KEYPOINT DETECTION AND DESCRIPTION ON IMAGE CURVES

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Ali KÖKSAL

July 2017
İZMİR

We approve the thesis of **Ali KÖKSAL**

Examining Committee Members:

Assoc. Prof. Dr. Derya BİRANT
Department of Computer Engineering , Dokuz Eylul University

Assoc. Prof. Dr. Yalın BAŞTANLAR
Department of Computer Engineering, İzmir Institute of Technology

---

Asst. Prof. Dr. Mustafa ÖZUYSAL
Department of Computer Engineering, İzmir Institute of Technology

14 July 2017

Asst. Prof. Dr. Mustafa ÖZUYSAL
Supervisor, Department of Computer Engineering
İzmir Institute of Technology

Assoc. Prof. Dr. Yusuf Murat ERTEN
Head of the Department of
Computer Engineering

Prof. Dr. Aysun SOFUOĞLU
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my supervisor Mustafa ÖZUYSAL for his motivation, patience, and understanding. Without his constant support and mentorship, this thesis work would have been a disappointing pursuit.

I would like to thank my colleagues especially members of Visual Intelligence Research Group (VIRG) for their support and valuable feedbacks.

To my family, I am grateful for their unconditional love and infinite support. Without their sacrifices, encouragement, and advising none of my success would be possible.

This thesis is partially supported by The Scientific and Technical Research Council of Turkey (TUBITAK) under the grant 113E496.

ABSTRACT

KEYPOINT DETECTION AND DESCRIPTION ON IMAGE CURVES

Image curves are one of the choices for representing interest points which also provide discriminative information about images. Boundary of regions and contour of shapes are real-time instances of image curves. In this thesis, we propose two approaches for keypoint detection and description on image curves. To extract keypoints on image curves, we compute the extrema curvature of region boundaries. This mechanism improves repeatability of keypoints on 3D data. For the description of image curves, shape contours are used. This is similar to approaches that describe the features based on shapes and image gradients. Unlike these approaches, we combine spatial and directional information of tangent directions to extract a feature vector that leads to improved matching and recognition on several standard computer vision tasks such as character and object recognition.

ÖZET

İMGE EĞRİLERİ ÜZERİNDE ANAHTAR NOKTA TESPİT VE BETİMLENMESİ

İmge eğrileri, imgeler hakkında ayırt edici bilgi sağlarken anahtar noktaların temsil edilmesinde kullanılmaktadır. Bölge sınırları ve şekil konturları, imge eğrilerinin gerçek zamanlı örneklerdir. Bu tez çalışmasında, imge eğrileri üzerinde anahtar nokta tespiti ve betimlenmesi için iki yöntem önerilmiştir. Bölge sınırlarının eğriliğinin uç noktaları, imge eğrilerinde anahtar nokta tespiti için hesaplanmıştır. Bu yöntem üç boyutlu nesnelerde, anahtar nokta tekrarlanabilirliğinin iyileşmesini sağlamıştır. İmge eğrilerinin tanımlanmasında ise şekillerin konturları kullanılmıştır. Bu yaklaşım, şekiller üzerinde tanımlama yapan yöntemler ve imge gradyanlarına benzeyip, bu yöntemlerden farklı olarak tanjant yönlerinin konumsal ve yönsel özellikleri, özellik vektörü hesaplanmasında bir arada kullanılmıştır. Özellik vektörleri karakter ve obje tanıma gibi çeşitli standart bilgisayarlı görü alanlarında tanıma ve eşleştirmede iyileşmeye yol açar.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

MSER	Maximally Stable Extremal Regions
SIFT	Scalable Invariant Feature Transform
GLOH	Gradient Location-orientation Histogram
SC	Shape Context
PCA	Principal Component Analysis
SPIN	Spin Images
JLA	Steerable Filters
KOEN	Differential Invariants
CF	Complex Filters
MOM	Moment Invariants
CC	Cross-correlation
GB	Geometric Blur
MR8	Maximum Response of Filter
CSS	Curvature Scale Space
CPDH	Contour Points Distribution Histogram
EMD	Earth Mover's Distance
IDSC	Inner Distance Shape Context
MCSS	Morphological Curvature Scale Space
MDS	Multidimensional Scaling
DP	Dynamic Programming
SCC	Skeletal Context

CHAPTER 1

INTRODUCTION

The main aim of the computer vision is to understand the content of the images or videos. Its applications such as motion estimation, object detection, recognition, and tracking utilize the low-level image features such as edges, corners, and curves instead of the whole image in order to understand the content. In many computer vision applications, image features are represented by interest points numerically. Some information about the feature like its orientation, scale, and intensity value is computed for each interest point. So in order to understand the content of the image, features are detected and described by using keypoints and their descriptors. In other words, keypoints are supporting factors of the algorithms. Because of that, any improvement on the keypoint detection and description algorithms affect the performance of the applications.

