AN ABSTRACT OF THE THESIS OF

Hongli Deng for the degree of Doctor of Philosophy in Computer Science presented on July 9, 2007.

Title: Image Feature Detection and Matching for Biological Object Recognition

Abstract approved: _____

Eric Mortensen

Image feature detection and matching are two critical processes for many computer vision tasks. Currently, intensity-based local interest region detectors and local feature-based matching methods are used widely in computer vision applications. But in some applications, such as biological object recognition tasks, within-class changes in pose, lighting, color, and texture can cause considerable variation of local intensity. Consequently, object recognition systems based on intensity-based interest region detectors often fail. This dissertation proposes a new structure-based local interest region detector called principal curvature-based region detector (PCBR) that detects stable watershed regions within the multi-scale principal curvature images. This detector typically detects distinctive patterns distributed evenly on the objects and it shows significant robustness to local intensity perturbation and intra-class variation. Second, this thesis develops a local feature matching algorithm that augments the SIFT descriptor with a global context feature vector containing curvilinear shape information from a much larger neighborhood to resolve ambiguity in matching. Moreover, this thesis further improves the matching method to make it robust to occlusion, clutter, and non-rigid transformation by defining affine-invariant log-polar elliptical con-

text and employing a reinforcement matching scheme. Results show that our new detector and matching algorithms improve recognition accuracy and are well suited for biological object recognition tasks.

Image Feature Detection and Matching for Biological Object Recognition

by

Hongli Deng

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented July 9, 2007
Commencement June 2008

Doctor of Philosophy thesis of Hongli Deng presented on July 9, 2007

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Hongli Deng, Author

ACKNOWLEDGMENTS

First of all, I would like to express my gratefulness to my advisor, Professor Eric Mortensen, whose support, encouragement and wisdom have helped me through my Ph.D. study. His rigorous attitude towards research and his kindness towards people set a good example for me. I have been truly honored to work with him.

I would also like to thank Professor Thomas Dietterich, who has always sparked me with inspiring ideas and has broadened my views of machine learning. He is one of the greatest researchers I have ever met. He led the NSF project upon which I finished this work and from which I received financial support.

I would like to thank Professor Linda Shapiro from University of Washington. She is a top expert in computer vision. The initial idea of the curvilinear-based detection related to this thesis came from her.

I would like to thank all my colleges, Madhu Srinivasan, Guoning Cheng, Wei Zhang etc. Studying and having discussion with them bring me knowledge and happiness as well.

I would give my deepest gratitude to my parents for their love and unconditional support. All my achievements would be impossible without their love. Most of all, I am utterly grateful to my wife, Zijuan, for her love, support and always being by my side.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

LIST OF TABLES

Image Feature Detection and Matching for Biological Object Recognition

1. INTRODUCTION

1.1. Background and Motivation

Environmental health can be measured by the population of insects and biodiversity, but these tasks are hampered by the lack of a way to easily measure them. Insect classification is difficult and expensive because there are very few entomologists who have the knowledge to classify them and fewer than a dozen taxonomic specialists in all of north America with the expertise to perform all of these analyses. "BUGID" is a system built to collect, manipulate, photograph and identify insects, very quickly, accurately and in large numbers. It brings advanced machine learning and computer vision techniques to the world of ecology and environmental protection. The work presented in this thesis are part of achievements of the "BUGID" project. The insects that we are trying to identify are stonefly larvae, which are known to be a sensitive indicator of stream health and water quality.

In terms of computer vision, this task expands the frontier of the traditional object recognition or pattern classification domain. Unlike a normal object recognition task such as identifying a car, insect identification is more complicated. Insects come in many shapes, sizes, colors and configurations, and there are a lot of them. For example, in Oregon, a square meter of soil can range from 100 to 300,000 individuals representing 2 to 200 species. Some of these insects

are quite different whereas some of them are difficult to tell apart even though they are different species. One distinctive challenge of this system is to exploit nuance between categories (e.g., taxa) while eliminating large variability within categories. Figure 1.1 shows two species of stoneflies (Calinueria and Doroneuria) that demonstrate large within-class variation and small between-class differences. Correct classification of these species is a difficult job even for human.

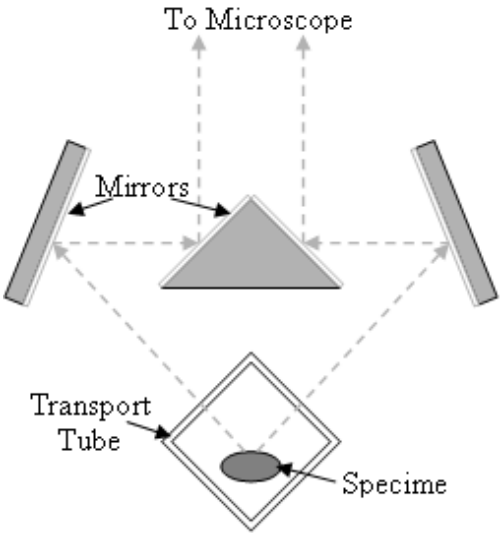
The first step of an object recognition system is to capture consistent, clear images. We have designed and constructed a software-controlled mechanical stonefly larvae transportation and imaging apparatus that positions specimens under a microscope, rotates them (to obtain views from various angles), and photographs them with a digital camera. Using this apparatus, imaging rates of a few tens of specimens per hour can be achieved. A minimum of eight images (from different viewing angles) are taken for each specimen. The imaging apparatus has a series of mirrors so that each image acquires two simultaneous views of a specimen from approximately 90 degrees apart. Light diffusers reduce glare and eliminate hard shadows. In summary, the apparatus can quickly acquire several images of a specimen from various angles with consistent imaging conditions across specimens and species. Figure 1.2 shows the imaging apparatus, including the mirror setup used for acquiring two simultaneous images of each specimen.

The second step of a object recognition system is feature detection and description. Feature detectors locate interest feature regions on images. These regions can be defined by shapes, textures, intensities etc., and come in global or local forms. Currently, local feature-based region detectors show their advantages for their robustness to occlusion and image deformation. They detect distributed interest image regions that provide a distinctive, compact representation of the image or object. Ideal interest regions should be informative, repeatably detectable

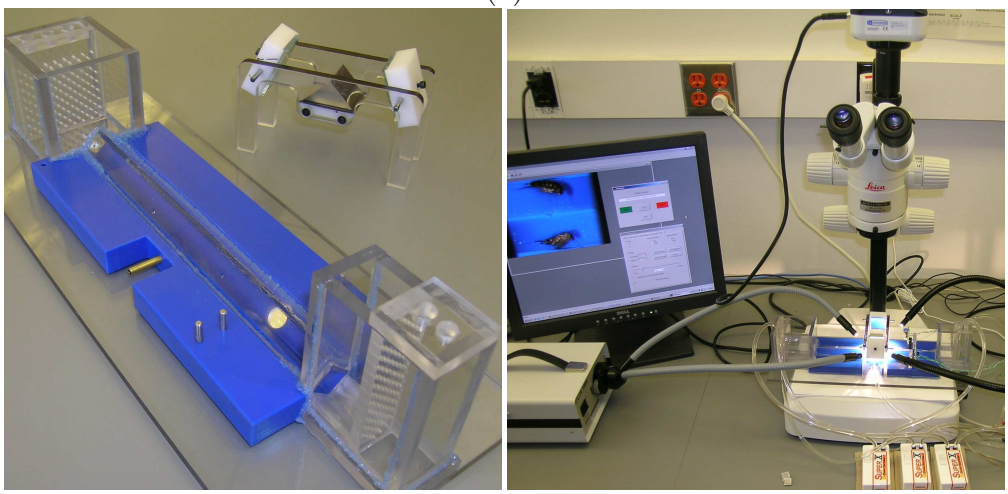
(a)

(b)

FIGURE 1.1. Example images of stonefly specimens taken with our imaging apparatus. (a) The original images. (b) A close look. (left column: *Calinueria californica*. right column: *Doroneuria baumanni*.)



(a)



(*b*) (*c*)

FIGURE 1.2. Transportation and imaging apparatus for stonefly larvae. (a). Diagram of mirror system for obtaining two simultaneous views of a specimen (from approx. 90° apart) in a single image. (b). Image of prototype mirror and transportation apparatus. (c). Image of entire stonefly transportation and imaging setup (with microscope and attached digital camera, light boxes, and computer controlled pumps for transporting and rotating the specimen.

and invariant to deformation. Detected interest regions can be directly used in classifiers using all pixel values in these regions or described by feature descriptors. Using all pixel values can be redundant or noisy while feature descriptors extract, filter and summarize the most distinctive features and build compact feature vectors. These descriptors can greatly facilitate classifiers with their compactness and discrimination.

For classification, images are divided into training images and testing images. All training images contain a label indicating their object category. One or more feature detectors and descriptors are applied to all training images. The resulting feature vectors are analyzed by various machine learning techniques. Based on these training images, a classifier is built and is then applied to test images. Given a test image, the classifier will generate a label indicating the object category that the image contains.

Feature detection and matching are two key steps in this classification process, and they are the main topics of this thesis. Feature detection and matching are important not only for this system, they are essential for many computer vision applications. As noted earlier, feature detection tries to identify the characteristic regions that form a good representation of the object. They should be discriminative across the object category while tolerant of variations within the category. Feature matching compares two set of detected features and determines their similarities. This process is also known as the correspondence problem. Unfortunately, local feature matching is often ambiguous. Choosing correctly matched pairs (inliers) and rejecting false matches (outliers) is called outlier rejection. For local feature-based matching, outlier rejection is inevitable because local features contain limited context. Other constraints such as spatial relations, epipolar constraints etc. must be added to resolve these ambiguities.

In terms of our "BUGID" project, automated recognition of stoneflies raises many fundamental challenges for feature detection and matching. Stonefly larvae are highly-articulated objects with many sub-parts (legs, antennae, tails, wing pads, etc.) and many degrees of freedom. Some taxa exhibit interesting patterns on their dorsal sides, but others are not patterned. Some taxa are distinctive whereas others are very difficult to identify. Finally, as the larvae repeatedly molt, their size and color change. Immediately after molting, they are light colored, and then they gradually darken. This variation in size, color, and pose provides considerable challenges for previous image feature detection and matching methods. We found in our previous experiments that existing region detectors and matching methods did not work very well on this dataset. We needed a new image feature detector that can handle significant variation and a new matching method that can resolve extensive ambiguities. This thesis is dedicated to solving these problems.

1.2. Contributions

This thesis makes two contributions. First, it proposes a new interest region detector—the principal curvature-based region detector (PCBR) for object recognition. Second, it proposes two new algorithms for feature matching by augmenting local image feature matching with global context information.

Compared with previous detectors, our new detector PCBR has the following improvements:

- PCBR is a new structure-based detector and it complements intensity-based detectors. PCBR uses principle curvature as a detection cue and is appropriate for objects whose shape patterns are more stable. It can handle

variations in local intensity values. Insects in our "BUGID" project and many other biological objects are in this category with variable local intensity. These insect images have huge variations on local appearance, and are highly articulated. Therefore, this detector is designed to explore the domain where previous detectors have failed and where a new detector is needed.

- PCBR improves previous structure-based detectors in several respects. First, it can handle both edges and curvilinear structures. Curvilinear structures are lines (either curved or straight) such as roads in aerial or satellite images or blood vessels in medical scans. Previous structure-based detectors only use edges as detection cues. PCBR uses both edges and curvilinear structures. Second, it provides cleaner regions than previous structure-based detectors. Previous structure-based detectors that used edges can not handle curvilinear structures very well. The gradient responses used by edge detection algorithms will generate two responses on both sides of a curvilinear structure. This makes the resulting image sketches complicated and regions fragmented. PCBR uses principal curvature response and only generates one response regardless if it is an edge or a curvilinear structure. This single response forms cleaner image sketches and regions. Third, PCBR uses an enhanced watershed segmentation algorithm to define regions, which is more efficient than circle or ellipse fitting used by previous structure-based detectors. As a result, previous structure-based detectors have to limit their detections to only scale invariant (SISF [37]) or some fixed points (EBR [76]) to decrease the search space. On the other hand, the enhanced watershed algorithm used by PCBR detects affine-invariant regions.

- PCBR achieves robust detections. We adopt two methods to improve the robustness of the watershed algorithm and the stability of detections. The first method is the new "eigenvector-flow" hysteresis thresholding. We find that the direction of the eigenvectors of Hessian matrix provides a strong indication of where curvilinear structures appear. Moreover they are more robust to intensity perturbation than is the eigenvalue magnitude. Therefore, directions of eigenvectors are used to help link image structures. The second method seeks stability over scales. We employ a similar scale space as SIFT [47] in the detection. Only regions that can be detected repeatedly in local consecutive scales are chosen as detected regions.

- PCBR is well suited for biological object recognition tasks. Biological objects such as insects or humans normally show apperance variation on different stages of life. PCBR typically detects distinctive structural patterns distributed evenly on the objects and these structural patterns are more robust detection cues than local intensity. Therefore, PCBR achieves significant robustness to local intensity perturbation and intra-class variation. We evaluate PCBR using various practical classification systems. Results show this detector outperforms other detectors on many biological object recognition tasks.

Our second contribution is two new algorithms on feature matching. These two new algorithms reject outliers effectively by including global context information into local feature matching. In the first algorithm, we enhance the SIFT descriptor [47] with shape context [5]. A bigger circular context bin constructed around every local feature summarizes curvilinear values in each bin. These summarized bin values build a global context feature vector which is attached to

the original SIFT descriptor. The new descriptor adds global curvilinear shape information from a much larger neighborhood, thus reducing mismatches when multiple local descriptors are similar. However we found that circular bins built in the first algorithm will cover different areas on two images with affine transformation. Further, this first algorithm is not robust to occlusion because it uses summarized bin values. If the bin is partially occluded, the summarized bin value will change, thus introducing errors to the feature vector. So, we designed the second matching algorithm to make it robust to affine transformation and occlusion. The new algorithm uses affine-invariant log-polar elliptical bins and employs a reinforcement matching scheme using distributed local regions. It rejects outliers effectively without losing the advantages of local feature-based matching since it is still robust to occlusion, cluttered backgrounds, and deformation. The two algorithms are tested on a standard test set [52] and results show they are comparable to or outperform other state-of-the-art outlier rejection methods (RANSAC and PROSAC). The work presented in this thesis is published in [59, 22, 23, 84, 60, 61].

1.3. Thesis Outline

A literature review of local feature detectors and feature matching methods is presented in Chapter 2. Chapter 3 describes our new principal curvature-based region operator. Chapter 4 evaluates the detector on repeatability using the published framework from INRIA, and information contents using maximum mutual information criteria. Chapter 5 tests the detector on various practical applications and demonstrates improved results. Beginning with Chapter 6, we move from feature detection to feature matching. A new outlier rejection method using

global context-enhanced SIFT descriptor is presented in this Chapter. Chapter 7 further improves the method in Chapter 6 to make it robust to occlusion, clutter background and image deformation. A comparison between the two new methods and another state-of-the-art outlier rejection method called PROSAC [16] is presented. Finally, this thesis concludes with my recommendations for future research directions in Chapter 8.

2. LITERATURE REVIEW

2.1. Overview

Feature detection and matching are two important tasks for many computer vision applications. Detection involves finding interest regions or points. Matching determines correspondence among features in different images.

Raw pixel values are rarely used directly in computer vision applications. Instead, image features extracted from these pixel values are widely used. These extracted image features are abstract representations of images and are normally more robust and easier to process than raw pixel values. Image features can be defined by using different detection cues, such as shape, texture and color etc. The cues used depends on the intrinsic characteristics of the images and applications. For example, shapes are more appropriate for representing structure-rich scenes while hair are better represented by their texture. Features can be local or global. Currently, local feature detectors are prevalent for their robustness to deformation and invariance to viewing condition change. Local feature detectors are divided into intensity-based and structure-based region detectors based on their detection cues. Intensity-based detectors depend on analyzing local intensity patterns to find regions that satisfy some uniqueness or stability criteria. Structure-based region detectors detect image structures such as edges, lines, etc.. These image structures are further analyzed in two dimensions to find corners, bumps, etc. or fitted with circles or ellipses to define regions. Performances of local feature detectors are evaluated through repeatability, detectability, distinctiveness, and robustness etc. Repeatability shows the portion of features that can be repeatedly detected on two different images of the same scene under different viewing conditions. Detectability evaluates the ratio between detected features and all

detectable features in an image. Distinctiveness calculates the similarity between a detected feature and other features to see how much ambiguity exist in these detections. Robustness deals with the stability of detectors in terms of image intensity variation and spatial deformation.

Feature matching finds corresponding features. Some of these matching methods find correspondence by comparing the similarity of features themselves. Other matching methods exploit the spatial constraints of features. Current research of feature matching focuses on how to robustly match features with appearance variation and spatial non-rigid deformation and how to resolve ambiguity in local feature matching.

2.2. Global Features vs Local Features

In terms of spatial extension, features can be local or global. Global features cover the whole image or a big portion of it. They are discriminative enough to represent the whole image. However, global features are not robust to clutter background, occlusion and deformation. In comparison, local features are easier to extract, more robust to noise, occlusion and clutter background. Unfortunately, due to the limited spatial extent of local features, ambiguities exist in matching due to a lack of context. As such, outlier rejection methods are needed to match these local features.

2.2.1. Global Feature

Some early computer vision applications directly used gray scale values of the image. The similarity between two images is calculated as the distance

between the pixel values of the two images. For example, image correlation calculates the Euclidean distance between two images. This type of application requires that images must be taken in a clear background; objects are well-aligned with no occlusion and no deformation. These conditions are hard to satisfy in practice, especially in object category recognition where objects come with many shapes and appearances. The global matching method can only be applied to matching the same object with similar viewing conditions.

To improve robustness to image variation, statistical methods can be used. Eigenfaces [75] are a set of "standardized face ingredients", derived from statistical analysis of many pictures of faces. Any human face can be considered to be a combination of these standard faces. To generate a set of eigenfaces, images with dimension $m * n$ are treated as mn-dimensional vectors whose components are the values of their pixels. The eigenvectors of the covariance matrix of the statistical distribution of these mn-dimensional vectors are then extracted. These eigenvectors are eigenfaces. The eigenfaces method generates compressed representations of images and makes the representation less sensitive to noise. For the eigenfaces method to work, the images need to be aligned before processing. However, this method is not robust to deformation, occlusion and background clutter.

In order to eliminate the influence of partial occlusion and background clutter, A.Leonardis and H.Bischoff [42] extract eigenfaces by a hypothesis-and-test paradigm using subsets of image points. Competing hypothesis are then subject to a selection procedure based on the minimum description length principle. Their method reduces the influence of occlusion and backgrounds clutter. However, it still exhibits sensitivity to image deformation.

2.2.2. Local Feature

Local features have many advantages. First, they are robust to global deformations. Second, they are robust to occlusion and background clutter. Third, their relative spatial relationships can be exploited. However, two problems must be solved for local feature based applications. One problem is how to detect them reliably under all kinds of conditions. Interest region detectors are used for this purpose. The other is how to resolve matching ambiguity due to the deficiency of global context. This is the problem of outlier rejection. Solving these two problems is very important for many computer vision applications, and they are the main topics of this thesis.

2.3. Local Interest Region Detector

Feature detectors are commonly used to extract stable and informative regions from images in order to reduce the computational complexity and improve the robustness to image deformation. Early detectors, called interest point detectors, only define the position of detections. Subsequently, interest region detectors were introduced to detect locations and define surrounding regions as well. According to [66], detectors can be divided into three types: (a) contour-based methods, (b) intensity-based methods, and (c) parametric-model based methods. The newer evaluation paper [52] by Mikolajczyk et al. divides interest region detectors into the two categories of intensity-based detectors and structure-based detectors.

Intensity-based detectors depend on analyzing local differential geometry or intensity patterns to find points or regions that satisfy some uniqueness and stability criteria. The Harris corner detector [33] finds points or pixels where both

eigenvalues of the second moment matrix are large by evaluating the simple-to-compute "Harris measure". The Harris-affine and Hessian-affine detectors [53, 55] compute maximum determinants of the second moment matrix and the Hessian matrix respectively across scale space and then apply Laplacian-based characteristic scale selection [43] and second-moment-matrix-based shape adaptation [44, 3]. MSER [50] uses a threshold selection process to detect stable regions that are either brighter or darker than the surrounding region. SIFT (i.e., the difference of Gaussian (DoG) extrema detector used by Lowe in [47]) finds local extrema across three consecutive difference-of-Gaussian scales and then removes spurious detections via a DoG-response threshold followed by a Harris-like metric to eliminate edge detections. Kadir's salient region detector [38] calculates the entropy of the probability density function (PDF) of intensity values over the scale and ellipse parameter spaces to find regions with entropy extrema. Other intensity-based detectors include SUSAN [71], intensity extrema-based regions (IBR) [77], and the work of Moravec [58] and Beaudet [4].

Structure-based detectors depend on structural image features such as lines, edges, curves, etc. to define interest points or regions. These detectors tend to be very computationally expensive and typically depend on reliable prior detection of structural features. Early structure-based detectors analyzed various 2D curves such as the curvature primal sketch or B-splines extracted from edges, ridges, troughs, etc. and then selected high curvature points, line or curve intersections, corners, ends, bumps, and dents as interest points [2, 51, 24, 69, 57]. Tuytelaar's edge-based region (EBR) detector [76] fits a parallelogram defined by Harris corner point and points on two adjacent edge contours (extracted by the Canny detector [11]). Scale-invariant shape features (SISF) [37] detects circles at different locations and scales by evaluating salient convex arrangements

of Canny edges based on a measure that maximizes how well a circle is supported by surrounding edges.

2.3.1. Intensity-based Region Detectors

- The Moravec detector [58] was the first local feature detector. This detector measures the directional variance over a local window as the windows is shifted in horizontal, vertical and two diagonals directions. If the minimum of these variances is greater than a threshold, an interest point is detected. As variances are sensitive to outliers, this method is sensitive to noise. The fixed-size local window is also not robust to image deformation.

- The Beaudet detector [4] uses the Hessian matrix 2.4 calculated from the second derivatives and a measure of the matrix determinant ($I_{xx}I_{yy} - I_{xy}^2$)to choose points. Points that are local maxima of the determinant values are chosen as detected points. This type of detector tends to detect uniform blobs.

- The Harris corner detector [33] looks for corners where the gradient changes in two directions. The direction of gradient does not affect the detection thus making the detection rotation invariant. The detector is somewhat robust to variations in illumination since gradient is insensitive to changes in brightness. The first order derivative-based second-moment matrix (see Eq. 2.1) is used for detection.

$$M = \mu(x, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(x, \sigma_D) & I_x(x, \sigma_D) I_y(x, \sigma_D) \\ I_x(x, \sigma_D) I_y(x, \sigma_D) & I_y^2(x, \sigma_D) \end{bmatrix} \tag{2.1}$$

where $I_x(x, \sigma_D)$, $I_y(x, \sigma_D)$ are the first-order derivatives of the image. σ_D is the Gaussian scale at which the first partial derivatives of the image are computed. $g(\sigma_I)$ is the integration Gaussian function used to average derivatives.

The second moment matrix in Eq. 2.1 is also called the auto-correlation matrix. It describes the gradient distribution in a local neighborhood of a point. The eigenvalues of this matrix represents two principal signal changes in the neighborhood. This property enables the extraction of corner points, for which both eigenvalues are significant. When one eigenvalue is big and the other is small, the image window conveys an edge. If both eigenvalues are small, it is a homogeneous area. The Harris measure (Eq. 2.2) avoids calculating the eigenvalues directly. It measures the "cornerness" by using the determinant and the trace of the matrix. Points that have Harris measure value greater than a threshold are selected as detected points.

$$det(M) - \alpha \cdot trace^2(M) > threshold \tag{2.2}$$

- SUSAN (Smallest Univalue Segment Assimilating Nucleus) corner detector [71] uses a totally different detection principle. USAN (Univalue Segment Assimilating Nucleus) is an area that has the same or similar brightness as the nucleus of a region mask. The area of USAN conveys information about the structure of the image in the region around a point. It reaches a maximum when the nucleus lies in a flat region of the image surface; it falls to half of this maximum when near to a edge, and falls even further when inside a corner. This property of the USAN's area is used to determine the presence of edges or corners. This integrating effect of the principle, together

with its non-linear response, provide strong noise rejection without the need to compute derivatives.

- Scale Invariant Feature Transform (SIFT) [47] is a very successful detector and descriptor. It improves detection stability in terms of scale changes, viewpoint changes, noise, and changes in illumination. It achieves almost real-time performance and the detected features are highly distinctive. SIFT also provides a very successful region descriptor for scene matching and object class recognition.

  The SIFT detector uses the difference of Gaussian (DoG) filter as an approximation to the normalized Laplacian, which is needed for true scale invariance [43]. Lindeberg [43] shows that under a variety of reasonable assumptions the only possible scale-space kernel is a Gaussian function. Maxima and minima of the normalized Laplacian over scales was found to be the most stable image features in terms of scales changes by Mikolajczyk [53]. For a given point, scales where these maxima and minima are located are called the characteristic scale at that point. This Laplacian-based scale selection process has been adopted by many detectors. The DoG scale space is sampled by blurring an image with successively larger Gaussian filters and subtracting each blurred image from the adjacent (more blurred) image. In this case, three levels of scale are created for each octave by blurring the image with incrementally larger Gaussian filters with scale steps of $\sigma = 2^{1/3}$. After completing one octave, the image with twice the initial σ is resampled by taking every other row and column pixel values. This resampled image becomes the first image in the second octave. The same smoothing and sampling process is repeated till a certain image size is reached. In-

terest points are characterized as the extrema (maxima or minima) in the $3D(x, y, \sigma)$ space. As such, each pixel is compared with its 26 neighbors in scale space and a pixel is selected as a feature point if its value is larger or smaller than all of its neighbors. Subsample accurate position and scale is computed for each extrema point by fitting a quadratic polynomial to the scale space function $D(x, y, \sigma)$ and finding the extremum, giving

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2}\frac{\partial D}{\partial X} \tag{2.3}$$

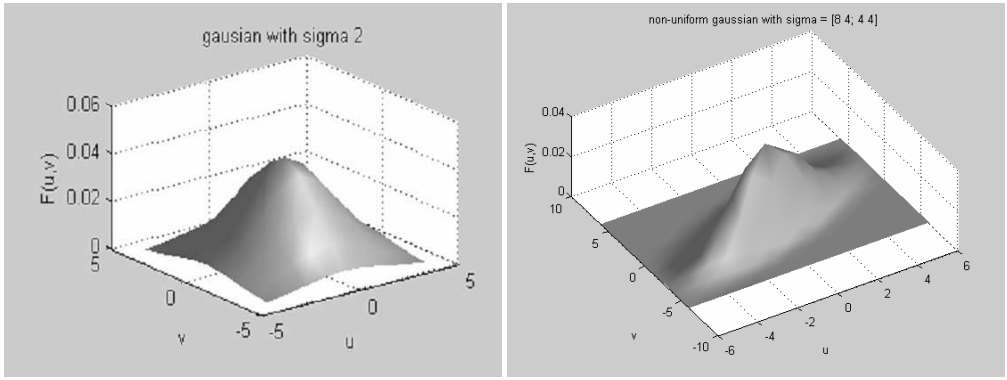
where $X = (x, y, \sigma)$ and $\hat{X}$ is the extremum position providing accurate position and scale. Finally, an orientation is assigned to each interest point that, combined with the scale above, provides a scale and rotation invariant coordinate system for the descriptor. Orientation is determined by building a histogram of gradient orientations from the key point's neighborhood, weighed by a Gaussian and the gradient magnitude. Every peak in the histogram with a height of 80% of the maximum produces a key point with the corresponding orientation. A parabola is fit to the peak(s) to improve accuracy.

- The Harris or Hessian-Affine Region detector [55] focuses on the problem of achieving viewpoint-invariant region detection using affine transformations as an approximation. The Harris-affine detector uses the second moment matrix (Eq.2.1) and the Hessian-affine detector selects interest regions by using the Hessian matrix given by (Eq.2.4).

$$H(x, \sigma_D) = \begin{bmatrix} I_{xx}(x, \sigma_D) & I_{xy}(x, \sigma_D) \\ I_{xy}(x, \sigma_D) & I_{yy}(x, \sigma_D) \end{bmatrix} \tag{2.4}$$

where $I_{xx}(x, \sigma_D)$, $I_{xy}(x, \sigma_D)$, $I_{yy}(x, \sigma_D)$ are the second-order partial derivatives of the image. σ_D is the Gaussian scale at which the second partial derivatives of the image are computed.

A true affine-invariant region detector should explore the affine Gaussian scale-space, which is represented by $g(\sigma) = \frac{1}{2\pi\sqrt{det\Sigma}} \exp^{-\frac{x^T \Sigma^{-1} x}{2}}$, where Σ is the covariance matrix of the Gaussian. An example of affine Gaussian is shown in (Fig. 2.1(b)). The uniform Gaussian space $g(\sigma) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}}$, where σ is the standard deviation of Gaussian, is shown in Fig.2.1(a)). Compared with the uniform Gaussian space, detection in the affine Gaussian



(a) (b)

FIGURE 2.1. Examples of (a)a uniform Gaussian and (b)an affine Gaussian.

scale-space is computationally prohibitive. To save computational time, this detector does not explore the whole affine Gaussian scale-space, thus it is not truly affine invariant. They start with points detected by the normal Harris or Hessian detector and the detection process then uses the second-moment matrix to iteratively adapt the shape while also using the Hessian matrix to search for maxima in scale space. Given a set of initial interest points, the detector estimates it shape based on the second moment matrix M. The

size (or scale) of the region is determined by its characteristic scale which, in turn, is given by the Laplacian-based scale selection process [43]. The region is normalized by the transformation given by the square root of the second moment matrix $M^{1/2}$. For example, if the neighborhood of corresponding points X_R and X_L are normalized by transformation $X'_R = M_R^{1/2} X_R$ and $X'_L = M_L^{1/2} X_L$, X'_R and X'_L are related by a rotation only. The rotation can be removed using the dominant orientation calculation [47]. After the normalization, the second moment matrix of the normalized region is rebuilt and this process continues till the two eigenvalues of the matrix are equal. This detector has high repeatability based on the evaluation framework proposed by Mikolajczyk [52].

- Rather than use derivitive filters, the Maximally Stable Extremal Region detector (MSER) [50] detects regions through thresholding instead of filtering. All pixels inside the MSER region have either higher or lower intensity than the pixels on its outer boundary. MSER regions are stable over a large range of thresholds. The maximal stability is measured by the relative area change as a function of thresholds. When the rate of change reaches a local minimum, the binarized (i.e.,thresholded) region is selected. The enumeration of regions is very fast. First, all pixels are sorted by intensity. Then, pixels are marked in the image and regions are defined using the union-find algorithm [68]. MSER is affine invariant as the definition of MSER region stability is independent of geometric transformations.

- The Intensity Extrema Region detector [77] starts with intensity extrema at multiple scales and defines the surrounding region as the extrema in intensity changes along rays emanating from the detected point. The resulting regions

can be any arbitrary shape. The extrema in the intensity change is defined as

$$f_I(t) = \frac{abs(I(t) - I_0)}{\max\left(\frac{\int_0^t abs(I(t)-I_0)dt}{t}, d\right)} \tag{2.5}$$

where t is the position along the ray, $I(t)$ is the intensity at position t, I_0 is the intensity value at the extremum and d is a small number to prevent a division by zero. Typically, the maximum of this function is reached where the intensity value suddenly increases or decreases. All maxima of this function are linked to define a region. As the function $f_I(t)$ and those extremum are affine invariant, this detector is affine invariant too.

- The Salient Region detector [38] explores the affine Gaussian scale space using different ellipses at different scales and orientations. The saliency measure Y_D, a function of scale s and position x is defined as

$$Y_D(s_p, x) = H_D(s_p, x)W_D(s_p, x) \tag{2.6}$$

where $H_D(s_p, x) = -\sum_I p(I) \log p(I)$ is the entropy of the intensities within an elliptical region at scale s_p and position x and $p(I)$ is the probability density function of the intensities within an ellipse. $W_D(s_p, x)$ is the inter-scale saliency and is defined by

$$W_D(s_p, x) = s \int \left| \frac{\partial}{\partial s} p(I, s, x) \right| \tag{2.7}$$

where $\frac{\partial}{\partial s} p(I, s, x)$ is the derivative of the probability density function. The salient region detector is affine invariant as it explores the entire affine Gaussian space. The detected maximum entropy regions seem suitable for object recognition to the extent that maximal entropy implies high information contents. Unfortunately, since this detector exhaustively searches the affine scale space, its calculation time is prohibitive.

2.3.2. Structure-based Region Detector

- There are several interest point detectors based on analyzing 2D curves extracted from image structures. In 1986, H. Asada and M. Brady [2] extract interest points from the curvature primal sketch, which are 2D curves detected by derivative filters. Corners, smooth joins, cracks, ends, bumps and dents are chosen based on the curvature of these 2D curves at multiple scales. G. Medioni and Y. Yasumoto [51] use B-splines to represent shapes. Interest points occur at the maxima curvature points of these B-splines. R. Deriche and G. Giraudon [24] use intersections of lines as detected points. E. Shilat etc. [69] use ridges or troughs instead of the traditional step edges in detection. High curvature points along these ridges or troughs, intersections of curves and minima points of image surfaces are detected. F. Mokhtarian and R. Suomela [57] detect points where edges have their maxima of absolute curvature. These points are detected in coarse scale and track down to fine scale to get more accurate position.

