the noise amplitude η approaches the difference between intensities in both regions $1 - h$. The addition of noise also causes that more than two boundary point configurations are detected. The fluctuations of curves in Figures 4.11 and 4.12 are caused by the random nature of the noise.

The third test scenario measures the accuracy of boundary detection of the region with non-homogenous intensity distribution which varies between 1 and I_{min} as shown in Figure 4.13. The test procedure measures the number of incorrectly detected boundary points depending on the I_{min}. The result shows that all boundary points are correctly detected if $I_{min} \geqslant 0.15$ $(1 - I_{min} = 0.85)$.

The detection of interest regions influences the performance of scene matching and object recognition which was shown in [42, 74]. The available comparisons provide evalu-

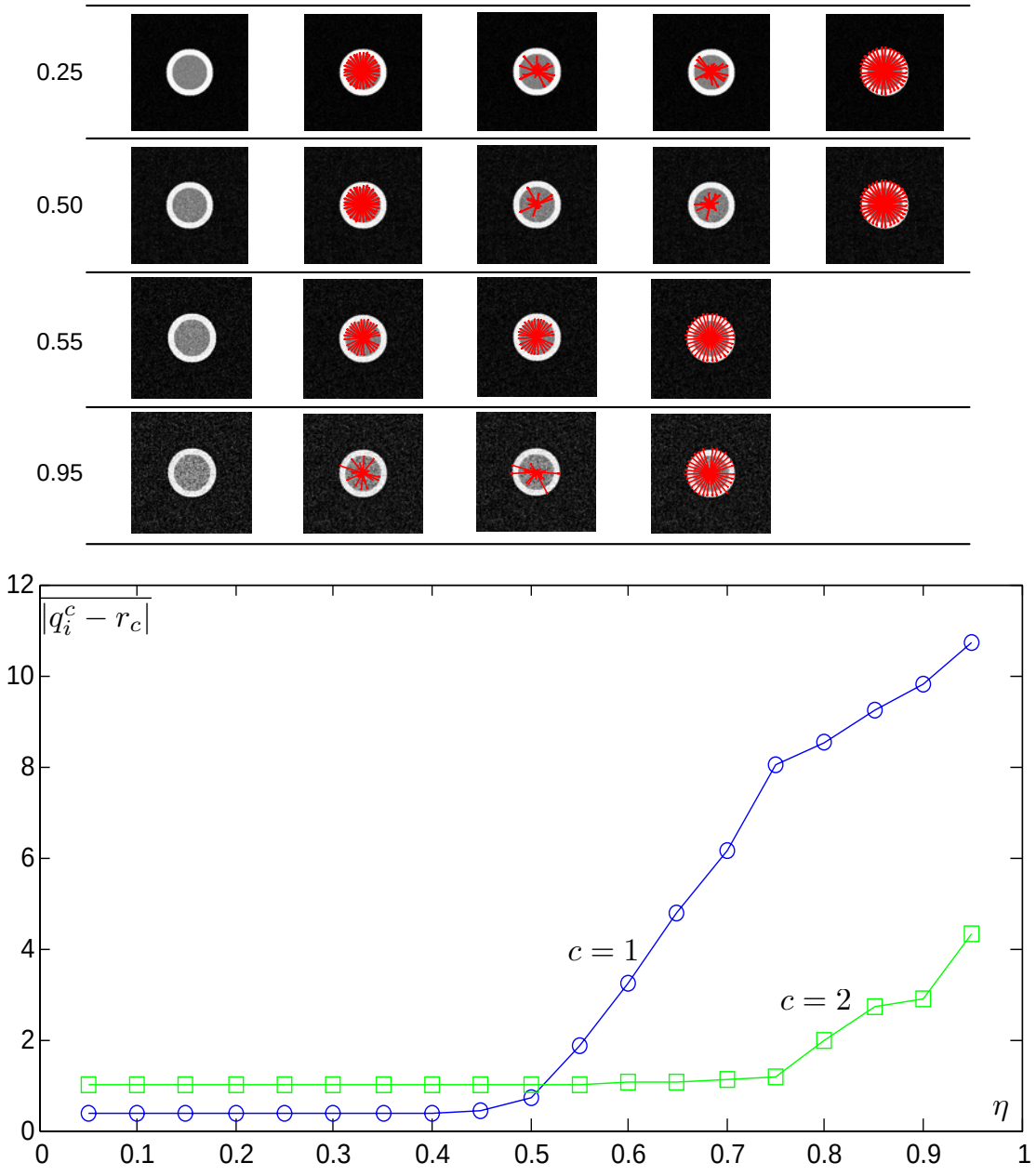


Figure 4.11: The result of boundary detection on the test image with $h = 0.5$ and Gaussian noise at amplitudes η applied (multiple radial configurations detected in each case). The inner circular structure is correctly detected for $\eta \leqslant 0.5$ ($\eta \lesssim h$).

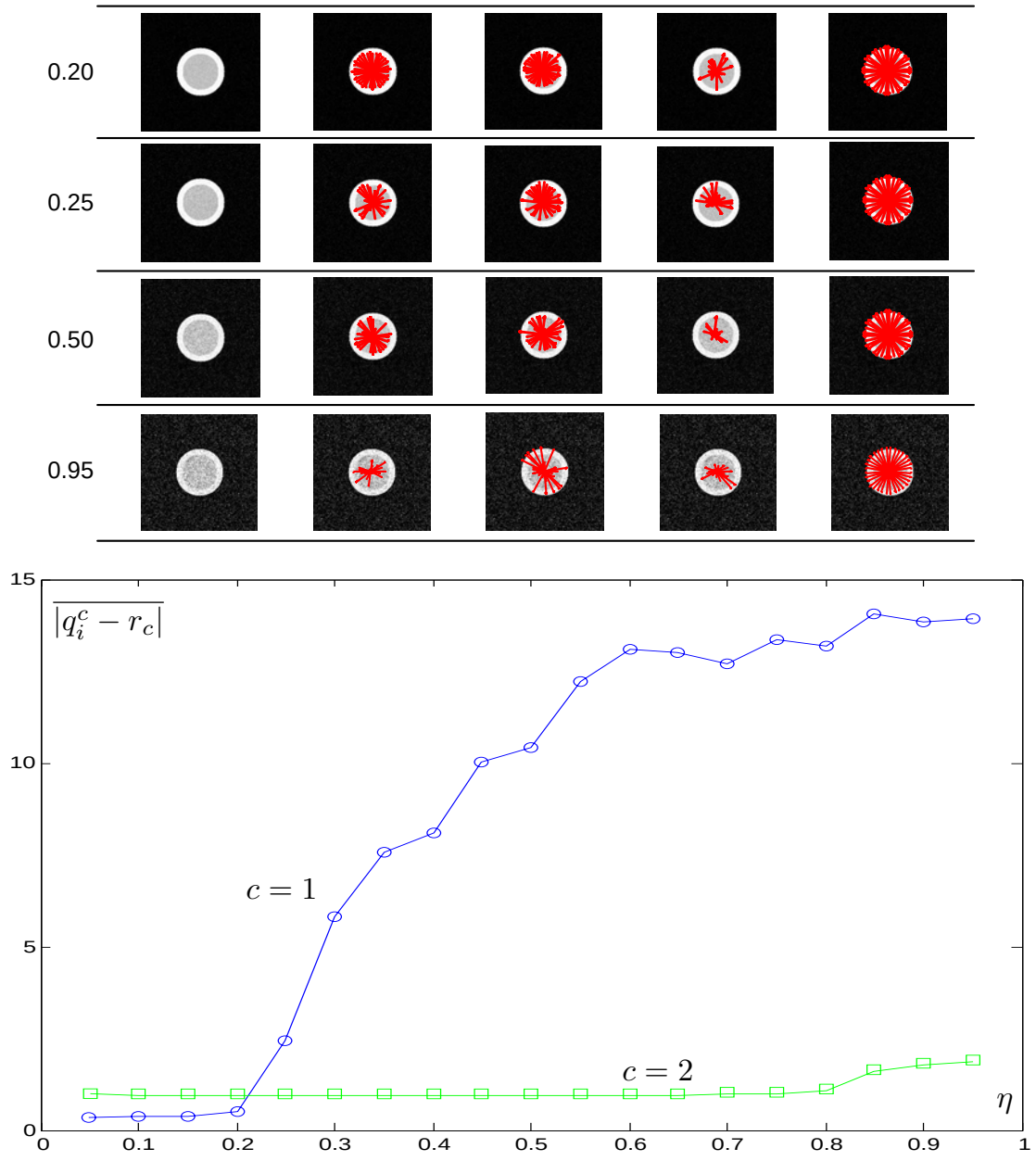


Figure 4.12: The result of boundary detection on the test image with $h = 0.75$ and Gaussian noise at amplitudes η applied. The inner circular structure is correctly detected for $\eta \leqslant 0.2$ ($\eta \lesssim h$).

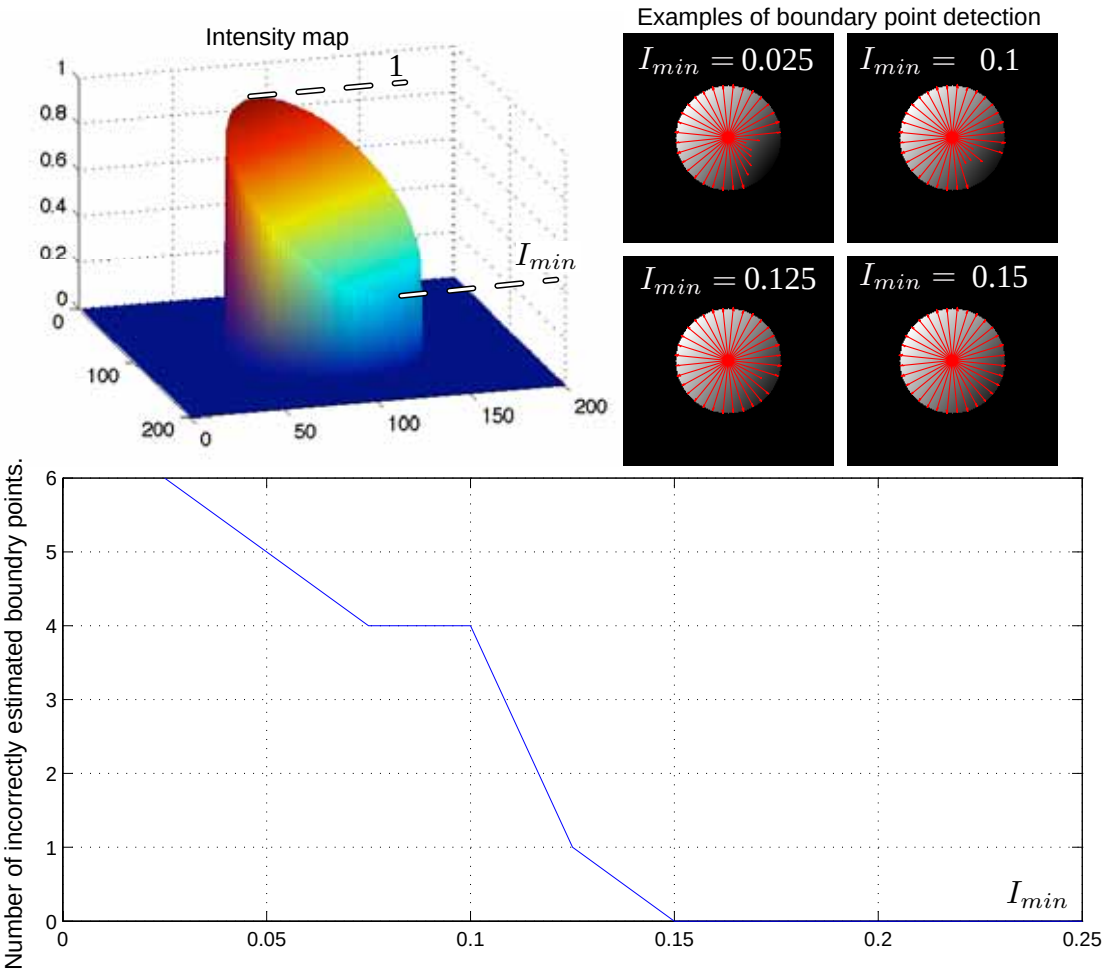


Figure 4.13: The result of boundary detection of the region with non-homogenous intensity distribution (top-right). The correct estimation of all boundary points is possible for $I_{min} \geqslant 0.15$.

ation of region detectors such as MSER, Harris corner detectors and blob detectors (see Section 2.6) which assume that the region is represented by a circle or an ellipse. The ORC descriptor produces an arbitrary convex region which as in the method of Tuytelaars and van Gool [70] could be approximated with an ellipse and compared to other detectors. This however would remove the information about the captured shape which differentiate ORC from other region detectors. The other possibility is to use the captured shape and the descriptor distance described in Section 4.2.8 but then not only the region detection is compared but also the descriptor matching. This approach is used for evaluation of ORC

performance in the matching of local image structures in the animal database described in Section 4.2.9.

4.2.8 Descriptor Distance

We discuss the measurement of scale and rotation invariant distance between ORC descriptors. The scale invariant distance is introduced first and then the technique for achieving invariance to rotation is presented.

ORC distance is calculated between two boundary point configurations $f(i)$ and $f(j)$. Each configuration is represented by an M row by N column matrix, where N is the number of sectors and M is the number of features per sector. In the subsequent evaluation N was set to 32, representing a trade off between computational complexity and the level of detail captured by the descriptor. We have used two features ($M = 2$) per sector, representing the boundary point radii values q_i^1 and q_i^2.