In the literature, there are several keypoint detection algorithms. Their approaches have a wide variety. For example, while an algorithm applies a local analysis that is focusing on the intensities of the neighborhood pixels [22], another algorithm applies a global analysis focusing on the intensities of the whole pixels of the image [24]. While some of them retrieve keypoints as an individual point, the others retrieve a group of points like regions. Due to this wide variety, in the applications one of them is selected to meet the requirements of the application.

Although keypoint detection algorithms have their own nature in a wide range, in the literature there are common metrics to measure their performance. Such metric is called repeatability. It is a well-accepted metric to compare different keypoint detection algorithms. Repeatability is measured on images that are taken from different views of an object. First, keypoints are detected on images, then they are transformed into the same image. And it is measured by counting how many of them are detected on both images.

After features on an image are detected, their descriptors are computed. Computing a descriptor for keypoints on the image is a way to describe the features. In the literature, there are many descriptors, and their approaches are different from each other like keypoint detectors. For example, while some of them define a patch around the keypoint by using its scale and computes the descriptor by using the intensity values of the pixel inside the patch [22, 28], others use the boundary points of the object [2, 30]. While an algorithm retrieves a vector of floating point numbers [22], the other creates a vector of binary numbers as a descriptor [5]. And also the dimension of the descriptor changes according to the algorithm. In spite of this variety, in order to the measure the performance of the descriptors, there is a metric that is called matching score or recognition rate. This metric allows to compare the success of the descriptors and observe their robustness. To measure matching score, keypoints are detected and their descriptors are computed on images that are taken from different views of an object. Then between images, a match for each descriptor is found by giving them to a classifier. After matches are found, if a match is the same with the ground truth correspondence, it is labeled as a correct match. Otherwise, it is labeled as incorrect. And the ratio between the number of correctly matched keypoints and the number of keypoints gives the matching score. So it means that keypoints are recognized with those ratios on the other views of the same object.

In this thesis, our main focus is image curves such as contour of shape of silhouettes, boundary of regions that are detected by region based keypoint detectors. We perform three distinct but related studies:

- Analysis of MSER Stability
- Detection along Extremal Region Boundary
- Description on Image Curves

1.1. Analysis of MSER Stability

1.1.1. Motivation

In the literature, the stability of MSER that is one of the region based keypoint detector is analyzed in [29, 31]. The result of these studies show that MSER is one of the most successful detectors. This part of the thesis covers a detailed analysis of MSER stability in order to understand the behavior of the detector and find its vulnerabilities.

1.1.2. Objectives and Contributions

The purpose of this part is designing a detailed and realistic stability analysis for regions that are detected by region based detectors especially by MSER. To obtain a detailed analysis, we generate synthetic images densely in the affine parameter space by deforming an image with three camera position parameters. Furthermore, to make the analysis realistic, regions are converted to convex hulls. This approximation provides a feasible and faster analysis than using regions directly and more sensitive than the ellipse approximation which is used in literature. So convex hull approximation provides a more realistic analysis.

There are three main contributions. First, generating synthetic images allows us to observe the relation between the stability of MSER and three camera position parameters. Second, the effect of the small amount of the deformations is measured by generating synthetic images with a small change of the parameters. This is important since small changes of the parameters give an opportunity to observe the success of detectors when they are used in object tracking applications. Third is using a convex hull approximation in the stability analysis.

Chapter 3 gives the details of the approach and the experimental results.

1.2. Detection along Extremal Region Boundary

1.2.1. Motivation

After analyzing the stability of the extremal region detector named MSER under various deformation, we make two observations. First, the boundary of the regions is more stable than their area. For example, when the image is deformed, the areas of the regions are changed drastically. On the other hand, their boundary remains stable enough to match and recognize them. Moreover, under partial occlusion, while at least a part of the boundary remains the same, the area is not stable enough to recognize the regions. Second, in the literature, MSER are used by their ellipse representation in general but this causes information loss because the boundary of the regions includes structures like intrusions and extrusions that are ignored due to the ellipse approximation.

1.2.2. Objectives and Contributions

The objectives of this part to make regions more stable and to prevent the information loss. To meet the objectives, we design a local analysis that focuses on the boundary of one region at a time. This approach takes the advantage of the stability of region boundaries and utilizes the boundary points in order to detect interest points along the boundary.

This part has two main contributions. First, the proposed approach increases the repeatability. Second, it increases the usability of regions in some field of computer vision especially in 3D operations because 3D operations require individual points instead of regions and points that are detected by the proposed approach has better performance than the center points of the regions.

1.3. Description On Image Curves

1.3.1. Motivation

In many computer vision applications, image features are represented by keypoints. To use them efficiently descriptors for keypoints are computed. During the computation, some information such as texture around the keypoint and shape of the keypoint is used. However, there are some objects that have significant shapes such as characters and such objects shape of the features is more important than the texture of the image. So for character recognition, texture inside and outside of the character and the other shapes around it has no valuable information and even sometimes they cause confusion when they are classified and recognized. To prove that we started to examine images of the characters. And we reached three main observations. First, there is enough information on the contours of the characters in order to recognize them. Second, when a descriptor is computed to match a character in a word, the effect of the characters before and after the targeted character can be ignored. And the last observation is that descriptors also should not be affected by the texture around the characters such as texture on the background.