- The Edge-Based Region detector [76] (Fig.2.2) starts from a Harris corner point p and connect it to two other points p_1, p_2 along adjacent Canny edges [11]. p_1, p_2 are on different sides of p and move away from p until the area l_i between the canny edge and the line pp_i reach some thresholds (see fig.2.2).

  The areas l_1, l_2 must be equal as the points p_1, p_2 move. This condition is an affine invariant criterion indeed. The parallelogram that is extremum in either of the following texture measure is chosen as the detected region.

$$
\begin{aligned}
Inv_1 &= abs\left(\frac{(p_1 - p_g)(p_2 - p_g)}{(p - p_1)(p - p_2)}\right)\frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)2}} \\
Inv_2 &= abs\left(\frac{(p - p_g)(q - p_g)}{(p - p_1)(p - p_2)}\right)\frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)2}}
\end{aligned} \tag{2.8}
$$

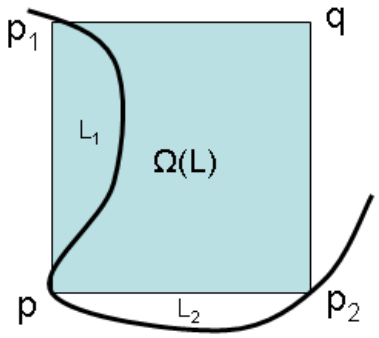


FIGURE 2.2. Edge-based Region Detector.

where $M_{pq}^n = \int_\Omega I^n(x,y)x^p y^q dxdy$ is the nth order, $p+q$ degree moment computed over the parallelogram Ω. $p_g = (\frac{M_{10}^1}{M_{00}^1}, \frac{M_{01}^1}{M_{00}^1})$ is the center of mass of the region.

- Scale-Invariant Shape Features (SISF) [37] first applies the Canny edge detector [11], the same as EBR, and then for every pixel in the image, circles with different size are defined and edge points that near the circle are found. The saliency measure is defined as the weighted sum of local contributions over these edge points, with the weights set to reflect closeness to the circle and alignment with its local tangent (see Fig. 2.3.2).

  The closeness weight is

$$w_i^d(c,\sigma) = \exp(-\frac{(\|p_i - c\| - \sigma)^2}{2(s\sigma)^2}) \tag{2.9}$$

  where s defines the locality of the detection and the tangency is measured by

$$w_i^o(c,\sigma) = \left| g_i \bullet \frac{p_i - c}{\|p_i - c\|} \right| = \|g_i\| \, cos\angle(g_i, P + i - c). \tag{2.10}$$

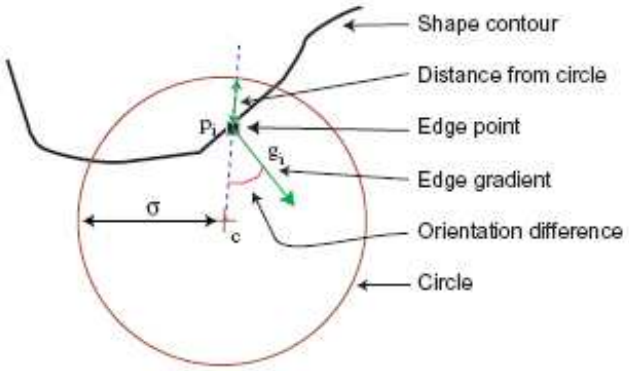


FIGURE 2.3. Quantities used for SISF Region extraction. (Image courtesy F. Jurie and C. Schmid)

The final weight $w_i(c,\sigma)$ is the product of the weights of closeness $w_i^d(c,\sigma)$ and alignment $w_i^o(c,\sigma)$. The final weights define the tangent edge energy

$$E(c,\sigma) \equiv \sum_{i=1}^{N} w_i(c,\sigma)^2 \tag{2.11}$$

To measure the extent to which the circles get support from a broad distribution of points around its boundary(and not just from a few points on one side of it), contour orientation entropy is defined as

$$H(c,\sigma) \equiv \sum_{k=1}^{M} h(k,c,\sigma) log h(k,c,\sigma) \tag{2.12}$$

And the final saliency metric is given by

$$C(c,\sigma) = H(c,\sigma)E(c,\sigma). \tag{2.13}$$

2.4. Evaluations of Local Interest Region Detectors

Local feature detectors can be evaluated by many criteria. Early evaluation methods employed visual inspections. For example, K. Bowyer et al. [9] used the

human generated ground truth feature for evaluation. However, ground truth features are hard to acquire or can only be decided subjectively. Compared with other methods, visual inspection [34] is easier but is subjective since different people may produce different results.

Localization accuracy [17] is important to tasks like camera calibration or 3D reconstruction. The localization accuracy is often measured by verifying that a set of 2D image points is coherent with the known set of corresponding 3D scene points.

Cordelia Schmid et al. [66] proposed repeatability and information content as evaluation criteria. The repeatability rate is defined as the percentage of the total observed corresponding points that are detected in both images. In the evaluation process, the repeatability conflicts with localization. Decreasing localization accuracy increases repeatability. The other criterion they used is information content which measures the distinctiveness of a feature. In this evaluation process, descriptors that characterize local features are generated, and entropy of these descriptors determine the information content. The more spread out the descriptors in feature space, the higher the entropy, whereas the lower the entropy, the closer the local features. When entropy is low, features do not convey enough information and there are ambiguities in matching.

Gustavo Carneiro [14] trained a discriminative classifier to select well behaved feature points based on distinctiveness, detectability and robustness. This classifier is applied prior to running a recognition system to reduce recognition time, improve accuracy and increase the scalability.

K. Mikolajczyk etc. [52] measured repeatability of images under various transformations to evaluate affine covariant region detectors. These transformations include viewpoint change, scale change, image blur, JPEG artifacts, and

light change. The ground truth of matching is calculated from the homography relating the two images. As affine region detectors output elliptical regions, the overlap error of two ellipses is used as the matching measure. When the overlap error is small enough, two regions are deemed to corresponded. The repeatability score for a given pair of images is computed as the ratio of the number of corresponding regions to the smaller of the total number of regions detected in the pair of images.

In this thesis, we evaluate our detector in Chapter 4 based on three criteria. The first is by visual inspection. The second is the repeatability from the framework given by Mikolajczyk [52]. The third uses the information content similar to Schmid [66].

2.5. Matching and Outlier Rejection

Local feature-based matching can be divided into two categories. One is to match the spatial relationships of points. It includes geometric hashing [40],iterated closest point [6], softassign [15], shape context [5] etc. The other category matches local appearance and rejects outliers using other constraints. This method includes Hough transform [10], RANSAC [30] and PROSAC [16], and voting by semi-local constraints [67].

Geometric hashing [40] is used in model-based matching problems. For every model, interest points are extracted and K random features are chosen to define the feature space. These K random features form a coordinate basis. Coordinates of the remaining features are calculated relative to this basis. A hash table is filled with the (model, basis) entries and indexed by feature coordinates. This process continues unti all possible bases of K feature are tried and the hash

table is built. During matching, interest points are first extracted, K features are chosen to build a basis and the coordinates of the other features are calculated relative to this basis. These coordinates are then used to access the hash table bins and thus rank models. The model receiving the most votes is assumed a good match.

Iterated closest point [6] uses an iterative process involving three steps. First, feature correspondences are established based on proximity. Second, the transformation matrix is calculated using these correspondences. Third, the transformation is applied to the first features and a new set of features are acquired. These new features are inputs to the first step and the algorithm continues till convergence.

Softassign [15] improves the iterated closest point algorithm so that it handles non-rigid transformations and is more tolerate to noise. They develop a TPS-RPM algorithm with the thin-plate spline as the parameterization of the non-rigid spatial mapping and the softassign for the correspondence.

The Hough transform [10] clusters features in pose space. For each matched hypothesis, the features vote for a transformation model in pose space. The transformation model that receives the most votes is selected. The matched points that disagree with this model are deemed as outliers and are thus removed from the matching list.

C. Schmid and R. Mohr [67] combine semi-local spatial constraints with local descriptors to do outlier rejection in image retrieval. In addition to checking the similarity of the local features' appearance, the spatial configurations of every feature's K nearest neighbors are also checked. This method uses the angle between two points to verify the spatial conformation.

There are too many matching methods in the literature to discuss them all thoroughly. Therefore, in the following paragraphs, we will mainly focus on shape context [5], RANSAC [30] and PROSAC [16], which are closely related to this thesis.

2.5.1. Shape Context

S. Belongie et al. [5] start with a collection of shape points (identified using the Canny edge detector, for example) and, for each point, build the relative distribution (Fig. 2.4) of the other points in log-polar space. The shape context is scale and rotation invariant and point differences are measured using the χ^2 statistic between the log-polar histograms. They use the Hungarian assignment algorithm to find the best global one-to-one assignment of point contexts between images. This is followed by a thin-plate spline fit to warp one shape to the other.

Shape context provides matching flexibility and allows for non-rigid transformations between images. As long as the points are in the same bin, they can be moved to anywhere in this bin and still get matched. Methods proposed in this thesis take advantage of shape context to do outlier rejection and extend this method to be affine-invariant by replacing the circular bins with elliptical bins. We also achieve rotation-invariance by transforming all points to normalized coordinates using the dominant orientation.

2.5.2. RANSAC and PROSAC

RANdom SAmple Consensus (RANSAC) [30] is an algorithm for robust model fitting in the presence of data outliers. Given a model with parameters X, RANSAC estimates these parameters using known data values.

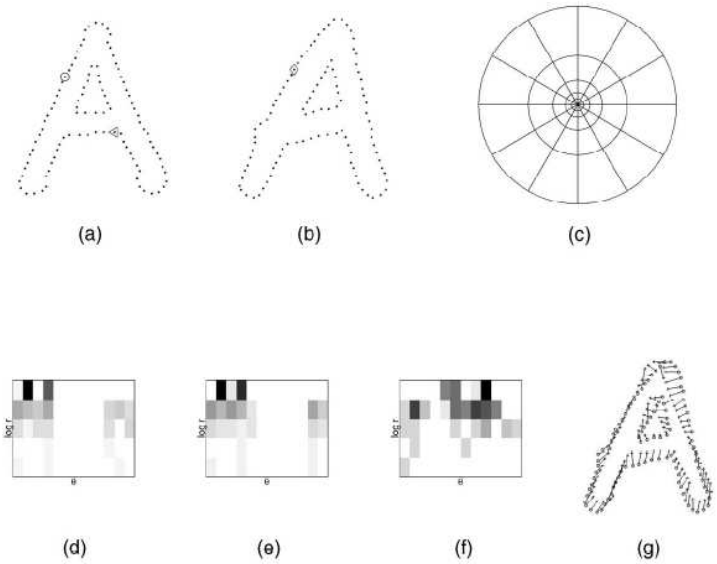


FIGURE 2.4. Shape context computation and matching. (a) and (b) Sampled edge points of two shapes. (c) Diagram of log-polar histogram bins used to compute shape context consists of five bins for $\log r$ and 12 bins for θ. (d), (e),and(f) Example shape contexts for reference samples marketed by $\circ$ $\diamond$ $\triangleleft$ in (a) and (b). Each shape context is a log-polar histogram of the coordinates of the rest of the point set measured using the reference point as the origin.(Dark=large value.) note the visual similarity of the shape context for $\circ$ and $\diamond$ which were computed for relatively similar points on the two shapes. By contrast, the shape context for $\triangleleft$ is quite different. (g) Correspondences found using bipartite matching, with costs defined by the χ^2 distance between histograms.(Image and caption courtesy S.Belongie etc.)

For example, let us assume the model can be estimated from N data items. There are $M > N$ total data value. The RANSAC algorithm is:

1. Select N data items at random from the M data values.
2. Using the N data items to estimate the parameters of the model.
3. Verify the model using the remaining $M - N$ data items by counting the number of data K that fit the model to within some user defined tolerance.
4. If K is large enough, accept the model and exit.
5. Otherwire, go back to step 1.
6. Fail after some user defined number of iterations.

The choice of K depends on what percentage of the data you think belongs to the model being fit and how many models you have in the image. Identifying the correct model depends on whether N inliers can be correctly chosen. For example, if there are W inliers in the M data, the probability $P_{inliers}$ of choosing N inliers is:

$$P_{inliers} = \frac{C_W^N}{C_M^N} = \frac{W!(M-N)!}{M!(W-N)!} \tag{2.14}$$

If this probability is high, there are more chances to get the correct model and the number of iterations is smaller and vice versa.

Progressive Sample Consensus (PROSAC) [16] exploits the linear ordering defined on the set of correspondences by a similarity function used in establishing tentative correspondences. Unlike RANSAC, which treats all correspondences equally and draws random samples uniformly from the full set, PROSAC samples are drawn from progressively larger sets of top-ranked correspondences. This

algorithm assumes that the similarity measure predicts correspondences of a match better than random guessing. As such, PROSAC is more computationally efficient as it is more likely to draw inliers.

RANSAC is widely used in rejecting outliers on matching images with homography. Z. Zhang et al. [83] use epipolar geometry to restrict the searching space to epipolar lines thus reducing ambiguity. Given a stereo pair of cameras (Fig 2.5), a point P in $3D$ space, and the centers of projection (O_1, O_2) of the two cameras, the epipolar plane π_p is defined. The lines (l_1, l_2) where π_p intersects the image plane are called epipolar lines. e_1 and e_2 are called epipoles. The corresponding point of the image point x_1 can only exist on the epipolar line l_2 and vice versa. This is known as the epipolar constraint.

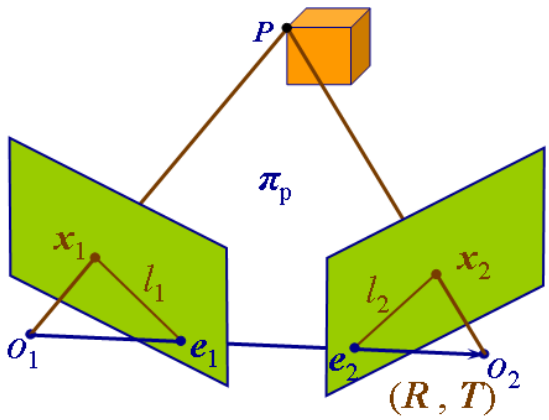


FIGURE 2.5. Illustration of epipolar geometry.

The fundamental matrix F defines the transformation between the two cameras. It satisfy the epiplar constraint equation

$$x_1 F x_2 = 0 \tag{2.15}$$

where x_1 is the projection of p in the left image and x_2 is the projection in the right image.

The fundamental matrix can be calculated using the eight-point algorithm [46], which, as the name suggests, require eight corresponding points. RANSAC algorithm is applied here to estimate the fundamental matrix. Eight random tentative corresponding points are selected and parameters of the fundamental matrix are estimated. This fundamental matrix is verified by other matching points. If the fitting score is high, this model is chosen as the correct transformation model. Otherwire, another eight points are chosen and a new model is estimated. This process continues until the correct model is found or the maximum number of iterations is reached.

Given F, the epipolar lines l_1, l_2 are given by $l_2 = F * x_1, l_1 = F * x_2$. Potential matches are only searched on or near the epipolar line.

3. PRINCIPLE CURVATURE-BASED REGION DETECTOR

3.1. Why a New Detector?

We developed this new detector for two reasons. First, previous intensity-based detectors prefer to detect distributed salient intensity patterns such as corners and blobs that are too local to capture characteristic parts of objects. Corners and blobs are suitable for image matching [52] on rigid objects. However, in object recognition system, these corners and blobs are too local and lack discriminative power. These systems rely on building various spatial models [27, 13, 19] or use bag of features [21] to improve recognition accuracy. However, spatial models have some drawbacks. Some of these spatial models are very computational expensive, e.g., the constellation model [27]. In order to lower the computational expense, the K-fan model [19] was developed to find a balance between computational complexity and the representational power. However, these models are not robust to spatial variation due to image deformations. They rely on probabilistic models to capture variation in spatial configurations. However, large spatial variations that exist in some object classes make the spatial cues unstable.

Second, in many object recognition tasks, within-class changes in pose, lighting, color, and texture can cause considerable variation in local intensity. Therefore, local intensity does not always provide a stable detection cue. As such, intensity-based interest operators (e.g., Harris, Kadir)–and the object recognition systems based on them (e.g., Opelts [62])–often fail to identify discriminative features. An alternative to local intensity cues is to capture semi-local structural cues such as edges and curvilinear shapes [72]. These structural cues tend to be more robust to intensity, color, and pose variation. They provide the basis for a more stable interest operator, which in turn improves object recognition accuracy.

Therefore, we introduce a new detector that exploits curvilinear structures to reliably detect interesting regions. The detector, called the Principal Curvature-Based Region (PCBR) detector, identifies stable watershed regions within the multi-scale principal curvature images.

Curvilinear structures are lines (either curved or straight) such as roads in aerial or satellite images or blood vessels in medical scans. They provide a kind of sketch of the objects appearing in images. These curvilinear structures can be detected over a range of viewpoints, scales, and illumination changes. The PCBR detector employs the first steps of Steger's curvilinear detector algorithm [72]. It computes the eigenvalues of the Hessian matrix at each pixel and then forms an image that is composed of one of the two eigenvalues. We call this the principal curvature image, as it approximates the principal curvature of the image intensity surface. This process generates a single response for both lines and edges, producing a clearer structural sketch of an image than is usually provided by the gradient magnitude image (see Fig. 3.1).

We develop a process that detects structural regions efficiently and robustly by applying the watershed algorithm to the principal curvature images across scale space. The watershed algorithm provides a more efficient mechanism for defining structural regions than previous methods of fitting circles, ellipses, and parallelograms [37, 76]. However, the watershed algorithm is sensitive to noise and other small image perturbations. To improve robustness to noise, we "clean" the principal curvature image with a grayscale morphological close operation followed by a new hysteresis thresholding method based on local eigenvector flow. The watershed transform is then applied to the cleaned principal curvature image. The resulting watershed regions (i.e., the catchment basins) define the PCBR regions. To achieve robust detections across multiple scales, the watershed is applied to

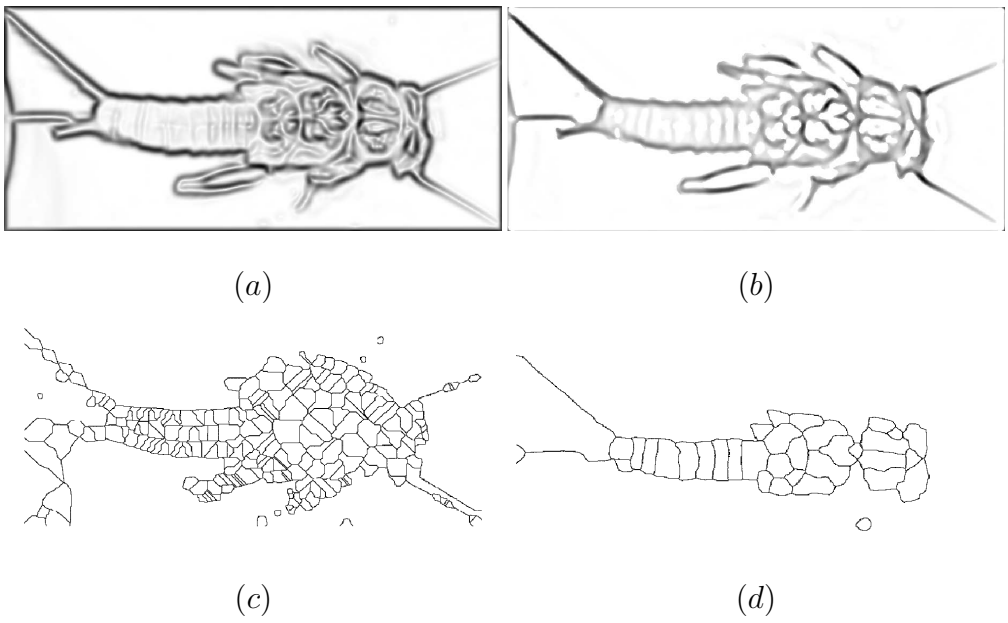


FIGURE 3.1. Comparision of Principal Curvature Response and Gradient Response. (a) Gradient Response, (b) Principal Curvature Response, (b) Watershed Boundaries from Gradient Response, (c) Watershed Boundaries from Principal Curvature Response.

the maxima of three consecutive images in the principal curvature scale space–similar to local scale-space extrema used by Lowe [47], Mikolajczyk and Schmidt [53], and others–and we further search for stable PCBR regions across consecutive scales–an idea adapted from the stable regions detected across multiple threshold levels used by the MSER detector [50].

In summary, this detector has two advantages. First, it can handle both edges and curvilinear structures. A normal edge detector will generate two responses for every single curvilinear structure on both of its sides, thus making regions fragmented. Fig. 3.1 (a)(b) shows the edge or gradient response and the principal curvature response of the stonefly image respectively. Fig. 3.1 (c,d) shows the watershed boundaries produced by the gradient magnitude and principal curvature images respectively. From the watershed boundaries, it is evident that the principal curvature response forms clearer regions. The principal curvature response generates only one response for both an edge or a curvilinear structure. If we treat an image as a 3D intensity surface, edges and curvilinear structures both have large curvature values in one direction and small curvature values in the other direction. Thus the principal curvature can detect both edge and curvilinear structures by choosing one of the eigenvalues of the Hessian matrix.

The second advantage is that the modified watershed algorithm used in our detector can achieve affine invariance more efficiently than circle or ellipse fitting that were used by other structure-based detectors. The SISF [37] detector requires an expensive search to fit circles to image structures. Furthermore, it is not affine-invariant and making it so would make it computationally intractable. EBR [76] limits the search space and achieves affine-invariant detection by starting with Harris corners and fitting parallelograms to nearby edges. This confines detection

to the detection results of another detector. PCBR is based on segmentation and the watershed region is a good approximation of the best fitted region. The detected regions are automatically affine-invariant because the watershed algorithm is also affine-invariant.

3.2. Principle Curvature-based Region Detector

The PCBR algorithm can be summarized as follows:

1. A curvature calculation algorithm similar to that of Stegers [72] is employed. But unlike Steger, we only generate the responses of curvature magnitude without thinning and post processing steps. We choose one of the maximum eigenvalues of Hessian matrix to form an eigenvalue image. They represent the principle curvature structures of this image surface.

2. Principle curvature images are calculated over the SIFT [47] scale space. Maximum of principle curvature values of three consecutive principle curvature images are calculated to form one image for processing.

3. Eigenvector-augmented hysteresis thresholding and morphological operations are applied on the maximal eigenvalue image to remove noise and help boost weak structural cues.

4. A watershed segmentation on the binary image is employed to form watershed regions.

5. An ellipses is fit to every watershed region.

6. Overlap errors of regions detected in two consecutive scales are calculated. Regions that can be detected in two consecutive scales and have overlap errors less than 30% are chosen as final detections.

Each of these steps is detailed in the following sections.

3.2.1. Principal Curvature Image

(a) (b)

FIGURE 3.2. Image Intensity Surface. (a) Original fingerprint image. (b) The 3D intensity surface

There are two types of structures that have high curvature in one direction and low curvature in the orthogonal direction: lines (i.e., straight or nearly straight curvilinear features) and edges. Viewing an image as an intensity surface, the curvilinear structures correspond to ridges and valleys of this surface (see Fig. 3.2). The local shape characteristics of the surface at a particular point can be described by the Hessian matrix,

$$\mathbf{H}(\mathbf{x}, \sigma_{\mathbf{D}}) = \begin{bmatrix} I_{xx}(\mathbf{x}, \sigma_{\mathbf{D}}) & I_{xy}(\mathbf{x}, \sigma_{\mathbf{D}}) \\ I_{xy}(\mathbf{x}, \sigma_{\mathbf{D}}) & I_{yy}(\mathbf{x}, \sigma_{\mathbf{D}}) \end{bmatrix}, \tag{3.1}$$

where I_{xx}, I_{xy} and I_{yy} are the second-order partial derivatives of the image evaluated at the point $\mathbf{x}$ and σ_D is the Gaussian scale at which the second partial derivatives are computed.

We note that both the Hessian matrix and the related second moment matrix have been applied in several other interest operators (e.g., the Harris [33], Harris-affine [52], and Hessian-affine [55] detectors) to find image positions where the local image geometry is changing in more than one direction. Likewise, Lowe's maximal difference-of-Gaussian (DoG) detector [47] also uses components of the Hessian matrix (or at least approximates the sum of the diagonal elements) to find points of interest. However, our PCBR detector is quite different from these other methods and is complementary to them. Rather than finding extreme "points", our detector applies the watershed algorithm to ridges, valleys, and cliffs of the image intensity surface to find "regions". Just like extreme points, the ridges, valleys, and cliffs can be detected over a range of viewpoints, scales, and appearance changes.

Many of the interest point detectors mentioned previously [33, 52, 55] apply the Harris measure (or a similar metric [47]) to determine a point's saliency. The Harris measure is given by $\mathbf{det}(\mathbf{A}) - \mathbf{k} \cdot \mathbf{tr^2}(\mathbf{A}) > \mathbf{threshold}$ where **det** is the determinant, **tr** is the trace, and the matrix $\mathbf{A}$ is either the Hessian matrix or the second moment matrix. The Harris measure penalizes (i.e., produces low values for) "long" structures for which the first or second derivative in one particular orientation is very small. One advantage of the Harris metric is that it does not require explicit computation of the eigenvalue or eigenvectors. However, computing the eigenvalues and eigenvectors for a 2×2 matrix requires only a single Jacobi rotation to eliminate the off-diagonal term, I_{xy}, as noted by Steger [72].

Our PCBR detector complements the previous interest point detectors in that we abandon the Harris measure and exploit those very long structures as detection cues. The principal curvature image is given by either

$$P(\mathbf{x}) = \max(\lambda_{\mathbf{1}}(\mathbf{x}), \mathbf{0}) \tag{3.2}$$

or

$$P(\mathbf{x}) = \min(\lambda_{\mathbf{2}}(\mathbf{x}), \mathbf{0}) \tag{3.3}$$

where $\lambda_1(\mathbf{x})$ and $\lambda_2(\mathbf{x})$ are the maximum and minimum eigenvalues, respectively, of H at $\mathbf{x}$. Eq. 3.2 provides a high response only for dark lines on a light background (or on the dark side of edges) while Eq. 3.3 is used to detect light lines against a darker background.

Principal curvature images are calculated in scale space in a fashion similar to that of SIFT [47]. We first double the size of the original image to produce our initial image, I_{11}, and then produce increasingly Gaussian smoothed images, I_{1j}, with scales of $\sigma = k^{j-1}$ where $k = 2^{1/3}$ and $j = 2..6$. This set of images spans the first octave consisting of 6 images, I_{11} to I_{16}. Image I_{14} is downsampled to half its size to produce image I_{21}, which becomes the first image in the second octave. We apply the same smoothing process to build the second octave, and continue to create a total of $n = log2(min(w, h)) - 3$ octaves, where w and h are the width and height of the doubled image, respectively. We build the following

image array:

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ 1.6 & 1.6k & 1.6k^2 & 1.6k^3 & 1.6k^4 & 1.6k^5 \\ I_{11} & I_{12} & I_{13} & I_{14} & I_{15} & I_{16} \\ I_{21} & I_{22} & I_{23} & I_{24} & I_{25} & I_{26} \\ ... & & & & & \\ I_{n1} & I_{n2} & I_{n3} & I_{n4} & I_{n5} & I_{n6} \end{array} \tag{3.4}$$

To save time on smoothing, we use increamental recursive filtering. The incremental *sigma* of a gaussian is:

$$\sigma = 1.6 * k^{j-1} * \sqrt{(k^2 - 1)} \tag{3.5}$$

We calculate a principal curvature image, P_{ij}, for each smoothed image above by computing the maximum eigenvalue (Eq. 3.2) of the second derivative Hessian matrix at each pixel to get the following image array:

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ 1.6 & 1.6k & 1.6k^2 & 1.6k^3 & 1.6k^4 & 1.6k^5 \\ P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & P_{26} \\ ... & & & & & \\ P_{n1} & P_{n2} & P_{n3} & P_{n4} & P_{n5} & P_{n6} \end{array} \tag{3.6}$$

Given the principal curvature scale space images, we calculate the maximum curvature over every three consecutive principal curvature images to form the following set of four images in each of the n octaves:

$$\begin{array}{cccc} MP_{12} & MP_{13} & MP_{14} & MP_{15} \\ MP_{22} & MP_{23} & MP_{24} & MP_{25} \\ & ... & & \\ MP_{n2} & MP_{n3} & MP_{n4} & MP_{n5} \end{array} \tag{3.7}$$

where $MP_{ij} = \max(P_{ij-1}, P_{ij}, P_{ij+1})$.

Figure 3.3(b), 3.4(b) shows one of the maximum curvature images, MP, created by maximizing the principal curvature at each pixel over three consecutive principal curvature images. From these maximum principal curvature images we find the stable regions via our watershed algorithm.

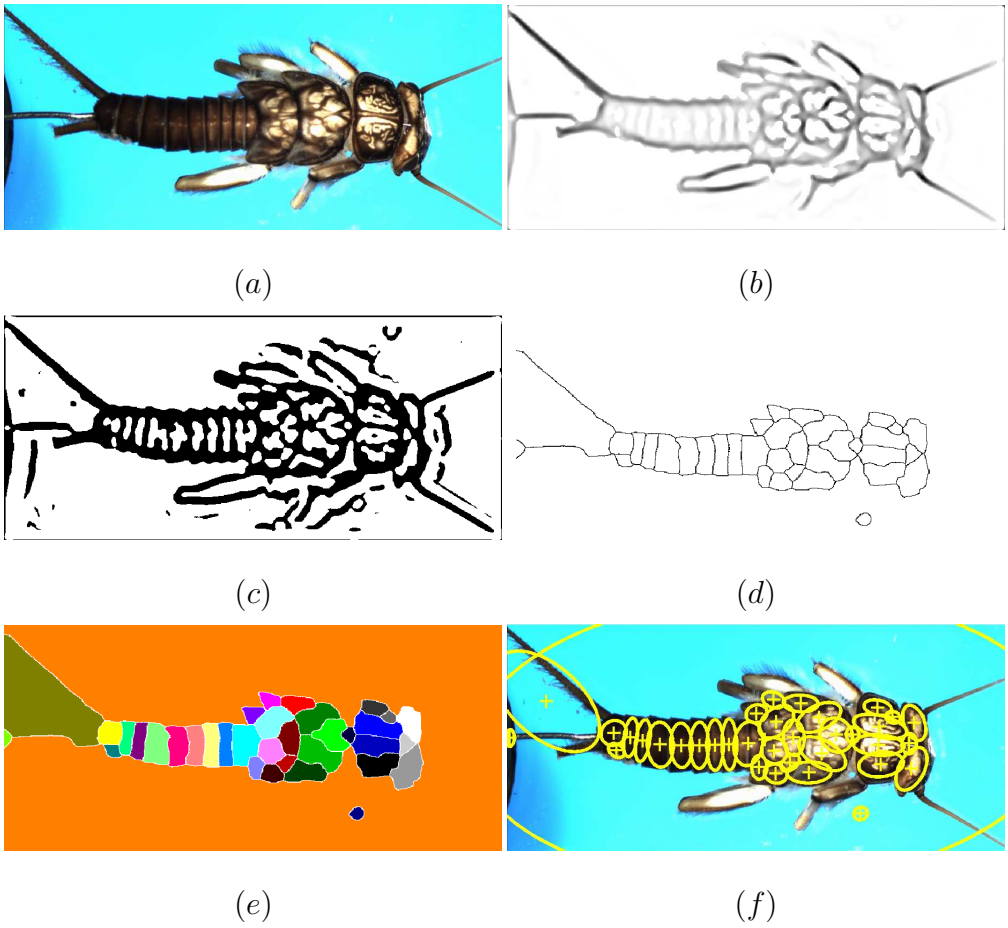
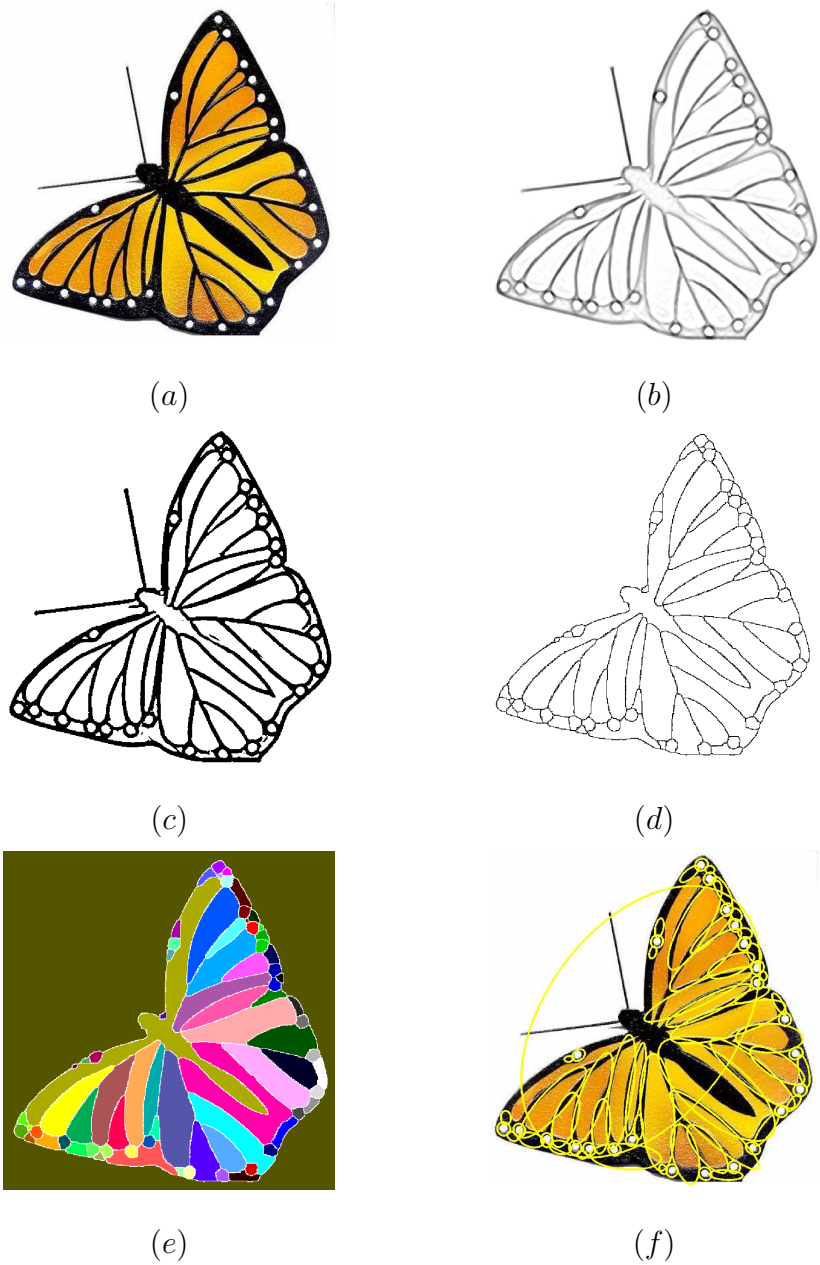


FIGURE 3.3. Interest Regions detected by the PCBR detector on a stonefly image. (a) Original stonefly image. (b) Principal curvature and (c) cleaned binary images. (d) Watershed Boundaries, (e) Watershed regions. (f) Detected regions represented by ellipses.