The Euclidean distance between descriptor i and descriptor j, rotated by p sectors, is calculated according to:

$$d(i,j,p) = \sum_{n=1}^{N}\sum_{m=1}^{M}\left(f(i)_{m,n} - \varsigma_{i,j}(p) f(j)_{m,(n+p) \bmod N}\right)^2 \tag{4.5}$$

where $\varsigma_{i,j}(p)$ is a relative scale between descriptors i and j, which is discussed later in this section. Equation (4.5) is equivalent to the horizontal rolling of the descriptor j matrix p times to the right.

Local image descriptors, such as SIFT, are extracted from an interest region, which provides an estimation of the local image structure scale. While this approach allows the simplification of the descriptor extraction and reduces matching complexity it also introduces dependence of the descriptor on the accuracy of the scale or interest region detection.

In our approach we have decided to avoid prior scale detection and replace it with the detection of relative scale between two descriptors, which is calculated during descriptor matching. The relative scale $\varsigma_{i,j}$ between descriptor i and j, for a given rotation p of the descriptor j, is calculated as follows:

$$\varsigma_{i,j}(p) = \frac{\sum_{n=1}^{N} f(i)_{1,n} f(j)_{1,(n+p) \bmod N}}{\sum_{n=1}^{N} \left(f(i)_{1,(n+p) \bmod N}\right)^2} \tag{4.6}$$

It can be proven, that the distance (4.5) reaches its minimum, for a given rotation p,

if descriptor j is scaled by $\varsigma_{i,j}(p)$.

Rotational invariance can be achieved by estimating local dominant orientation, either from local gradient covariance or by fitting an ellipse to the ORC boundary. This strategy produces consistent results for elongated or anisotropic structures but also produce unstable results for structures with no dominant orientation. We therefore apply a rotationally invariant distance measure, which is consistent in both cases:

$$d(i,j) = \min_{p}(d(i,j,p) : p = 1..N, M = 1) \tag{4.7}$$

As in the case of scale invariance we opt rather for the detection of relative orientation between two descriptors than prior orientation estimation for each descriptor due to estimation stability as previously discussed. The relative orientation between two configurations p_{min} is found from the minimum configuration distance calculated for $M = 1$ (taking into account only q_i^1 values).

The distance between two boundary point configurations depends on the consistency of interest point positioning. The distance calculated between two ORC descriptors originating from the same structure but calculated at different interest point positions is not equal to 0. Avoiding this dependency requires an iterative optimization of the distance, where one of the descriptors is iteratively translated to minimize the final distance. The results of ORC testing presented in Section 4.2.9 were generated without refining interest point positions.

4.2.9 Local Structure Matching - Performance Evaluation

The proposed testing procedure evaluates the matching of local structures found in the following animals: tiger, zebra, giraffe and leopard. Tigers and zebras contain mostly elongated features e.g. straight and bent stripes, while leopards and giraffe contain a mixture of round and elongated shapes. Every animal type represents a distinctive pattern of local structures, suitable for matching using local descriptors.

Mikolajczyk et al. [42] compare descriptors by evaluation of their performance in matching local structures between transformed images of the same scene. For the evaluation presented here, we show the ability of the descriptor to generalize while still being discriminative. That is we measure the accuracy of matching of the same type of structure in different scenes, e.g. the giraffe skin pattern in many images containing different giraffes.

The test procedure operates on groups of images (36 groups in total), each containing

five images with different animals and backgrounds. The images contain only one type of animal each and were manually segmented to separate the background from the animal. The first two images in the group contain an animal of the same type and the remaining three images contain other animals e.g. (zebra 1, zebra 2, giraffe, leopard, tiger). Descriptors are extracted for the region corresponding to the animal (obtained from the manual segmentation) in the first image in the group. These descriptors are then matched with all descriptors from the rest of the images in the group to find the closest match for each descriptor. This way the percentage of closest matches inside the same type of animal as well as inside other animal types and backgrounds can be recorded.

The results of local structure matching were obtained from 36 image groups (180 images). The image groups are divided into 4 categories, each containing one type of animal occurring twice in every group. Figures 4.16 show examples of descriptor matching and Figure 4.17 shows a set of interest point correspondences obtained from descriptor matching. Figure 4.15 shows the example of descriptor matching in image groups – the first column contains images that are the source of descriptor models extracted from the animal while other images show the interest point locations that correspond to one of the interest points in the first image in the row. Figure 4.14 shows the average percentage (within each category) of descriptors from the animal in the first image matched inside animals in other images in the group (black and gray bars) as well as matched anywhere in other images (white bars). The summarized results, showing only the percentage of descriptors matched in the same kind of animal are presented in Table 4.1.

The results expose limitations of SIFT and ORC associated with their locality – images used for testing contain a variety of backgrounds and as one could expect there is a high chance of finding local structures in one of the image backgrounds that are similar to the structures present in the animal e.g. most of the tiger descriptors are matched with the fence in the background of the image containing a giraffe in Figure 4.15 (second row). These matches are shown in more detail in Figure 4.16. The ORC method produces a higher percentage of descriptor correspondences within the same animal category compared to SIFT by 10–20%. The difference in the matching performance can be attributed to two primary factors: the detection of homogenous regions in images used by ORC is more consistent than the DoG based scale selection and the orientation invariant distance allows to avoid orientation ambiguity in round structures which shows the importance of the invariant matching.

We compare the results of the proposed ORC descriptors with SIFT descriptors*.

Category	Group count	ORC (%)	SIFT (%)
leopard	10	43.6	36.2
giraffe	9	40.4	29.2
tiger	9	21.4	12.0
zebra	8	64.1	40.3

Table 4.1: Animal matching results per animal category.

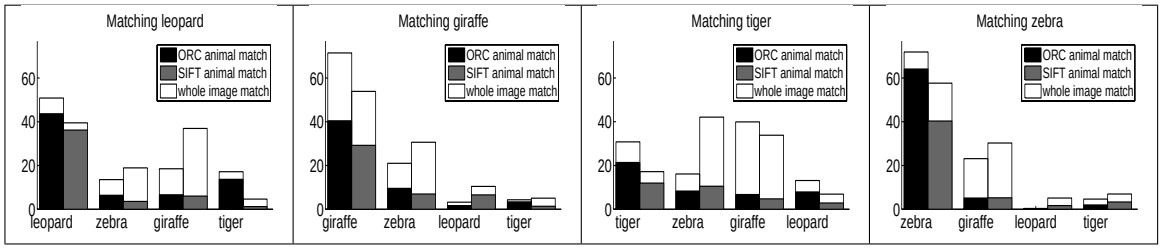


Figure 4.14: Animal matching results per animal category. Black and gray bars show the percentage of category descriptors matched inside corresponding animals for ORC and SIFT respectively. White bars show the percentage of category descriptors matched anywhere in the corresponding animal images.

4.2.10 Discussion

The evaluation of ORC and SIFT† descriptors applied to the local image structure matching has shown that both descriptors are comparable with ORC performing 10-20% better than the standard SIFT in the experiments on images of four types of animal. It has also shown that the applicability of both local descriptors to general object recognition tasks is limited by the lack of global context. Significant improvements are possible only if the co-occurence and spatial relationship of local structures are taken into account.

In [46] the SIFT descriptor is extended with by adding a "global context" that allows to capture image structure beyond the local image patch used by SIFT. The advantages of using spatial configurations of local descriptors are demonstrated in [6, 52]. The ORC

*SIFT descriptors and interest points were generated using code available from http://vision.ucla.edu/vedaldi/code/sift/sift.html by A. Vedaldi.

†the descriptor comparison provided by Mikolajczyk et al. [42] does not indicate differences in matching performance that would generate better results than ORC, however precise comparison is yet to be performed.

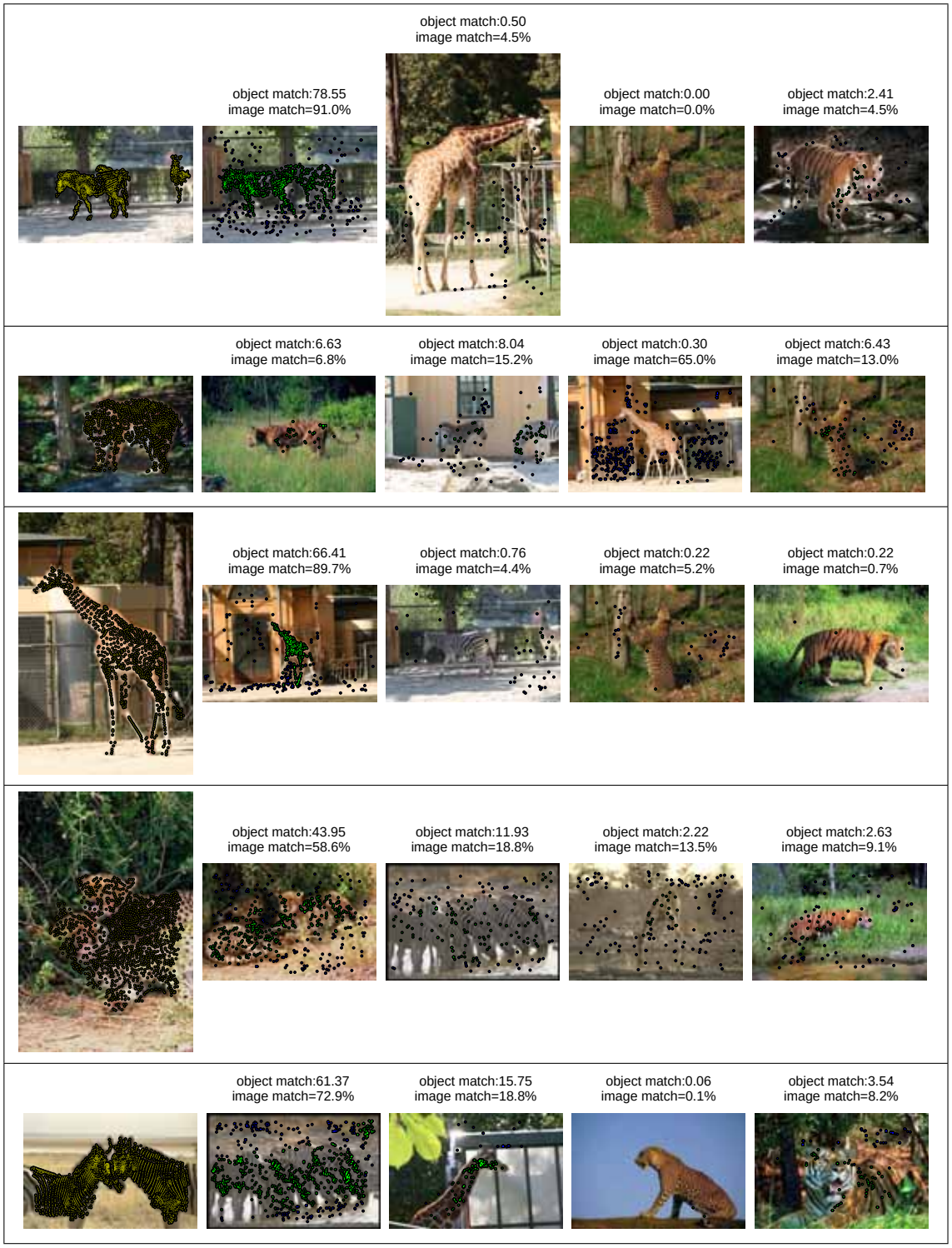


Figure 4.15: Example of descriptor matching in the image sequences (in rows). Images in the first column are the source of model descriptors extracted at locations marked by dots. Images in other columns show the locations of closest matches to model descriptors. The "object match" and "image match" provide the percentage of model descriptors matched inside the corresponding animal and image respectively.