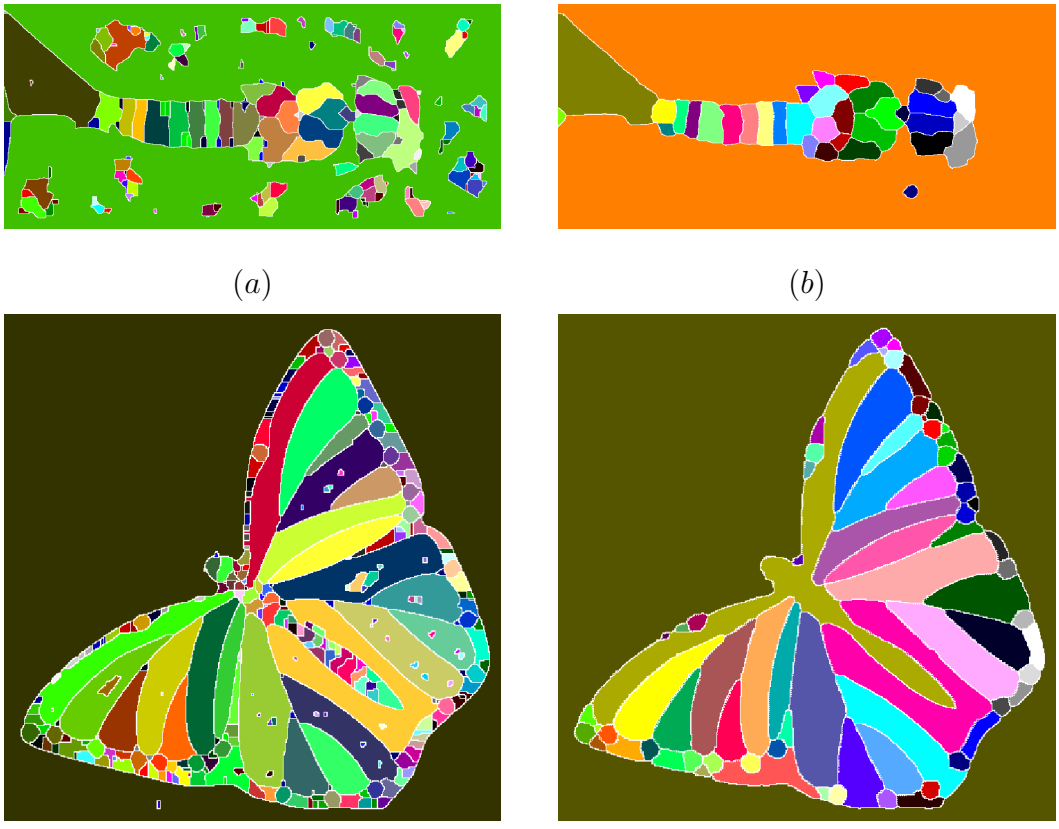
(a) (b) (c) (d) (e) (f)

FIGURE 3.4. Interest Regions detected by the PCBR detector on a butterfly image. (a) Original butterfly image. (b) Principal curvature and (c) cleaned binary images. (d) Watershed Boundaries, (e) Watershed regions. (f) Detected regions represented by ellipses.

3.2.2. Enhanced Watershed Regions Detection

The watershed transform is an efficient technique that is widely employed for image segmentation. It is normally applied either to an intensity image directly or to the gradient magnitude of an image. We instead apply the watershed transform to the principal curvature image. However, the watershed transform is sensitive to noise (and other small perturbations) in the intensity image. A consequence of this is that the small image variations form local minima that result in many, small watershed regions. Figure 3.5(a)(c) shows the oversegmentation results when the watershed algorithm is applied directly to the principal curvature image in Figure 3.3(b) and in Figure 3.4(b)). To achieve a more stable watershed segmentation, we first apply a grayscale morphological closing followed by hysteresis thresholding. The grayscale morphological closing operation is defined as $f \bullet b = (f \oplus b) \ominus b$ where f is the image MP from Eq. 3.7, b is a disk-shaped structuring element, and $\oplus$ and $\ominus$ are the grayscale dilation and erosion, respectively. The closing operation removes small "potholes" in the principal curvature terrain, thus eliminating many local minima that result from noise and that would otherwise produce watershed catchment basins.

However, beyond the small (in terms of area of influence) local minima, there are other variations that have larger zones of influence and that are not reclaimed by the morphological closing. To further eliminate spurious or unstable watershed regions, we threshold the principal curvature image to create a clean, binarized principal curvature image. However, rather than apply a straight threshold or even hysteresis thresholding–both of which can still miss weak image structures–we apply a more robust eigenvector-guided hysteresis thresholding to help link structural cues and remove perturbations. Since the eigenvalues of the



(a) (b)

(c) (d)

FIGURE 3.5. (a)(c) Watershed segmentation of original principal curvature image (Fig. 3.3b and Fig. 3.4b). (b) Watershed segmentation of the "clean" principal curvature image (Fig. 3.3c and Fig. 3.4c).

Hessian matrix are directly related to the signal strength (i.e., the line or edge contrast), the principal curvature image may, at times, become weak due to low contrast portions of an edge or curvilinear structure. These low contrast segments may potentially cause gaps in the thresholded principal curvature image, which in turn cause watershed regions to merge that should otherwise be seperate. However, the directions of the eigenvectors provide a strong indication of where curvilinear structures appear and they are more robust to these intensity pertubations than is the eigenvalue magnitude.

In eigenvector-flow hysteresis thresholding, there are two thresholds (high and low) just as in traditional hysteresis thresholding. For this application, we have set the high threshold at 0.04 to indicate a strong principal curvature response. Pixels with a strong response act as seeds that expand out to include connected pixels that are above the low threshold. Unlike traditional hysteresis thresholding, our low threshold is a function of the support that each pixel's major eigenvector receives from neighboring pixels. The low threshold is set on every pixel by comparing the direction of the major (or minor) eigenvector to the direction of the adjacent pixels' major (or minor) eigenvectors. This can be done by taking the absolute value of the inner product of a pixel's normalized eigenvector with that of each neighbor. If the average dot product over all neighbors is high enough, we set the low-to-high threshold ratio to 0.2 (for a low threshold of $0.04 \cdot 0.2 = 0.008$); otherwise the low-to-high ratio is set to 0.7 (giving a low threshold of 0.028). These ratios were chosen based on experiments with hundreds of images.

Figure 3.6 illustrates how the eigenvector flow supports an otherwise weak region. The red arrows are the major eigenvectors, and the yellow arrows are the minor eigenvectors. To improve visibility, we draw them at every fourth

pixel. At the point indicated by the large white arrow, we see that the eigenvalue magnitudes are small and the ridge there is almost invisible. Nonetheless, the directions of the eigenvectors are quite uniform. This eigenvector-based active thresholding process yields better performance in building continuous ridges and in handling perturbations, which results in more stable regions (Fig. 3.5(b)(d)).

The final step is to perform the watershed transform on the clean binary image (Fig. 3.3(c),Fig. 3.4(c)). Since the image is binary, all black (or 0-valued) pixels become catchment basins and the midlines of the thresholded white ridge pixels become watershed lines if they separate two distinct catchment basins. To define the interest regions of the PCBR detector in one scale, the resulting segmented regions are fit with ellipses, via principal component analysis(PCA), that have the same second-moment as the watershed regions (Fig. 3.3(e,f),Fig. 3.4(e,f)).

3.2.3. Stable Regions over Scales

Choosing maximum principal curvature images is only one way to achieve stable region detections. To improve robustness further, we adopt a key idea from MSER [50] and keep only those regions that can be detected in at least two consecutive scales (see Fig 3.7). In a method similar to the process of selecting stable regions via thresholding in MESR, we select regions that are stable across local scale changes. To achieve this, we compute the overlap error of the detected regions across each triplet of consecutive scales in every octave. Mikolajczyk et al. [52] calculate overlap error by checking every pixel inside the ellipses and count the number of pixles that fall into both ellipses. We use a much faster algorithm that approximates ellipses with polygons and analytically calculate the area of the overlaped polygons (Fig. 3.8).

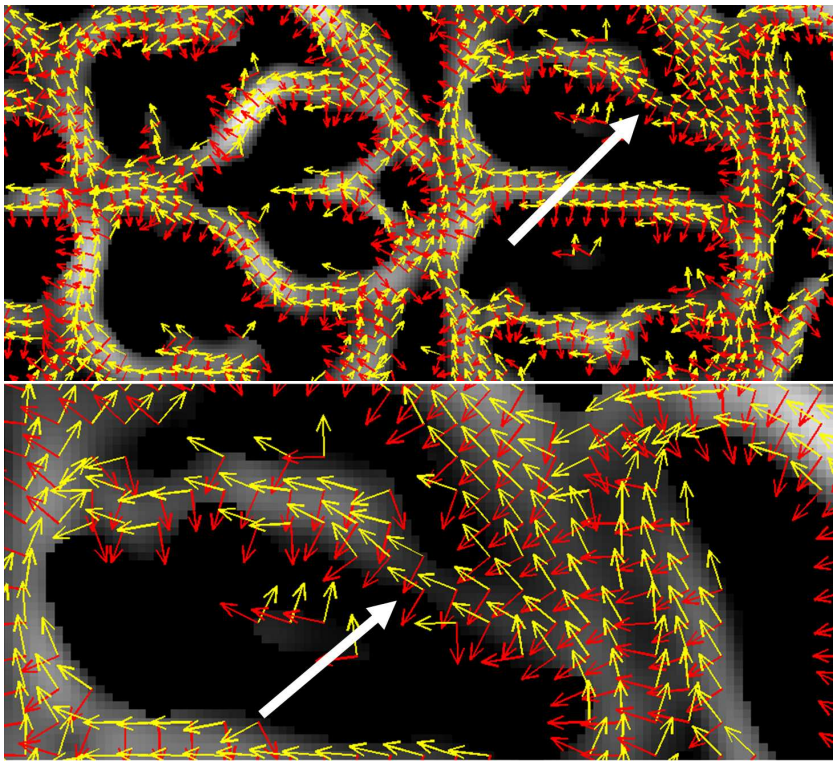


FIGURE 3.6. Illustration of how the eigenvector flow helps overcome weak principal curvature responses.

The overlap error is caculated as

$$1 - \frac{e1 \cap e2}{e1 \cup e2} \tag{3.8}$$

where $e1, e2$ are two ellipses.

Overlapping regions that are detected at different scales normally exhibit some variation. This kind of variation is valuable for object recognition because it provides multiple descriptions of the same pattern. An object category normally exhibits large within-class variation even in the same area. Since detectors usually have difficulty in locating the interest area accurately, rather than attempt to find a single region and extract a single descriptor vector, it is better to identify

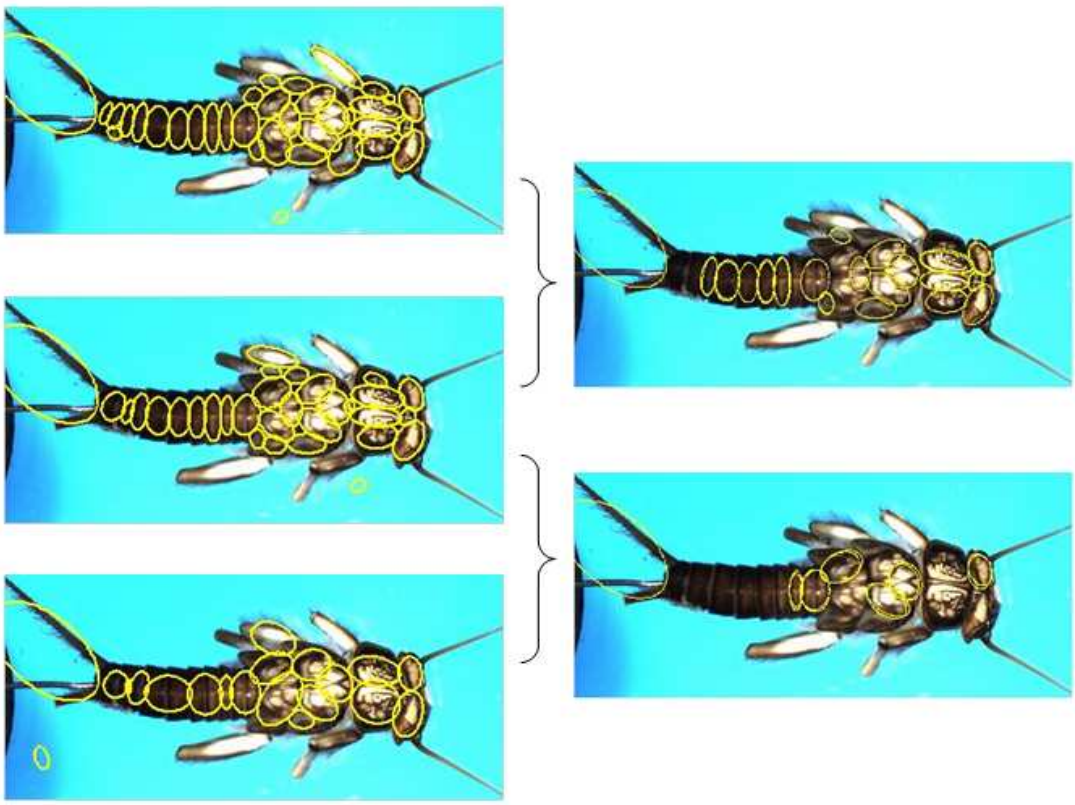


FIGURE 3.7. Choosing overlap ellipses across scales.

multiple overlapping regions and extract multiple descriptor vectors, provided that these multiple vectors can be handled properly by the classifier.

To determine a threshold value for the permitted amount of overlap, we analyze the sensitivity of the SIFT descriptor. Three transformations (translations from 1 to 10 pixels, rotations from 2 to 20 degrees and minor axis enlargements from 1 to 10 pixels) are applied on all detected regions in the Inria dataset [52]. Overlap errors and similarities of SIFT descriptors between the transformed regions and the originals are calculated. To keep regions that can be detected over local scales, only regions with overlap error less than 30% are chosen. However,

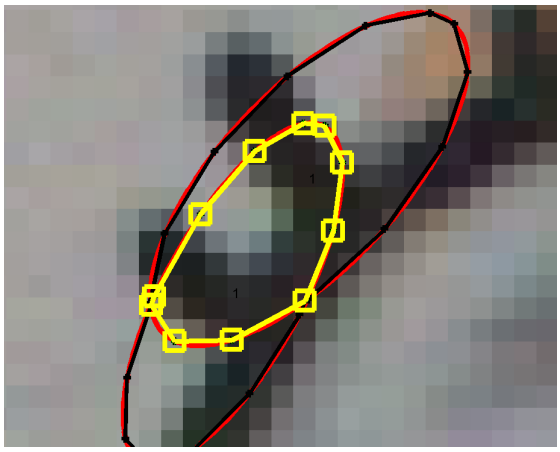


FIGURE 3.8. Calculate overlap errors by polygon approximation.

as indicated in Figure 3.9, SIFT similarity decreases to 70% for regions that have an overlap error of 30%. As such, we keep all stable regions with an overlap error less than 30% to maintain more descriptions for similar regions. We also notice that the similarity of the SIFT descriptors is above 90% when overlap error is less than 10%. These very similar regions are merged into a single region.

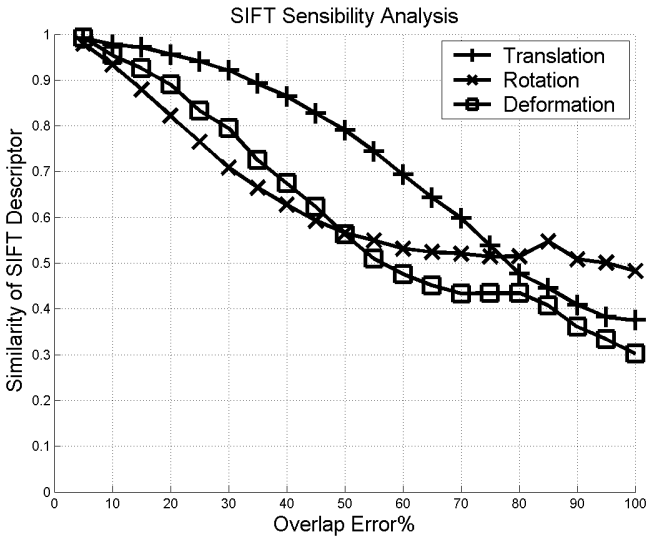


FIGURE 3.9. Sensitivity analysis of SIFT descriptor.

4. EVALUATIONS

We evaluate our PCBR detector in three ways: 1) qualitative visual inspection, 2) quantitative repeatability using a published framework [52], and 3) quantitative evaluation on information contents.

4.1. PCBR Regions—A Visual Inspection

To provide a visual evaluation of PCBR, we show PCBR detection results on a variety of different types of images. Fig. 4.1 shows PCBR detections on two graffiti images from the INRIA dataset [52]. In Figure 4.2 shows detection results for face, motorbike, and cars (rear) images from the Caltech dataset. In Figure 4.2, we remove background detections to improve visibility. From these images we note that PCBR detections appear to be evenly distributed, highly consistent, and robust to intra-class variations.

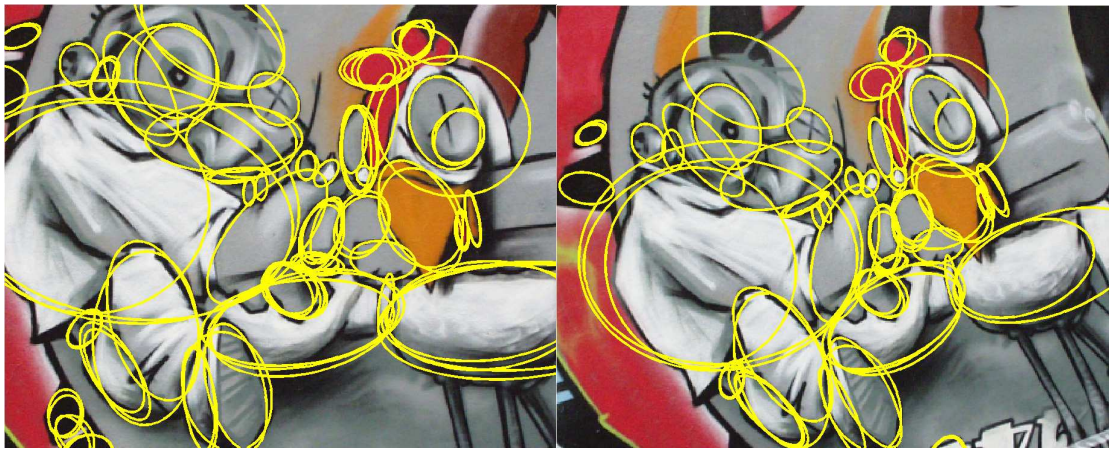


FIGURE 4.1. PCBR detections on the first and second graffiti images from the INRIA dataset

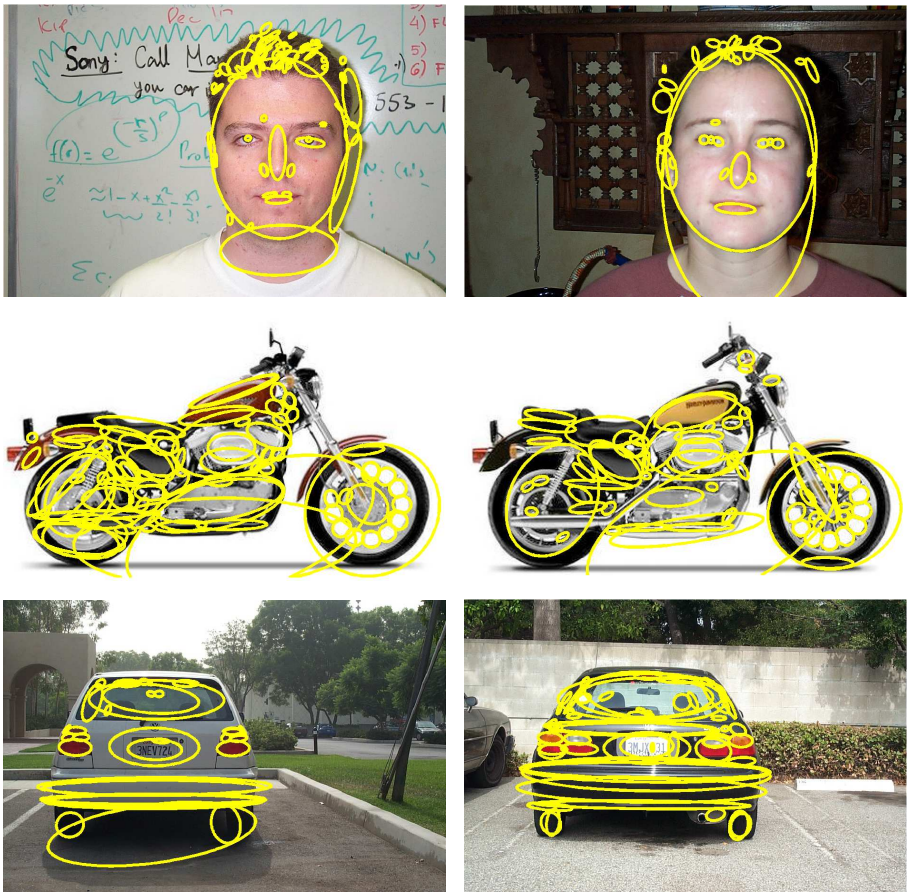
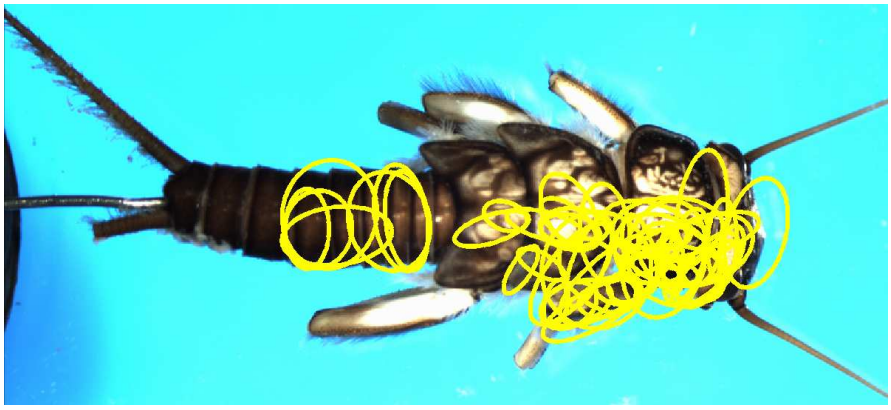


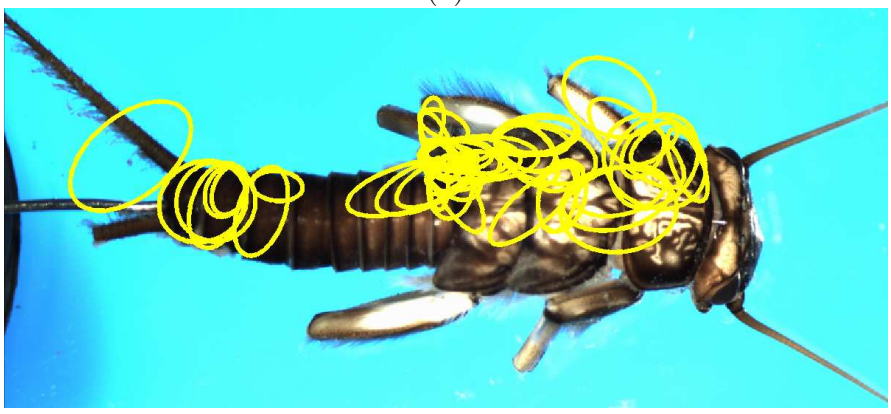
FIGURE 4.2. PCBR detections on faces, cars (rear), and motorbikes from the Caltech dataset.

4.2. Comparison with Other Detectors on a Stonefly Image

To give a visual impression on how other detectors' results on stonefly images to facilitate comaprision, we give examples in Figures 4.3, 4.4, 4.5, 4.6. As some detectors output too many regions, we show them in mutiple images to improve visibility. In these detectors, Hessian affine, MSER, Kadir are intensity-based detector while EBR and PCBR are structure-based detector. Compared with other detectors, PCBR regions are more visually meaningful.



(a)



(b)

FIGURE 4.3. EBR detections (146 regions total). (a) Regions 1-50, (b) Regions 100-146.

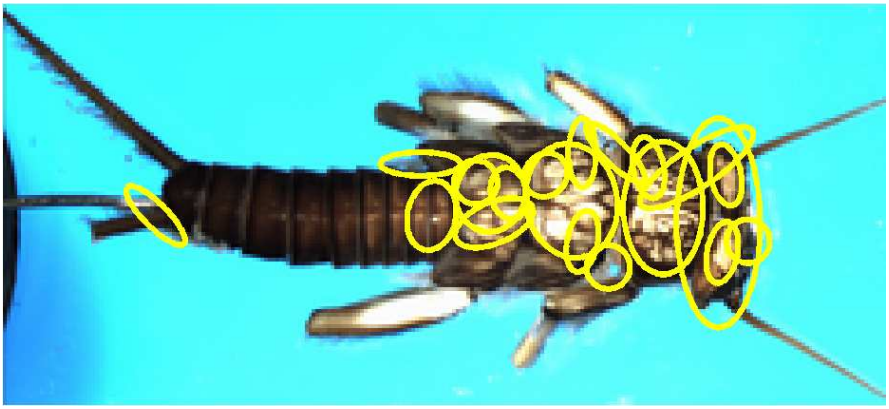
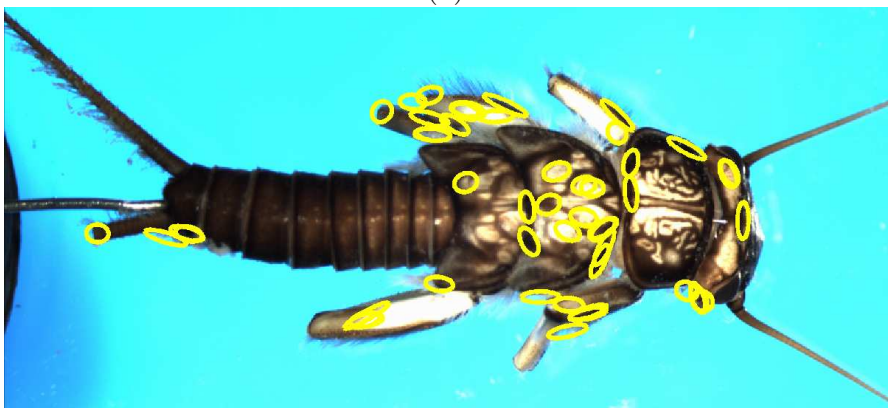


FIGURE 4.4. Kadir region detector (20 regions total)

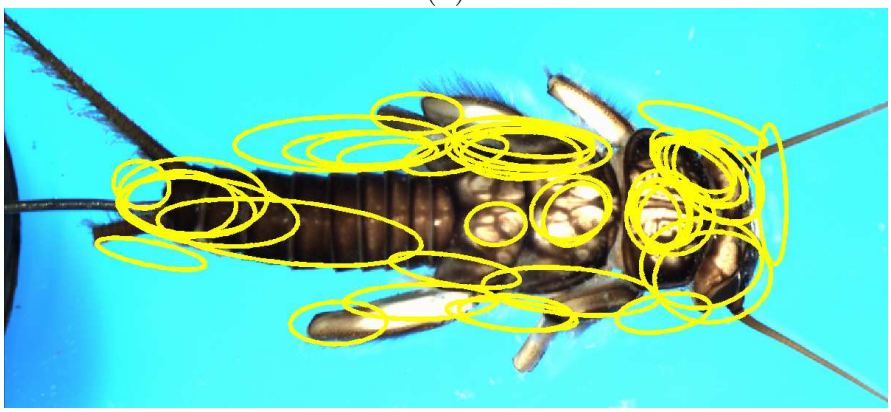
4.3. Evaluation on Repeatability

Although the PCBR detector is designed for object recognition rather than wide baseline matching, we still evaluate its repeatability and compare it to other detectors. Mikolajczyk et al. [52] provide a comparison of the performance of several popular affine-invariant interest region detectors. They focus on the repeatability of detections under various transformations. A similar comparison is performed on non-planar scenes by Fraundorfer and Bischof in [32]. The INRIA dataset and code [52] are used for these experiments. The INRIA dataset consists of six sets of images representing six types of transformations (viewpoint change, zoom-rotation, image blur, JPEG compression and lighting change). Each set contains six images with increasing degrees of transformation (Fig. 4.7). Results show that our detector (Curvilinear/PCBR) ranks within the best three on structured images (Fig. 4.8 (a-c)). This is reasonable, because our detector is based on structural cues. On textured images (Fig. 4.8 (d,f)), performance decreases. Compared with other structure-based detector (EBR), PCBR is much better.

(a)

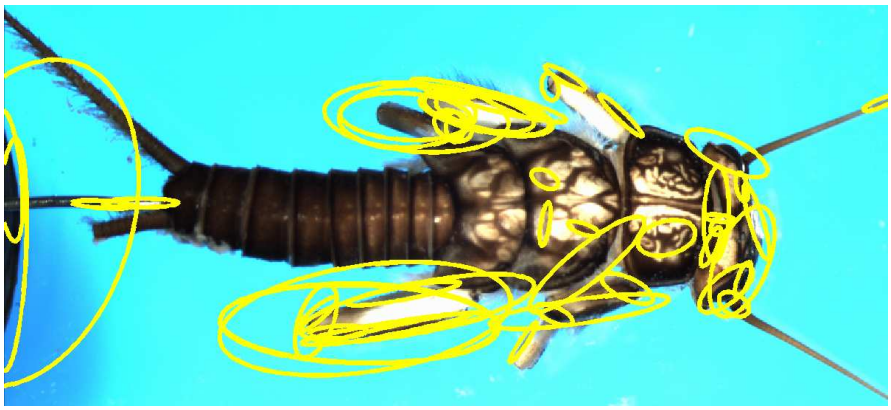


(b)

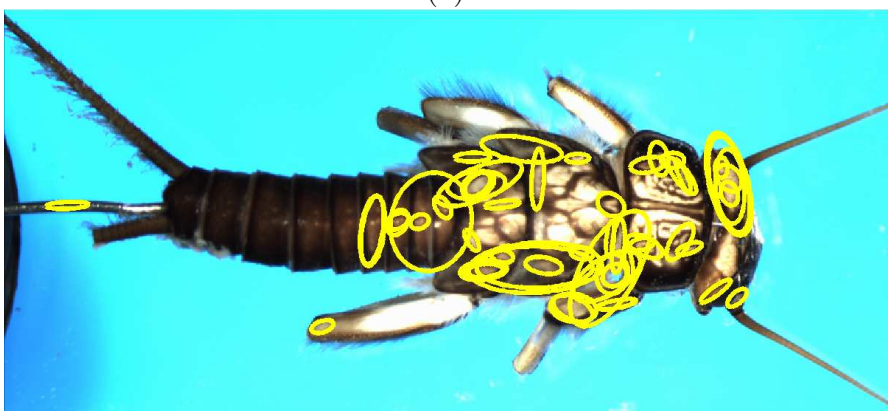


(c)

FIGURE 4.5. Hessian affine region detector (912 regions total). (a) Region 1-50, (b) Region 500-550, (c) region 870-912.



(a)



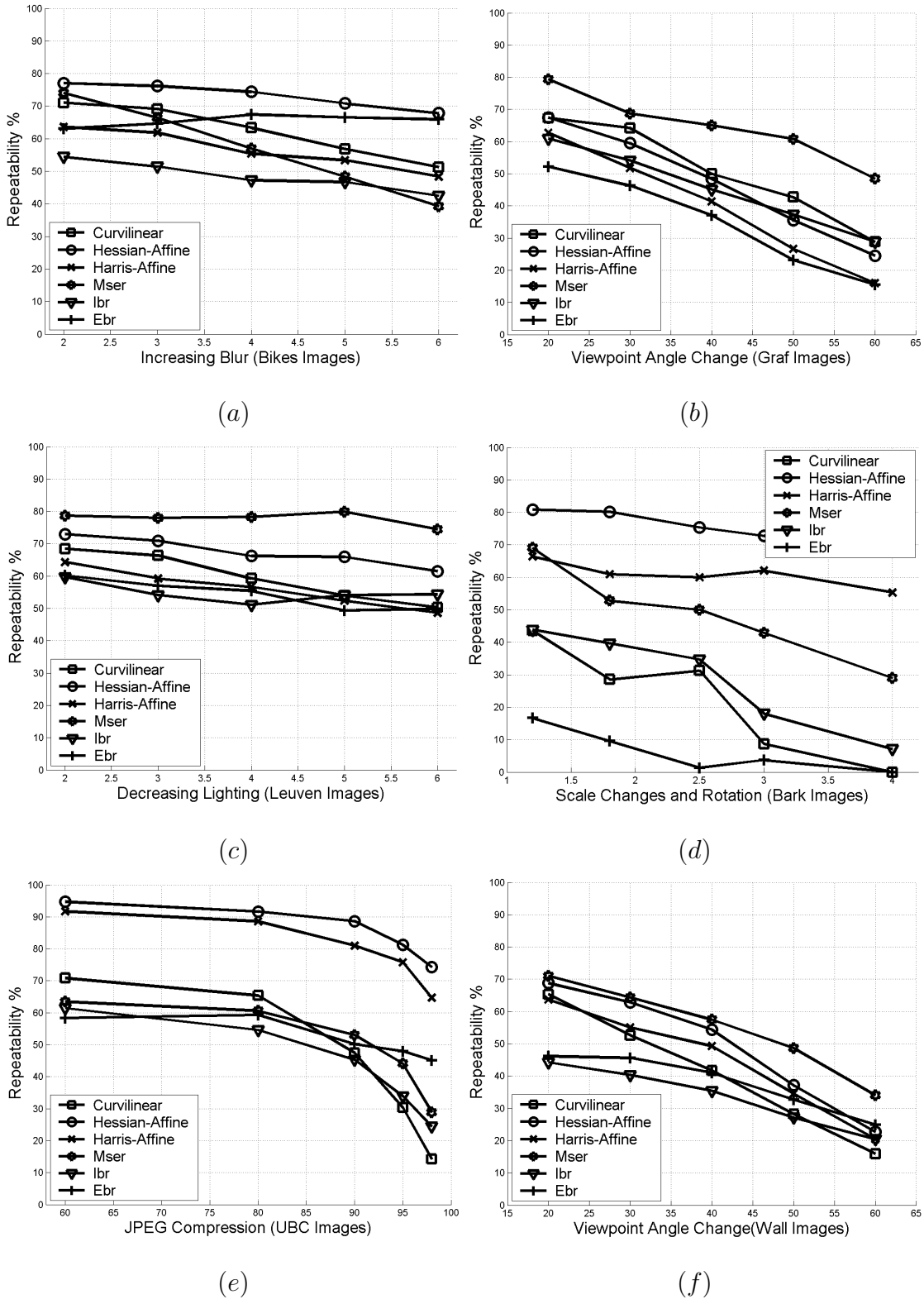
(b)

FIGURE 4.6. Mser region detector (184 regions total). (a) Region 1-50, (b) Region 130-184.

(a)

(b)

FIGURE 4.7. Inria dataset. Images and code can be downloaded from: http://www.robots.ox.ac.uk/∼vgg/research/affine/index.html



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

FIGURE 4.8. Comparisons of repeatability on images of rigid objects.

4.4. Evaluation on Information Content

For object recognition problems, a detector that achieves high repeatability on an object does not necessarily detect regions that are valuable for classification. Detector performances for object recognition can be evaluated directly by combining each detector with some recognition algorithm and testing on specific problems. But the choice of recognition algorithm can be problematic. Different choices can affect the evaluation results significantly. In this section, we propose to evaluates detectors directly based on their detections rather than on classifiers. In our method, the best detector is the one that can consistently detect the most discriminative structural or textural patterns (e.g. two eyes in human faces and rear lights of cars) in object images. The discriminating power of a pattern is its ability to discriminate objects of one class from objects of another class. This criteria can be evaluated efficiently using Maximum Mutual Information (MMI) scores. This evaluation can be performed on any object recognition dataset without the requirement for prior knowledge of homographies. MMI curves clearly reveal the characteristics of detectors for the specific object recognition problem. Additionally, the shapes of the curves are valuable guidelines for the design of classification algorithms or object models in object recognition systems. The basic idea of this method is similar to the part classifier approach developed by Dorko and Schmid [25].

4.4.1. Clustering Object Pattern

The positive images in each image dataset are randomly divided into two non-overlapping sets of equal sizes, one of the sets serve as clustering (training) images, and all the remaining images are used as evaluation (testing) images.

Each region detector is then applied to all the images. A SIFT (Scale Invariant Feature Transform) [47] descriptor is computed for each detected region. The SIFT descriptors from the training images are then clustered by fitting a Gaussian mixture model (GMM) via EM [65]. A GMM is expressed as:

$$P(x) = \sum_{i=1}^{k} P(x|C_i)P(C_i) \tag{4.1}$$

$$P(x|C_i) = N(x|\mu_i, \Sigma_i) \tag{4.2}$$

where $N(x|\mu_i, \Sigma_i)$ denotes the normal (Gaussian) distribution with mean μ_i and covariance matrix Σ_i (which is assumed to be diagonal). Each fitted Gaussian $C_i : \mu_i, \Sigma_i$ is interpreted to be a "pattern" that is analogous to a "part" in [25].

4.4.2. Maximum Mutual Information (MMI) Score

Each Gaussian pattern is described by its mean and covariance; we calculate the distance of this pattern to each evaluation image. If this distance is small, it means that the image contains a detected region that is very similar to this Gaussian pattern. Based on these distances, we sort evaluation images and train decision stumps. The discriminating power of this pattern is measured by the mutual information of the best decision stump classifier with the class labels attached to evaluation images.

There are three steps in the MMI score computation:

1. Compute distances from patterns to evaluation images. Given a Gaussian pattern $C_i : \mu_i, \Sigma_i$ and the SIFT feature sets from the N_E evaluation images, $\{F_E\} = \{F_1, \cdots, F_e, \cdots, F_{N_E}\}$, we classify the evaluation images using pattern C_i only. Inspired by the work of Opelt et al. [62], we first calculate the

distance from C_i to each evaluation image I_e. This is defined as the minimum Mahalanobis distance between C_i and the evaluation SIFT descriptor set F_e.

2. Distance-based sorting of evaluation images. Then evaluation images are sorted according to the distances calculated above. These images are labeled by +1 denoting a positive example and -1 denoting a negative example. Thus, a sorted label array $L_{i,\pi}$ consisting of positive and negative images is formed.

3. Calculate the maximum mutual information scores. The sorted label array illustrates the discriminating power of pattern C_i, A perfect pattern should have all of the positive images (+1) ranked in front followed by all of the negative images (−1). The discriminating power can be quantitatively measured by the maximum mutual information [18] (MMI) between the classification results of its pattern classifiers $\{C_{i,s}\}$ and the true class labels. The MMI score of pattern C_i is given by

$$MMI_i = max_s(MI(L_{i,\pi}, s)) \tag{4.3}$$

where the mutual information is computed by

$$\begin{aligned} MI(L_{i,\pi}, s) = & P(\bar{C_{i,s}}, \bar{O}) \log \frac{P(\bar{C_{i,s}}, \bar{O})}{P(\bar{C_{i,s}})P(\bar{O})} \\ & + P(C_{i,s}, \bar{O}) \log \frac{P(C_{i,s}, \bar{O})}{P(C_{i,s})P(\bar{O})} \\ & + P(\bar{C_{i,s}}, O) \log \frac{P(\bar{C_{i,s}}, O)}{P(\bar{C_{i,s}})P(O)} \\ & + P(C_{i,s}, O) \log \frac{P(C_{i,s}, O)}{P(C_{i,s})P(O)} \end{aligned} \tag{4.4}$$

$C_{i,s}$ is a decision stump [65] with the threshold set at position $s(1 < s < N_E)$ in the sorting array. The images before s are classified as positive, those after s are

classified as negative. The function $MI(L_{i,\pi}, s)$ calculates the mutual information between the classification results of $C_{i,s}$ and true class O. The joint probabilities in Eq.4.4 can be calculated efficiently by counting the number of true positives, false positives, and so on.

Since we don't need to detect background patterns in positive images, we require that

$$P(C_{i,s}, O) > P(C_{i,s})P(\bar{O}) \tag{4.5}$$

If **??**eq:MMI3) is not satisfied, $MI(L_{i,\pi}, s)$ returns 0. In summary, the MMI score measures the discriminating power of patterns in terms of the number of bits of information that a pattern provides about the true class label.

4.4.3. MMI Curves

We calculate the MMI scores for all the detectors under investigation; then we plot the MMI scores for each detector in descending order of mutual information. We call the resulting plots MMI curves (Fig.4.9). The performance and characteristics of a detector for object recognition can be described by the area under the curve (AUC) and shape of the MMI curve. A perfect detector would produce a set of perfectly-discriminative patterns. The corresponding MMI curve would be a horizontal line with mutual information of 1 bit (and maximum AUC).

The shape of the MMI curve can provide guidance for the choice of learning algorithm to apply. If a detector has an MMI curve that is above average but relatively flat, such as the MMI curve of the Harris-Affine detector in Fig.4.9(d), this indicates that most of the detected regions are fairly discriminative, and only a small proportion of the detections are very noisy. Under this situation, learning

algorithms such as Fisher linear discriminants, perceptrons, or neural networks, which tend to assign equal weight to all input attributes, will likely give high recognition performance.

On the other hand, if a detector generates a curve that has very high scores for the top ranking patterns but relatively low scores for the following patterns, for example, the MMI curve of PCBR (or curvilinear) detector in Fig.4.9(b), it shows that the detector can only find a few highly consistent and discriminative regions while at the same time producing many relatively low-performance detections. Under this situation, learning algorithms such a decision trees (combined with boosting, bagging, or randomization) would probably perform better, because they incorporate feature selection (typically based on mutual information) as a fundamental part of the algorithm [25, 62].

In summary, the heights and shapes of MMI curves help us understand the discriminating power of detections. This provides valuable guidance for the selection of appropriate detectors and the choice of classification algorithms.

4.4.4. Evaluation Results

We experimented with some of the standard datasets studied by the computer vision community: the Caltech set used in [25]. Four object classes (leaves, faces, cars markus, cars brad) with different characteristics are selected. These image sets are used as the positive images in the evaluation framework. Negative images are the corresponding background images provided in the datasets. IBR and EBR do not provide reasonable number of detections in the cars brad image set. As advised by their authors, we resized the images to three times their original size. Detectors are applied on the resized images without any other preprocessing.

GMM clustering was performed with $K = 50$ clusters. To test the robustness of the MMI curves to the choice of the cluster number K, we repeated the evaluation experiments with K = 100. For all the datasets; the relative rankings of detectors remained unchanged.

The MMI evaluation results are shown in Fig.4.9. We can see that the performance of PCBR(curvilinear) detector is very good on the leaves, faces, cars markus and cars brad image set (Fig.4.9(a-d)).

The MMI curves of the PCBR detector usually start with high scores but soon drop because of noise detections. On most objects, the PCBR detector is able to find several highly distinctive and discriminative patterns which can represent the local or global characteristic features of objects. For example, note the scores of the top 5 patterns in Fig. 4.9(a) and (b). This implies its potential utility when combined with feature selection algorithms and constellation models.

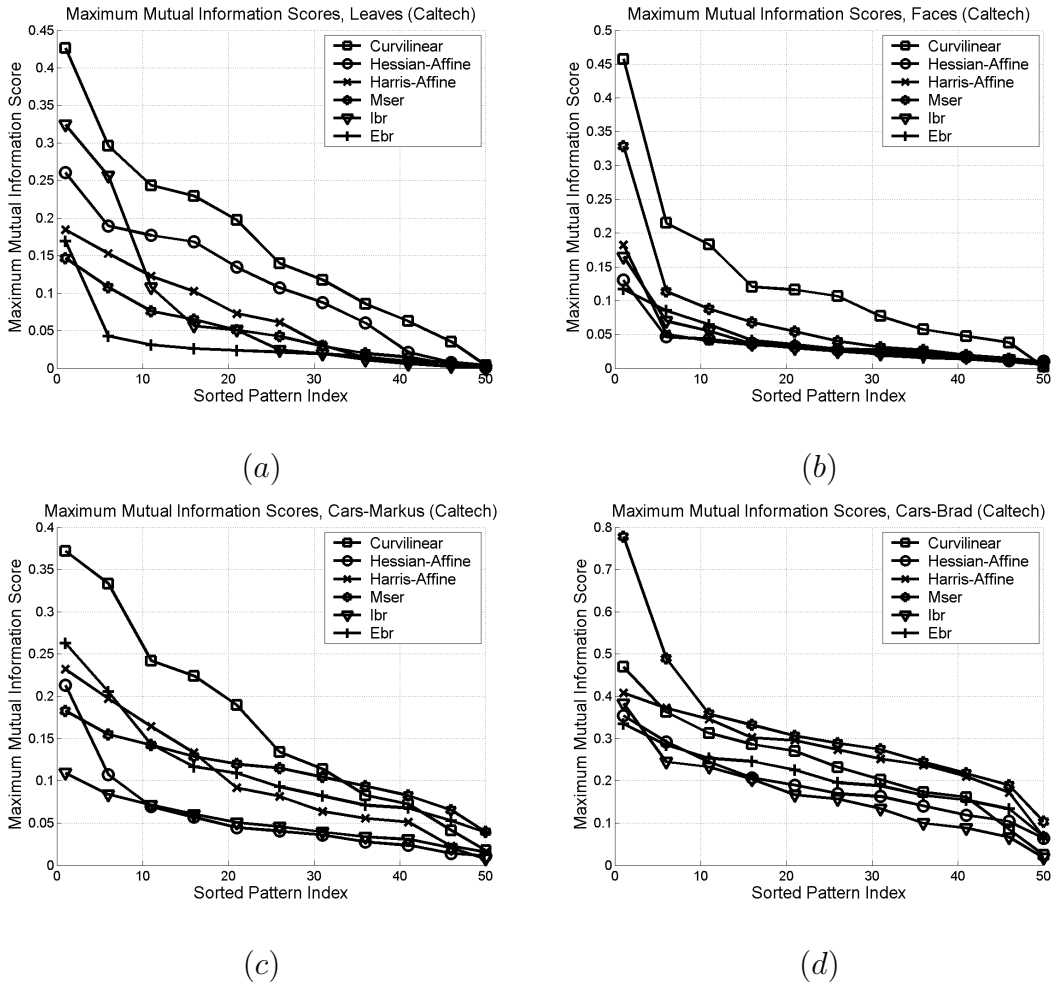


FIGURE 4.9. Maximum mutual information evaluation results. (a)leaves, (b)faces, (c)cars markus, and (d)cars brad.

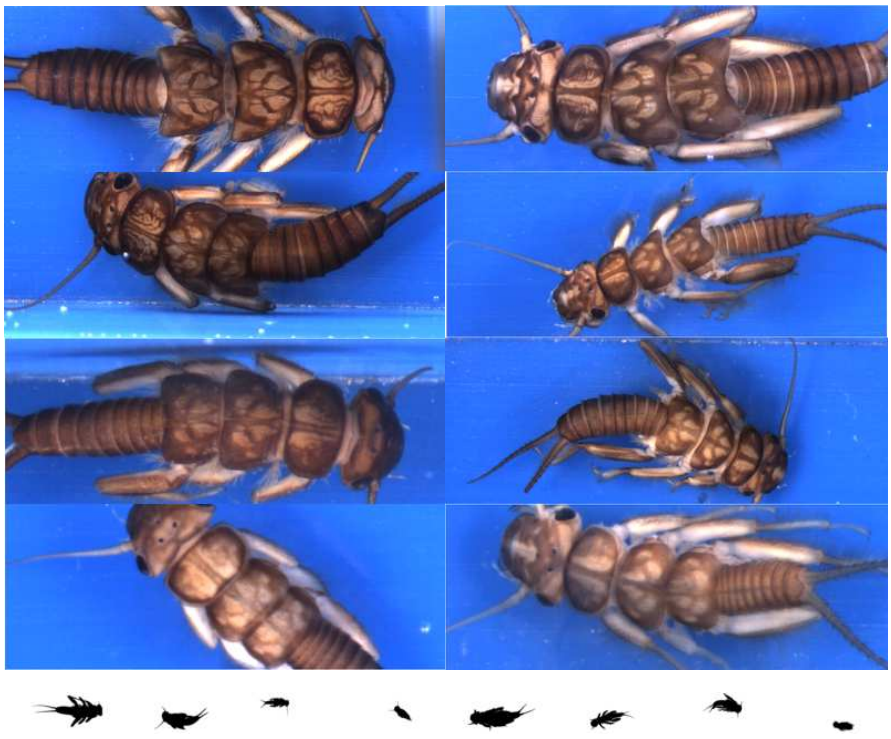
5. APPLICATIONS

5.1. Object Recognition on Stonefly Dataset

Population counts of larval stoneflies inhabiting stream substrates are known to be a sensitive and robust indicator of stream health and water quality. Consequently, automated classification of stonefly larva can make great strides in overcoming current bottlenecks–such as the considerable time and technical expertise required–to large scale implementation of this important biomonitoring task. As such, we evaluate the effectiveness of our PCBR detector on a more fine-grained object-class recognition problem, that of distinguishing between two related species of stonefly larva, *Calineuria californica* and *Doroneuria baumanni*. These two stonefly species are from the same taxonomic family and, as such, are very similar in appearance. Indeed, this problem is challenging even for humans and is akin to visually distinguishing between nearly identical car models. As such, this problem is more difficult than differentiating between faces and airplanes as per the Caltech dataset.

Figure 5.1 (a-b) shows images of four specimens (and their relative sizes) from each of the two taxa. To verify the difficulty of discriminating these two taxa, we conducted an informal study to test the human classification accuracy of *Calineuria* and *Doroneuria*. A total of 26 students and faculty were trained on 50 randomly-selected images of *Calineuria* and *Doroneuria*, and were subsequently tested with another 50 images. Most of the subjects (21) had some prior entomological experience. The mean human classification accuracy is 78.6% correctly identified (std. dev. = 8.4).

We compare PCBR with the Kadir salient region detector [38] and the Hessian-affine detector [52] on the stonefly recognition problem. All classification

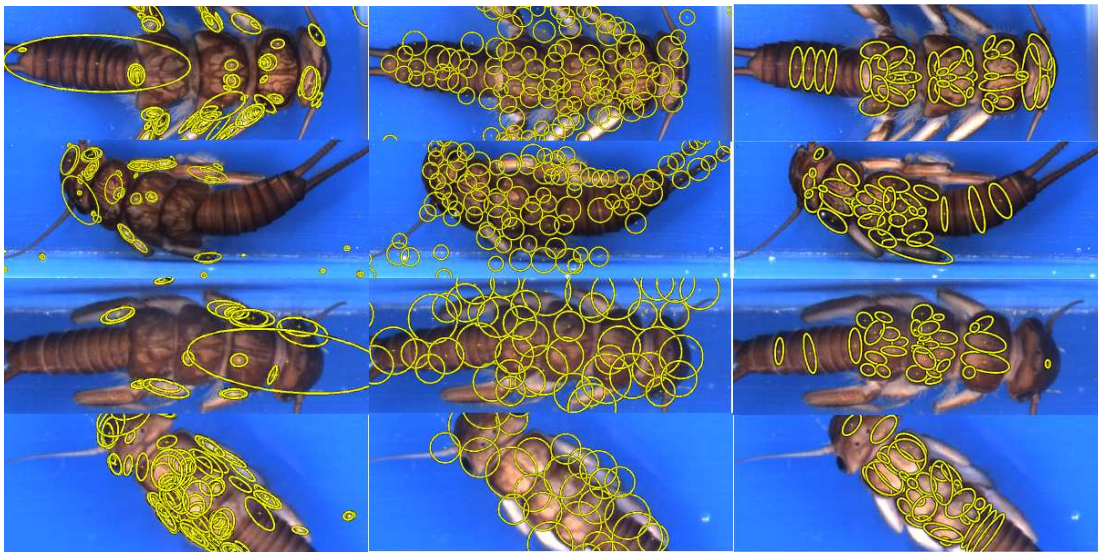


(*a*) (*b*)

FIGURE 5.1. Visual comparison of *Calinueria* and *Doroneuria* and their relative specimen sizes. (a) Four different *Calinueria* and (b) *Doroneuria* specimens.

settings are identical except for the detector. Figure 5.2 shows the detections for the four *Calinueria* images in Fig. 5.1(a). Notice again how well distributed and consistent the PCBR detections.

We apply two state-of-the-art object-class recognition algorithms to the stonefly dataset: logistic model trees (LMT) by Landwehr [41] and Opelt's method [62]. We use our own LMT implementation and use Opelt's Matlab code (adapted to use other detectors). The number of specimens and images used in this experiment is listed in Table 5.1 while Table 5.2 summarizes the classification accuracy



(*a*) (*b*) (*c*)

FIGURE 5.2. Comparison of three detectors on *Calinueria* images. (a) Hessian-affine, (b) Kadir salient regions, and (c) PCBR

Taxon	Specimens	Images
Calineuria	85	400
Doroneuria	91	463

TABLE 5.1. Specimens and images employed in the study.

for this two-class recognition problem. As can be seen, both classifiers yield better recognition accuracy with the PCBR detector than with the other two detectors.

5.2. A Hierarchical Object Recognition System based on PCBR

We also constructed a hierarchical object recognition system using multiscale PCBR regions. This system is composed of layer classifiers using an im-

Hessian Affine	Kadir Entropy	PCBR	Accuracy[%]	
			Opelt [62]	LMTs [41]
√			60.59	70.10
	√		62.63	70.34
		√	67.86	79.03

TABLE 5.2. *Calineuria* and *Doroneuria* classification rates comparison of different detectors when applied with Opelt's method and LMTs. A √ indicates that the corresponding detector is used.

proved boosting feature selection method [62]. These layer classifiers select the most discriminative features and use them to classify. All layer classifiers are then combined to give the final classification. This system is tested on various object recognition problems. Experimental results show that the new hierarchical system outperforms the comparable solutions on most of the datasets tested. The good performance of this classifier also shows the power of the PCBR detector.

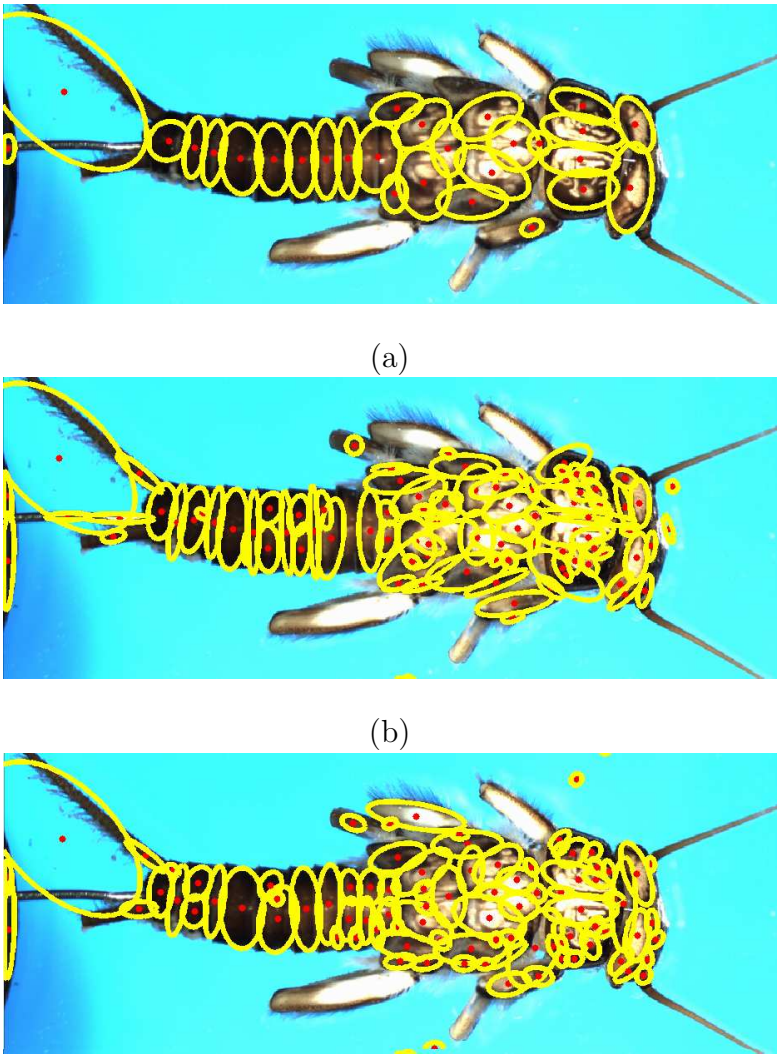
5.2.1. PCBR detection

The description of object classes is a crucial issue in the design of object recognition systems. Previous description methods include single-scale fragment-based [78] and local interest-region [1, 25, 27, 62] approaches. While fragment or part features are usually very informative for object categories, they can be too class-specific and are not transform invariant. Interest regions are more generic and more robust to occlusion and transformations, but they are too local and often noisy. Probabilistic constellation models [27] and clustering-based methods

[25] have been proposed to recognize image categories based on these fragments or interest regions. Our PCBR region detector belongs to local interest-region detector, it is not only robust to occlusion and transformation as other local interest region detectors but also not too local and has fewer noisy regions. As such, it is a good detector for object category recognition.

In this project, rather than describe objects at a single scale, we represent objects at multiple scales. These multiscale descriptions are biologically motivated - the human visual system selects and combines both coarse (global) and detailed (local) object features for recognition. Shokoufandeh et al. [70] use saliency map graphs to capture the salient image structure with multi-scale wavelet transforms. Epshtein and Ullman [26] propose feature hierarchies based on mutual information feature selection and parameter adaptation. The work of Bouchard and Triggs [8] models each object as a hierarchy of parts and subparts with partial transformations (translation and scale transformations) that softly relate the parts and sub-trees to their parents. But there is a common weakness existing in these hierarchical object descriptions: all these descriptions are highly concrete models (trees or graphs). Applying these types of descriptions to classification requires graph matching [70] or model instantiation [8, 26] algorithms. In section 4.4 PCBR regions have been showned to be appropriate for feature selection-based classifiers. Based on its strong detections, we don't need to build complicated model to achieve good performance.

To detect interest regions, we use outputs of the PCBR detector in multipile fixed scales. Figure 5.3 shows an example of detections with $\sigma = 4, 2, 1$ respectively.



(a)

(b)

(c)

FIGURE 5.3. PCBR detections in different scales. (a) $\sigma = 4$, (b) $\sigma = 2$, and (c) $\sigma = 1$

5.2.2. PCBR Region Descriptions

PCA-SIFT [39] and statistical measurements [85] are both used in describing PCBR regions. The PCA-SIFT features are 36-dimensional and have been demonstrated to be more compact and distinctive than the SIFT descriptor [39]. The statistical feature combines the coefficient of variation (Eq. 5.1), skewness (Eq. 5.2), kurtosis (Eq. 5.3), and moment invariants (Eq. 5.5) to form a 9-dimensional feature vector for each region. These statistical measures are a good complement to PCA-SIFT as they are more robust to variations and image transformations. The coefficient of variation, skewness, and kurtosis are given by

$$C_v = \frac{\sigma}{\mu}, \tag{5.1}$$

$$\gamma = \frac{\sum_{y=0}^{m-1}\sum_{x=0}^{n-1}(I(x,y)-\mu)^3}{(m*n-1)C_v^3}, \tag{5.2}$$

and

$$\beta = \frac{\sum_{y=0}^{m-1}\sum_{x=0}^{n-1}(I(x,y)-\mu)^4}{(m*n-1)C_v^4}, \tag{5.3}$$

respectively, where σ is the standard deviation, μ is the mean value, and $I(x,y)$ is intensity value for a pixel in position (x,y).

The moment invariants are given by

$$\mu_{pq} = \sum_{y=0}^{m-1}\sum_{x=0}^{n-1}(x-\bar{x})^p(y-\bar{y})^q I(x,y)$$
$$M_1 = \mu_{20} + \mu_{02}$$
$$M_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$
$$M_3 = (\mu_{30} - 3\mu_{12})^2 + 3(\mu_{21} - \mu_{03})^2$$
$$M_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$
$$M_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2)$$
$$+(3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})((3(\mu_{30} + \mu_{12})^2) - (\mu_{21} + \mu_{03})^2)$$
$$M_6 = (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$$
$$M_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2)$$
$$+(\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) \tag{5.4}$$

$$\beta_1 = \frac{\sqrt{M_2}}{M_1}$$
$$\beta_2 = \frac{M_3\mu_{00}}{M_1 M_2}$$
$$\beta_3 = \frac{M_4}{M_3}$$
$$\beta_4 = \frac{\sqrt{M_5}}{M_4}$$
$$\beta_5 = \frac{\sqrt{M_6}}{M_1 M_4}$$
$$\beta_6 = \frac{\sqrt{M_7}}{M_5} \tag{5.5}$$

where

$$\bar{x} = \frac{\sum_{y=0}^{m-1}\sum_{x=0}^{n-1} x * I(x,y)}{\sum_{y=0}^{m-1}\sum_{x=0}^{n-1} I(x,y)},$$
$$\bar{y} = \frac{\sum_{y=0}^{m-1}\sum_{x=0}^{n-1} y * I(x,y)}{\sum_{y=0}^{m-1}\sum_{x=0}^{n-1} I(x,y)} \tag{5.6}$$

are the center of gravity. In addition to using the region descriptors themselves, we characterize the spatial configuration of the regions with bins-based cluster index

distribution histograms. The construction of spatial relation features involves three steps. First, we cluster the PCA-SIFT features from the positive training images using E-M algorithm to fit a Gaussian mixture model (GMM) with $C = 16$ clusters. Second, for each region in the training and testing images, we compute the index of the Gaussian cluster that is most likely to generate its PCA-SIFT vector. And third, we discretize the distances and directions between regions into $M = 36$ bins with 12 directions and 3 distance ranges. The sizes of the bins are fixed relative to the image sizes. Thus, the spatial configuration of regions in each image is described by a histogram R composed of $D = CMC = 16*36*16 = 9216$ feature elements. An element $R_{i,m,j}$ in R records the number of times a region with cluster index j falls into bin m with center region index i.

5.2.3. Hierarchical Object Recognition System

Using our new object descriptions, we design a hierarchical object recognition system which uses multi-scale image analysis to do classification. This system is illustrated in Figure 5.4. From the top layer to the bottom, we train layer classifiers $L_1, \ldots, L_n$ based on the region features obtained at scales $s_1, \ldots, s_n$, which are in decreasing order (global to local). We then combine the outputs of layer classifiers to predict the class labels of new images.