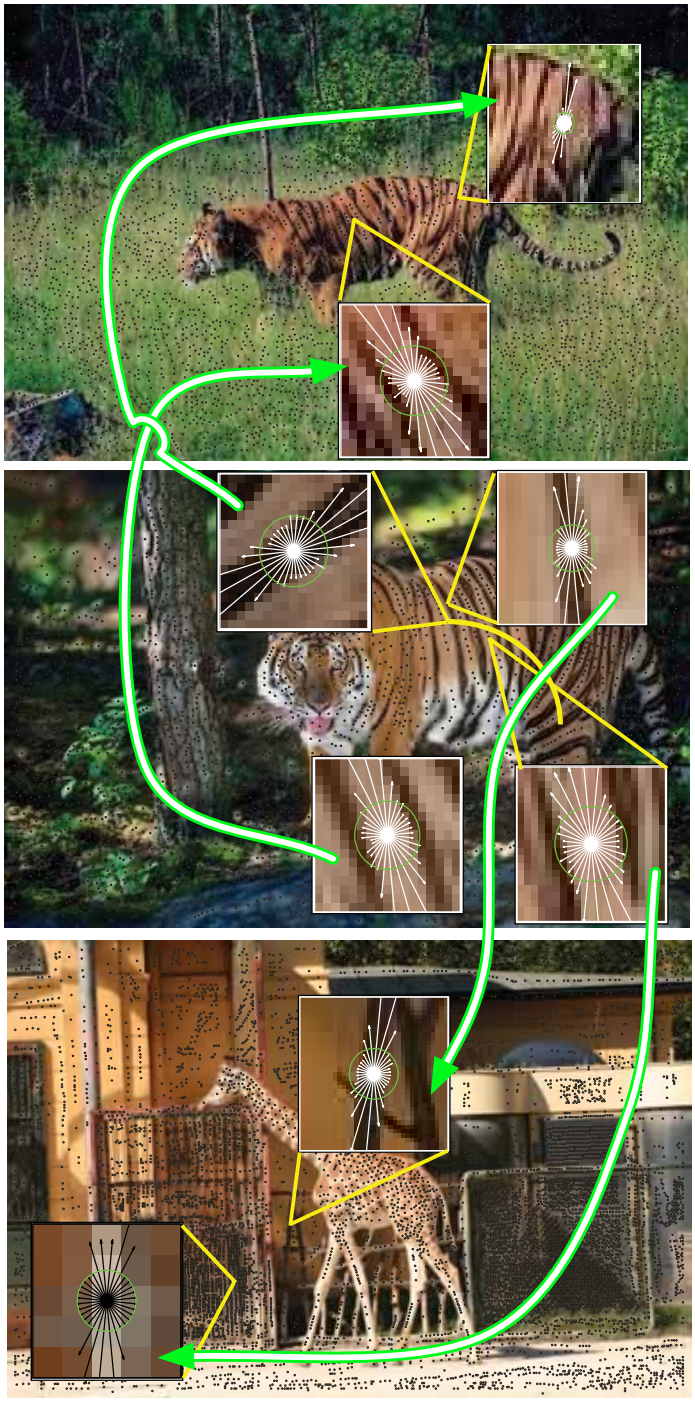
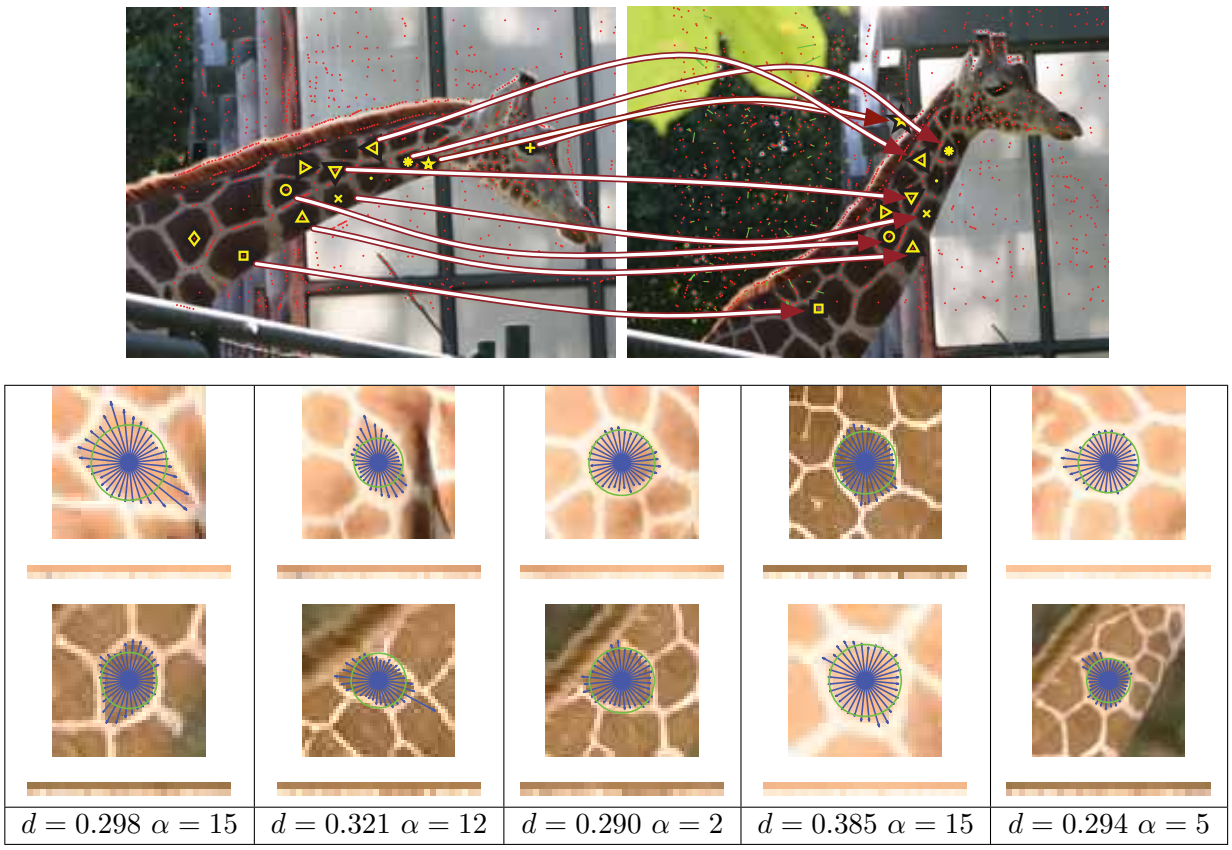


Figure 4.16: Examples of ORC descriptors originating from the image in the middle matched to the descriptors in the other two images. Some of the local structures in the lower image are similar to the tiger stripes on a local scale.



$d = 0.298$ $\alpha = 15$	$d = 0.321$ $\alpha = 12$	$d = 0.290$ $\alpha = 2$	$d = 0.385$ $\alpha = 15$	$d = 0.294$ $\alpha = 5$

Figure 4.17: Five ORC matches exhibiting the lowest distance among all possible combinations of descriptor matches in two images. The distance d and relative orientation α between two descriptors (counted in sectors) are provided below each pair of descriptors in the bottom row.

descriptor provides a natural way for constructing spatial configurations of multiple descriptors, since adjacency of descriptors can be extracted from the boundaries shared by neighboring structures. This is a topic of future research.

4.3 Radial Edge Configuration for Semi-Local Image Structure Description.

Edges are an intuitive way to represent shape information, but the problems associated with the detected edge fragmentation, missing edges due to occlusions and low contrast as well as changes in object scale and orientation affect the final result based on edge matching or classification. To overcome this problem we introduce a novel semi-local shape descriptor which represents the shape of an image structure by means of edges and their configurations. Our *Radial Edge Configuration*-descriptor (REC) encodes edges found in a neighborhood of an interest point as a sequence of radial distances in a polar coordinate system (centered on the interest point). Thus, the similarity of shape is assessed by the comparison of local edge configurations. Here, our main contribution is the definition of a rotation and scale-invariant distance measure between edge configuration descriptors that is able to match multiple edges, preserving their spatial relationships, and reject outlier edge pairs at the same time. This allows for a comparison of image structures across different scales, with only partially established correspondences. Another particularity of the chosen approach is that scale and orientation are not estimated during descriptor extraction. Instead they are established as relative entities between two REC descriptors during the distance calculation, which leads to more stable results.

4.3.1 Edge Matching in Polar Coordinates

The complexity of edge matching is primarily associated with the difficulty in assigning a scale to the edge – a part of one edge may be matched to another edge or to itself at a larger scale (e.g. straight edges or fractal like structures). Polar coordinates allow the definition of an edge scale locally, based on the relative position to the origin of a coordinate system. However, the matching of a part of an edge to a part or whole of another edge is still admissible.

The origin of the coordinate system is associated with the interest point location. We use the symmetry based interest point detector RST introduced in Chapter 3 and the Canny edge detector for obtaining edges around interest points. Examples of interest point distribution and edges detected in a hand X-Ray are shown in the Figure 4.18.

The REC descriptor consists of a variable number of K continuous edges. The k-th edge Γ_k is encoded as an ordered list of radial boundary points, each representing the distance $r_{k,i}$ along the i-th ray from the origin of the polar coordinate system:

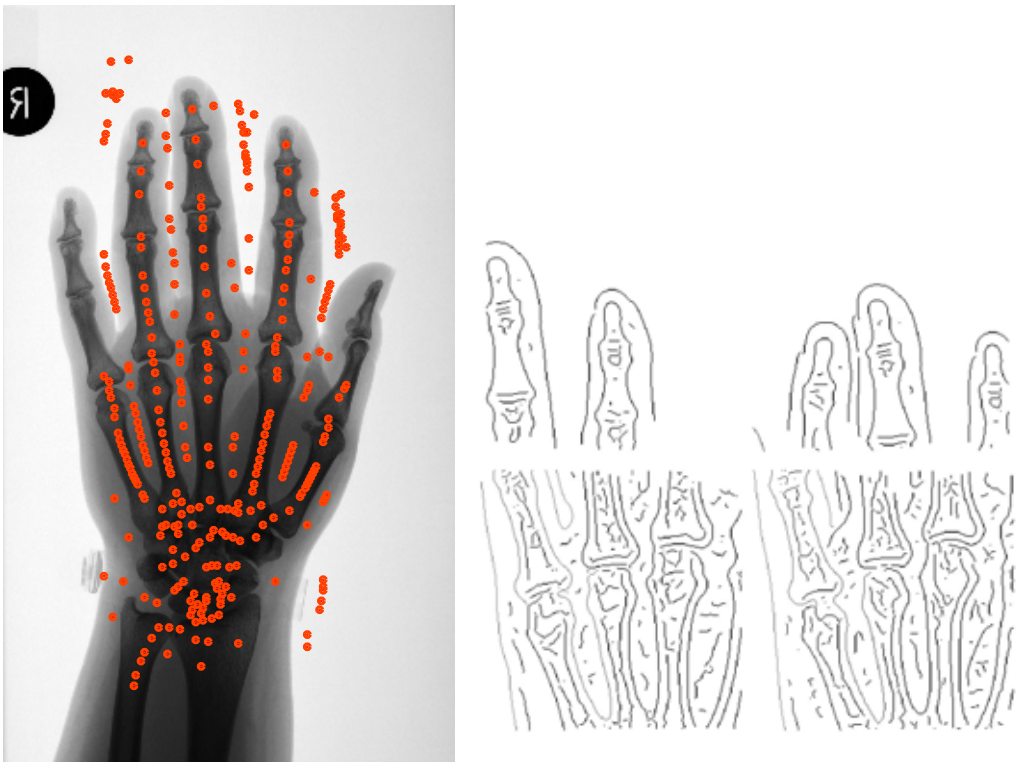


Figure 4.18: Left: examples of interest point distribution. Right: examples of edge detection.

$$\Gamma_k = \{r_{k,i} : i \subset \mathbb{N}_0^+; i = (b_k ... b_k + n_k) \bmod N\} \tag{4.8}$$

where b_k denotes the index of the first ray and n_k is the number of rays the edge occupies. The modulo operation is used to ensure that index $i < N$, where N describes the total number of rays (polar resolution) and in all our experiments is set to 64, which we found to offer a good compromise between accuracy and computational cost.

Calculating the distance between two REC descriptors involves finding correspondences between multiple edges. We describe a method to find the best fit between two edges, assuming one of the edges can be rotated and scaled relative to the origin of the polar coordinate system associated with the interest point (as shown in Figure 4.19). This operation is a prerequisite for the estimation of distance between two REC descriptors.

Fitting one edge to another corresponds to finding a transformation (rotation and scaling) which globally minimizes the spatial distance between corresponding boundary points of the two edges. It is important to note that while the scaling of an edge is

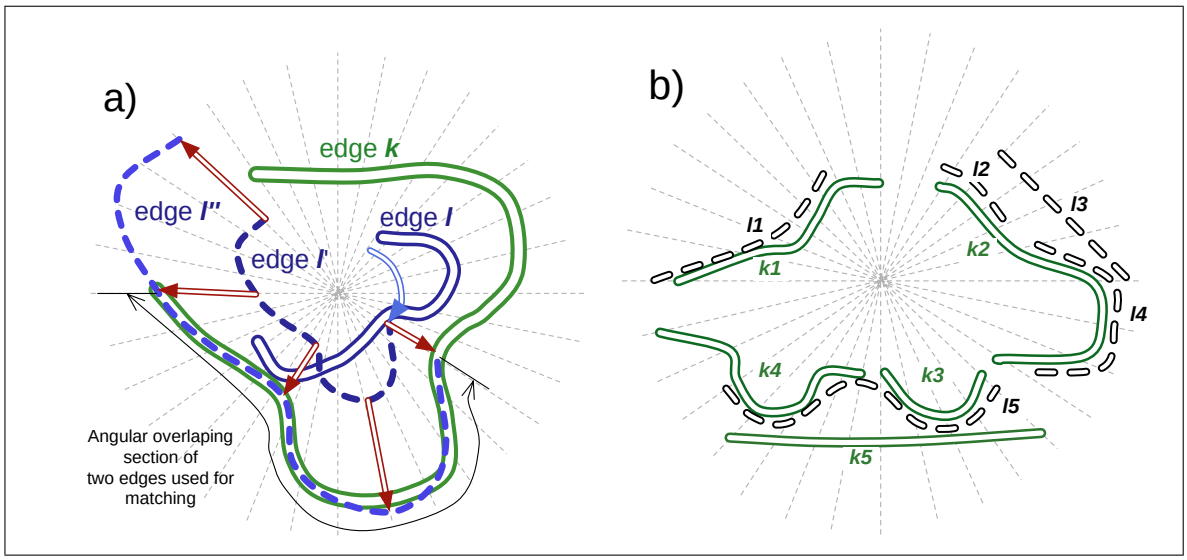


Figure 4.19: a) example of matching edge k and l in polar coordinates. Edge l' is a rotated version of l and l'' is scaled version of l' relative to the origin of the coordinate system. b) example of edge correspondences in two descriptors (edges k and l).

performed in the continuous domain, the relative rotation is quantized into N rays. The relative scale $\varsigma_{k,l}^{a,b}$ between edge k belonging to the descriptor a and edge l belonging to the descriptor b, rotated by α rays, is calculated as follows:

$$\varsigma_{k,l}^{a,b}(\alpha) = \left(\sum_{i=b_{kl}}^{b_{kl}+n_{kl}} r_{k,i}^{a} r_{l,\bar{i}}^{b} \right) / \left(\sum_{i=b_{kl}}^{b_{kl}+n_{kl}} (r_{l,\bar{i}}^{b})^2 \right) \tag{4.9}$$

where b_{kl} is the first ray containing boundary points of both edges, n_{kl} is the number of consecutive rays containing boundary points from both edges for a given rotation α and $\bar{i} = (i - \alpha) \bmod N$. It is important to note that this scheme allows for partial edge matching, which means that only the overlapping section of the two edges is matched (as shown in Figure 4.19). However, only combinations of α for which $n_{kl} \geqslant \tau$ (in our experiments τ=5) are used, due to the fact that extremely short sections of an edge usually carry less information, which is made worse by the quantization process. It can be easily proven that the spatial distance between corresponding boundary points of the edges k and l, for a given rotation α, is minimized when edge l is scaled (multiplied) by $\varsigma_{k,l}^{a,b}(\alpha)$.