5.2.3.1. Layer Classifier

Using our new description method above, object images are described by normal feature vectors of three types (intensity statistical features, PCA-SIFT, and spatial relation features) rather than concrete models. This permits standard classification algorithms to be employed as layer classifiers. According to our

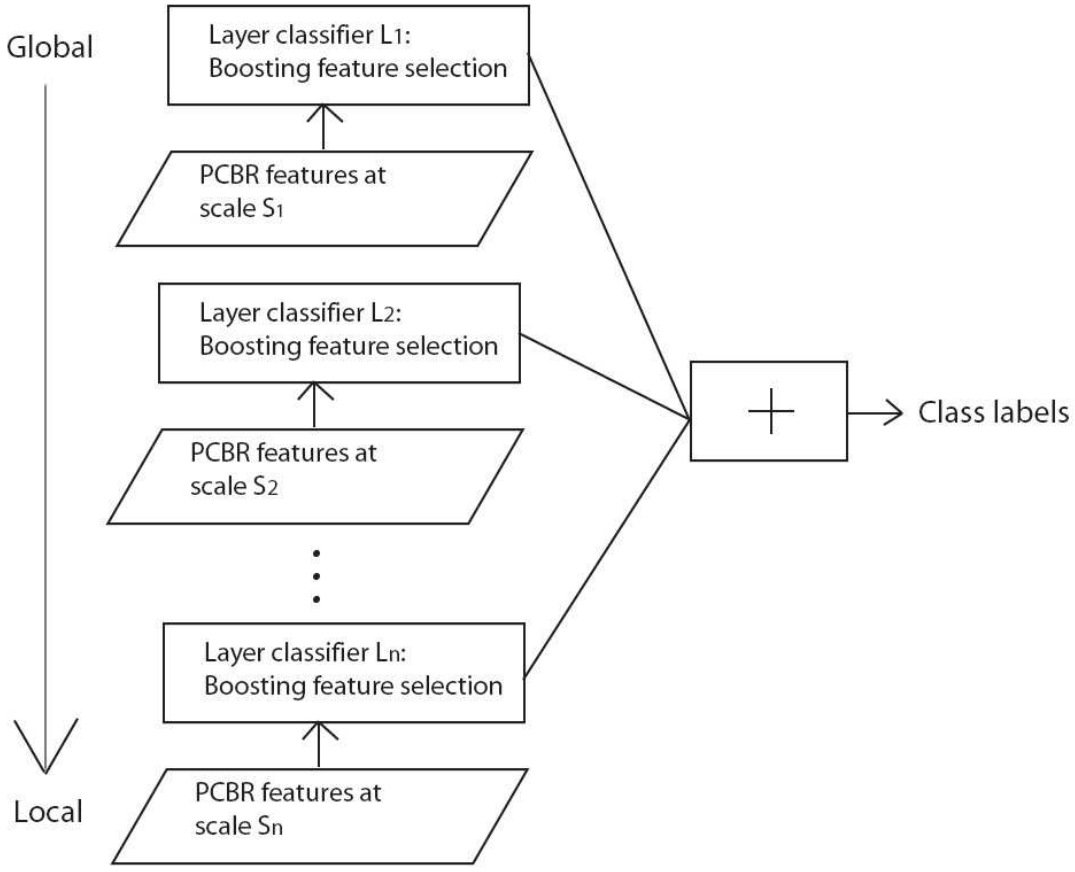


FIGURE 5.4. Hierarchical Object Recognition System.

experiments, we notice that for most of the image sets, only a small portion of the image features are useful for classification. We therefore employ and improve the boosting feature selection algorithm proposed by Opelt et al. [62] that searches among all the available features and automatically selects the most stable and discriminative ones to form the final classifier.

The layer classifiers are learned using the AdaBoost algorithm which maintains a weight for each training image. In iteration t of AdaBoost, all the unselected feature vectors of the training images are evaluated based on the current image weights to find the most discriminative feature.

We evaluate the statistical intensity features and the PCA-SIFT features in the same way as Opelt et al. [62]. The stability and discriminating power of a feature vector v_f is evaluated in three steps. First, calculate the distance from v_f to each of the training images. This is done by finding the minimum distance between v_f and all the feature vectors of the same type in the training image. We use the Mahalanobis distance metric for the statistical intensity feature and the Euclidean distance for PCA-SIFT. Second, sort the training images into ascending order according to their distances to v_f. Third, we apply the scanline algorithm [62] to the sorted distance array to determine a threshold θ_f that maximizes the weighted accuracy of using v_f as a weak classifier. The maximal weighted sum is adopted as the evaluation of v_f.

Evaluating the spatial relation features is simpler because there is no need to calculate the feature-to-image distances. The training images are directly sorted according to their spatial relation feature values. More specifically, all the spatial relation features of K training images are assembled into a $D \times K$ matrix A (where D is the dimension of the spatial configuration histogram). Then for each row of A, training images are sorted by decreasing order of their corresponding feature

values. Finally, the scanline algorithm scans the sorted array and outputs the optimal threshold and the maximal weighted sum evaluation for the row, which indicates the significance of the specific spatial configuration for classification.

A perfect feature should have all of the positive images (+1) sorted before all the negative images (-1) so that the feature vector gives a weak classifier that is perfectly discriminative. The feature and threshold v^*, θ^* which has maximal score among all the available feature vectors is selected as the weak classifier for iteration t. We construct T weak classifiers for each layer. All these T weak classifiers are then combined into a strong classifier (called the layer classifier) using standard AdaBoost. The output of a strong classifier L_i is given by

$$y_t = \sum_{t=1}^{T} (\ln \beta i, t) h_{i,t}(I) \tag{5.7}$$

with

$$\beta i, t = \sqrt{\frac{1 - \epsilon_{i,t}}{\epsilon_{i,t}}} \tag{5.8}$$

where $h_{i,t}(I)$ represents the output of the t^{th} weak classifier of layer classifier L_i. $\epsilon_{i,t}$ is the weighted classification error rate of the t^{th} weak classifier computed based on the AdaBoost weights.

For presence/absence 2-class object recognition problems, it is not plausible to use negative features to recognize positive examples. So we modified the original algorithm in [62] to select only among the features from positive images.

5.2.3.2. Final Classification

The final result of the hierarchical system is simply the sign of the sum of the outputs of layer classifiers, which is given by

$$Y = sign(\sum_{i=1}^{n} y_i). \tag{5.9}$$

In our tentative experiments, we also tried to set weights for layer classifiers, and use the Voted Perceptron algorithm to adapt the weights to minimize the classification error on training images, but it overfits the data and the performance degrades.

5.2.4. Experimental results

We did experiments on various 2-class object recognition image sets in order to test the performance of our system. We employed a four-layer system with scales of $4.0, 3.0, 2.0$, and 1.0 and $T = 100$ boosting iterations. The system is tested on six object classes in the Caltech dataset : airplanes (1074), cars (rear) (526), cars (side) (123), faces (450), leaves (186) and motorbikes (826). The background set in Caltech contains 451 images. We also tested on a stonefly larva set containing 70 *Doroneuria* images (positive) and 57 *Hesperoperla* images (negative). Examples of Caltech images and stonefly images are shown in Figure 5.5. Half of the images in each set are used for training, and the rest are held out for testing. Recognition performance is evaluated by ROC equal error rates.

The hierarchical system based on the new descriptions is tested on these datasets and compared with the constellation model of Fergus et al. [27] and the boosting feature selection approach by Opelt et al [62]. The results are summarized in Table 5.3. The comparison indicates that our hierarchical object recognition system outperforms the other methods on most of the comparable datasets.

In order to test the value of our hierarchical structure, we compared the equal error rates of the entire 4-layer system (denoted as 4-layer with spatial)

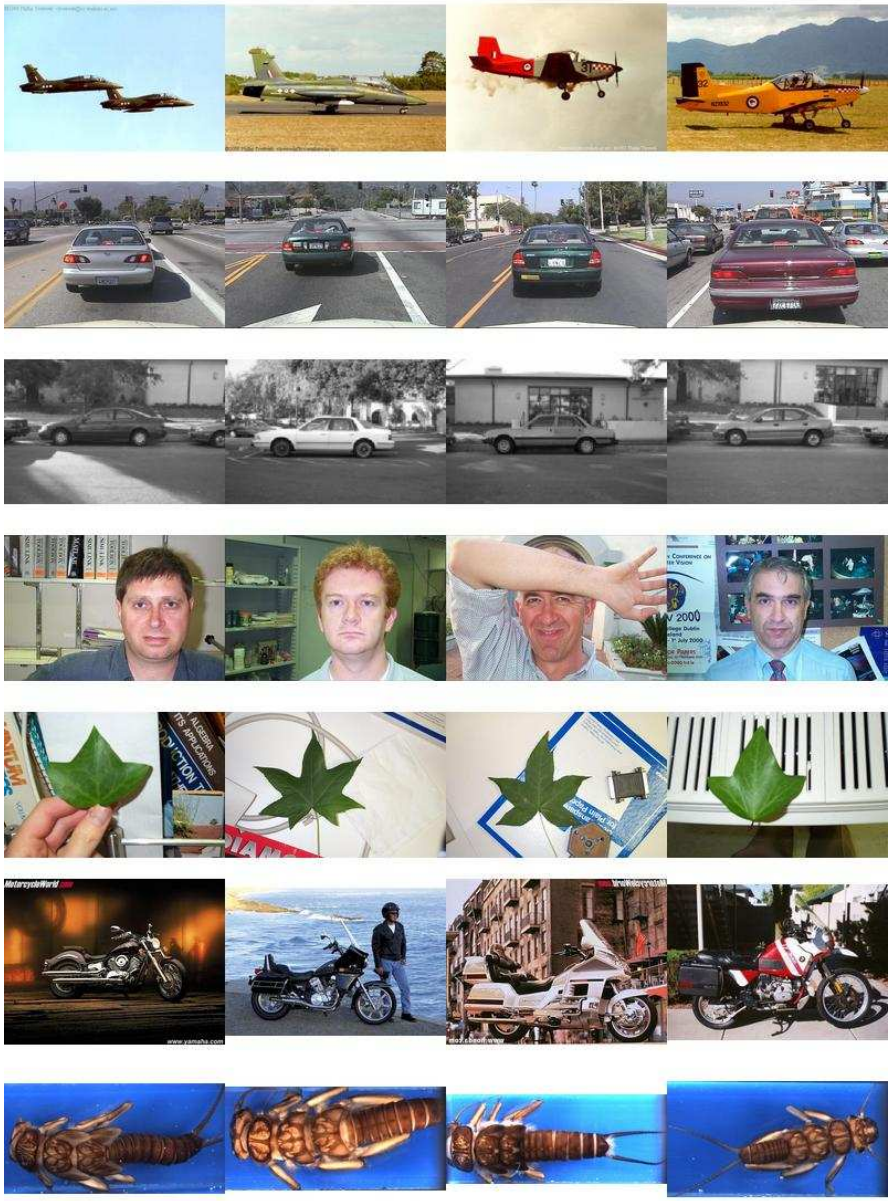


FIGURE 5.5. Sample images from Caltech and stonefly larva dataset with rows corresponding to: airplanes, cars (rear), cars (side), faces, leaves, motorbikes and stonefly images)

Dataset	Ours	Fergus [27]	Opelt [62]
Airplane	90.6	90.2	88.9
Cars(rear)	94.3	90.3	/
Cars(side)	83.6	88.5	83.0
Faces	98.8	96.4	93.5
Leaves	97.5	/	/
Motorbikes	94.3	92.5	92.2
Stoneflies	88.6	/	/

TABLE 5.3. ROC equal error rates of our approach and other approaches

to the best single layer classifier (1-layer). The results are summarized in the second and third columns of Table 5.4. In the forth column of Table 5.4, we show the performance of the 4-layer system without spatial relation features (4-layer without spatial) to test the utility of the spatial configuration descriptor.

We noticed that on all these datasets, there are significant gaps between the performance of the multi-layer system and that of the best one-layer classifier. This demonstrates that the multi-scale object description is more generic and informative for object classes than single scale description.

On most of the datasets, spatial relation features improve the performance of the system, thus supporting our claim that spatial configurations of detected regions are also valuable cues for recognition.

Dataset	4-layer with spatial	1-layer	4-layer without spatial
Airplane	90.6	89.0	90.0
Cars(rear)	94.3	91.0	89.2
Cars(side)	83.6	81.6	80.3
Faces	98.8	97.2	98.8
Leaves	97.5	96.0	97.3
Motorbikes	94.3	92.0	93.5
Stoneflies	88.6	80.0	82.9

TABLE 5.4. ROC equal error rates of our full implementation compared to single-layer (with spatial relation) and 4-layer (w/o spatial relation).

5.3. Symmetry Detection

Symmetry is quite common in biological and artificial objects. Symmetry detections have been used in various computer vision applications such as, image analysis [56], reconstruction [81], object detection [82], etc. Since our PCBR detector detects robust structure-based interest regions, it is also good at detecting symmetrical regions in images containing objects with bilateral symmetry. To demonstrate this, we combine the PCBR detector with the SIFT-based symmetry detection method and test it on various images. Similar work is also done by Loy and Eklundh [49].

We apply symmetry detection to choose good dorsal (or back side) views of stonefly larvae from among the various poses. Dorsal views exhibit more bilateral symmetry than do other poses. As such, symmetry detection is a useful mechanism for identifying those images that are best for classification. Figure 5.6 shows

various poses of the stoneflies as contained in the database. The PCBR detector is better at finding bilaterally-symmetric regions in the stonefly images than are other detectors.

FIGURE 5.6. Different object poses in the stonefly database.

Given the PCBR detections (Fig. 5.7(b)), two regions are symmetrical, for example, region 2 and 6 in 5.7(b), if one SIFT descriptor is similar to the others mirrored descriptor. The SIFT descriptor [47] computes the gradient vector for each pixel in a feature regions neighborhood and builds a normalized histogram of gradient directions. Fig. 5.7 (c)(d) show normalized regions 2 and 6. Their SIFT descriptors are shown in the upper two figures in Fig. 5.7 (e) respectively. A regions' SIFT descriptor can be mirrored by rearranging the 128 descriptor entries according to the following reordering indices:

97 104 103 102 101 100 99 98 105 112 111 110 109 108 107 106 113 120 119 118 117 116 115 114 121 128 127 126 125 124 123 122 65 72 71 70 69 68 67 66 73 80 79 78 77 76 75 74 81 88 87 86 85 84 83 82 89 96 95 94 93 92 91 90 33 40 39 38 37 36 35 34 41 48 47 46 45 44 43 42 49 56 55 54 53 52 51 50 57 64 63 62 61 60 59 58 1 8 7 6 5 4 3 2 9 16 15 14 13 12 11 10 17 24 23 22 21 20 19 18 25 32 31 30 29 28 27 26

These reordering indices are computed by mirroring the SIFT bins about the dominant gradient orientation about which each SIFT descriptor is constructed. The reordered SIFT descriptors are shown in the lower two images in Fig. 5.7(e). From Figure 5.7(e) we can see that the SIFT descriptor of region 2 is very similar to the mirrored or reflected SIFT descriptor of region 6 and vice versa.

The reordering of SIFT descriptors provides an effective way of comparing symmetrical regions. If a region's descriptor is very simiar to the reordering of another region's descriptor, they are symmetrical regions. As the SIFT descriptor is rotationally invariant, this symmetrical detection is also rotationally invariant. This is very helpful in our domain because our symmetrical regions normally have different orientations.

Based on this principle of symmetrical region detection, the whole process of detecting dorsal views is as follow:

1. Apply the PCBR region detector to an image.

2. Generate SIFT descriptors and their mirrors for all detected n regions.

3. Build an $n * n$ symmetrical similarity distances matrix M. Each matrix entry is the Euclidean distance between the SIFT descriptor of a region and the reordered(or mirrored) SIFT descriptor of another region. The diagonal entries of this matrix are set to a large number as we don't need to compare a region with itself.

4. Choose the entries in M that are less than a threshold. These entries define potential matched symmetrical regions.

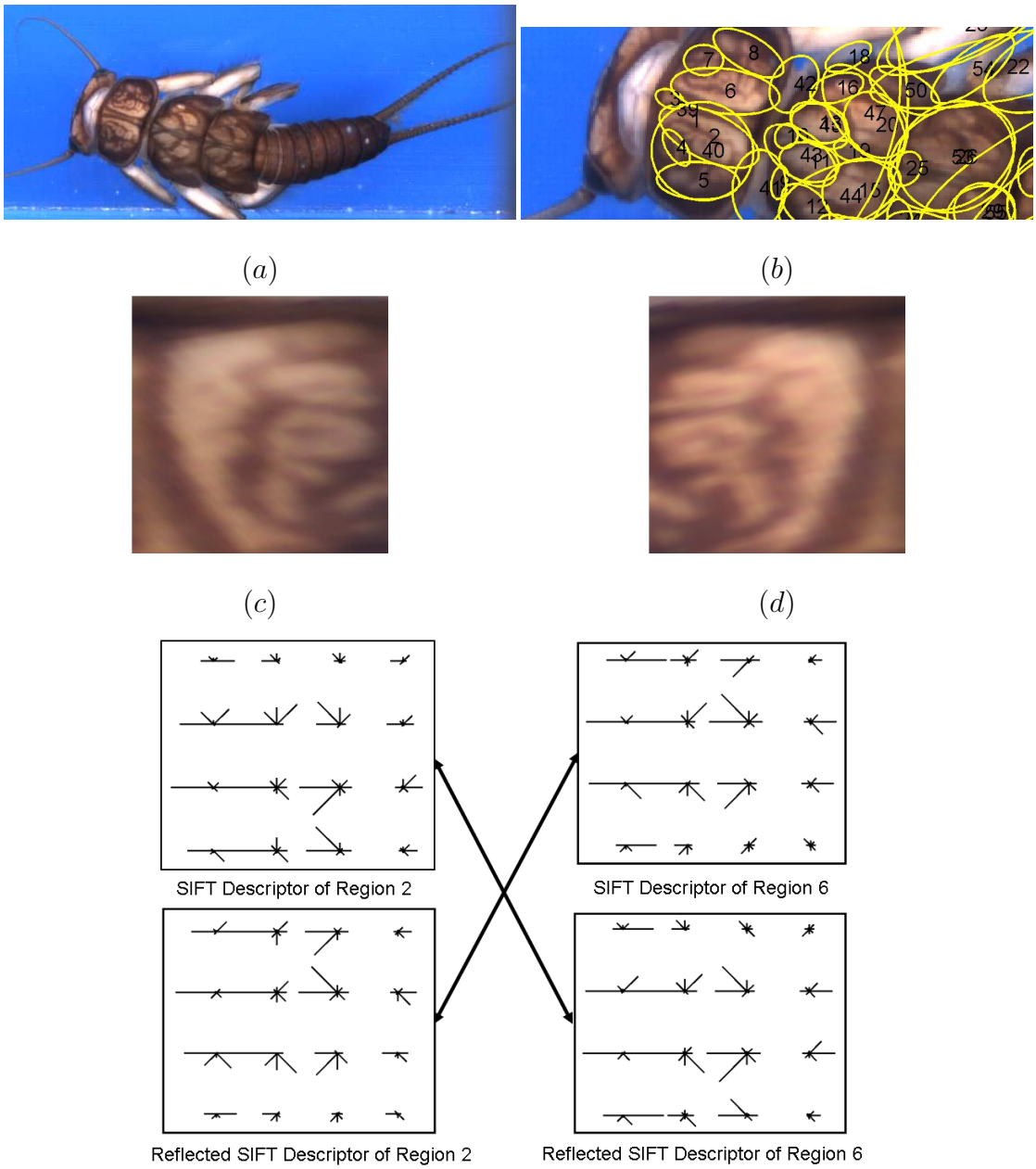


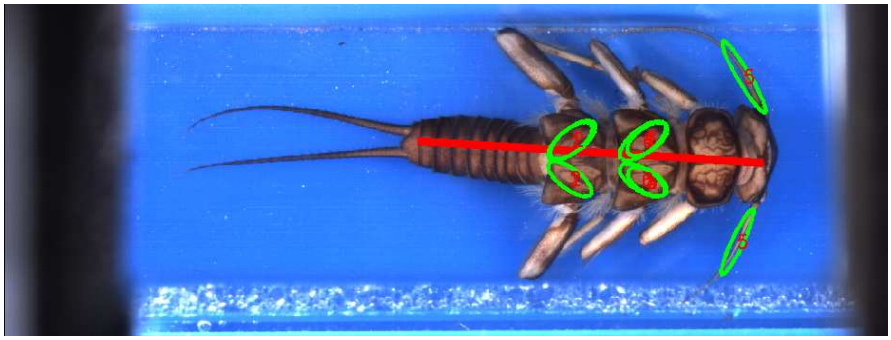
FIGURE 5.7. Symmetrical region detection by reflected SIFT descriptor. (a) The original image, (b) Regions detected by PCBR, (c) Normalized region 2, (d) Normalized region 6, (e) SIFT and reflected SIFT of region 2 and 6.

5. Check all these regions, remove those whose size is larger than half size of the image.

6. Calculate the stonefly's orientation (red line in Fig. 5.8) by computing the principal component of the segmented stonefly image(Algorithm 1).

7. Check all the regions that are below the threshold and calculate the angle between the connecting line of the corresponding regions and bug's principal orientation. If the angle is greater than 60°, eliminate this pair.

8. Count the number of matched pairs. If it is greater than a predefined number, this image is classified as a good dorsal view image, otherwise, it is not.



(a)

FIGURE 5.8. Orientation of stonefly (red line).

With the use of the PCBR detector, detecting symmetrical regions is a very effective technique for finding dorsal view in images. Figure 5.9 shows detected images with good dorsal views out of the images in Figure 5.6.

We also apply this method to other images, Figure 5.10 shows the symmetrical detections in several other images. We can see that the detected symmetrical

FIGURE 5.9. Good dorsal views selected using bilateral symmetry detection with PCBR.

regions are quite accurate and distinctive and provide valuable cues for the detection and recognition of symmetrical objects.

Algorithm 1 Calculate Bug's orientation

```
1: procedure BUGORIENTATION(B, m, n)          ▷ B is the segmented bug image, m:height, n:width
2:     k ← 0, sumX ← 0, sumY ← 0
3:     for i ← 1, m do
4:         for j ← 1, n do
5:             if B(i, j) > 0 then
6:                 X(k) ← j
7:                 Y(k) ← i
8:                 sumX ← sumX + j
9:                 sumY ← sumY + i
10:                k ← k + 1
11:            end if
12:        end for
13:    end for
14:    cX ← sumX/(m ∗ n)
15:    cY ← sumY/(m ∗ n)
16:    for i ← 1, k − 1 do
17:        X(i) ← X(i) − cX
18:        Y(i) ← Y(i) − cY
19:    end for
20:    m11 ← 0, m12 ← 0
21:    m21 ← 0, m22 ← 0
22:    for i ← 1, k − 1 do
23:        m11 ← m11 + X(i) ∗ X(i)
24:        m12 ← m12 + X(i) ∗ Y(i)
25:        m21 ← m21 + Y(i) ∗ X(i)
26:        m22 ← m22 + Y(i) ∗ Y(i)
27:    end for
28:    V ← Eigenvector(m11, m12, m21, m22)
29:    bugOrientation ← (atan2(V(2, 2), V(1, 2))) ∗ 180/pi;
30:    return bugOrientation, cX, cY
31: end procedure
```

FIGURE 5.10. Bilateral symmetry detection using PCBR.

6. A SIFT DESCRIPTOR WITH GLOBAL CONTEXT

6.1. Overview

In previous chapters, we discussed local feature detections and their applications. In this chapter, we will move to feature matching, which is another critical task for computer vision applications. Feature matching is also refered to as the correspondence problem, which tries to find corresponding features in two or more images. Correspondence is a necessary step for many computer vision applications such as image registration, object tracking, 3D reconstruction, and object recognition. Currently, local feature-based matching is popupar due to its robustness to both clutter and occlusion. However, a primary shortcoming of local features is its deficiency of global information that can cause ambiguity in matching. In our insect identification project, insects are normally highly articulated with repeated patterns. Matching of local features only often fails due to these ambiguities. Therefore, resolving ambiguity and augmenting local feature matching with more information are the primary topics of this chapter.

Local features combined with global relationships convey much more information. We propose a method of including flexible global context information into local feature-based matching by augmenting the local feature descriptor. In the matching process, feature descriptors are normally built for every detected local feature and are used to match descriptors in other images. It is important that a point's description is as unique as possible while also allowing for various image transformation due to difference in lighting, object movement, and change in camera pose.

This chapter presents a new feature descriptor that combines a local SIFT descriptor [47] with a global context feature vector similar to the shape context

[5]. The global context helps discriminate local features that have similar local appearance. We believe that this technique more closely matches the process of human feature matching in that humans are able to augment local regions with the big picture that provides an overall reference to help disambiguate multiple regions with locally similar appearance.

Figure 6.1 illustrates the primary difficulties that this work will address. In particular, an image may have many areas that are locally similar to each other (such as the checkerboard pattern). Further, an object such as the stonefly larva in Fig. 6.3 (a-b) may have a complex shape and thus exhibits non-affine distortions due to out-of-plane rotations or other articulated or non-rigid object movements. Multiple locally similar areas produce ambiguities when matching local descriptors while non-rigid distortions produce difficulties when matching groups of feature points with assumed 2D rigid body or affine transformations.

6.2. Local Feature Detection

As noted before, the first step in point correspondence is feature (or interest) point detection. To more accurately quantify performance of the feature descriptors without introducing variability due to differences in interest point detectors, we use the scale-space DoG extrema detection code available from David Lowe [47] that provides both interest points and the SIFT descriptor at each feature.

The DoG is an approximation to the normalized Laplacian, which is needed for true scale invariance [43]. DoG scale space is sampled by blurring an image with successively larger Gaussian filters and subtracting each blurred image from the adjacent (more blurred) image. In this case, three levels of scale are created for

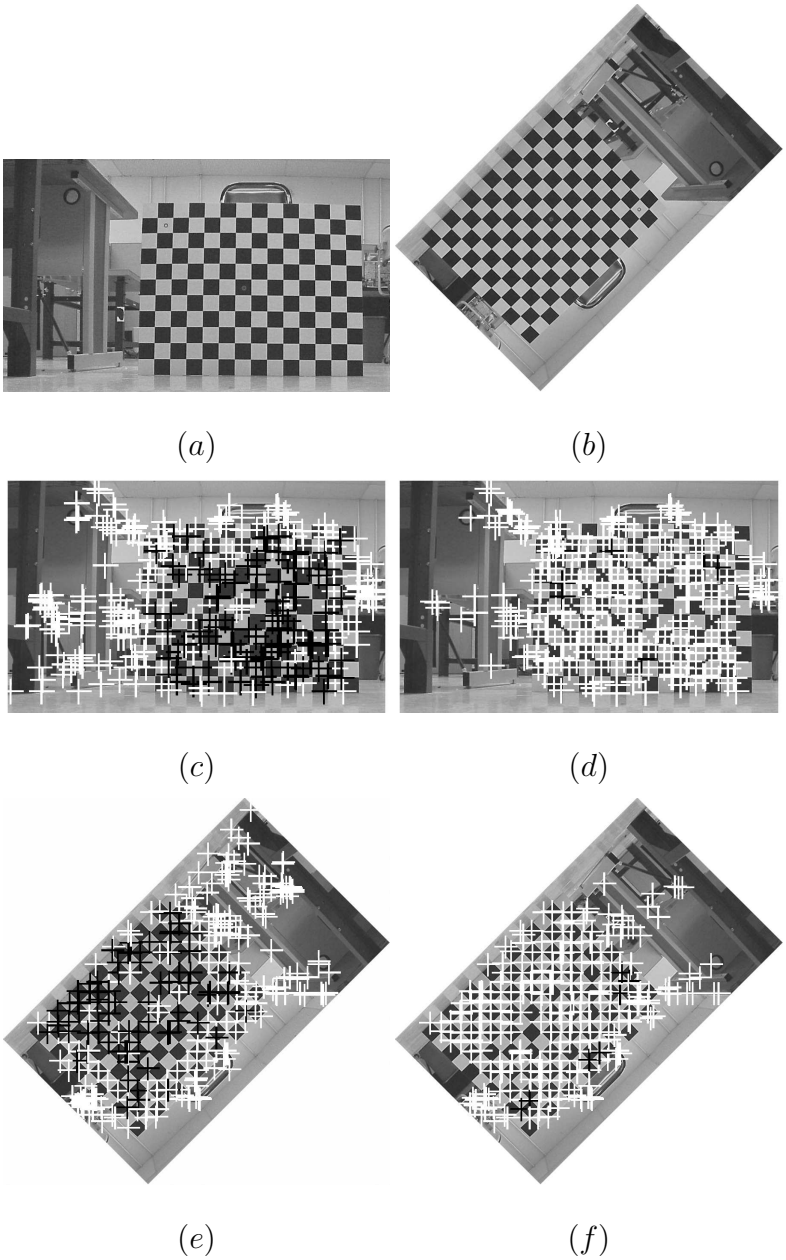


FIGURE 6.1. Comparison of matching results. (a) Original checkerboard image. (b) Rotated 135°. (c-f) Matches (white) and mismatches (black) using ambiguity rejection with (c,e) SIFT alone-268/400 correct matches (67%)-and (d,f) SIFT with global context-391/400 correct (97.75%).

each octave by blurring the image with incrementally larger Gaussian filters with scale steps of $\sigma = 2^{1/3}$. After completing one octave, the image with twice the initial scale is resampled by taking every other row and column and the smoothing process is repeated for the next octave, thus reducing computation. Interest points are characterized as the extrema (maxima or minima) in the $3D$ (x, y, σ) space. As such, each pixel is compared with its 26 neighbors in scale space and a pixel is selected as a feature point if its value is larger or smaller than all of its neighbors. Subsample accurate position and scale is computed for each extrema point by fitting a quadratic polynomial to the scale space function $D(x, y, \sigma)$ and finding the extremum, giving

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D}{\partial x} \tag{6.1}$$

where $x = (x, y, \sigma)$ and $\hat{x}$ is the extremum position providing accurate position and scale.

Finally, an orientation is assigned to each interest point that, combined with the scale above, provides a scale and rotation invariant coordinate system for the descriptor. Orientation is determined by building a histogram of gradient orientations from the key points neighborhood, weighed by a Gaussian and the gradient magnitude. Every peak in the histogram with a height of 80% of the maximum produces a key point with the corresponding orientation. A parabola is fit to the peak(s) to improve accuracy.

6.3. Feature Descriptor

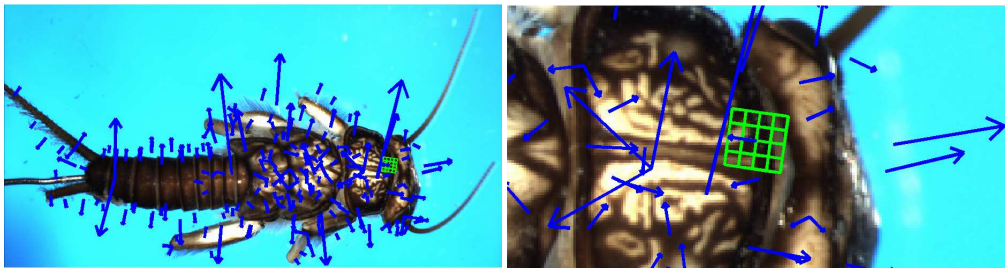
For every interest point detected, we build a two-component vector consisting of a SIFT descriptor representing local properties and a global context vector to disambiguate locally similar features. Thus, our vector is defined as

$$F = \begin{bmatrix} \omega L \\ (1-\omega)G \end{bmatrix} \tag{6.2}$$

where L is the 128-dimension local SIFT descriptor, G is a 60-dimension global context vector, and ω is a relative weighting factor.

6.3.1. SIFT Descriptor

The SIFT (Scale Invariant Feature Transform) [47, 48] has been shown to perform better than other local descriptors [54]. Given a feature point, the SIFT descriptor computes the gradient vector for each pixel in the feature point's neighborhood and builds a normalized histogram of gradient directions. The SIFT descriptor creates a $16 * 16$ neighborhood that is partitioned into 16 subregions of $4 * 4$ pixels each (Fig 6.2(b)). For each pixel within a subregion, SIFT adds the pixel's gradient vector to a histogram of gradient directions by quantizing each orientation to one of 8 directions and weighting the contribution of each vector by its magnitude. Each gradient direction is further weighted by a Gaussian of scale $\sigma = n/2$ where n is the neighborhood size and the values are distributed to neighboring bins using trilinear interpolation to reduce boundary effects as samples move between positions and orientations. Figure 6.3 shows the SIFT descriptor created for a corresponding pair of points in two stonefly images and a non-matching point.



(a) (b)

FIGURE 6.2. SIFT detections on an stonefly image (a) Detections on the whole image. (b) The neighborhood area used for calculating SIFT descriptor(green grid).

6.3.2. Global Context Descriptor

We use an approach similar to shape contexts [5] to describe the global context of each feature point. Like SIFT, shape contexts also create a histogram, but in this case they count the number of sampled edge points in each bin of a log-polar histogram that extends over a large portion of the image. Rather than counting distinct edge points, detection of which can be sensitive to changes in contrast and threshold values, we compute the maximum curvature at each pixel. Given an image point (x, y), the maximum curvature is the absolute maximum eigenvalue of the Hessian matrix 3.1. Thus, the curvature image is defined as

$$C(x, y) = |\alpha(x, y)| \tag{6.3}$$

where $\alpha(x, y)$ is the eigenvalue of Hessian matrix(Eq.3.1) with the largest absolute value. As noted in [72], $\alpha(x, y)$ can be computed in a numerically stable and efficient manner with just a single Jacobian rotation of the matrix to eliminate

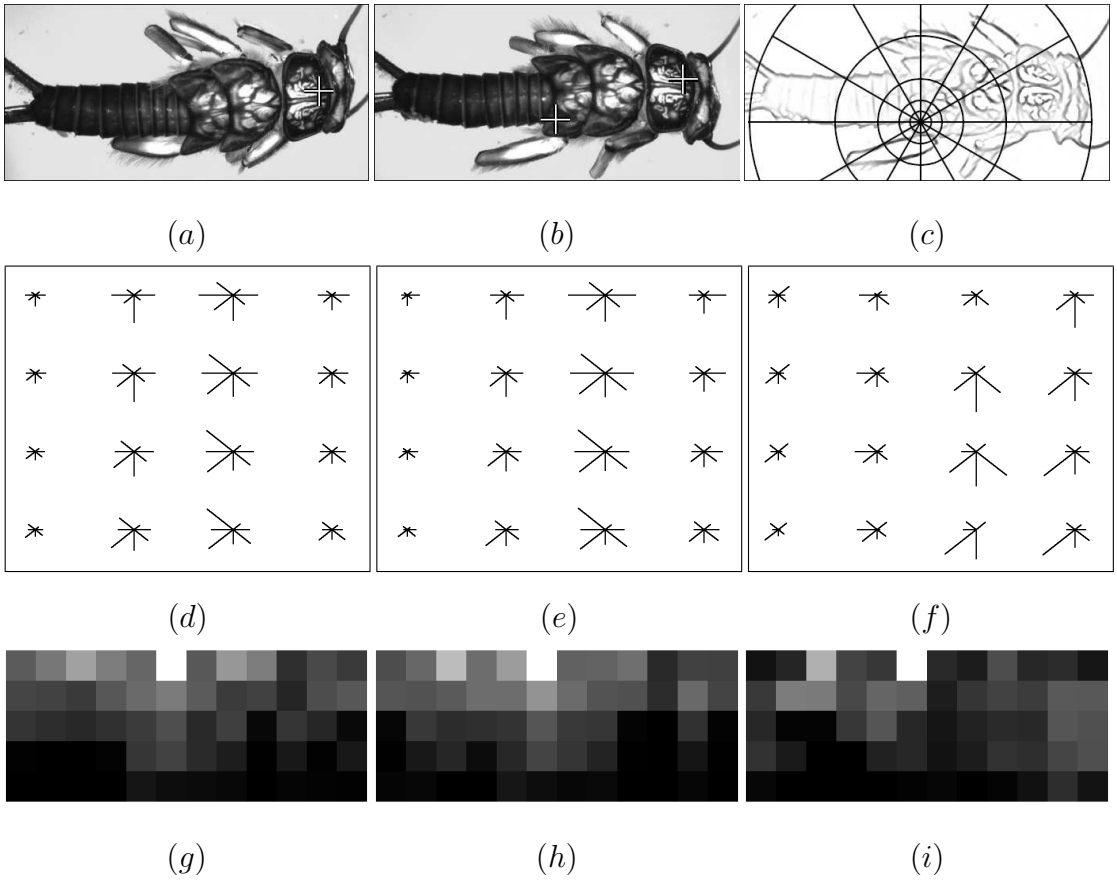


FIGURE 6.3. (a-b) Original images with selected feature points marked. (c) Reversed curvature image of (b) with shape context bins overlaid. (d) SIFT of point marked in (a), (e) SIFT of matching point in (b), (f) SIFT of a random point in (b), (g) Shape context of point marked in (a), (h) Shape context of matching point in (b), (i)Shape context of random point in (b).

the I_{xy} term. Figure 6.3(c) shows the curvature image (reversed for printing) $C(x, y)$, resulting from the insect image in Fig. 6.3(b).

For each feature, the global shape context accumulates curvature values in each log-polar bin. The diameter is equal to the image diagonal and, like [5], our shape context is a $5 * 12$ histogram. Our implementation is not exactly log-polar since the radial increment of the center two bins are equal, thus, the bins have radial increments

$$\frac{r}{16}, \frac{r}{16}, \frac{r}{8}, \frac{r}{4} \text{ and } \frac{r}{2}, \tag{6.4}$$

where r is the radius of shape context. The curvature value of each pixel is weighted by an inverted Gaussian and then added to the corresponding bin. The larger a pixel's curvature measure (shown as darker pixels in Fig. 6.3, the more it adds to its bin. The Gaussian weighting function is

$$w(x, y) = 1 - \exp^{-((x-x_f)^2+(y-y_f)^2)/2\sigma^2} \tag{6.5}$$

where (x_f, y_f) is the feature point position and σ is the same scale used to weight the SIFT features neighborhood. In this way, the weighting functions places more importance on features beyond the neighborhood described by SIFT and provides a smooth transition from the local SIFT descriptor to the global shape context.

To reduce boundary effects as pixels shift between bins and to improve computational efficiency, the curvature image is reduced by a factor of 4 with a low-pass Harr wavelet filter and the resulting image is further smoothed with a Gaussian filter of scale $\sigma = 3$ pixels. The shape context samples this reduced and smoothed image. Finally, the global context vector is normalized to unit magnitude so that it is invariant to changes in image contrast.

More specifically, if $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})^T$ is the feature point position with orientation θ,then

$$\alpha = \left\lfloor \frac{6}{\pi} \left(\arctan\left(\frac{y - \tilde{y}}{x - \tilde{x}} \right) - \theta \right) \right\rfloor \tag{6.6}$$

and

$$d = \max\left(1, \left\lfloor \log_2 \left(\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{r} \right) + 6 \right\rfloor \right) \tag{6.7}$$

are the angular and radial-distance bin indices, respectively, for a point $(x, y)^T$, where $\|\bullet\|$ is the L_2 norm and r is the shape context radius as used in Eq. 6.4. Let $N_{\alpha,d}$ be the neighborhood of points with bin indices α and d, then bin $\dot{G}_{a,d}$ of the unnormalized histogram is computed by

$$\dot{G}_{a,d} = \sum_{(x,y) \in N_{a,d}} C^{'}(x, y) \tag{6.8}$$

where $C^{'}$ is the reduced and smoothed curvature image from Eq. 6.3 as described previoously. Finally, the normalized global shape context is given by

$$G = \frac{\dot{G}}{\left\| \dot{G} \right\|} \tag{6.9}$$

In practice, G is computed by scanning the shape context's bounding box, computing the indices α and d for each pixel and incrementing the corresponding bin by $C^{'}(x, y)$, and finally normalizing it to unit magnitude.

6.3.3. Rotation and Scale Invariance

Our combined feature descriptor, F, in Eq. 6.2 is rotation invariant since both the SIFT descriptor and the global context are constructed relative to the key point's orientation. Further, the SIFT descriptor is scale invariant since it is

constructed in the key point's scaled coordinate frame. However, the size of the global context vector is a function of the image size rather than the interest point's scale and, as such, is not fully scale invariant, although some scale invariance is afforded by the smoothing and by the logpolar construction in that the log radial bins allow for increasing uncertainty as relative distance increases.

There are two reasons why the shape context size is not relative to the interest point scale. First, in our insect ID project, we only have minor scale changes. As such, we don't have the need for large scale invariance. Second, the range of scales returned by the feature detector is on the order of a couple of pixels up to hundreds of pixels. To capture enough global scope for the feature's with the smallest scale, the radius of the shape context would need to be many (perhaps a hundred or more) times larger than the feature's scale. This would be impractical for the features with large scale since (a) a shape context that large would extend well beyond the image boundary and (b) the larger features do not really need the global context, as they already describe a large neighborhood.

As it is, the weighting function in Eq. 6.5 balances the contributions of the fixed-size shape context with the variable-size SIFT descriptor. When the SIFT scale is small, the shape context extends well beyond the SIFT descriptor's neighborhood to give the small neighborhood a global scope. For large local features that already describe large portions of the image, the shape context size is proportionally much smaller and Eq. 6.5 further reduces its relative contribution.

For our insect recognition project, we have explored a more robust option for achieving rotation and scale invariance. Since we already segment the insect prior to feature matching (the blue background simplifies automatic segmentation), we compute the principal axis of the segmented insect using principal component analysis (PCA) and build our feature descriptor relative to this global

orientation. The principle axis is much more robust to noise and local transformations that would otherwise effect the local orientation computation described in Section 6.2. We also achieve scale invariance by constructing our shape context relative to the magnitude of the principal axis.

6.4. Matching

Given two or more images, a set of feature points that can be reliably detected in each image, and robust descriptors for those features, we next match feature points between images. Since our descriptor already includes global shape information, we don't need to perform expensive groupwise or global consistency checks when matching. Consequently, we compare descriptors with a simple nearest neighbor distance or nearest neighbor with ambiguity rejection metric with a threshold on the match. If two or more points match to a single point in another image, we keep the pair with the best match and discard the other(s).

Given the definition of our feature descriptor in Eq. 6.2 and two descriptors, F_i and F_j, our distance metric is a simple Euclidean distance metric

$$d_L = |L_i - L_j| = \sqrt{\sum_k (L_{i,k} - L_{j,k})^2} \tag{6.10}$$

for the SIFT component, L, of the feature vector and a χ^2 statistic

$$d_G = \chi^2 = \frac{1}{2} \sum_k \frac{(h_{i,k} - h_(j,k)^2}{h_{i,k} + h_(j,k)} \tag{6.11}$$

for the shape context component, G. The χ^2 measure is appropriate since it normalizes larger bins so that small differences between large bins, which typically have much greater accumulated values, produce a smaller distance than a small

difference between the small bins (which have small values to begin with) [5]. The final distance measure value is given by

$$d = \omega d_L + (1 - \omega) d_G \tag{6.12}$$

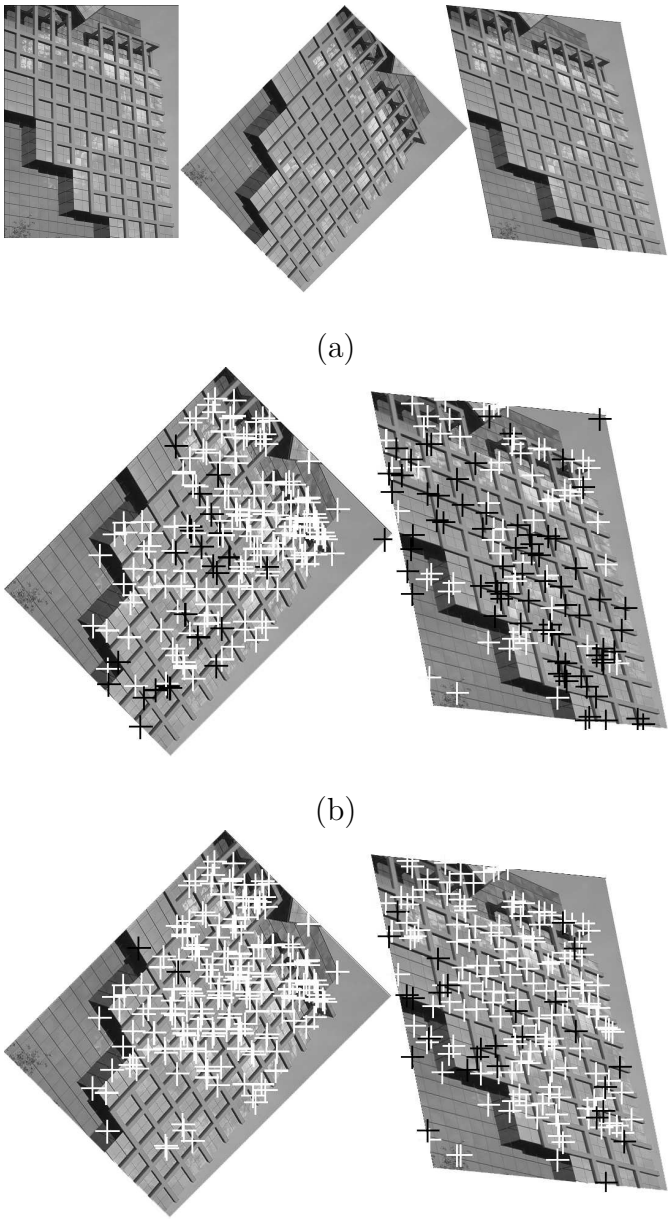
where ω is the same weight used in Eq. 6.2. For the results presented here, we use a value of $\omega = 0.5$.

Finally, we discard matches with a distance above some threshold T_d. Since the components of our feature vector, F, are normalized, we can apply a meaningful threshold that will be consistent across multiple images and transformations. In this work, we use $T_d = 0.5$.

6.5. Results

To assess matching rate, we artificially transform images so as to automatically determine if a match is correct. Figures 6.1, 6.4 - 6.6 compare the matching rate between SIFT alone and SIFT with global context (SIFT+GC). For a given descriptor (SIFT or SIFT+GC), we match each feature point in the original image with feature points in the transformed image using both nearest neighbor (NN) and ambiguity rejection (AR). Like [47], ambiguity rejection throws out matches if the ratio of the closest match to the second closest match is greater than 0.8. The resulting matches for both NN and AR (after discarding ambiguous matches) are then sorted from best (lowest matching distance) to worst and the best $50, 100, 200, 300, 400$, etc. matches are chosen for comparison. A match is correct if it is within 4 pixels of its predicted position.

In Figure 6.4, SIFT alone correctly matches some of the windows since the reflection of clouds disambiguates the otherwise similar local features. Note that the SIFT scale for both the checkerboard squares in Fig. 6.1 and the windows



(a)

(b)

(c)

FIGURE 6.4. (a) Original and transformed images. Matching results in transformed images using nearest neighbor with (b) SIFT only-rotate: 170/200 correct (85%); skew: 73/200 correct (37%);-and (c) SIFT with global context- rotate: 198/200 correct (99%); skew: 165/200 correct (83%). The corresponding matching points from the original image are not shown.

in Fig. 6.4 are large enough to include neighboring squares or windows. Thus, SIFT correctly matches squares on the edge of the checkerboard since the feature neighborhoods extend beyond the edge of the checkerboard; likewise for the windows. Despite this, SIFT+GC still increases the matching rate significantly for these images.

Figure 6.5 plots the matching rate as a function of the number of matched points for SIFT and SIFT+GC using both NN and AR matching. Matching rates are computed using the artificially transformed images in Figures 6.1, 6.4 and 6.6 — four images each for rotation, skew, and both rotation and skew. Note that SIFT+GC has a consistently higher matching rate for a given matching technique and, in many cases, SIFT+GC using NN matching produces a higher matching rate than SIFT alone using AR matching.

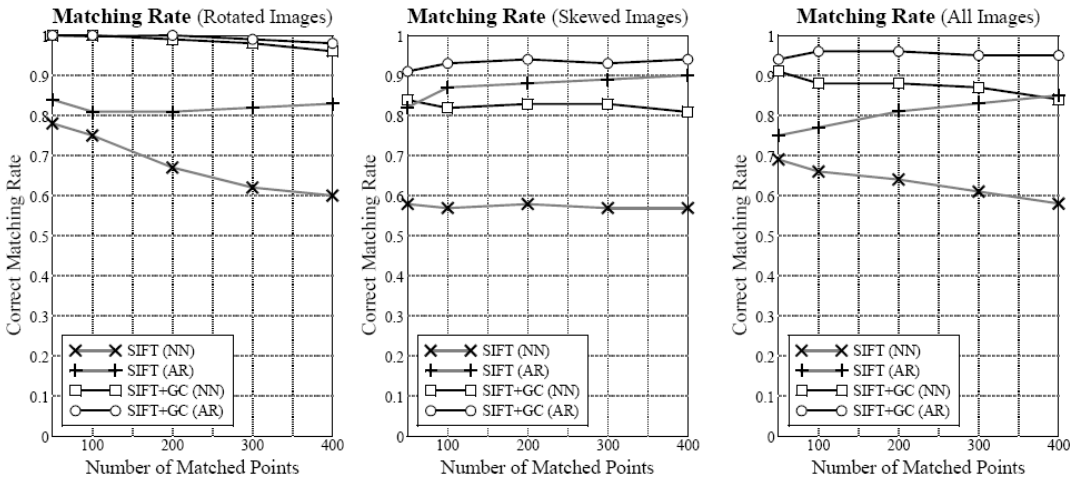


FIGURE 6.5. Matching rate as a function of matched points for the (left) rotated images (see Fig. 6.4), (middle) skewed images, and (right) all images (including images with both rotation and skew). Matching rate is computed for SIFT alone and SIFT with global context (SIFT+GC) using both nearest neighbor matching (NN) and ambiguity rejection (AR).

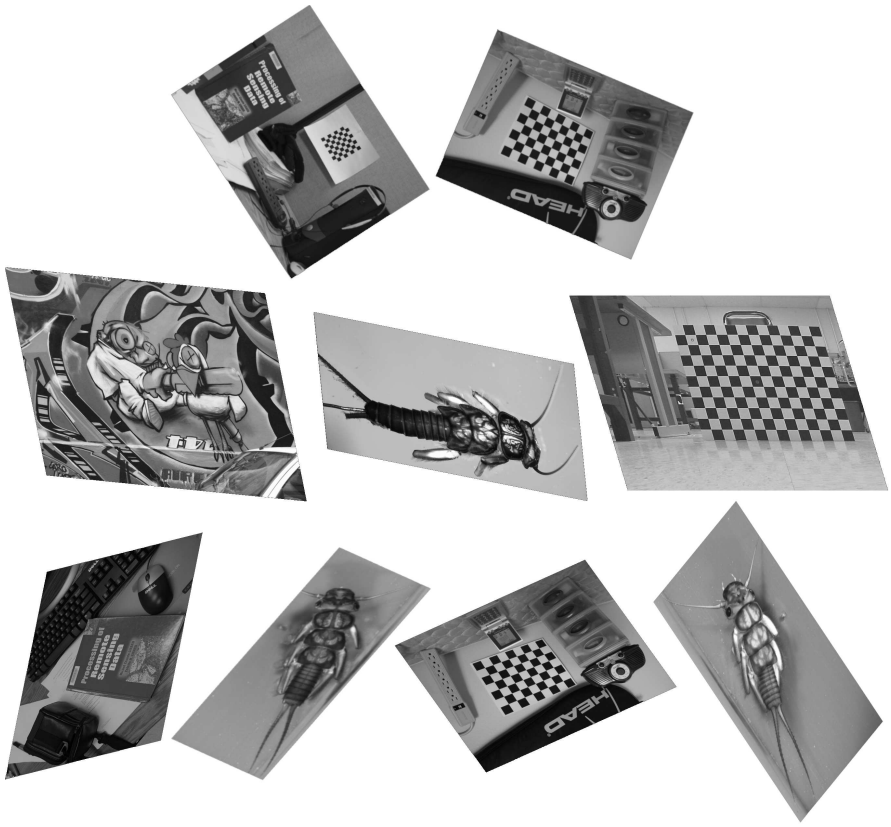


FIGURE 6.6. Images used to compute matching rates shown in Fig. 6.5.

Finally, Fig. 6.7 plots the matching rate of SIFT+GC as a function of the relative weighting factor, ω, used in Eqs. 6.2 and 6.12 for the images in Figures 6.1 and 6.4 as well as the average over all images. As noted earlier, we use a value of $\omega = 0.5$ in all our results.

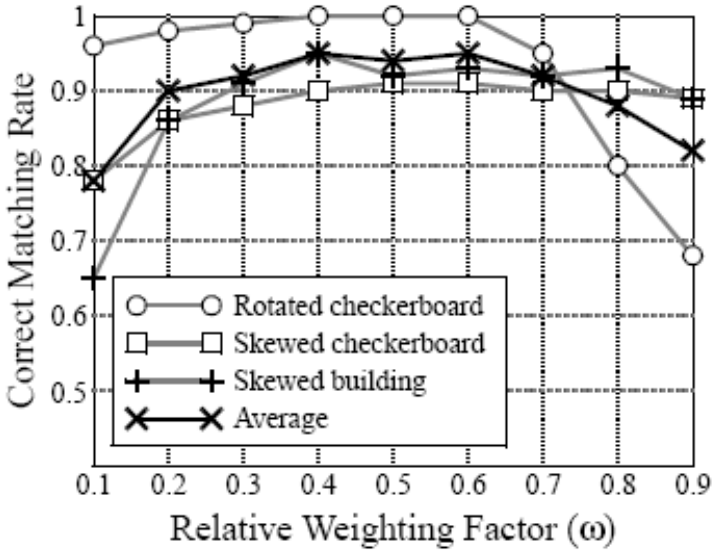


FIGURE 6.7. Correct matching rate for 200 matching points as a function of the relative weighting factor (ω) as used in Eqs. 6.2 and 6.12.

7. REINFORCEMENT MATCHING WITH GLOBAL CONTEXT

7.1. Overview

In the previous chapter, we proposed a method of augmenting the local feature descriptor with global context information. We have seen that combining local features with global relationships is very effective for outlier rejection. But there are problems with this method. It is not robust to occlusion and/or non-rigid transformation. In this chapter, we propose a new framework for including global context information into local feature matching, while still maintaining robustness to occlusion, clutter, and non-rigid transformation. To generate global context information, we extend previous fixed-scale, circular-bin methods by using affine-invariant log-polar elliptical bins. Further, we employ a reinforcement matching scheme that provides greater robustness to occlusion and clutter than previous methods that non-discriminately compare accumulated bins values over the entire context. We also present a more robust method of calculating a feature's dominant orientation. We compar this new method to three existing matching method: nearest neighbor matching without region context, the enhanced local feature descriptor method discussed in the previous chapter and the robust matching method (RANSAC and PROSAC).

In general, feature matching methods can be used in three types of applications. The first application domain determines feature correspondences between multiple images of the same scene under different viewing conditions for tasks such as 3D reconstruction or recovering camera motion in a static scene. These applications usually need to recover the epipolar geometry or solve for a rigid 3D motion model to find a consistent set of matching features. Since these applications already assume a transformation model for feature motion, the RANSAC

[30] method is often used because it is good for solving problems with known models. The second application category involves non-rigid object tracking. The matching object is the same object but deformable. It has similar local appearance for matching features but non-rigid spatial geometry. The third category is object class recognition, where there is typically no rigid transformation model to recover and the spatial relationships between features, as well as the descriptors identifying matching object "parts", can have considerable variation.

No matter what type of matching is used in an application, local feature-based matching normally has ambiguities. Therefore, global constraints are needed to resolve ambiguities. In augmenting local matching with global consistency, three types of methods have been used previously. These include the epipolar constraint [10, 64, 83], graph-based models [13, 19, 27], and spatial binning models [5, 12, 59].

The epipolar constraint is by far the most common method for the first application domain. Using epipolar constraints, matching candidates are confined to the epipolar line. This gives a strong constraint on matching and can reduce ambiguities on images with rigid view transformations. RANSAC methods are widely used in this domain to determine parameters of the transformation model. RANSAC samples a subset of matching points randomly to build a transformation model and verify the correctness of the model by a majority vote. The model that fits the largest number of matching points is selected. RANSAC can be very effective but it has three problems. The first is that when the ratio of outliers is high, the probability of getting correct samplings that represent the true transformation model is low, thus requiring a large number of sampling iterations. PROSAC [16] (progressive sample consensus) is a recent RANSAC techniques which addresses the sampling speed problem. RANSAC treats all correspondences equally and

draws random samples uniformly while PROSAC draws samples from a progressively larger set from among the top-ranked correspondences. This underlying assumption is that correct matches have a greater chance to be among the top-ranked correspondences. The second problem with RANSAC is that it provides only 1D epipolar line constraints which might not be strong enough, especially for texture images with many repeated patterns. An example is shown in Section 7.4 (Fig. 7.11) that there are ambiguities in the epipolar line. The third problem is that the transformation model may be unknown in practice. Without knowing the transformation model, RANSAC has to keep guessing models until it finds the correct model. This takes a lot of time and there is no guarantee that the correct model can be found. Especially for the second and third applications, there are not many models that can be described easily. In the few models that can be described easily, determining the correct number of models is yet another challenge.

Graph-based models are the most common in the second and third application for their flexibility. The various graph-based models include constellation [27], star shape [28], K-fan [19], tree [29], bag of feature [21, 79], hierarchical [8], and the sparse flexible model [13]. These models are illustrated in Figure 7.1. The advantage of graph models is that they formulate the global context problem as a spatial relationship graph and then apply a variety of graph techniques. We note that most of the previous work focuses on the structure of the model and typically use test images with small transformations and stable spatial relations among features. However, in many situations, these graph models are unsuitable.

The method presented in this chapter can be viewed as an extension to spatial binning method [5, 12]. Belongie et al. [5] start with a collection of shape points and builds, for each point, a histogram describing the relative distribution

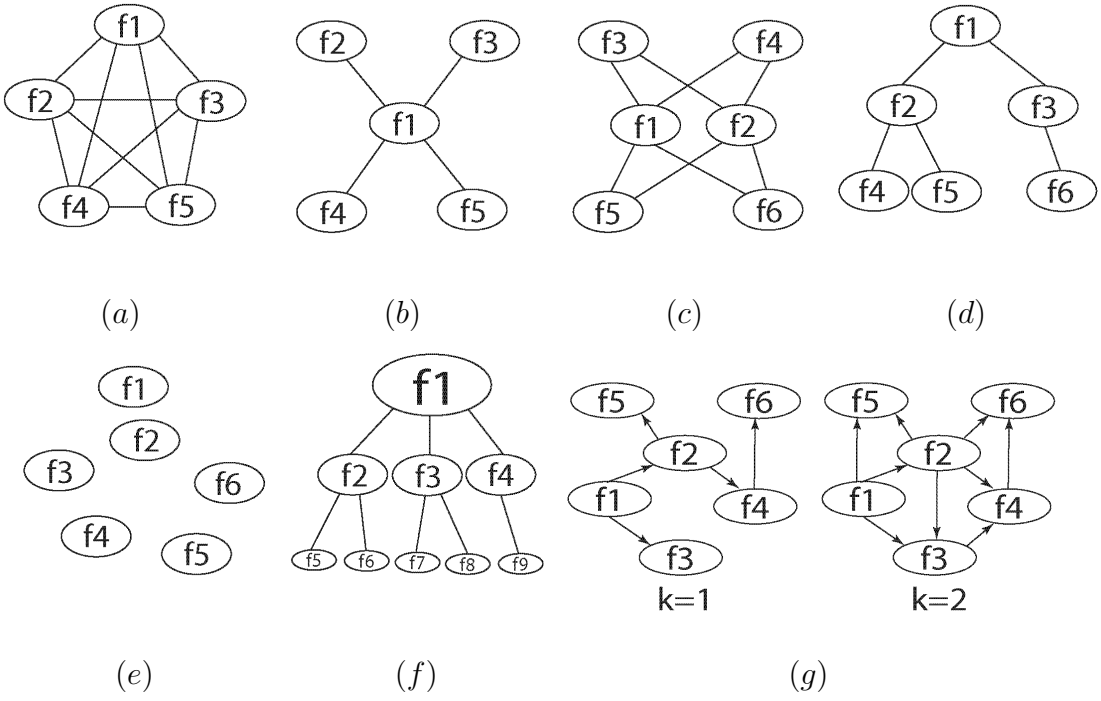


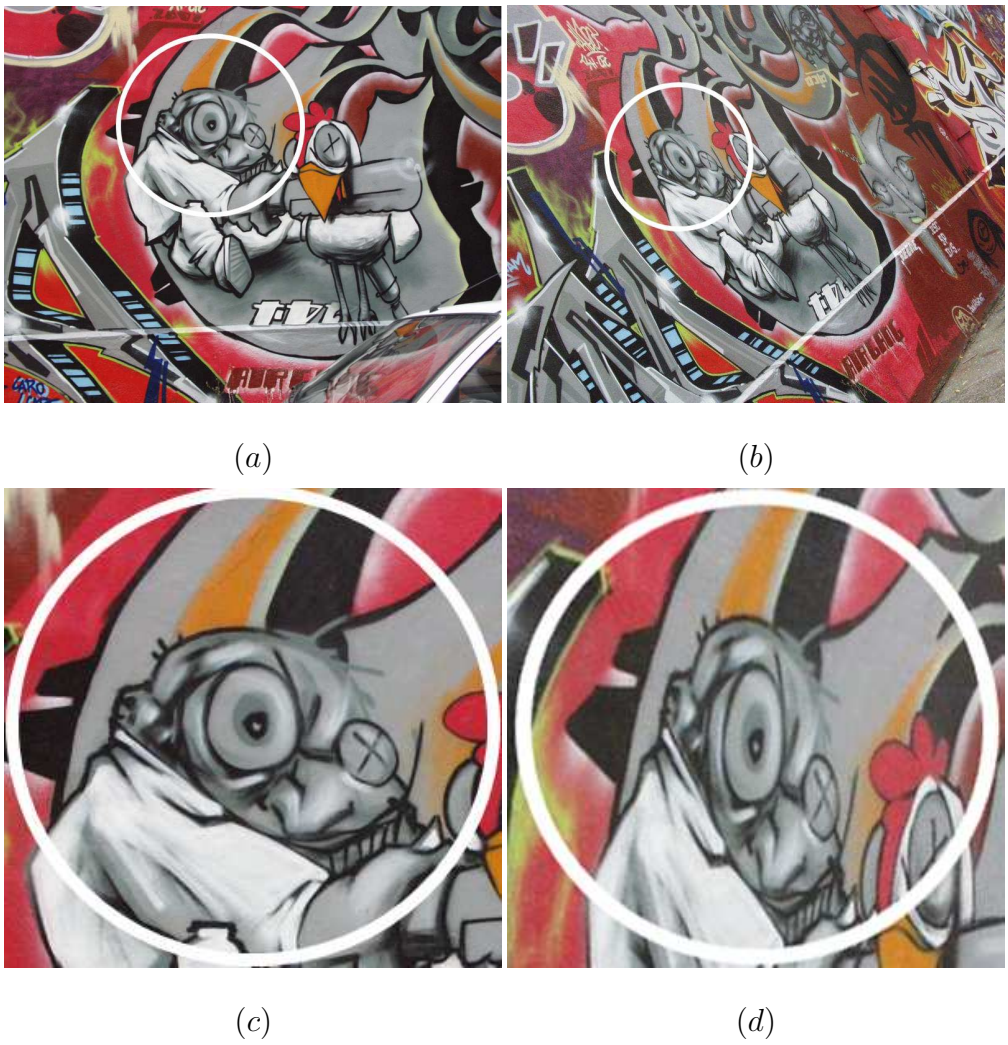
FIGURE 7.1. Various types of spatial models. (a) Constellation model. (b) Star model. (c) K-fan model(k = 2). (d) Tree model. (e) Bag of features. (f) Hierarchy model. (g) Sparse flexible model.

of the other points in log-polar space. Carneiro and Jepson [12] build log-polar bins around each feature and accumulate the weighted count of other features within each bin. Our SIFT with global context method [59] presented in chapter 6 augments local descriptors to include global context information to develop a feature vector that includes both local features and global curvilinear information.

In our previous spatial binning method [59], a global context vector is built to augment the SIFT descriptor and reduce mismatches when multiple local descriptors are similar. However, there are two problems with this approach. First, the global context bins assume a very limited set of transformations between images, as such, under more general transformations, the two circular bins will cover different areas. This introduces errors in the global context histogram. Figure 7.2 shows two corresponding circular regions with the same relative size to their original detected feature scale. Figures 7.2 (c, d), show that the two circular regions cover quite different areas. We solve this problem by using elliptical bins as can be seen in Figure 7.3, the two elliptical bins capture almost the same area, or context, after normalization.

The second problem with previous spatial binning methods is in their inability to handle occlusion and background clutter. All previous spatial binning methods use histograms to represent global context. They sum up all pixel or sample values within each bin to get accumulated histogram values (Fig. 6.3). These accumulated values are not robust to occlusion, background clutter, and detection error.