One way of estimating how well two edges fit together is to calculate the variation of relative scale between the corresponding boundary points:

$$\epsilon_{k,l}^{a,b}(\alpha) = \frac{1}{n_{kl}} \sum_{i=b_{kl}}^{b_{kl}+n_{kl}} \left| \log^2\left(\frac{r_{k,i}^{a}}{r_{l,\bar{i}}^{b}}\right) - \log^2\left(\varsigma_{k,l}^{a,b}(\alpha)\right) \right| \tag{4.10}$$

This equation is a scale independent fitting distance between two edges for a given relative rotation α. The $\log^2()$ operation is used to avoid impairment associated with the $\frac{r_{k,i}^{a}}{r_{l,\bar{i}}^{b}}$ measure. The relative rotation giving the best fit of the two edges is the one which minimizes the distance $\epsilon_{k,l}^{a,b}$:

$$\epsilon_{k,l}^{a,b} = \min_{\alpha}\left(\epsilon_{k,l}^{a,b}(\alpha) : n_{kl} \geqslant \tau\right) \tag{4.11}$$

Finding the transformation resulting in the best fit between two edges requires $\epsilon_{k,l}^{a,b}(\alpha)$ to be evaluated for all α (for which $n_{kl} \geqslant \tau$ holds).

4.3.2 Descriptor Distance

The REC descriptor contains a set of edges that are the result of edge detection around the corresponding interest point. In reality we should expect that some perceptible edges may be missing or fragmented due to weak gradients and noise. An additional problem is related to the fact that only a subset of edges in the two descriptors may correspond well, while others are related to non-similar image structures. For example we can find patches on a giraffe skin with a high shape similarity at a local scale, but the random distribution of the patches makes shape comparison irrelevant on a large scale. Thus we have to search for a subset of edges in both descriptors, which together give a low fitting error, while other edges are rejected as outliers.

The primary idea behind the matching of multiple edges in the descriptors a and b is summarized below:

1. Perform edge fitting for admissible edge pair combination k and l, resulting in P putative transformations.
2. Repeat multiple edge fitting for P transformations. Choose the one which gives the lowest overall fitting error for the descriptor.
   (a) Rotate and scale all edges in descriptor b according to the current transformation and find the edge correspondences between two descriptors.
   (b) Remove outliers and calculate the final distance from all corresponding edge pairs.

The most computationally demanding task is finding edge correspondences for a given relative scale and rotation. The difficulty is associated with the possibility that a single edge in one descriptor may correspond to more than one non-overlapping edge in the other descriptor. An example of such multi-correspondences is shown in the Figure 4.19-b – edge $k2$ corresponds to edges $l2$ and $l4$, while edges $k4$ and $k3$ correspond to edge $l5$. Note that edge $l3$ could be also matched to the edge $k2$, but it overlaps with edges $l2$ and $l4$, which produce a better fit with edge $k2$. The process of finding edge correspondences can be divided into several steps:

1. Find overlapping edge pairs in a: $\phi^a_{k1,k2} = \begin{cases} 1, & \text{if } k1 \text{ and } k2 \text{ overlap } \geqslant \tau \\ 0, & \text{otherwise} \end{cases}$
2. Find overlapping edge pairs in b: $\phi^b_{l1,l2} = \begin{cases} 1, & \text{if } l1 \text{ and } l2 \text{ overlap } \geqslant \tau \\ 0, & \text{otherwise} \end{cases}$
3. Find overlapping edge pairs between a and b: $\phi^{ab}_{k,l} = \begin{cases} 1, & \text{if } k \text{ and } l \text{ overlap } \geqslant \tau \\ 0, & \text{otherwise} \end{cases}$
4. Find edge correspondence. The edge l is correspondent to edge k if:

$$\epsilon^{a,b}_{k,l} = \min_{f,g}\Big(\epsilon^{a,b}_{f,g} : f \in \{\phi^{ab}_{f,l} = 1 \wedge \phi^a_{f,k} = 1\}; g \in \{\phi^{ab}_{k,g} = 1 \wedge \phi^b_{l,g} = 1\}\Big) \tag{4.12}$$

   which means that edges k and l correspond when the distance $\epsilon^{a,b}_{k,l}$ is the minimum among all combinations of edges f and g which overlap with k and l. This condition allows the association of multiple non-overlapping edges in one descriptor with a single edge in another descriptor.

The final distance between two descriptors a and b is a weighted sum of individual edge-pair (k, l) distances:

$$\epsilon^{a,b} = \frac{1}{\sum_{k,l} \upsilon^a_k \upsilon^b_l} \sum_{k,l} \upsilon^a_k \upsilon^b_l \epsilon^{a,b}_{k,l} \tag{4.13}$$

where the weights υ_k and υ_l describe the confidence of edge match:

$$\upsilon_k = \frac{\widehat{s}^a_k}{s^a_k} \tag{4.14}$$

where s^a_k is the total length of edge k in descriptor a and $\widehat{s}^a_k$ is the length of all edge fragments that were matched to edges in the descriptor b. The edge match confidence reaches 1 if it was completely matched to other edge or edges and is 0 if it was not matched to any edges.

During our matching tests we found that a simple outlier removal scheme helped to improve results when only a part of the structure in the two descriptors was found to correspond.

Examples of finding similar image structures through the edge matching are presented in Figures 4.24, 4.25 and 4.26. Figure 4.27 show a number of closest matches between descriptors corresponding to distinctive structures in four images of hand X-Rays. The majority of descriptors are matched to similar structures despite differences in scale, orientation and slight deformations of shape.

4.3.3 Matching Characteristics

The performance of matching structures in real images is presented in Section 5.2. In this section we show the dependency of REC descriptor matching on the interest point drift, scale changes and shape deformations (matching is performed on a square structure). As a reference we use the distance $\epsilon^{a,b} = 0.0808$ between two different structures: the descriptor extracted at the center of a 100×100 pixel square and the second one extracted at the center of a circle that covers the same area as the square.

The interest points serve as the origins of polar coordinates in the REC descriptor and provide the reference points that enable the scale and rotation invariant matching of the REC descriptors. However, problems such as interest point drift, already discussed in Chapter 2, affect the edge representation in the REC descriptor. This means that matching two REC descriptors that encode identical sets of edges but extracted at different positions does not produce a distance equal to 0. The dependency of the matching distance to interest point drift is shown in Figure 4.20, where the descriptor extracted at the center (x_c, y_c) of the 100×100 pixel square is matched to the descriptors extracted at positions $(xc + \Delta x, yc + \Delta y)$. The Figure also shows the difference between descriptors extracted at position (x_c, y_c) and $(x_c + 15, y_c + 15)$. The maximum interest point drift at which the matching distance is smaller than the reference distance between square and circle is approximately equal to 8 pixels or 16% of the circle radius. However the distance $\epsilon^{a,b}(\Delta x, \Delta y)$ is monotonically decreasing toward 0 at $(\Delta x = 0, \Delta y = 0)$. This means the iterative optimization of the distance by finding $(\Delta x, \Delta y)$ which minimizes the distance is possible at the expense of computational complexity. This optimization is currently not used.

Let us analyze now the sensitivity of the descriptor to the change in scale. In this test the descriptor extracted at the center of a 100×100 pixel square is matched with the scaled

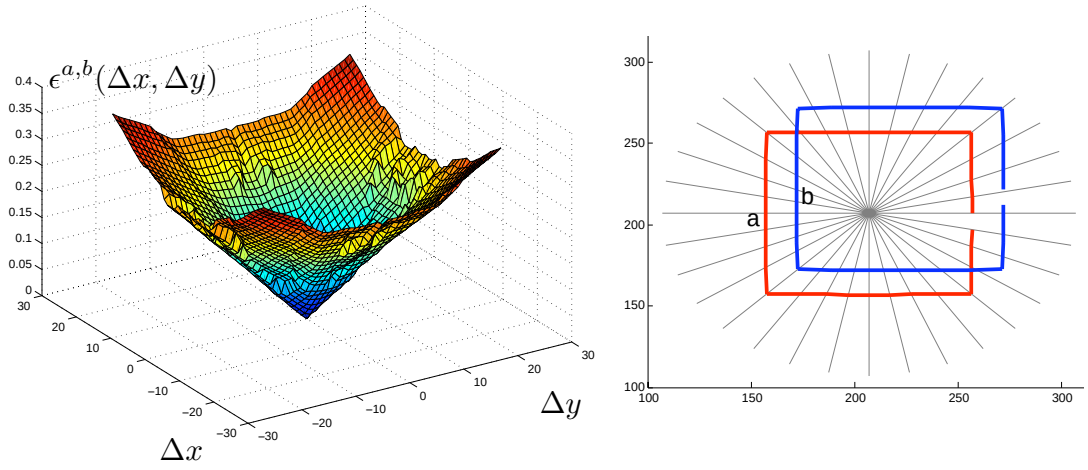


Figure 4.20: Left: The distance between descriptor **a** extracted at the center (x_c, y_c) of a 100×100 pixel square matched to the descriptors **b** extracted at positions $(x_c+\Delta x, y_c+\Delta y)$. Right: The polar representation of edges extracted at (x_c, y_c) and $(x_c + 15, y_c + 15)$.

down version extracted at the center of $\zeta^{a,b}(100 \times 100)$ pixel square. The relative scale $\zeta^{a,b}$ is varied from 0.1 to 1. Figure 4.21 shows the dependency of distance $\epsilon^{a,b}$ on the relative scale $\zeta^{a,b}$. Even when one square is 10 times smaller than the other the distance is two times smaller than the reference distance. Note that the use of the relative scale $\epsilon^{a,b} < 0.1$ would produce boundary points at a radius smaller than 5 pixels. The results indicate that the matching distance approaches infinity when the size of the structure approaches the resolution limit of the image which agrees with the Equation (4.10) describing the distance between edges.

Figure 4.22 shows the sensitivity of the descriptor to orientation changes. Two 100×100 pixel squares are matched and one of the squares is rotated by α degrees within the range 0..359 degrees. The result shows the dependency of the matching distance on the orientation α which is a periodic function containing 32 minima and 32 maxima (32 is the number of rays used in the REC descriptor). The minima are associated with orientations $\alpha = (0..31)\frac{2\pi}{32}$ for which the positions of boundary points along the boundary of the rotated square are identical (within the quantization error range) to the boundary points in the non-rotated square. For other orientations the boundary points in the rotated square correspond to locations between boundary points in the non-rotated square and therefore generate different distances from the square center.

The matching of REC descriptors is scale and rotation invariant which means that

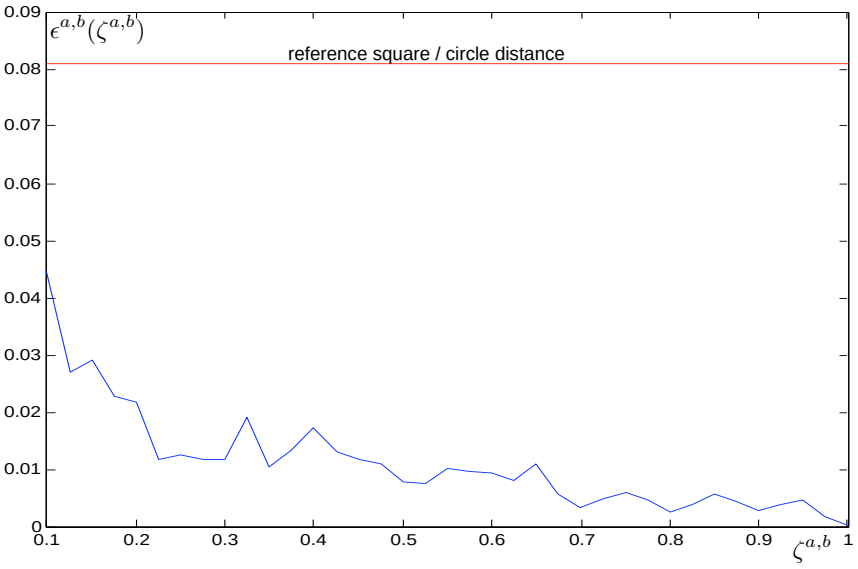


Figure 4.21: The dependency of distance $\epsilon^{a,b}$ on the relative scale $\zeta^{a,b}$. The distance is calculated between between descriptor **a** extracted at the center of a 100×100 pixel square and the descriptor extracted at the center of $\zeta^{a,b}(100 \times 100)$ pixel square. The fluctuations are caused by quantization errors after conversion to polar coordinates.