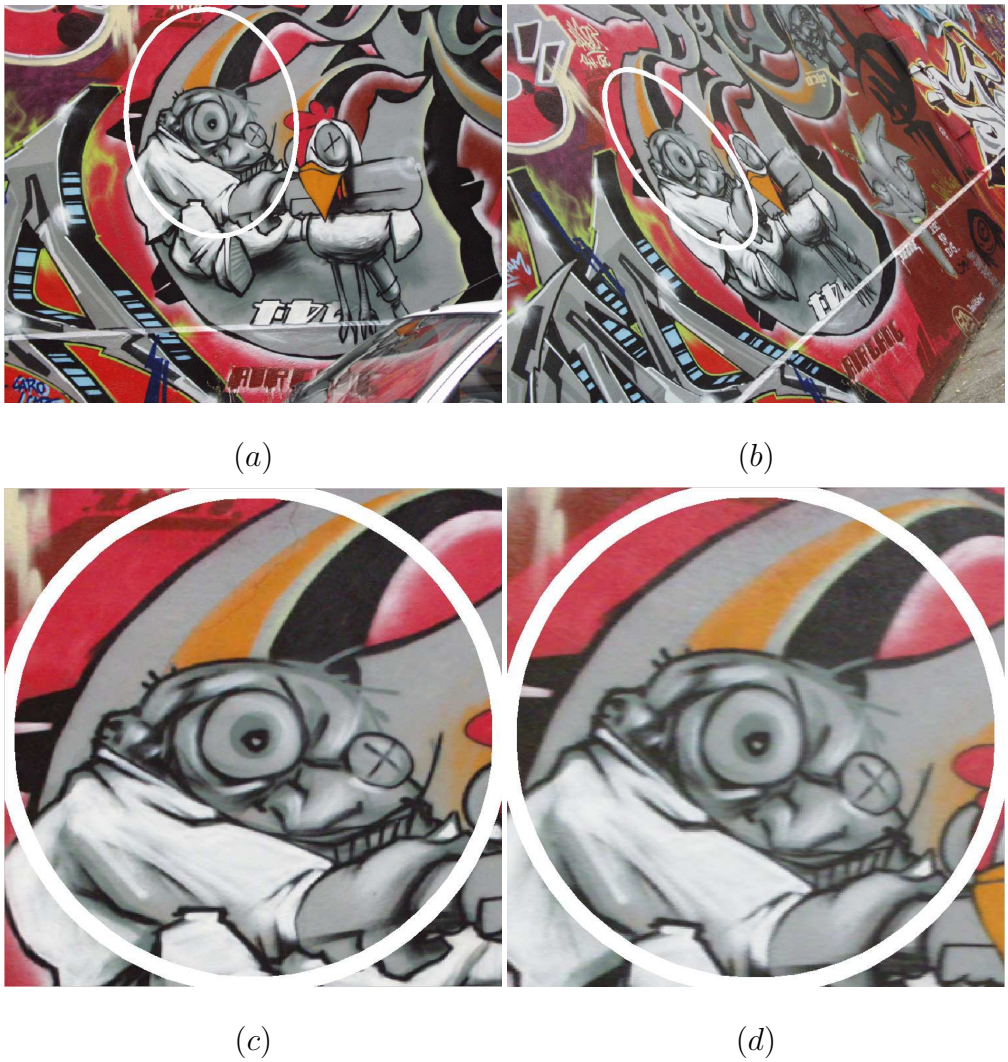
Figures 7.4 (a, b) shows the global context for two corresponding regions. The context regions here are 16 times larger than the original detected regions (small ellipses in Figures 7.4 (c, d)). Figures 7.4 (e, f) shows the normalized context regions for the two areas. The differences in the coverage area in the



(a) (b)

(c) (d)

FIGURE 7.2. Circular regions used for computing global context histogram. (a)(b) Circular bins for two corresponding features. (c)(d) THe normalized regions demonstrate that circular bins fail to capture similar context.



(a) (b)

(c) (d)

FIGURE 7.3. (a)(b) Elliptical regions for two corresponding features. (c)(d) The normalized elliptical regions capture similar context.

normalized region is due to two factors. First, there is occlusion in the first image (the windshield of the car). Second, the original detected regions have variation. They are not at exactly the same position and scale and their shapes are not exactly the same. These variations are magnified in the process of defining the global context from the ellipse of the original feature detections. These problems result in difference in the areas covered by the elliptical context; although they are still much better than circular bins. When building accumulated histograms from these context regions, the two histograms will exhibit differences due to the occlusion and mismatched coverage. We address this problem by using distributed regions rather than accumulated pixels values.

When compared to RANSAC, or more recently PROSAC, our method has two advantages. First, our method doesn't need a consistency model (e.g., epipolar geometric constraints) and consequently, our method works on any reasonable (including non-rigid/non-linear) transformation without requiring a constraint model, and consequently a sample set size. Second, when there is a high percentage of outliers, RANSAC is much less likely to select a sample set from among the inliers—which is necessary to compute the correct transformation. On the other hand, our reinforcement matching scheme is more tolerate by effectively ignoring outliers.

7.2. Elliptical Global Context

For comparison, we use the Hessian-affine interest operator developed by Mikolajczyk and Schmid in [53, 55] due to its performance, repeatability and affine invariant properties. We use the SIFT [47] descriptor to describe each detected region. Our reinforcement matching algorithm can be summarized as follows:

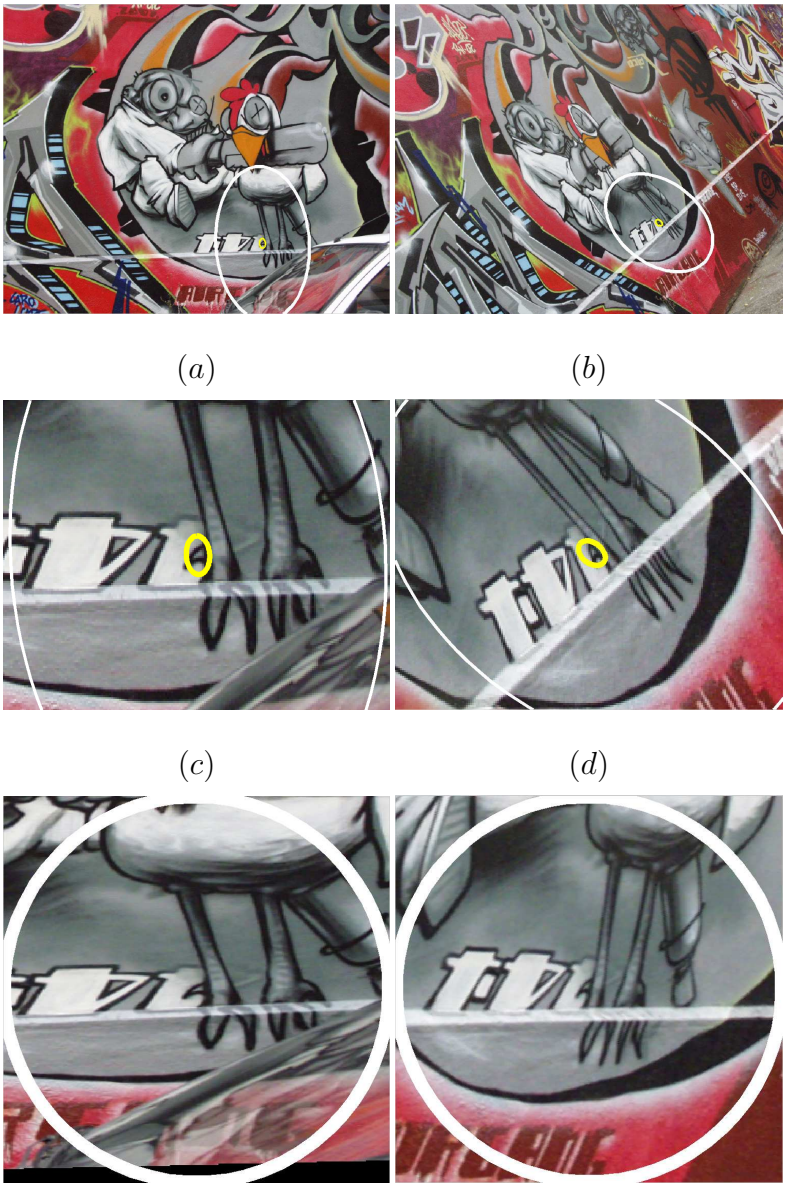


FIGURE 7.4. Influences of global context by occlusion and detection variations. (a)(b) Global context for two corresponding regions. (c)(d) The original detected regions (small ellipses). (e)(f) Normalized global context.

1. For each detected region, calculate the dominant gradient orientation and use it to choose the reference orientation of the ellipse region.

2. Scale the detected affine region (i.e., the innermost ellipse in Figure 7.5) to obtain two additional regions that are 8 and 16 times larger (the outer two ellipses in Figure 7.5). The features detected within these enlarged ellipses form the "region context" for the center feature.

3. Normalize the enlarged regions, including the positions of all the contained context features. Define context bins for each normalized region and construct, for each bin, a list of context features that fall within that bin.

4. Construct the initial matching distance matrix using Euclidean distance and the local descriptors only. From this matrix, a fixed fraction of one-to-one best matches are chosen to form "anchor regions".

5. Compute the final match score between each pair of regions by combining the Euclidean distance match score with the context score, which is computed by counting, for corresponding bins in the context of the two regions, the number of matching anchor regions they contain.

The details of our matching procedure are given below.

7.2.1. Building Region Context

To build the feature's region context for a detected feature, we enlarge the feature's original affine region while maintaining its elliptical shape. This methodology is based on the belief that the deformation of the area around the detected region is somewhat similar to the deformation of the center region. Figures 7.5

show samples of corresponding context regions for a pair of images. The innermost ellipse is the original detected region. The second one is used to calculate the SIFT descriptor. The outer two ellipses—which are eight and sixteen times larger than the inner ellipse—are used to build context bins. The size of the context bins follows the log-polar bin design of [5, 12, 59]; thereby allowing for image deformations due to perspective and non-rigid transformation.

(a) (b)

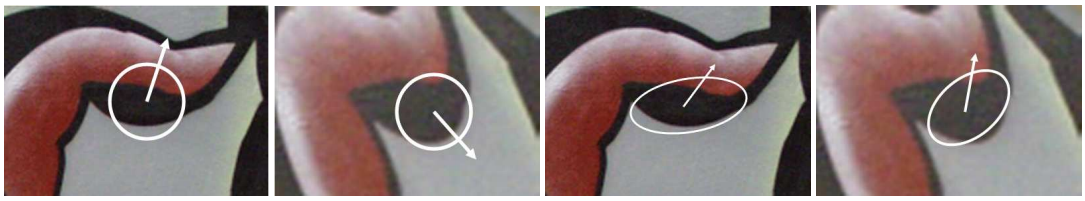
FIGURE 7.5. Global region context for two corresponding regions.

7.2.2. Dominant Orientation Calculation

A stable and robust reference orientation is critical to ensure rotation invariance for both the SIFT descriptor and the region context. Both Lowe [47] and Mikolajczyk [54] compute dominant gradient orientation in a small circular neighborhood around each keypoint. The size of the circular neighborhood is determined by the keypoint's scale. The gradient vector of every pixel in the circular region is used to build a histogram of gradient angles weighted by the

gradient magnitude, and the orientation corresponding to the largest histogram bin is chosen as the dominant gradient.

Using a circular region is not affine invariant. Using a circular region assumes that, although there are global deformations, a small local region should still looks similar. But this is not always the case. When matching images with a wide baseline, the local region can be deformed considerably (Figure 7.6).



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

FIGURE 7.6. (a-b) Illustration of how the dominant orientation for a local feature can be affected by using circular neighborhoods that enclose different areas. (c-d) The dominant orientation computed using the affine detected region is more stable.

For images with a uniform scale change, in-plane rotation, and even minor affine deformations, computing the gradient orientation from a circular region is acceptable. However, using a circular region in the presence of large affine transformations does not produce a stable dominant orientation (Figure 7.6(a-b). On the other hand, calculating the dominant orientation using the original elliptical regions is more stable since the enclosed areas more closely match (Figure 7.6(c-d)).

To sample the gradient within an affine region, we use an efficient scan-line algorithm to determine the pixels contained within the ellipse. Centering the coordinate axis on the keypoint, the implicit equation of the ellipse is

$$Ax^2 + Bxy + Cy^2 = 1 \tag{7.1}$$

Given the eigenvalues (λ_1, λ_2) and eigenvectors (v_1, v_2) of the matrix

$$M = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \tag{7.2}$$

the vertical range of scanline of the ellipse is given by

$$\begin{aligned} y_{max} &= \sqrt{A \cdot r_a^2 \cdot r_b^2} \\ y_{min} &= -y_{max} \end{aligned} \tag{7.3}$$

where

$$r_a = \frac{1}{\sqrt{\lambda_2}}, r_b = \frac{1}{\sqrt{\lambda_1}} \tag{7.4}$$

are scale factors for the ellipse's major and minor axes, respectively.

For each horizontal scanline in the range $y_{min} \leq y \leq y_{max}$, the starting, x_{min}, and ending, x_{max}, points are obtained by solving the implicit ellipse equation 7.1 with known y value. Once the gradient values within the ellipse have been computed, the same histogram building process as in [47] is used to find the dominant gradient. In Section 7.4 we note that using the elliptical region to compute dominant orientation achieves better matching results than using the circular region with all other settings identical (Figure 7.15). After computing the dominant orientation, θ_D, we form a unit vector, $v_D = [\cos(\theta_D), \sin(\theta_D)]^T$, and use it to choose the orientation of the ellipse. Since the dominant orientation tends to point in the direction of the minor axis, v_1 or $-v_1$, we choose the reference orientation as the direction along the major axis, v_2 or $-v_2$, that produces a positive cross product with v_D. In other words, our reference orientation, α , is defined as

$$\alpha = \arctan\left(sgn(v_2 \times v_D)\frac{v_{2,y}}{v_{2,x}}\right) \tag{7.5}$$

7.2.3. Region Context Selection

Given the equation for an ellipse in Eq. 7.1, a point (x, y) is within an ellipse that is S times larger if

$$Ax^2 + Bxy + Cy^2 \leq S^2. \tag{7.6}$$

In Figure 7.7, the second innermost ellipse corresponds to $S = 3$ and it is used to calculate the SIFT descriptor (as noted earlier). The third and fourth ellipses correspond to $S = 8$ and $S = 16$, respectively. Since the region inside the second ellipse is already described by the SIFT descriptor, the region context consists of all the features between the second ($S = 3$) and fourth ($S = 16$) ellipses. The black dots are all detected regions. Yellow crosses are anchor features. These features, of course, also represent an elliptical region with its own sizes and orientations, but are shown as crosses and dots to improve visibility.

7.2.4. Normalization of Region Context Bins

To ensure that each context feature maps to the correct context bin, we normalize the region context by using the ellipse parameters from the keypoint's second moment matrix. The transformation that maps the reference orientation to the x-axis and the inner ellipse ($S = 1$) to a unit circle is

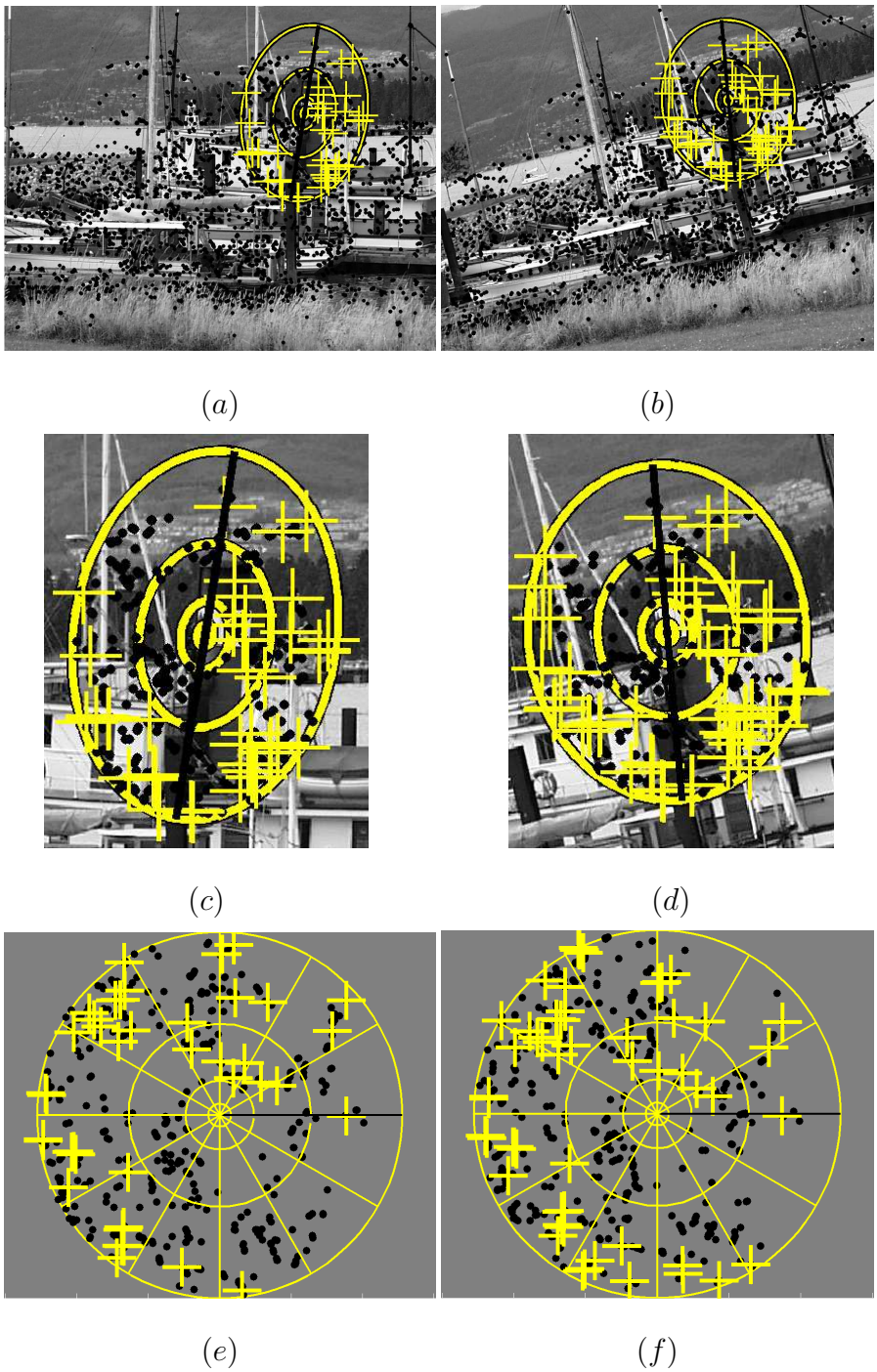


FIGURE 7.7. Regions Context Normalization. (a, b) global context for two corresponding regions. (c, d) A close look. (e, f) Normalized region context (Yellow crosses are anchor regions. Black dots are other regions in the context.

$$M^{'} = \begin{bmatrix} \lambda_2 & 0 \\ 0 & \lambda_1 \end{bmatrix}^{-\frac{1}{2}} \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$
$$= \begin{bmatrix} r_a & 0 \\ 0 & r_b \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \quad (7.7)$$

where λ_1, λ_2 are eigenvalues of Eq. 7.2, r_a, r_b are scale factors defined in Eq. 7.4 , and α is the reference orientation from Eq. 7.5. The position of each context feature, x, that falls within the region context is then mapped to its normalized position,

$$x^{'} = M^{'}x, \quad (7.8)$$

and the context feature is added to the context bin that it falls in, as determined by the radial and angular position of $x^{'}$ in the normalized space. Rather than simply accumulate a count of the number of features in each bin, each context bin maintains a list of features (i.e., a list of SIFT descriptor indices). Given a feature, the feature's region context tells us what other features are near it and at what angle and distance. These context bin lists are the key to reinforcement matching since corresponding bins can be compared to determine the number of matching features in each bin while ignoring features that don't match.

7.3. Reinforcement Matching

The goal of reinforcement matching is to use the region context to efficiently improve matching accuracy by increasing the confidence of a good match between two features if they have a similar spatial arrangement of neighboring features. We first compute the $m * n$ matching cost matrix that contains the Euclidean

distance, $c(i, j)$ for $1 \leq i \leq m$, $1 \leq j \leq n$, between each pair of SIFT descriptors, where m is the number of features in the first image and n is the number in the second image. From these correspondences, we find the best matches along all rows. These best matches are sorted and a portion (e.g., 20% of $\min(m, n)$) of it are selected. The selected matches are called anchor features. Note that this produces a one-to-one mapping. Figure 7.7 illustrates the two types of regions: anchor features (indicated with crosses) and other features (indicated with dots).

For this method, quickly comparing the global context is critical. We achieve fast comparisons by checking whether the one-to-one anchor feature mappings are in corresponding bins or not. We construct a data structure (Figure 7.8) to quickly compare the global context of two features. Every global context feature has 24 bins. Two arrays are attached to every bin. The first array is the anchor feature index array. It records all anchor features that fall into this bin. The second is the feature bit set array. It marks the entries of anchor regions in this bin to 1. Using bit set array, we avoid the need to search the anchor feature index array; thus achieving $O(1)$ for every check at the expense of using more memory.

For example, if we want to compare a context A (Fig. 7.8(a)) with another context B (Fig. 7.8(b)), we look at the first bin in A and find anchor feature 5 is there. The one to one anchor feature mapping table (Fig. 7.8(c)) is then checked and find that the anchor feature 200 is the matched anchor feature in the second image. Without searching the whole anchor feature index array, we look at the bit set array in B. If the 200th entry is 1, this means feature 200 is in the corresponding bin. We treat this as a match. The total number of such feature matches is counted and the final matching distance is

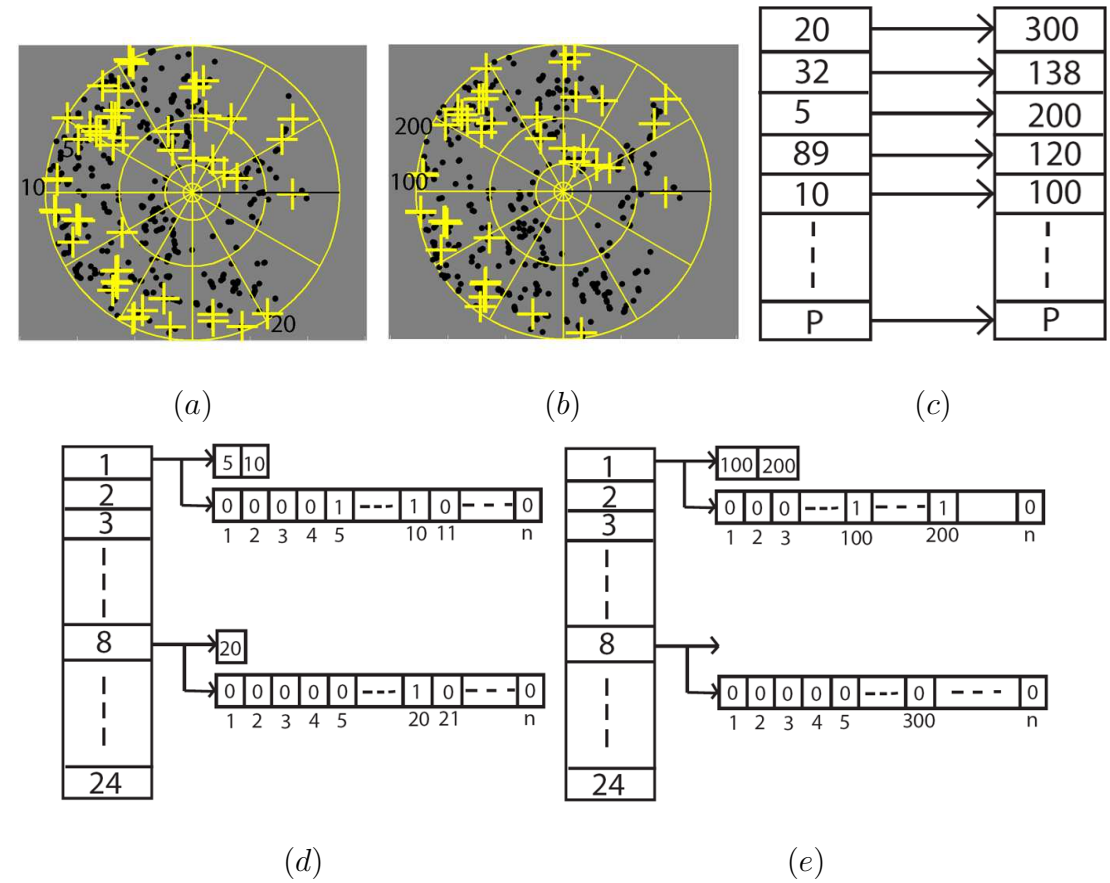


FIGURE 7.8. Data Structure of global context. (a,b) Global context for two corresponding regions. (c) Anchor features map. (d,e) Data structures for the two corresponding regions.

$$c^{'}(i,j) = \frac{c(i,j)}{\log_{10}(10 + num_{support})} \tag{7.9}$$

where $num_{support}$ is the number of matched anchor features. If there are no matched anchor features in any of the context bins, then the denominator is unity and the central feature match is not reinforced. However, as the number of context matches increases, these matches reinforce the central match by increasing the denominator and thus lowering the final matching distance. This equation is chosen empirically. The worst case computational complexity for n features is $O(n^3)$, but this only occurs when all the anchor features are in a single bin for all the matches. In practice, the complexity is $O(n^2 m)$ where $m << n$ is the average number of anchor points in a bin.

Figure 7.9 illustrates how this matching methodology is robust to occlusion and changes in background. If some of features are occluded in one or more bins, or if a context bin contains background that can change from one image to another, the missing features in those bins do not penalize the final matching distance (other than to reduce the support number) while other matches in other bins still contribute to sufficiently reinforce the central feature match. Note that this strategy provides a distinct advantage over global support methods that simply accumulate a single value in each bin. For example, if each bin simply summed up the number of feature/shape points [5] or gradient/curvature pixel values [59] in each bin, then bins that are occluded or contain differing background imagery would actually increase the matching distance since the difference of accumulated bin values can be significant in these examples. Thus, using a single accumulated value in these cases can often lead to reduced matching rates. Our experiment results will show this in section 7.4.

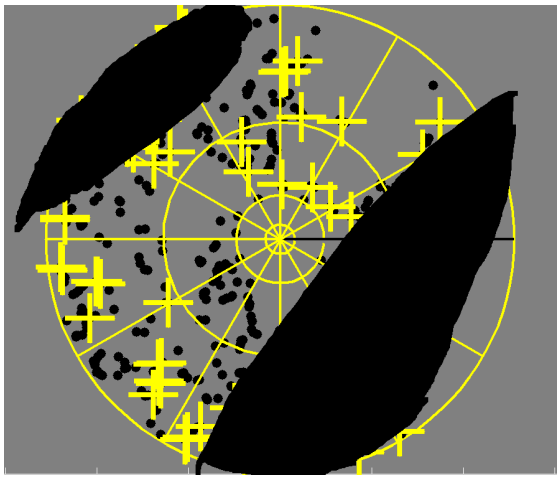


FIGURE 7.9. Illustration of how the region context is robust to occlusion. Reinforcement matching counts the number of matching features in corresponding bins. If a feature is occluded, it is simply ignored and features in other bins still provide sufficient support to reinforce the central feature match.

7.4. Results

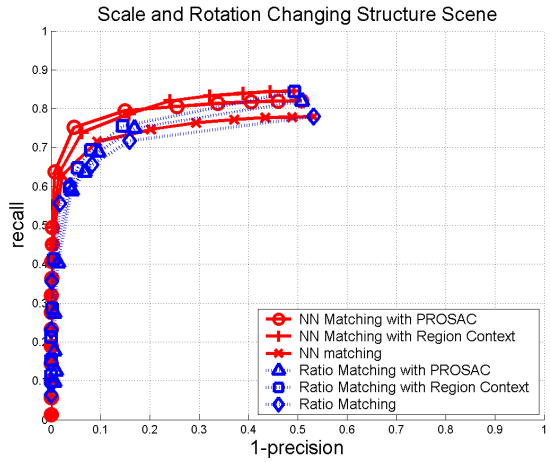
To evaluate performance, we use the INRIA dataset [52] that contains eight image sets representing five transformations (viewpoint change, zoom-rotation, image blur, JPEG compression, and lighting change). Each set contains six images at various degrees of transformation (Fig. 4.7).

We compare our method with PROSAC (a recent, RANSAC-style robust matching method that uses progressive sample consensus) [16] and the previous spatial binning method described in chapter 6. Since the INRIA image sets all represent homographies, they are well suited to RANSAC-style matching using epipolar geometric constraints. We use the same matching performance framework provided by Mikolajczyk and Schmid [52] (recall vs. 1 - precision curves) to evaluate matching performance for two different matching strategies: nearest

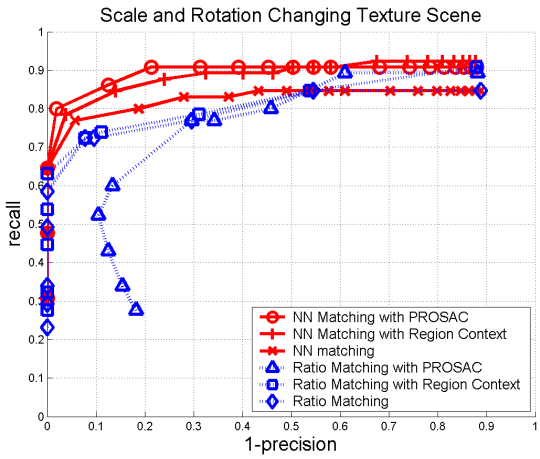
neighbor (NN) and nearest-neighbor-ratio (NNR)-which finds the highest ratio of the nearest neighbor to the second nearest neighbor. The same experiments are done in all image sets. In every image set, images 2 through 6 are matched to the first image in their respective set. For each of these two matching strategies, we measure performance with (using $c^{'}(i,j)$) and without (using $c(i,j)$) reinforcement matching and with PROSAC (using $c(i,j)$). Test results show that reinforcement matching provides higher accuracy than matching without region context on all images and is comparable to PROSAC with NN and better than PROSAC with NNR (Fig. 7.10).

One reason that reinforcement matching provides better matching rates than RANSAC methods is that, like shape context [5], our method provides for general-purpose two-dimensional constraints with some degree of positional flexibility (in that a reinforcing match can fall anywhere within a corresponding bin) while transformational constraints in RANSAC methods are typically more rigid and, in the case of epipolar geometry, only constrain matches to one-dimensional epipolar lines. However, Figure 7.11 demonstrates how highly textured images can still produce many similar patterns even along a one-dimensional epipolar line. Figure 7.10(f) shows the recall vs. 1 - precision curves for matching this image with the first image from this (the tree image) set.

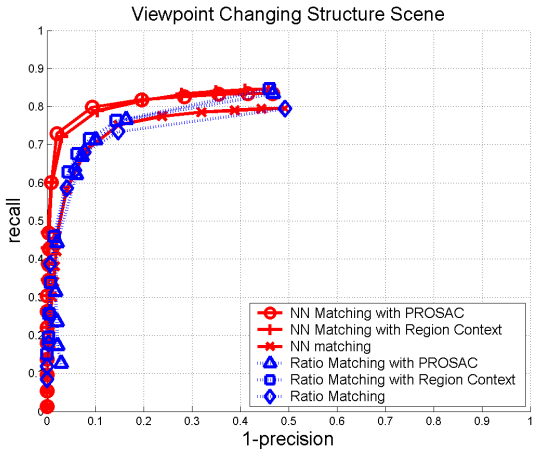
We also compare reinforcement matching with the spatial binning method in chapter 6. Results show that reinforcement matching outperforms spatial binning on most images with NN or NNR matching (figure 7.12). However, the spatial binning method performs better on structured images with blur and JPEG compression. This could be due to the fact that those images normally have less local detail—and increased blur further reduces number of local feature. Since rein-


Scale and Rotation Changing Structure Scene
recall
1-precision
NN Matching with PROSAC
NN Matching with Region Context
NN matching
Ratio Matching with PROSAC
Ratio Matching with Region Context
Ratio Matching

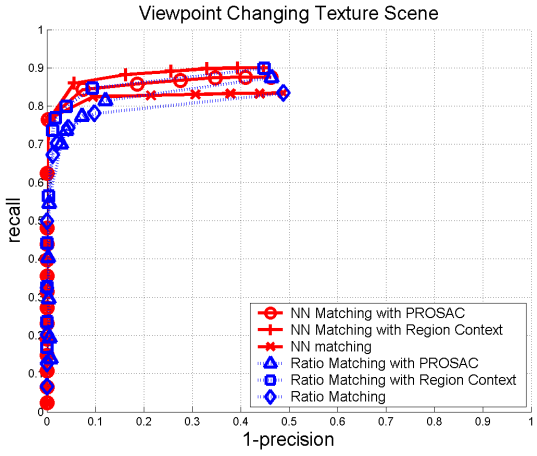
(a)


Scale and Rotation Changing Texture Scene
recall
1-precision
NN Matching with PROSAC
NN Matching with Region Context
NN matching
Ratio Matching with PROSAC
Ratio Matching with Region Context
Ratio Matching

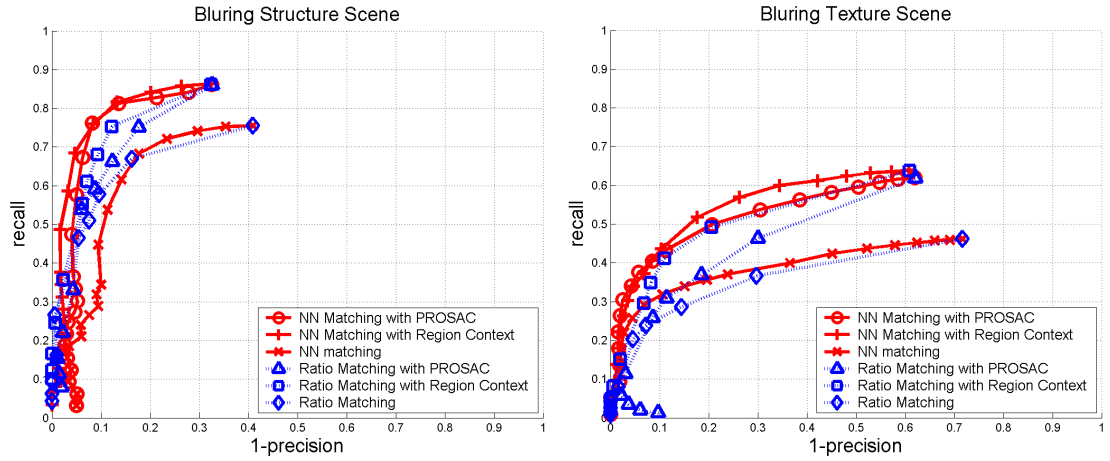
(b)


Viewpoint Changing Structure Scene
recall
1-precision
NN Matching with PROSAC
NN Matching with Region Context
NN matching
Ratio Matching with PROSAC
Ratio Matching with Region Context
Ratio Matching

(c)

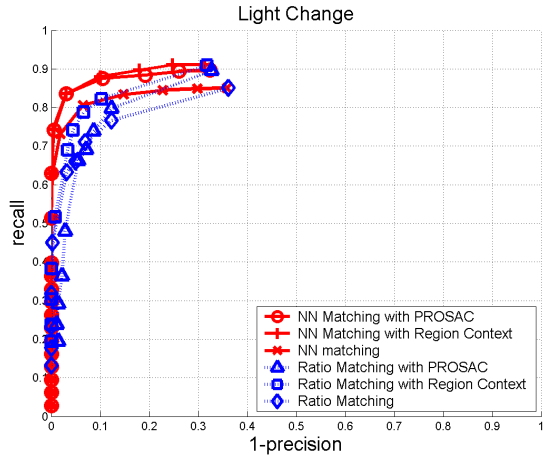

Viewpoint Changing Texture Scene
recall
1-precision
NN Matching with PROSAC
NN Matching with Region Context
NN matching
Ratio Matching with PROSAC
Ratio Matching with Region Context
Ratio Matching

(d)

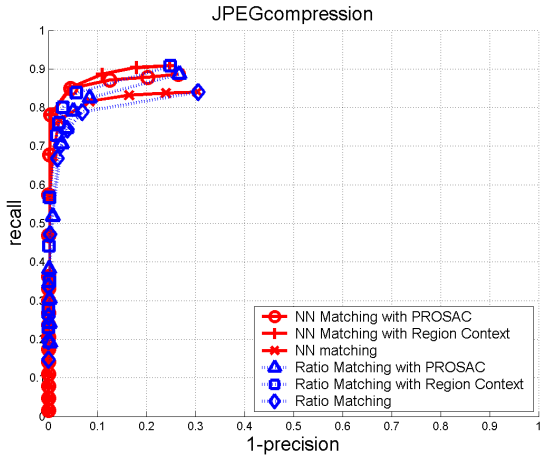


(e)

(f)



(g)



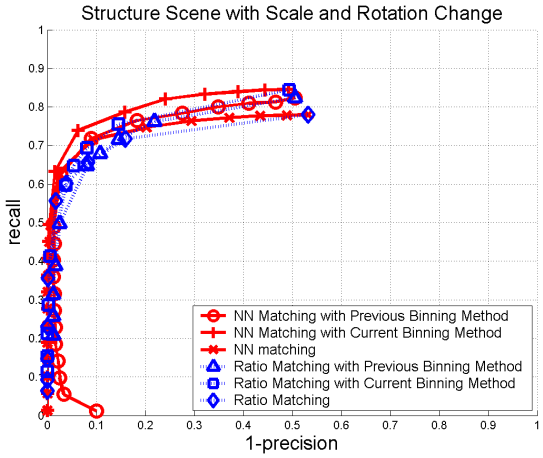
(h)

FIGURE 7.10. Comparison of matching performance with and without region context and with PROSAC for two matching strategies using six types of image transformations: (a) boat (previous page), (b) bark (previous page), (c) graffiti (previous page), (d) wall (previous page), (e) bike(previous page), (f) trees(previous page), (g) Leuven, (h) UBC. Images can be downloaded from: http://www.robots.ox.ac.uk/ vgg/research/affine/index.html.