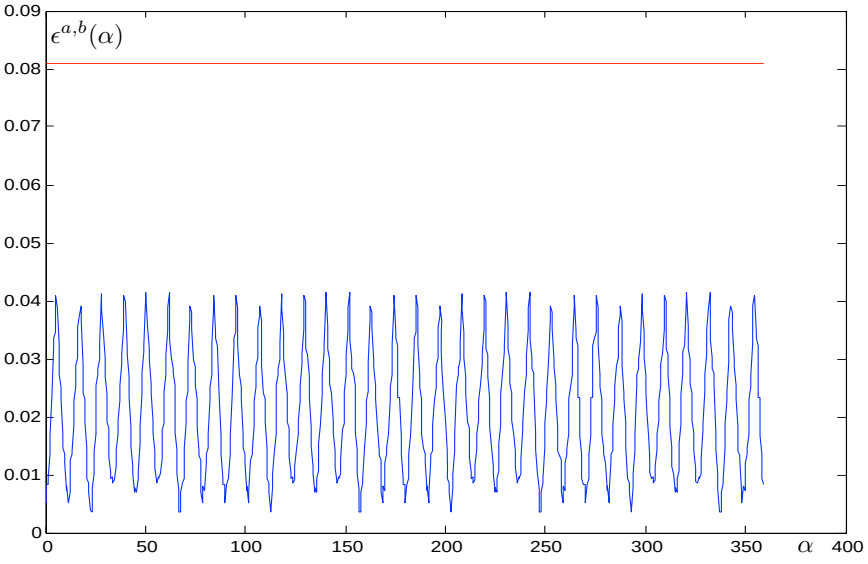


Figure 4.22: The dependency of distance $\epsilon^{a,b}$ on the rotation of the rectangle **b**.

it is designed for matching rigid structures. In this test we will examine the tolerance of the matching to anisotropic shape deformations. Matching will be performed between the descriptor extracted at the center of the 100×100 pixel square and the second one extracted at the center of the $100 \times \tau^b 100$ pixel rectangle (that can be caused by affine transformation). The value of τ^b is varied from 0.1 to 1. Figure 4.23 shows the dependency of distance $\epsilon^{a,b}$ on the relative scale τ^b. The result shows that the rectangles with $\tau^b < 0.9$ produce a higher matching distance than the reference distance between square and circle.

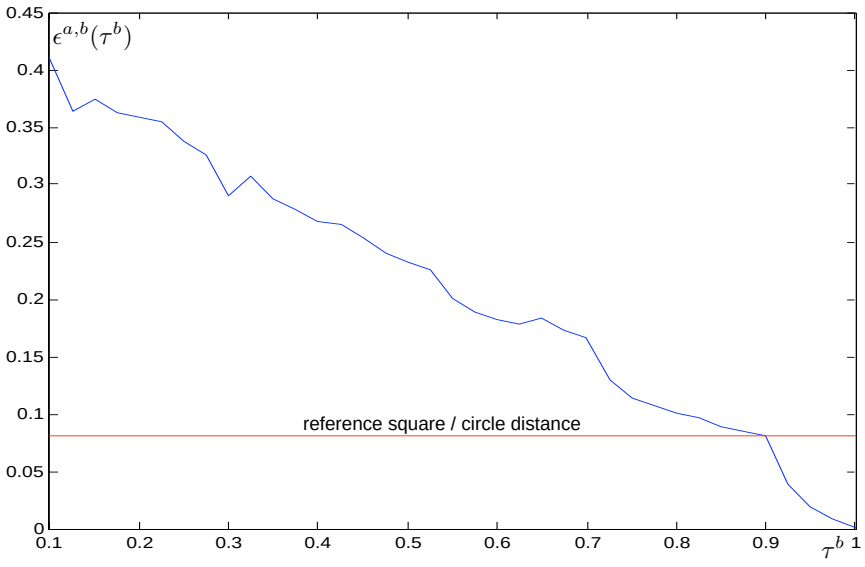


Figure 4.23: The dependency of distance $\epsilon^{a,b}$ on the ratio τ^b between the sides of rectangle **b**.

The REC descriptor is designed to operate on fragmented and incomplete object boundaries. Since the number of possible edge fragmentation cases is infinite a single test scenario cannot provide a comprehensive evaluation of descriptor sensitivity to fragmented or missing edges. An indication of matching performance in real images in which detected image structures contain fragmented edges is given in Chapter 5. The sensitivity to missing edges depends on the contents of the matched descriptors such as the amount of random (non-corresponding) appearances, the informativeness of the edge parts (straight vs. curved) and the correspondent boundary completeness. Again, the number of possible configurations is infinite and therefore we can provide only an indication of the matching performance in the real image database. For example the supervised model extraction presented in Chapter 5 operates on structures that contain random appearances e.g. the

neighborhood of the spinal cord in MRI image contain edge parts that are not repeated in other MRI images of the same type.

4.3.4 Discussion

The REC descriptor and a technique for matching REC descriptors introduce an idea of invariant matching which operates on non-invariant features. Each REC descriptor encodes a set of edge parts around the interest point that do not represent a particular scale or orientation since we do not differentiate between an object and the background at this stage. However, by comparison of two REC descriptors we find a subset of edges as well as relative rotation and scale such that these subset of edges minimize the distance between the two descriptors. We solve two problems at once: finding a subset of similar edges and finding a relative transformation between REC descriptors. This invariant matching is the primary difference between our approach and other methods described in Chapter 3.

This idea is further extended in Chapter 5 where repeatable image structures are extracted from training images using shape clustering based on invariant REC matching.

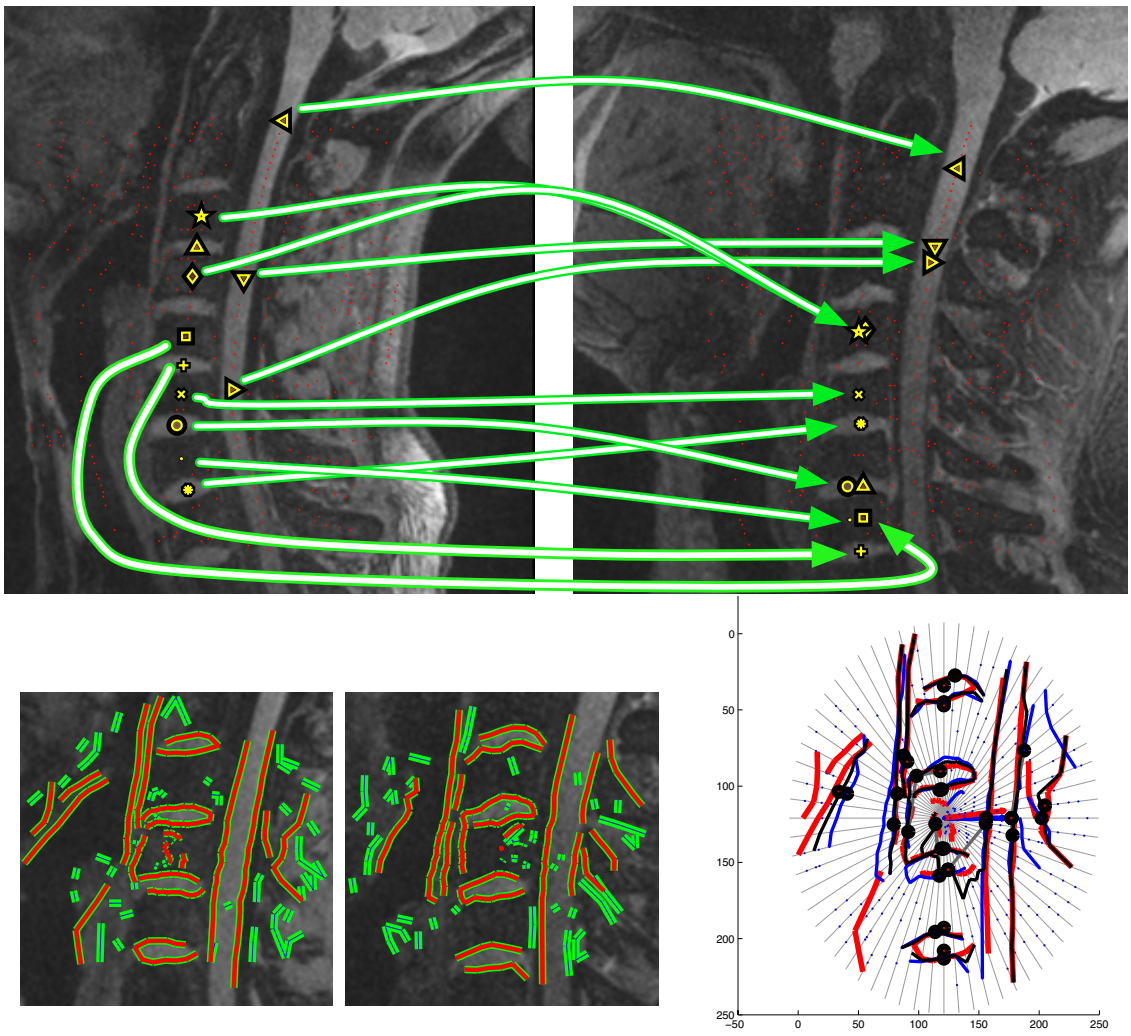


Figure 4.24: Top row: example of descriptor matching between different MRI images. Only a representative subset of interest point matches is shown to avoid clutter. Bottom row: example of two similar image structures matched. The first two images show corresponding image patches and the extracted edges (edges which length falls below configurable threshold are not used for matching and marked with a green color). The third image shows correspondence of edges from two descriptors (red and blue respectively) and the resulting mean edges after descriptor merging (black). Note that not all edges have been matched. We strongly advise to view all images in color.

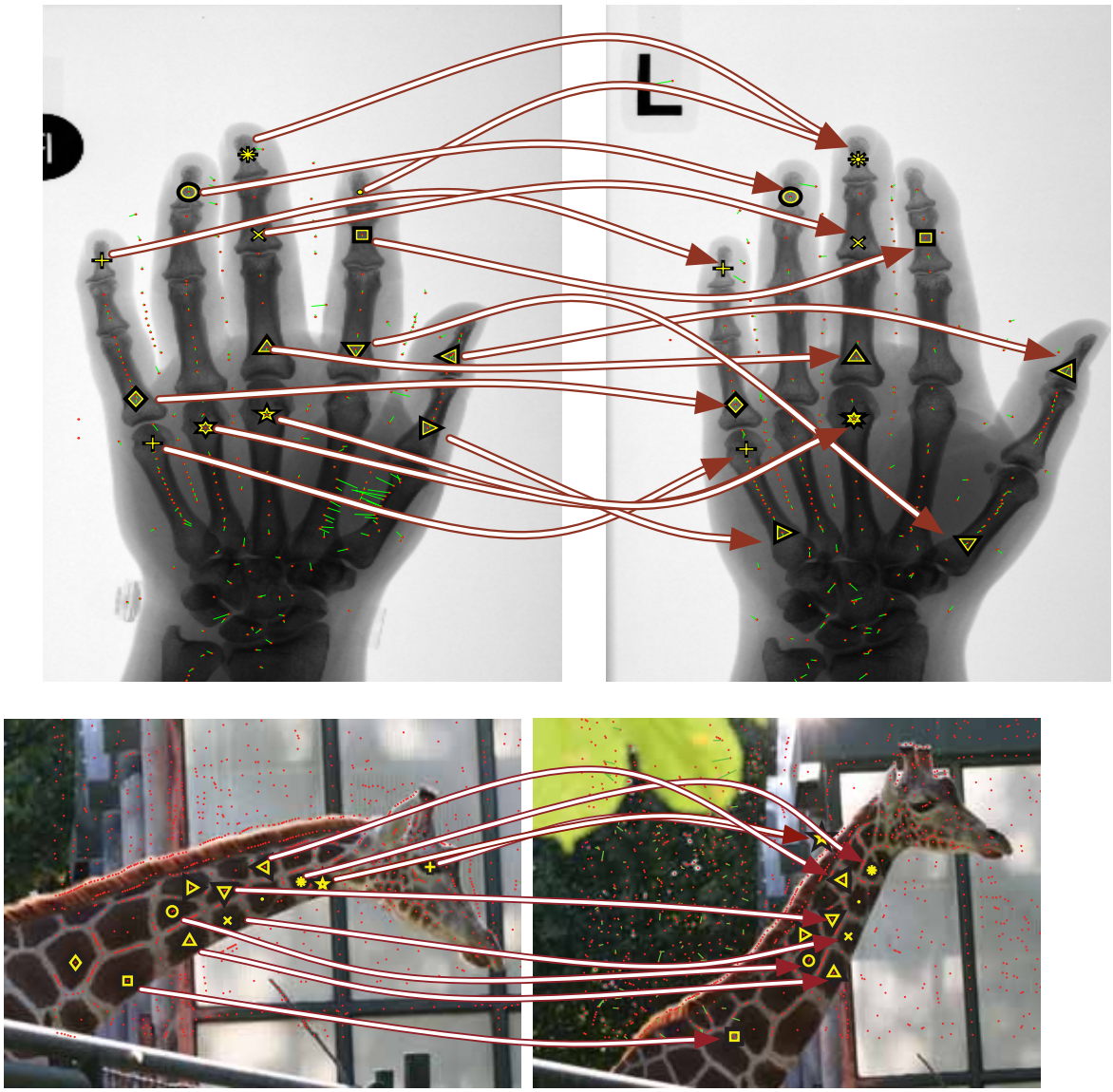


Figure 4.25: Examples of descriptor matching. Corresponding descriptor locations are connected with arrows and marked with a unique symbol.

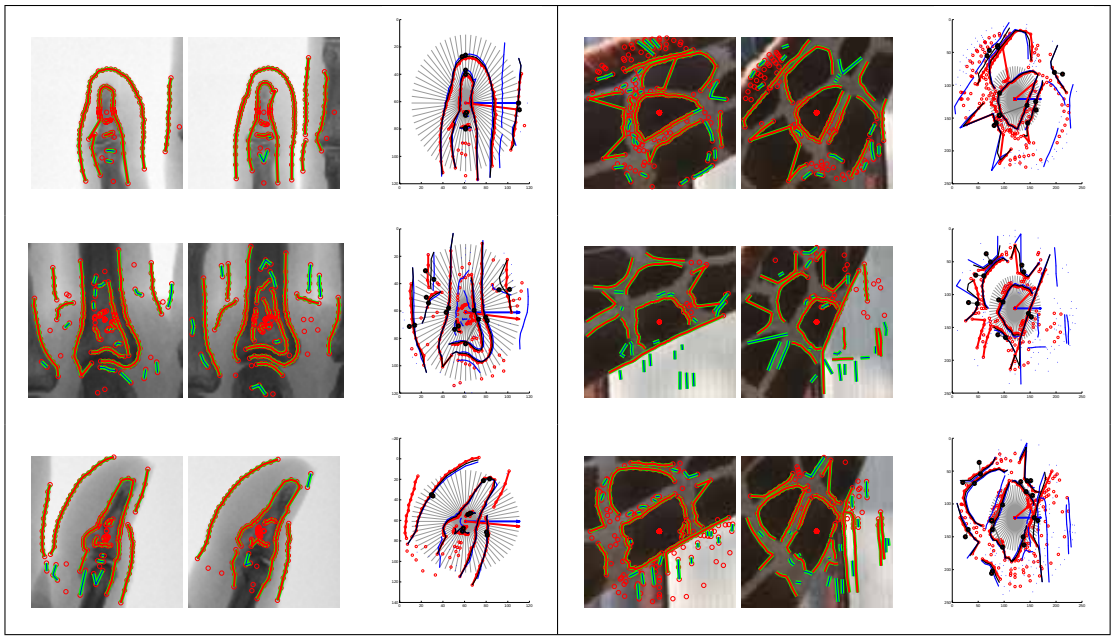


Figure 4.26: Examples of edge matching in X-Ray images of hands (left) and the giraffe skin (right). The first two columns contain corresponding image regions whose center is aligned with the center of the REC coordinate system. Detected edges are visible as red lines. The last column shows the edge correspondence between two descriptors with black lines depicting mean edges - a result of descriptor merging. Note that in each pair of matched descriptors only a subset of edges corresponds well – frequently some edges present in one of the descriptors are missing in another one due to imperfect edge detection or differences in local appearance.

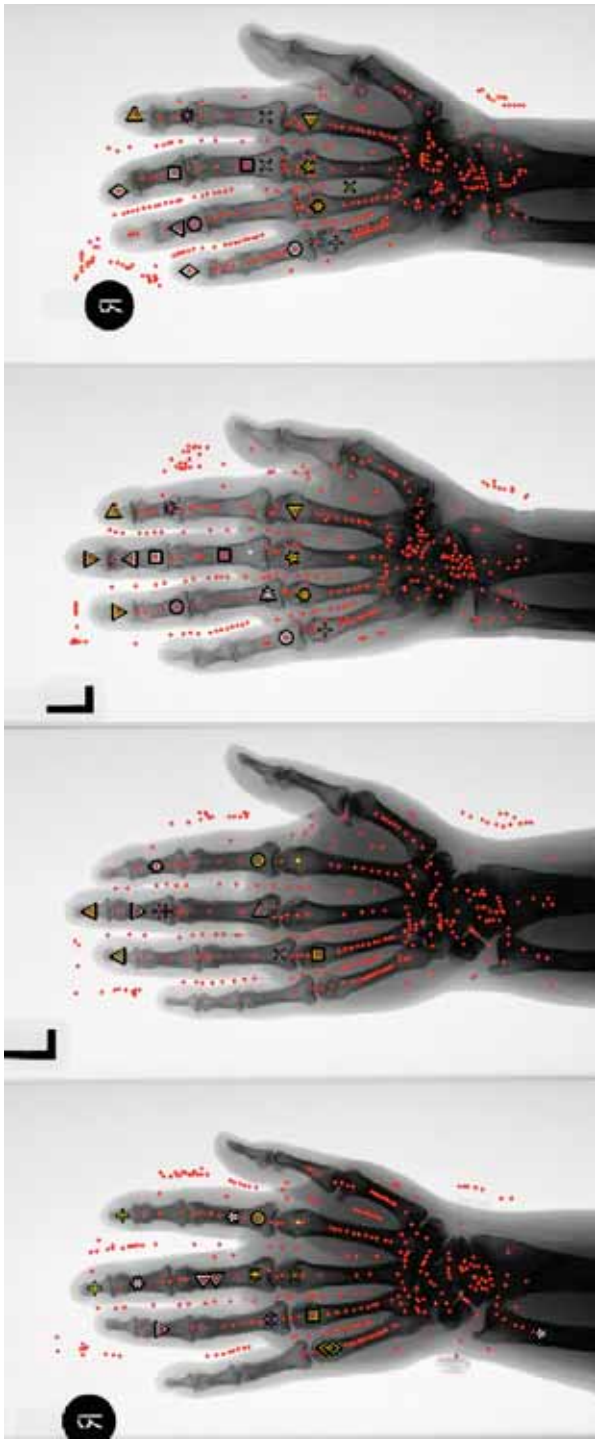


Figure 4.27: Examples of REC descriptor matching. The descriptors were extracted from four images of hand X-Rays and all combinations of two descriptors matched. Identical symbol/color pairs refer to the closest matches between corresponding descriptors. The number of correspondences shown was chosen for clarity due to the limited number of different symbols.

Chapter 5

Shape Clustering

Chapter 4 presents a novel semi-local image descriptor REC that is capable of encoding shapes as spatial configurations of edge fragments around an interest point. In this chapter we focus on unsupervised and weakly supervised learning of structure models that are represented by a set of REC descriptors with individual edges weighted accordingly to their repeatability and similarity within the same category of structures. The structure model learning is achieved through shape clustering presented in Section 5.1. The quality of extracted structure models is evaluated in two test scenarios: weakly supervised learning of characteristic structures in MRI spine images described in Section 5.2 and unsupervised learning using hand X-Ray images described in Seciton 5.3.

The shape clustering is related to agglomerative hierarchical clustering but operates on variable length feature vectors, specifically Radial Edge Configurations. The result of shape clustering are "mean" edge fragment configurations, also represented by REC descriptors, that can be used to locate similar structures in the image.

5.1 Clustering of Radial Edge Configurations

Clustering of local image descriptors (e.g. SIFT) is the basis of object recognition techniques such as "bag of keypoints" [74] and was used in extraction of part based models [30]. It has also been used for extraction of object shape fragments as described in Section 3.2.5. In these cases clustering allows for a compact (low-dimensional) representation of distinctive image structures. Among the most popular clustering methods are hierarchical, k-means and kd-tree clustering. The first difference between clustering of typical image descriptors and clustering of the REC descriptor is that the later produces

a variable length feature vector (the number of edges can vary significantly). This prevents the use of k-means and kd-tree clustering which require constant dimensionality of the feature vectors. The second difference is that the clustering of REC descriptors assigns weights to edges and individual boundary points along the edges that depend on the edge repeatability across training instances of the same structure type and the amount of variability an edge exhibits across the training instances.

The REC descriptor is clustered using agglomerative hierarchical clustering [11] based on the REC distance defined in Section 4.3.2. Clustering starts with finding the closest pairs between a set of descriptors extracted from the training data set labelled as clustering level $t = 0$. The closest pairs are merged into nodes at the next clustering level and the same procedure is repeated on these nodes. The closest descriptor pairs are merged only if the matching distance between them does not exceed the threshold τ. Therefore clustering is performed until no more pairs can be merged. Parameter $\tau = 0.4$ was experimentally chosen and used in all tests presented in this chapter. The merging of two descriptors is an operation which generates a single edge for each set of corresponding edges in two descriptors as described in Section 4.3.2. Recall that a single edge in one descriptor can correspond to several edges in another descriptor and that some edges do not have any correspondences and are down-weighted in the merged descriptor. The edge kl, which is a result of merging of edges k and l, is obtained by averaging the boundary point positions from both edges:

$$\Gamma_{kl} = \{0.5(r_{k,i} + r_{l,i-\alpha \bmod N_0^+}) : i \subset \mathbb{N}; i = (b_{kl}...b_{kl} + n_{kl}) \bmod N\} \tag{5.1}$$

In addition, each boundary point is assigned the weight that is corresponding to the distance between two merged boundary points and includes the boundary point weights from the previous clustering level. This way edges are prioritized according to their similarity across the clustering levels.

$$w_{kl}^t(i) = \omega_p(w_k^{t-1} + w_l^{t-1}) + \omega_d \exp\left(- \left(1 - \frac{\max\left(r_{k,i}^a, \varsigma_{k,l}^{a,b} r_{l,\bar{i}}^b\right)}{\min\left(r_{k,i}^a, \varsigma_{k,l}^{a,b} r_{l,\bar{i}}^b\right)} \right)^2 / \sigma^2 \right) \tag{5.2}$$

where σ was set to 0.25 in all experiments and regulates the down-weighting depending on the local edge deformation – the difference between relative boundary point scale and the relative descriptor scale. The parameters ω_p and ω_d regulate the influence of edge weights from previous cluster level $t - 1$ (history) and the differences between merged

edges (deformation) respectively onto the final weight $w^t_{kl}(i)$. These were set to $\omega_p = 0.25$ and $\omega_d = 0.75$ in all experiments which prioritizes the influence of "deformation" over the "history". The edges without correspondences are copied into the merged descriptor and the corresponding weights are divided by two – if such an edge consequently has no correspondences at multiple clustering levels its weight is reduced to approximately 0.

At clustering level $t = 0$ all boundary point weights are set to 1 which means that all edges in every descriptor have identical priority.

The result of clustering is a set of REC descriptors, which contain edges resulting from edge merging across a number of clustering levels. The weights assigned to the edges are then used during matching cluster nodes (structure models) to descriptors in the test data set. The edge distance (4.10) is then replaced with:

$$\epsilon^{a,b}_{k,l}(\alpha) = \frac{\sum_{i=b_{kl}}^{b_{kl}+n_{kl}} w^a_{k,i} \left| \log^2\left(\frac{r^a_{k,i}}{r^b_{l,\bar{i}}}\right) - \log^2\left(\varsigma^{a,b}_{k,l}(\alpha)\right) \right|}{\sum_{i=b_{kl}}^{b_{kl}+n_{kl}} w^a_{k,i}} \tag{5.3}$$

where descriptor a corresponds to the cluster node and weights for descriptor b corresponding to the detected structure are set to 1.

5.2 Supervised Model Extraction in MRI Spine Images

The intention behind this test scenario is to show the discriminative capabilities of structure models obtained from shape clustering. The evaluation is performed on MRI spine images, that contain characteristic structures such as vertebrae, disks and the spinal cord. Figure 5.1 shows examples of MRI images used in this evaluation as well as examples of the manual structure annotation that assigns structure type labels to the symmetry based interest points. The annotation of a single image can be performed in less than one minute – the annotation of structure boundaries is not needed. The MRI image database consists of 30 images in total but only 4 images are used to extract structure models.

The localization of vertebrae, disks and spine has a medical application of providing landmarks for image segmentation and global structure localization [10].

The structure model extraction is performed using shape clustering described in Section 5.1. The training descriptor database is obtained from 4 training images (approximately 10% of all images) and grouped into categories according to the manual image annotation (see Figure 5.1). Every category is then clustered which produces cluster trees containing structure models. Figure 5.2 shows an example of structure models obtained

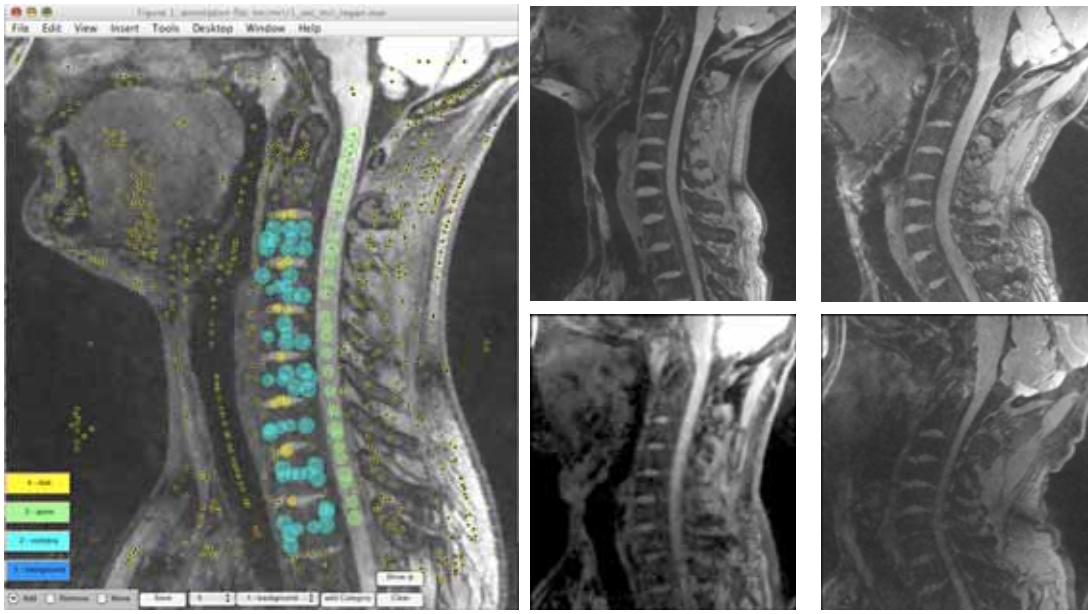


Figure 5.1: Left: Example of MRI annotation. The categories represent 3 characteristic structures (visible as color disks covering corresponding interest points) and the background (interest points that were not annotated). Right: Examples of test images. Note that the scale and orientation of these structures is not constant across different images.