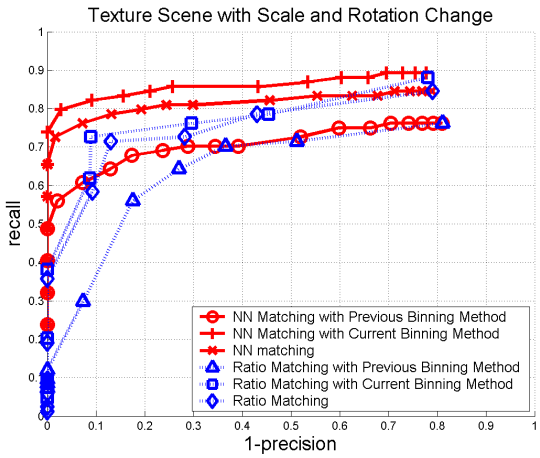
FIGURE 7.11. Example of how matching ambiguity can still exist even with 1-D epipolar constraints.

forcement matching relies on support from surrounding local features, a lack of neighborhood support diminishes its performance.

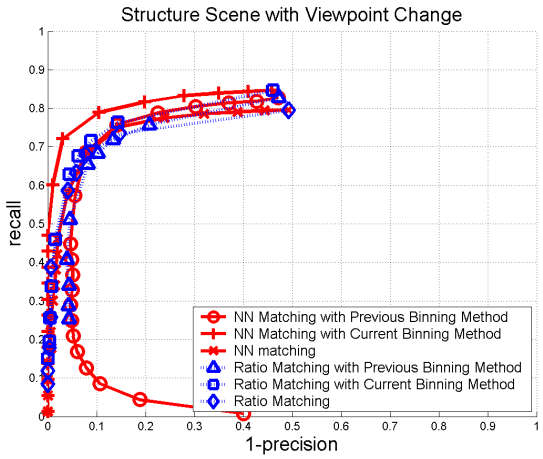
Another advantage of our method over RANSAC methods is that reinforcement matching doesn't need a transformation model and is therefore more flexible in that it can handle non-rigid or unknown transformations. We demonstrate this flexibility by matching images that have undergone affine, projective, polynomial, piecewise linear and sinusoidal transformations (Fig. 7.13). All seven transformations are applied to every image in the INRIA data set and compared with the first, untransformed image from each corresponding set. Results show that our method can increase the matching rate 8% on average over matching without region context (Fig. 7.14). We do not show results using RANSAC or PROSAC since an epipolar model is clearly incorrect and, consequently, these methods typically fail to arrive at a correct consensus. While we could apply


Structure Scene with Scale and Rotation Change
recall
1-precision
NN Matching with Previous Binning Method
NN Matching with Current Binning Method
NN matching
Ratio Matching with Previous Binning Method
Ratio Matching with Current Binning Method
Ratio Matching

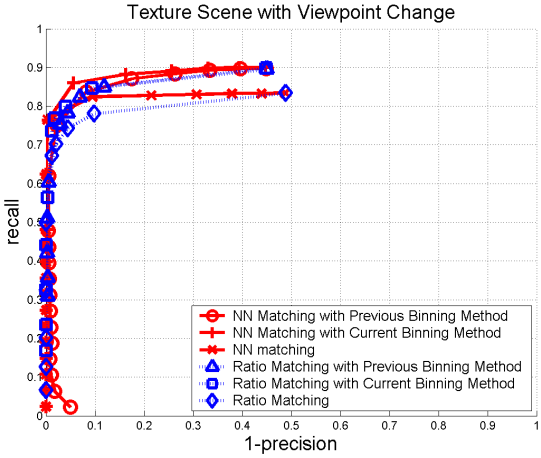
(a)


Texture Scene with Scale and Rotation Change
recall
1-precision
NN Matching with Previous Binning Method
NN Matching with Current Binning Method
NN matching
Ratio Matching with Previous Binning Method
Ratio Matching with Current Binning Method
Ratio Matching

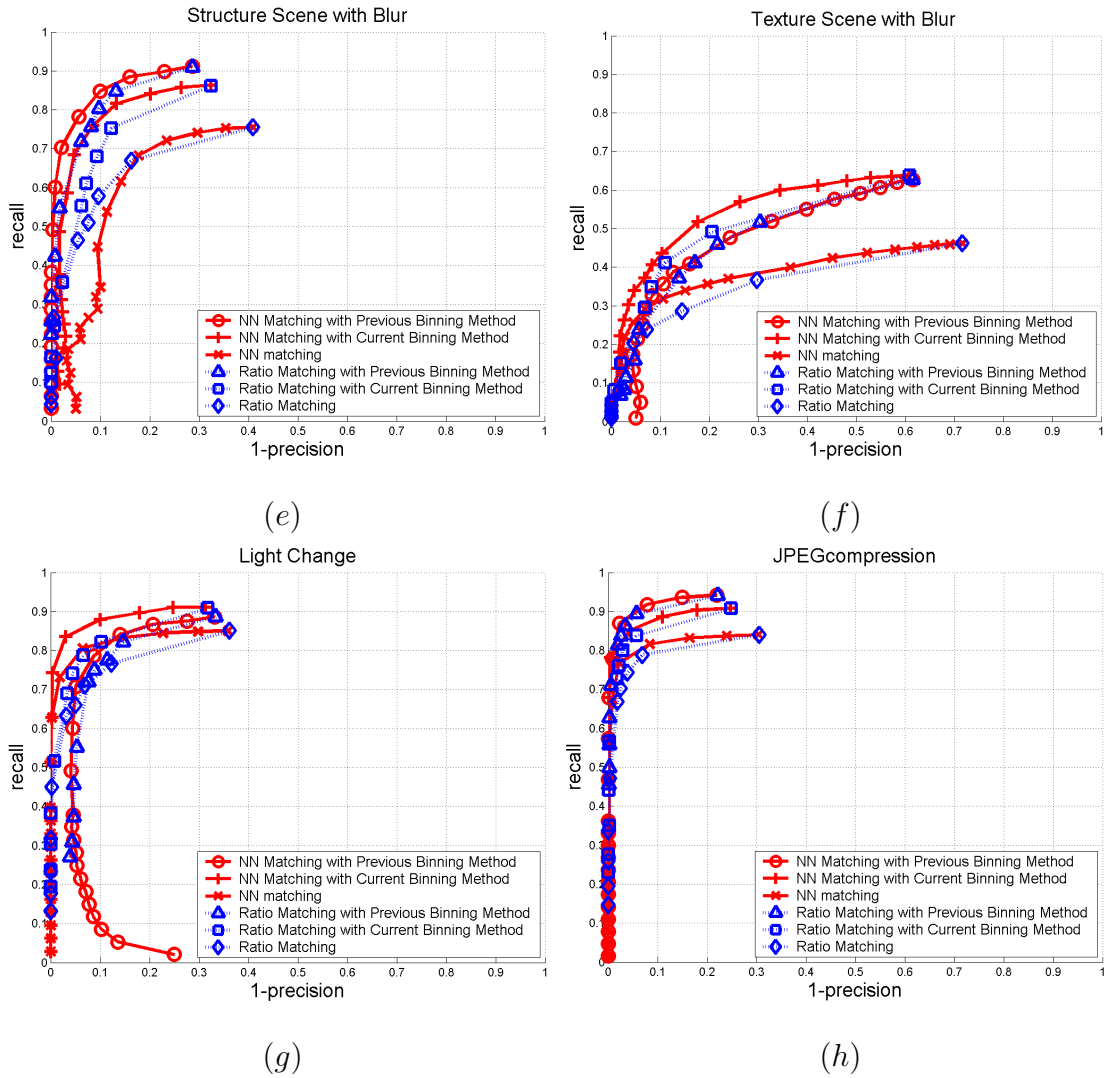
(b)


Structure Scene with Viewpoint Change
recall
1-precision
NN Matching with Previous Binning Method
NN Matching with Current Binning Method
NN matching
Ratio Matching with Previous Binning Method
Ratio Matching with Current Binning Method
Ratio Matching

(c)


Texture Scene with Viewpoint Change
recall
1-precision
NN Matching with Previous Binning Method
NN Matching with Current Binning Method
NN matching
Ratio Matching with Previous Binning Method
Ratio Matching with Current Binning Method
Ratio Matching

(d)



(e) (f)

(g) (h)

FIGURE 7.12. Comparison of matching performance with previous spatial binning method (a) boat (previous page), (b) bark (previous page), (c) graffiti(previous page), (d) wall(previous page), (e) bike, (f) trees, (g) Leuven, (h) UBC.

the correct transformation, since it is known, a different consistency model would have to be applied for each of the seven transformations. On the other hand, reinforcement matching does not require a transformation model and, as such, can be applied directly to all of the images regardless of the transformation. We use the known transformation model only for getting the ground truth of matching and didn't use them as pre-knowledge in the matching process.

To evaluate the performance of our new dominant gradient calculation, we compared the new method with the standard method on all images in the INRIA dataset with all other settings identical. On images without large affine changes, matching performance using our new method is the same or slightly better than that of the previous method. For images with large affine changes, the performance of our method is noticeably better (Fig. 7.15).

To evaluate the influence of the number of bins, we measured performance using configurations with 8, 16 and 24 bins. The configuration with 24 bins provides the best performance, but the difference between 24 bins and 16 bins is marginal. We examined the effect of using $5\%, 10\%, 20\%, 40\%, 60\%, 100\%$ of the total matched features as anchors. Variations of the recall score can be as large as 10%. The basic trend is that a higher percentage of anchor features improves performance- however, we achieve a rate of diminishing returns at about 20%. An exception to this increasing trend is that on some images with large zoom and rotation, the best-matched features have many errors, resulting in decreased performance with increased percentage of anchor features.

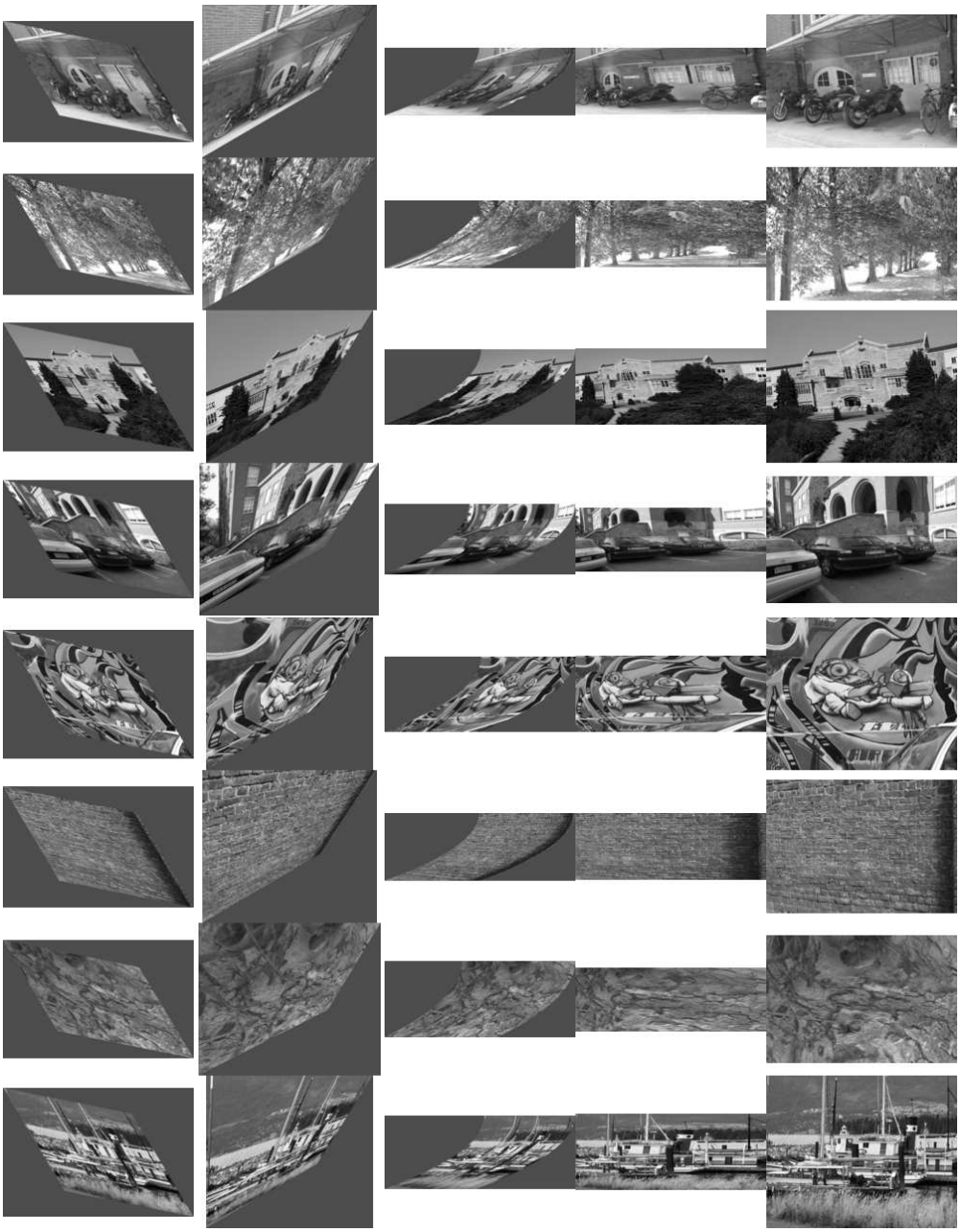


FIGURE 7.13. Deformed Inria Dataset.

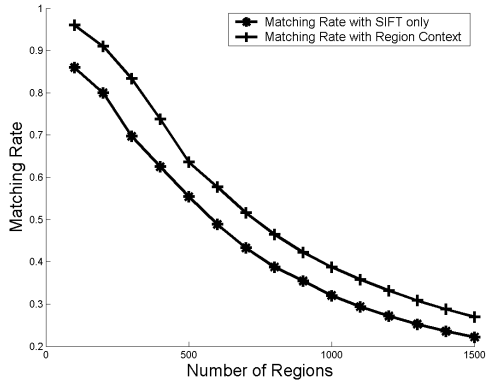


FIGURE 7.14. Matching rate on changing viewpoint images of a structured scene.

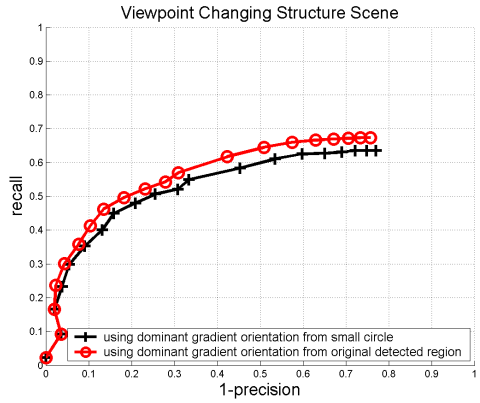


FIGURE 7.15. Comparison of two methods for dominant orientation calculation.

8. CONCLUSION AND FUTURE WORKS

This thesis proposed a novel approach for detecting image interest regions and two new methods for matching features. The first contribution is the feature detector, which is robust to local intensity perturbation. The second contribution is the: two feature matching methods that resolve ambiguities by including global context information into local feature matchings. In the following sections we present the conclusions and opportunities for future work.

8.1. Conclusion

This thesis has presented a new local interest region detector that is based on principal curvature and an enhanced watershed algorithm. The motivation behind this detector is the need to handle within-class variations for biological object recognition tasks. Previous intensity-based region detectors are sensitive to image intensity perturbations due to image noise and object within-class variation. In contrast, our PCBR detector is based on semi-local image structures which are more robust to image intensity perturbation. Our detector outperforms previous structure-based detectors for two reasons. First, the principal curvature detects both edges and curvilinear structures, thus providing for cleanner structural cues and clearer image sketches. Second, the watershed algorithm is more efficient than ellipse fitting for building affine regions. Fitting circles only achieves scale invariance and fitting ellipses to unstructured local edge cues is computationally prohibitive. PCBR also fits ellipses. The enhanced watershed algorithm provides a good approximation to the best fitted regions and it is also affine-invariant. For improved robustness, we adopted Eigenvector-flow based hysteresis thresholding and repeated detections over multiple scales.

The second contribution of this thesis is the development of two novel methods used to reject outliers while matching feature descriptors by using global context information. The first method incorporates a modified shape context into the local feature descriptor. Our new shape context builds the histogram by accumulating the principal curvature filter response within each bin. This histogram is more robust than the original shape context histogram built from thresholded edge points. The second matching method improves upon the first method by making it affine invariant and robust to occlusion. It employs elliptical bins to achieve affine invariance and distributed feature regions to achieve robustness to occlusion using a reinforcement matching scheme.

We apply these techniques to several applications to demonstrate the effectiveness and versatility of our approach. We considered applications of object recognition, especially biological object recognition, symmetry detection and image registration. In all applications, our methods provided good results in terms of robustness, accuracy and effectiveness.

8.2. Future Work

The principal curvature-based region detector and feature matching with global context presented in this dissertation provide many advantages over previous interest region detectors and matching methods. However, there is still plenty of room for additional improvements to this work. Some of the future directions to explore and possible extensions to this work include:

- *Building domain-specific interest region detectors.* Most of the current interest region detectors are application independent. Our principal curvature-based region detector is a kind of application-specific detector as it applies

to biological objects. However, this detetor is powerful only on objects with distinctive structure patterns and can not handle all biological objects. Current interest region detectors only use low level image information such as local intensity or color, which is incapable of solving current complicated computer vision problems. Some detection problems can only be solved by using high level application knowledge. For example, to detect the license plate of a vehicle, we can include the ratio of height and width of the rectangular license plate and the relative location of the license plate to the vehicle. This high level application knowledges can be combined with low level image information such as the intensity of characters on the plate in order to obtain the best detection. In my opinion, an all-purpose interest region detector that is suitable for all applications does not exist. Developing domain specific interest region detectors is more promising.

- *Fusing multiple detection cues.* The PCBR detector currently only detects edge and curvilinear structures in intensity (i.e.,grayscale) images (using the principal curvature as a detection cue). Images used in real applications are very complicated with a variety of features. Therefore, one future extension includes integrating all usable image cues such as texture, color, intensity, and edges because it can broaden the application domain of the detector and enhance its robustness to handle different images.

- *Incorporating machine learning techniques into interest region detector.* The PCBR detector is limited to the use of low level image information and fixed detector behavior. It has limited power to adapt to different images. Therefore, machine learning techniques must be included to make the detection more flexible and robust. Two directions can be pursued to apply machine

learning techniques to feature detection. The first is treating the detection problem as a classification problem. If we have training images containing the target object, a classifier can be built using these images. The problem of detecting this object is transformed to the problem of classifying images that contain the object as opposed to the background images. The second direction to pursue is to build self-adaptive feature detectors. Current interest region detectors have fixed behaviors, such as searching for corners, blobs etc. PCBR is the same in this respect because it only detects curvilinear structures with predefined parameters. Therefore, PCBR can not adapt to other distinctive features. Current computer vision tasks are so complicated that distinctive features can change. For example, in the application of tracking a person, when the person is close to the surveillance camera, interest region detectors like SIFT are appropriate to use. But when the person moves away from the camera, SIFT is not able to extract useful points from the person. In this situation, the detector should switch to another image or context information to locate the target. Machine learning techniques can make the detector learn and select the best detection cue, build strong detections based on many weak detections and adapt current detections based on performance. Overall, the detector should be self-adaptive and performance-oriented, which is more effective than the current local image feature-oriented detectors with fixed behaviors.

BIBLIOGRAPHY

[1] S. Agarwal and A. Awan and D. Roth. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26(11), page 1475–1490, 2004.

[2] H. Asada and M. Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 8(1), pages 2–14, 1986.

[3] A. Baumberg. Reliable feature matching across widely separated views. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.

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

[5] S. Belongie and J. Malik and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. *Proceedings of Neural Information Processing Systems*, pages 831–837, 2000.

[6] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 14(2), pages 239–256, 1992.

[7] M. Blaschko, G. Holness, M. Mattar, D. Lisin, P. Utgoff, A. Hanson, H. Schultz, E. Riseman, M. Sieracki, W. Balch, and B. Tupper. Automatic In Situ Identification of Plankton. *IEEE Workshop on Applications of Computer Vision*, pages 79–86, 2005.

[8] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 710–715, 2005.

[9] K. Bowyer and C. Kranenburg and S. Dougherty; Edge detector evaluation using empirical ROC curves. *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 23–25, 1999.

[10] M. Brown and D. G. Lowe. Invariant features from interest point groups. *The British Machine Vision Conference*, pages 656–665, 2002.

[11] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 8, pages 679–698, 1986.

[12] G. Carneiro and A. Jepson. Pruning Local Feature Correspondences using Shape Context. *IEEE International Conference of Pattern Recognition*, volume 3, pages 16–19, 2004.

[13] G. Carneiro and D. G. Lowe. Sparse flexible models of local features. *European Conference on Computer Vision*, volume 3, pages 29–43, 2006.

[14] G. Carneiro and A. Jepson. The distinctiveness, detectability, and robustness of local image features. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–301, 2005.

[15] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 44–51, 2000.

[16] O. Chum and J. Matas. Matching with PROSAC-Progressive Sample Consensus. *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 220–226, 2005.

[17] C. Coelho and A. Heller and J. Mundy and D. Forsyth and A. Zisserman. An experimental evaluation of projective invariants. *Geometric invariance in computer vision*, Pages 87–104, ISBN:0-262-13285-0, 1991.

[18] T. M. Cover and J. A. Thomas. Elements of Information Theory. *John Wiley and Sons, Inc*, 1991.

[19] D. Crandall, P. Felzenszwalb and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10–17, 2005.

[20] A. D. Cross and E. R. Hancock. Graph matching with a dual-step EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1236–1253, 1998.

[21] G. Csurka, C. Bray and C. Dance L. Fan. Visual categorization with bags of keypoints. *Workshop of European Conference on Computer Vision*, 2004.

[22] H. Deng, E. Mortensen, L. Shapiro, T. Dietterich Reinforcement Matching using Region Context. *IEEE Conference on Computer Vision and Pattern Recognition Beyond Patches Workshop*, 2006.

[23] H. Deng, W. Zhang, E. Mortensen, T. Dietterich, L. Shapiro Principal curvature-based regions for object recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[24] R. Deriche and G. Giraudon A computational approach for corner and vertex detection. *International Journal of Computer Vision*, Volume 10(2), pages 101–124, 1992.

[25] G. Dorko and C. Schmid. Object class recognition using discriminative local features. *IEEE Transactions on Pattern Analysis and Machine Intelligence submitted*, 2004.

[26] B. Epshtein and S. Ullman. Feature hierarchies for object classification. *IEEE International Conference on Computer Vision*, Volume 1, pages 220–227, 2005.

[27] R. Fergus, P. Perona and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2, pages 264–271, 2003.

[28] R. Fergus, P. Perona and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, Volume 1, pages 380–387, 2005.

[29] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition, *International Journal of Computer Vision*, volume 61(1), pages 55–79, 2005.

[30] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communication of the ACM*, volume 24(1), pages 381–395, 1981.

[31] W. Frstner. A framework for low-level feature extraction. *European Conference on Computer Vision*, volume 2, pages 383–394, 1994.

[32] F. Fraundorfer and H. Bischof. Evaluation of local detectors on non-planar scenes. *Proc. of the 28th Workshop of the Austrian Association for Pattern Recognition*, pages 125–132, 2004.

[33] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–151, 1988.

[34] M. Heath and S. Sarkar and T. Sanocki and K. Bowyer. Robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19(12), pages 1338–1359, 1997.

[35] Y. A. Hicks and D. Marshall and P. L. Rosin and R. R. Martin and D. G. Mann and S. J. M. Droop. A model of diatom shape and texture for analysis, synthesis and identification. *Machine Vision and Applications*, pages 1–11, 2006.

[36] G. E. Hinton, C. K. Williams and M. Revow. Adaptive elastic models for hand-printed character recognition. *Proceedings of Neural Information Processing Systems*, pages 512–519, 1992.

[37] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 90–96, 2004.

[38] T. Kadir, A. Zisserman and M. Brady. An affine invariant salient region detector. *European Conference on Computer Vision*, pages 228-241, 2004.

[39] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 506-513, 2004.

[40] Y. Lamdan and H.J. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. *IEEE International Conference on Computer Vision*, pages 238–249, 1988.

[41] N. Landwehr and M. Hall and E. Frank. Logistic model trees. *Machine Learning*, Volumn 59, pages 161–205, 2005.

[42] T. Lindeberg and J, Garding. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *Image and Vision Computing*, pages 415–434, 1997.

[43] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, volume 30(2), pages 79–116, 1998.

[44] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *Image and Vision Computing* , pages 415–434, 1997.

[45] G. Lohmann and D. Y. Cramon. Automatic labeling of the human cortical surface using sulcal basins. *Medical image analysis* , pages 179–188, 2000.

[46] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, volume 293(10), pages 133–135, 1981.

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

[48] D. G. Lowe. Object recognition from local scaleinvariant features. *IEEE International Conference on Computer Vision*, pages 682–688, 1999.

[49] G. Loy and J-O. Eklundh. Detecting symmetry and symmetric constellations of features. *European Conference on Computer Vision*, pages 508–521, 2006.

[50] J. Matas and O. Chum and M. Urban and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, volume 22(10), pages 761–767, 2004.

[51] G. Medioni and Y. Yasumoto. Corner detection and curve representation using cubic B-splines. *Graphical Model and Image Processing*, volume 39, pages 267–278, 1987.

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

[53] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *European Conference on Computer Vision*, volume 1(1), pages 128–142, 2002.

[54] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 257–264, 2003.

[55] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, volume 60(1), pages 63–86, 2004.

[56] S. Mitra and Y. Liu. Local facial asymmetry for expression classification. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 889–894, 2004.

[57] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20(12), pages 1376–1381, 1998.

[58] H. Moravec. Towards automatic visual obstacle avoidance. *International Joint Conference on Artificial Intelligence*, pages 584, 1977.

[59] E. Mortensen, H. Deng and L. Shapiro. A SIFT descriptor with global context. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 184–190, 2005.

[60] N. Larios, H. Deng, W. Zhang, M. Sarpola, J. Yuen, R. Paasch, A. Moldenke, D. A. Lytle, S. Ruiz Correa, E. Mortensen, L. G. Shapiro, T. G. Dietterich Automated insect identification through concatenated histograms of local appearance features. *IEEE Workshop on Applications of Computer Vision*, pages 26, 2007.

[61] N. Larios, H. Deng, W. Zhang, M. Sarpola, J. Yuen, R. Paasch, A. Moldenke, D. A. Lytle, S. Ruiz Correa, E. Mortensen, L. G. Shapiro, T. G. Dietterich

Automated Insect Identification through Concatenated Histograms of Local Appearance Features. *Journal of Machine Vision and Applications*, 2007.

[62] A. Opelt, M. Fussenegger, A. Pinz and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. *European Conference on Computer Vision*, pages 71–84, 2004.

[63] J. Pilet and V. Lepetit and P. Fua. Real-time Non-Rigid Surface Detection. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 822–828, 2005.

[64] P. Pritchett and A. Zisserman. Wide baseline stereo matching. *IEEE International Conference on Computer Vision*, pages 754–760, 1998.

[65] S. Russell, P. Norvig. Artificial intelligence - a modern approach. *Prentice Hall*, 2003.

[66] C. Schmid and R. Mohr and C. Bauckhage Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, volume 37(2), pages 151-172, 2000.

[67] C. Schmid and R. Mohr Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19(5), pages 530-535, 1997.

[68] R.Sedgewick. Algorithms. *Addison-Wesley*, 2nd edition, 1988.

[69] E. Shilat and M. Werman and Y. Gdalyahu. Ridge's corner detection and correspondence. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 976–981, 1997.

[70] A. Shokoufandeh and I. Marsic and S. J. Dickinson. View-based object recognition using saliency maps. *Image and Vision Computing*, volume 17(5-6), pages 445-460, 1999.

[71] S. Smith and J. Michael Brady SUSAN a new approach to low level image processing. *International Journal of Computer Vision*, volume 23(1), pages 45–78, 1997.

[72] C. Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20(2), pages 113–125, 1998.

[73] D. Tell and S. Carlsson. Wide Baseline Point Matching using Affine Invariants Computed from Intensity Profiles. *European Conference on Computer Vision*, pages 814–828, 2000.

[74] L. Torresani and D. Yang and E. Alexander and C. Bregler. Tracking and Modeling Non-Rigid Objects with Rank Constraints. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 493–498, 2001.

[75] M. Turk and A. Pentland. Face recognition using eigenfaces. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[76] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, volume 59(1), pages 61–85, 2004.

[77] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. *The British Machine Vision Conference*, pages 412–425, 2000.

[78] S. Ullman and E. Sali and M. Vidal-Naquet. A fragment-based approach to object representation and classification. *International Workshop on Visual Form, Berlin: Springer*, page 85–100, 2001.

[79] N. Vasconcelos Bayesian models for visual information retrieval. PhD thesis, MIT., 2000.

[80] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13(6), pages 583–598, 1991.

[81] A. Y. Yang and S. Rao and K. Huang and W. Hong and Y. Ma. Geometric segmentation of perspective images based on symmetry groups. *IEEE International Conference on Computer Vision*, pages 1251–1258, 2003.

[82] H. Zabrodsky and S. Peleg and D. Avnir. Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17(12), pages 1154–1166, 1995.

[83] Z. Zhang and R. Deriche and O. Faugeras and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, volume 78(1), pages 87–119, 1995.

[84] W. Zhang and H. Deng and T. Dietterich and E. Mortensen. A hierarchical object recognition system based on multi-scale principal curvature regions. *IEEE International Conference on Pattern Recognition*, volume 1, pages 778–782, 2006.

[85] S. Y. Zhu and G. Schaefer. Springer Lecture Notes in Computer Science. *LNCS* , volume 3337, pages 182–187, 2004.