Model	**Vertebra**		**Disk**		**Spine**		**Background**	
	tp	fp	tp	fp	tp	fp	tp	fp
Unclustered	0.6809	0.0413	0.8875	0.0212	0.8511	0.0615	0.8596	0.0242
Clustered $\tau = 0.4$	0.8723	0.2402	0.8625	0.0234	0.9149	0.0465	0.6555	0.0139

Table 5.1: The first row in the table contains results of matching descriptors extracted from the training data set (MRI image database) to the descriptors extracted from the evaluation data set. The second row shows the matching results for structure models obtained from descriptor clustering. The results are provided separately for each anatomical structure as true positives (tp) and false positives (fp).

from shape clustering. The structure models are then evaluated using test images that have also been annotated. The descriptors from test images are matched against structure models – each descriptor is classified as the category that corresponds to the category of the structure model exhibiting the minimum matching distance to the descriptor. Table 5.1 shows the results of the evaluation in the form of ratio of true positives (tp) to all points belonging to each category and ratio of false positives (fp) to all descriptors.

The results in Table 5.1 show that clustering improves detection accuracy (vs. un-

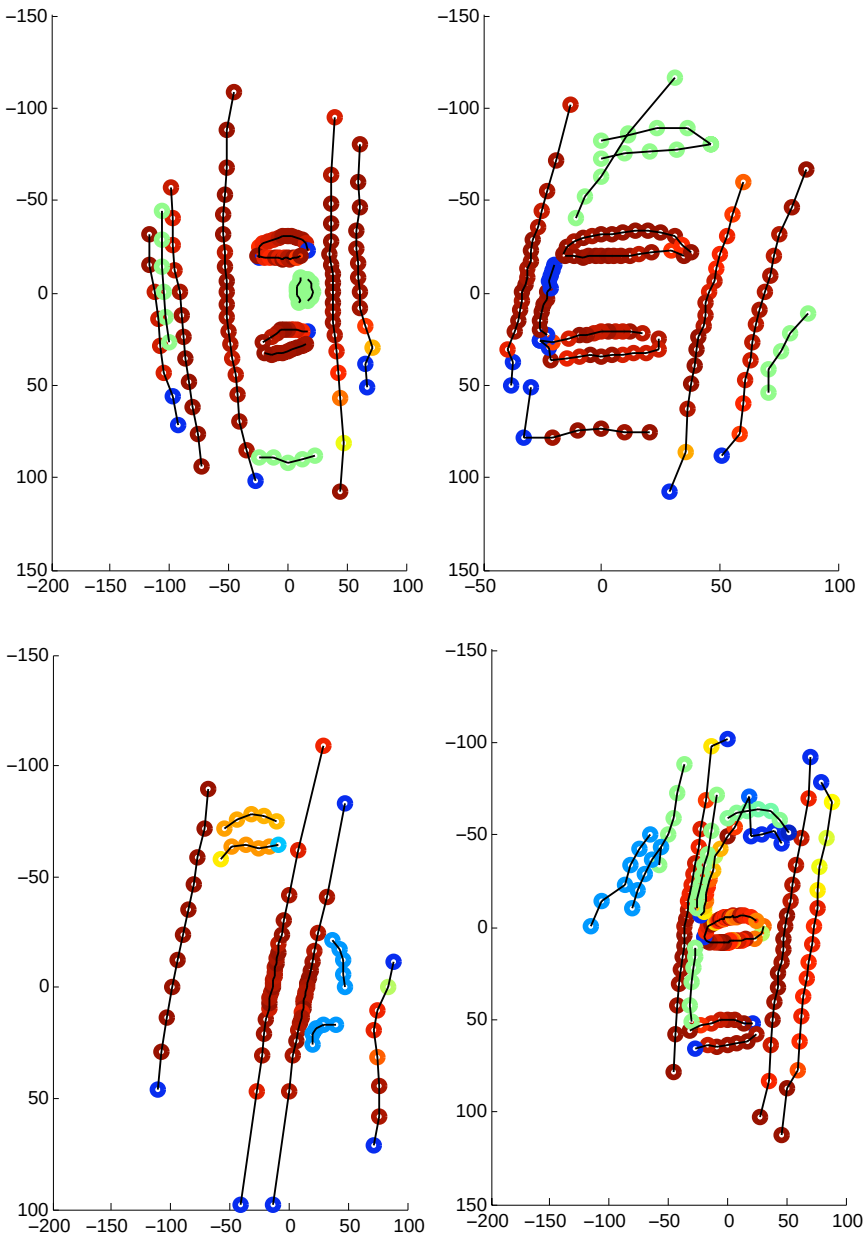


Figure 5.2: Example of weights assigned to boundary points in the process of shape clustering. The top row contains two examples of vertebrae structures while the bottom row contains models of spinal cord and the disk. The weight values represented by colors increase from blue to red.

clustered models) for all categories except the background e.g. the models of vertebrae obtained from descriptor clustering are correctly matched to 87% vertebrae related de-

scriptors in the evaluation data set while without clustering only 68% of descriptors are correctly matched. The clustered descriptors have weights assigned to the encoded edges that describe repeatability of them among examples in the training data set while these weights are set to 1 in unclustered descriptors. This explains why repeatable structures such as vertebrae, disc and spine are better detected by structure models obtained from descriptor clustering. Background detection however shows the opposite trend due to higher variability of background related structures than in the case of other categories e.g. compare the structures behind the spine in examples in Figure 5.2. The improvement of background matching is possible either by using training data set that contains majority of structures occurring in the test data set or by learning and detecting spatial relationship between detected structures (e.g. [10]).

5.3 Unsupervised Model Extraction in Hand X-Ray Images

Our testing strategy is focused on investigating the discriminative power of the REC descriptor. Tests are conducted on X-Ray images of human hands, and we want to assess how well the REC descriptor discriminates between four categories related to different finger bone types plus one background category. Figure 5.3 shows an example of the bone annotation in a human hand X-Ray image and a selection of images from the image test database. To this end, descriptors extracted from the training set (10 images) are clustered and the resulting clusters are matched against descriptors from the test set (20 images) with assigned category labels (obtained by manual annotation of a test set). The clustering process is expected to create consistent representations of similar shapes (shape alphabet) from the training data set (which also correspond to similar interest point locations). As clusters are themselves represented by single RECs, one can assess their representative quality by simply comparing them to labeled descriptors (see Section 5.1). We consider a shape cluster as highly consistent when it exhibits a majority of closest matches to descriptors stemming from a single category. However, the primary difficulty related to category discrimination is associated with the choice of categories and the fact that different types of bones (see category 4 and 3) are similar (especially on a local scale). Also taking into account inconsistency in interest point distribution and edge fragmentation, one can not expect that all clusters will be representative for one distinguished category. Therefore, we divide clusters in two groups: a “highly informative” group, with more than 50% matches within a single category and an “uninformative group” with the remaining clusters.

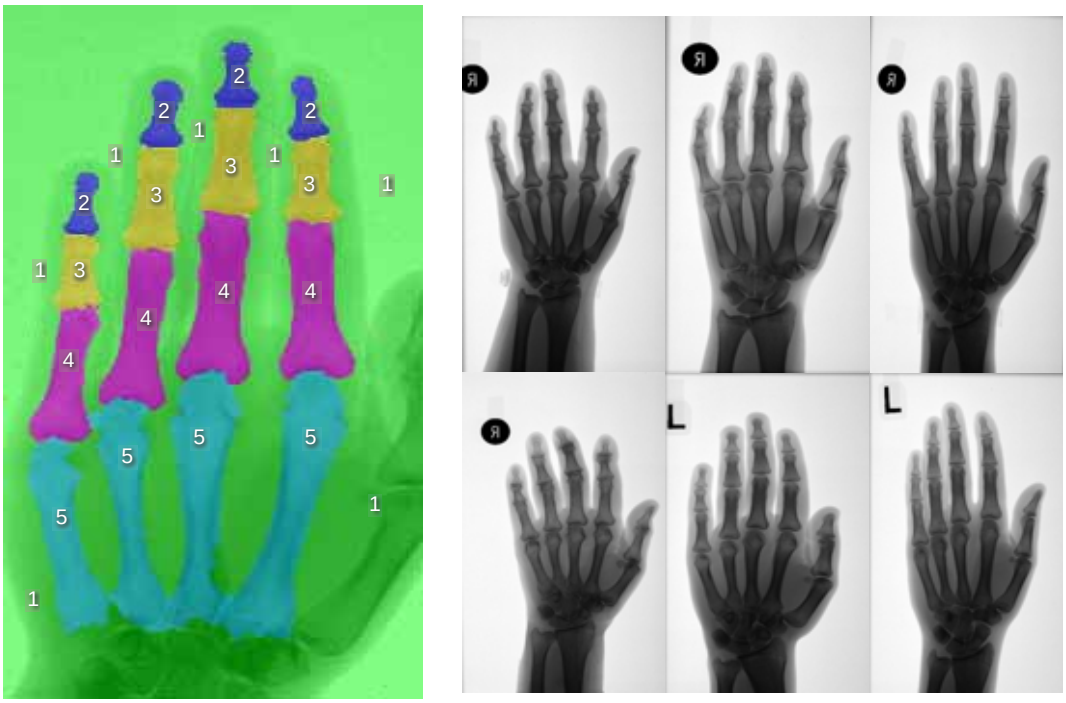


Figure 5.3: Left: Example of hand X-Ray annotation. The categories represent 5 perceptible shapes, plus background. Note that this coarse annotation is not used during the training, but only during evaluation. The annotation defines the association of interest points with shape categories. Right: A subset of images used for clustering and testing. The shape and size of the bones varies between different images.

Another problem is the choice of the number of clusters that produces the highest number of "highly informative" models. Since each category covers four similarly shaped bones repeated in 10 training images, we should expect approximately 40 similar shapes per specific relative interest point location within a category. This corresponds to clustering level 5 and 6 where each clustering node corresponds to maximally 32 or 64 original REC descriptors merged together. We have chosen the 5-th clustering level, which results in over 600 clusters out of the original approximately 5000 interest points.

Table 5.2 shows the average percentage of "highly informative" cluster matches in each category in the first column. The second column shows the percentage of interest points in each category, which correspond to the "highly informative" cluster matches.

The positive aspect revealed during these tests was the consistency of highly informative clusters. with approximately 80% of the cluster matches within a single corresponding category each. However, we found that in several cases less than 50% of interest points

Category	Avg. cluster matches within same (correct) category [%]	Amount of descriptors matched to highly informative clusters [%]
1 - background	89.4	81.8
2 - Distal phalanges	68.6	33.6
3 - Intermediate phalanges	79.2	41.3
4 - Proximal phalanges	75.7	46.2
5 - Metacarpals	86.2	57.4

Table 5.2: The cluster matching statistics for the database of hand X-Ray images corresponding to the highly informative group of clusters. The second column can be interpreted as the probability of detecting the corresponding category when the related cluster has been matched to the descriptor associated with the interest point.

were matched against those clusters, which can be attributed to the following factors:

- The number of descriptors in each category differ e.g. category 2 accounts for only 2% of all interest points, while categories 3-5 correspond to 6, 9 and 23% of all interest points respectively. This negatively affects the clustering stage; clusters related to lowly populated categories are more likely to be merged with the clusters corresponding to highly populated categories. This is partially prevented by the use of distance threshold τ.
- Deviations in interest point positions increased the number of clusters needed to describe image structures related to the same category.
- Local shape similarity between categories – the central locations inside elongated bones, which account for the majority of interest points are similar to each other across multiple categories. This is one of the primary reasons why only 30-40% of all clusters are highly discriminative for categories 2 and 4.

5.4 Conclusions

We have presented a method for clustering shapes that uses an edge based semi-local shape descriptor (REC) together with a robust scale and rotation invariant distance measure. This allows us to perform clustering of the descriptors in order to obtain a consistent representation of similar image structures.

The two test scenarios presented in Sections 5.2 and 5.3 show the applicability of the REC descriptor to detection of image structures in medical images. The MRI images used for supervised learning of characteristic anatomical structures as well as hand X-Rays

used for unsupervised extraction of bone models contain structures that differ in scale and orientation while edge detection performed on these images produces fragmented structure boundaries due to low image contrast and noise (see examples in Figures 4.18 and 4.24). Despite these problems and the inconsistency of interest point detection the supervised learning of anatomical structures in MRI images produced structure models that were able to correctly detect more than 80% of corresponding structures in the validation data set.

Future research will concentrate on the replacement of symmetry based interest points with edge key points corresponding to high curvature locations along detected edges. These key points are significantly less exposed to the interest point drift affecting symmetry interest points and blob detectors, as discussed in Chapter 2. An additional advantage of using these key-points is their ability to estimate the descriptor orientation from a local edge orientation, thereby reducing the search for relative orientation between two descriptors and overall computational complexity. Finally the descriptor distance will be altered to make it affine invariant with the ability to control the amount of affine transformation allowed.

Chapter 6

Conclusion

6.1 Object Recognition Considerations

In this thesis we have presented several distinctive method classes to tackle the object recognition problem, based on matching local image structures, extraction of features from image segments and shape matching. The presented results as well as reviewed literature clearly indicate that neither of the method classes alone is sufficient to provide a general solution. As an evidence of this statement let us consider an approach presented in [52] (see also Section 3.1.4), where the use of local descriptor SIFT achieves remarkable accuracy in finding different views of the same urban scenes within large image database. This is possible because local image patches and their local spatial configurations provide sufficiently discriminative features. However, the same local descriptor, applied to the matching of local structures in the image database containing animals (see Section 4.2.10) reveals less promising performance. In case of many objects shape is more discriminative than local appearance. Neither shape nor local features provide sufficient means to learn and recognize any arbitrary set of objects. It is clear that better results can be achieved only when one considers a combination of features related to texture, local appearance, shape and possibly global features.

6.2 Contributions

The long term goal of the thesis author is an object recognition using multiple cues. This thesis however concentrates rather on details of various feature type extraction and matching that are crucial components of many object recognition methods.

Our primary contributions are rotation and scale invariant shape based detector (Chapter 4, Section 4.3) and a method to learn repeatable shape structures in the training images with and without supervision (Chapter 5). Achieving rotational and scale invariance in shape detectors is a challenging task as none of the approaches described in Chapter 3 fully offer both properties simultaneously. These approaches extract and learn shape features, such as image edge fragments, that are not scale nor rotational invariant. These features and their geometric relationships are directly learned from training images with the expectation that objects occur at similar scale and orientation. A typical technique for obtaining invariant features, e.g. used for extraction of local features, is prior estimation of orientation, scale or affine regions in the image (see Chapter 2). However that approach has several drawbacks, the reliable "apriori" scale estimation is an extremely difficult task (see Section 2.3), orientation estimation of structures in images works well for simple elongated structures but round or more complex structures may not have a dominant orientation. **We propose a different philosophy: instead estimating "apriori" scale or orientation during feature extraction, we detect relative scale and rotation during feature matching** (Chapter 4). This difference allows the construction rotation and scale invariant matching applicable for shape based features. Detection of relative orientation and scale allows us to build a shape clustering approach (Chapter 5), that can extract a repeatable image structures which undergo scaling and rotation in both training and test images.

6.3 Outlook

The future work will be initially concentrated on several improvements to the shape detector presented in Section 4.3 and the shape clustering presented in Chapter 5. The current dependency on symmetry based interest points decreases matching accuracy due to inconsistency in the interest point positioning. These points can be replaced with the key points corresponding to curvature maxima along the detected edges which are also used for feature extraction. This solution has two advantages: much better point position stability and possibility of using local edge orientation for more reliable and more efficient estimation of relative rotation during feature matching. The matching of shape descriptors will include the measure of spatial similarity between correspondences of neighboring descriptors, which can be also viewed as matching constellations of neighboring descriptors. Finally the cluster coherency will be used as a measure of shape variation within a cluster which will help to to stop clustering of individual shapes when their variability reaches

predefined threshold.

All the planned improvements are a prerequisite to building a shape and part based object recognition approach. The long term plan assumes also introduction of local features into the object recognition approach.

Appendix A

Acronyms and Symbols

List of Acronyms

LoG	Laplacian of Gaussian
DoG	Difference of Gaussian
DoH	Determinant of Hessian
DoHA	Affine Determinant of Hessian
RST	Radial Symmetry Transform
FRST	Fast Radial Symmetry Transform
PCA	principal component analysis
PDF	probability density function
ROI	region of interest
SIFT	Scale Invariant Feature Transform
ORC	Orientation-invariant Radial Configuration
REC	Radial Edge Configuration

List of Symbols

F_x	Derivative with respect to x
j	Imaginary number, $\sqrt{-1}$
δ	Dirac delta function
$\delta_\triangle$	Kroneker delta function
$\mathrm{x} * \mathrm{y}$	Convolution
x^*	Conjugate complex
∇	Nabla operator
$\triangle$	Laplace operator
$\mathbf{I}$	Identity matrix
$\mathcal{N}(\mu, \sigma)$	Normal distribution

Bibliography

[1] Amores, J., Sebe, N., and Radeva, P. (2007). Context-based object-class recognition and retrieval by generalized correlograms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1818–1833.

[2] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522.

[3] Borgefors, G. (1986). Distance transformations in digital images. *Comput. Vision Graph. Image Process.*, 34(3):344–371.

[4] Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698.

[5] Carmichael, O. and Hebert, M. (2004). Shape-based recognition of wiry objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(12):1537–1552.

[6] Carneiro, G. and Lowe, D. (2006). Sparse flexible models of local features. In *Proceedings of ECCV'06*, volume 3, pages 29–43.

[7] Chui, H. and Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.*, 89(2-3):114–141.

[8] Comanicu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 24, pages 603–619.

[9] Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graha, J. (1995). Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

[10] Donner, R., Micusik, B., Langs, G., Szumilas, L., Peloschek, P., Friedrich, K., and Bischof, H. (2007). Object localization based on markov random fields and symmetry interest points. In *Proceedings of MICCAI'07*, volume 2, pages 460–468.

[11] Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience.

[12] Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. volume 2, pages II–264–II–271 vol.2.

[13] Ferrari, V., Jurie, F., and Schmid, C. (2007). Accurate object detection with deformable shape models learnt from images. In *Proceedings of IEEE CVPR'07*, pages 1–8.

[14] Forsyth, D. A. and Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference.

[15] Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906.

[16] Geusebroek, J.-M., van den Boomgaard, R., Smeulders, A. W. M., and Gevers, T. (2003). Color constancy from physical principles. *Pattern Recogn. Lett.*, 24(11):1653–1662.

[17] Hall, D., Leibe, B., and Schiele, B. (2002). Saliency of interest points under scale changes. In *Proceedings of BMVC'02*, page Poster Session, Cardiff, UK.

[18] Harris, C. and Stephens, M. (1988). A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151.

[19] Heath, M., Sarkar, S., Sanocki, T., and Bowyer, K. (1997). A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(12):1338–1359.

[20] Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

[21] Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):4–37.

[22] Jurie, F. and Schmid, C. (2004). Scale-invariant shape features for recognition of object categories. In *International Conference on Computer Vision & Pattern Recognition*, volume II, pages 90–96.

[23] Kadir, T., Zisserman, A., and Brady, M. (2004). An affine invariant salient region detector. In *Proceedings of ECCV'04*, volume 1, pages 228–241.

[24] Ke, Y. and Sukthankar, R. (2004). Pca-sift: a more distinctive representation for local image descriptors. In *Proceedings CVPR'04*, volume 2, pages 506–513.

[25] Koenderink, J. (1984). The structure of images. *Biol. Cybernetics*, 50(5):363–370.

[26] Koenderink, J. J. and van Doom, A. J. (1987). Representation of local geometry in the visual system. *Biol. Cybern.*, 55(6):367–375.

[27] Kosslyn, S. M. (1990). Mental imagery. In Osherson, D. N., Kosslyn, S. M., and Hollerbach, J. M., editors, *Visual Cognition and Action: An Invitation to Cognitive Science (Volume 2)*, pages 73–97. MIT Press, Cambridge, MA.

[28] Lamdan, Y., Schwartz, J., and Wolfson, H. (1988). On recognition of 3d objects from 2d images. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1407–1413.

[29] Leibe, B., Cornelis, N., Cornelis, K., and Gool, L. J. V. (2007). Dynamic 3d scene analysis from a moving vehicle. In *Proceedings of CVPR'07*, pages 1–8.

[30] Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, Czech Republic.

[31] Leitner, R. and Bischof, H. (2005). Recognition of 3d objects by learning from correspondences in a sequence of unlabeled training images. In *Proceedings of DAGM'05*, volume 3663, pages 369–376.

[32] Leordeanu, M. and Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *Proceedings of ICCV'05*, volume 2, pages 1482–1489.

[33] Leordeanu, M., Hebert, M., and Sukthankar, R. (2007). Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Proceedings of CVPR'07*, pages 1–8.

[34] Li, W., Zhang, A., and Kleeman, L. (2005). Fast global reflectional symmetry detection for robotic grasping and visual tracking. In *Proceedings of ACRA'05*.

[35] Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *J. of Applied Statistics*, 21(2):224–270. (Supplement on Advances in Applied Statistics: Statistics and Images: 2).

[36] Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116.

[37] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157.

[38] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110.

[39] Loy, G. and Zelinsky, A. (2002). A fast radial symmetry transform for detecting points of interest. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 358–368, London, UK. Springer-Verlag.

[40] Loy, G. and Zelinsky, A. (2003). Fast radial symmetry for detecting points of interest. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(8):959–973.

[41] Matas, J., Chum, O., Martin, U., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, London.

[42] Mikolajczyk, K. and Schmid, C. (2004a). Comparison of affine-invariant local detectors and descriptors. In *Proceedings ESPC'04*.

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

[44] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72.

[45] Moravec, H. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. In *tech. report CMU-RI-TR-80-03 and doctoral dissertation, Robotics Institute, Carnegie Mellon University, Stanford University.* Available as Stanford AIM-340, CS-80-813 and republished as a Carnegie Mellon University Robotics Institue Technical Report to increase availability.

[46] Mortensen, E. N., Deng, H., and Shapiro, L. (2005). A SIFT descriptor with global context. In *Proceedings of the CVPR*, volume 1, pages 184–190.

[47] Mundy, J. (2006). *Object Recognition in the Geometric Era: A Retrospective*, volume 1, chapter 1, pages 3–28.

[48] Murase, H. and Nayar, S. K. (1995). Visual learning and recognition of 3-d objects from appearance. *Int. J. Comput. Vision*, 14(1):5–24.

[49] Ommer, B. and Buhmann, J. M. (2007). Learning the compositional nature of visual objects. In *Proceedings of CVPR'07*, pages 1–8.

[50] Opelt, A., Pinz, A., and Zisserman, A. (2006a). Fusing shape and appearance information for object category detection. In *Proceedings of the BMVC'06*, volume 1, pages 117–123.

[51] Opelt, A., Pinz, A., and Zisserman, A. (2006b). Incremental learning of object detectors using a visual shape alphabet. In *Proceedings of the CVPR'06*, volume 1, pages 3–10.

[52] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of CVPR'07*, pages 1–8.

[53] Ponce, J., Hebert, M., Schmid, C., and Zisserman, A., editors (2006). *Toward Category-Level Object Recognition*. Lecture Notes in Computer Science, volume 4170, Springer.

[54] Quinlan, R. J. (1993). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann.

[55] Reisfeld, D., Wolfson, H., and Yeshurun, Y. (1994). Context free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision, Special Issue on Qualitative Vision*, 14(2):119–130.

[56] Schaffalitzky, F. and Zisserman, A. (2002). Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 1, pages 414–431.

[57] Schiele, B. and Crowley, J. L. (2000). Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50.

[58] Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172.

[59] Schwartz, J. T. and Sharir, M. (1987). Identification of partially obscured objects in two and three dimensions by matching noisy characteristic. *Int. J. Rob. Res.*, 6(2):29–44.

[60] Sebe, N. and Lew, M. (2001). Salient points for content-based retrieval. In *Proceedings of BMVC'01*, pages 401–410.

[61] Steger, C. (1998). An unbiased detector of curvilinear structures. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(2):113–125.

[62] Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1):11–32.

[63] Szumilas, L., Donner, R., Langs, G., and Hanbury, A. (2007a). Detection of local image structures with orientation-invariant radial configuration. In *Proceedings of CVPR'07*, pages 1–8. IEEE Computer Society.

[64] Szumilas, L. and Hanbury, A. (2006). Color pair clustering for texture detection. In *ISVC (2)*, pages 255–264.

[65] Szumilas, L., Mičušikk, B., and Hanbury, A. (2006). Texture segmentation through salient texture patches. In *Proceedings of CVWW'06*, pages 111–116.

[66] Szumilas, L., Wildenauer, H., Hanbury, A., and Donner, R. (2007b). Radial edge configuration for semi-local image structure description. In LNCS, editor, *ISVC 2007*, volume 4841, pages 633–643.

[67] Thodberg, H. and Rosholm, A. (2001). Application of the active shape model in a commercial medicals device for bone densitometry. In *BMVC*, pages 43–52.

[68] Torralba, A., Murphy, K. P., and Freeman, W. T. (2004). Sharing features: Efficient boosting procedures for multiclass object detection. In *Proceedings CVPR'04*, volume 02, pages 762–769.

[69] Turk, M. and Pentland, A. (1991). Face recognition using eigenfaces. *Proceedings of CVPR '91*, pages 586–591.

[70] Tuytelaars, T. and van Gool, L. (2003). Matching widely separated views based on affine invariant regions. *IJCV*, 59(1):61–85.

[71] Van Gool, L. J., Moons, T., and Ungureanu, D. (1996). Affine/ photometric invariants for planar intensity patterns. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, pages 642–651, London, UK. Springer-Verlag.

[72] Wang, H. and Brady, M. (1995). Real-time corner detection algorithm for motion estimation. *Image Vision Comput.*, 13(9):695–703.

[73] Witkin, A. P. (1983). Scale-space filtering. In *8th Int. Joint Conf. Artificial Intelligence*, volume 2, pages 1019–1022.

[74] Zhang, J., Marszałek, M., Lazebnik, S., and Schmid, C. (2007). Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238.