1.3.2. Objectives and Contributions

The aim of this part is designing a shape based descriptor which recognizes objects that have significant and discriminative shape. To fulfill this aim, we design a descriptor by gathering and combining the directional and spatial information from the contour of the shape. The proposed approach has two main phases which are orientation estimation and descriptor computation. When orientation is estimated, the dominant direction is calculated from tangent directions as opposed to gradient information of each contour point. And then the shape is normalized with the estimated orientation to prepare the shape for descriptor computation. In descriptor computation, the location of the contour points of normalized shape gives the spatial information and the tangent direction of them gives the directional information.

In this part, there are two main contributions. First, the proposed approach computes descriptors by considering only the contour of the shape of the objects. So any information that is gathered from any other source is not used in the descriptor. Because this leads to classifying shapes that have distinctive contours. Relatedly, the dominant direction of the shape and also directional information that is used in the descriptor is computed with the tangent direction of contour points instead of the gradient information that is exploited by the majority of descriptors.

1.4. Thesis Outline

This thesis organized as follows. In the following chapter, literature overview and background information are given. Chapter 3 covers a detailed analysis of the stability of MSER which is one of the keypoint detection algorithms. In the next chapter, we suggest an approach in order to detect interest points on extremal region boundary. In Chapter 5, a novel shape based descriptor is proposed. The last chapter gives the final remarks and future work.

CHAPTER 2

RELATED WORK AND BACKGROUND

2.1. MSER and Analysis of Interest Point Stability

One of widely used image features is extremal regions and they are affine invariant and sensitive to change of lightning. Matas et al. [24] propose an algorithm to detect extremal regions. According to the algorithm, the intensity levels of the image are added from white to black and vice versa step by step and connected components are defined. After each step, the stability of connected components is analyzed. By considering the analysis, enough stable extremal regions are selected as maximally stable extremal regions. So MSER are the subset of extremal regions and the intensity values of the pixels around the regions are either darker or brighter than the intensity values of the pixels inside them. After MSER are detected, they are retrieved as ellipses, lines, and extended boundaries. In the literature, their ellipse representation is used in general. Some of MSER that are extracted from the first image of the Graffiti image sequence of Oxford dataset are shown in Figure 2.1. During the detection, Matas' executable is run and 540 extremal regions are detected as maximally stable. Randomly selected 50 MSER among them are indicated on the top image of Figure 2.1 to obtain better visualization. On the bottom of the figure, nine randomly selected regions are zoomed.

MSER algorithm has five parameters for gray scale and they are listed and described below:

- Minimum size: This parameter adjusts the size of output region and eliminates the regions that are smaller than minimum area. Its default value is 60.
- Maximum size: This parameter provides for eliminating the regions that are bigger than maximum area and its default value is 14400.
- Delta: It compares the area of the current connected component and the connected component after another possible threshold is added. So this parameter is related to

the variation of the regions. Its default value is five and when it is getting higher, the number of MSER is getting lower.

- Maximum variation: It is related to the variation of regions like the parameter delta. If the variation of a region is higher than this parameter, it means that this region might be relatively stable and not absolutely stable enough to be an MSER. And The default value for this parameter is 0.25 and smaller maximum variation means a decrease in the number of MSER.

- Minimum diversity: This parameter provides for eliminating too similar regions. Compare the size of two similar regions. And it is the ratio of the difference the size of the regions and the size of the smaller region. Its default value is 0.2 and larger minimum diversity causes a decrease in the number of the MSER.

In the literature, there is a comparison strategy for region based detectors which is proposed by Mikolajczyk et. al. [29]. This approach is evaluated on six types of region based detectors which are Harris-Affine region detector [27, 33], Hessian-Affine region detector [27], maximally stable extremal region detector [24], edge based region detector [42], intensity extrema-based region detector [43], and salient region detector [11]. They are explained briefly in Section 2.2. In the evaluation of the detectors, two types of comparisons are performed which are repeatability and matching score measurement and Oxford dataset is used.

To compute the repeatability of elliptical regions, below steps are followed:

- Projecting elliptical regions from the reference image to the deformed image

- Removing regions from outside of the common part of images

- Computing overlap error which is calculated by subtracting the overlap ratio of regions from one. The overlap ratio is the ratio of the intersection over the union of the regions.

- Selecting one to one correspondences

In the experiments, the overlap ratio threshold is fixed to 40%. So if the overlap ratio error of two regions is lower than 40%, they are accepted as repeated. The repeatability calculation is followed for increasing transformations which are viewpoint, scale, blur, and light change and JPEG artifacts.

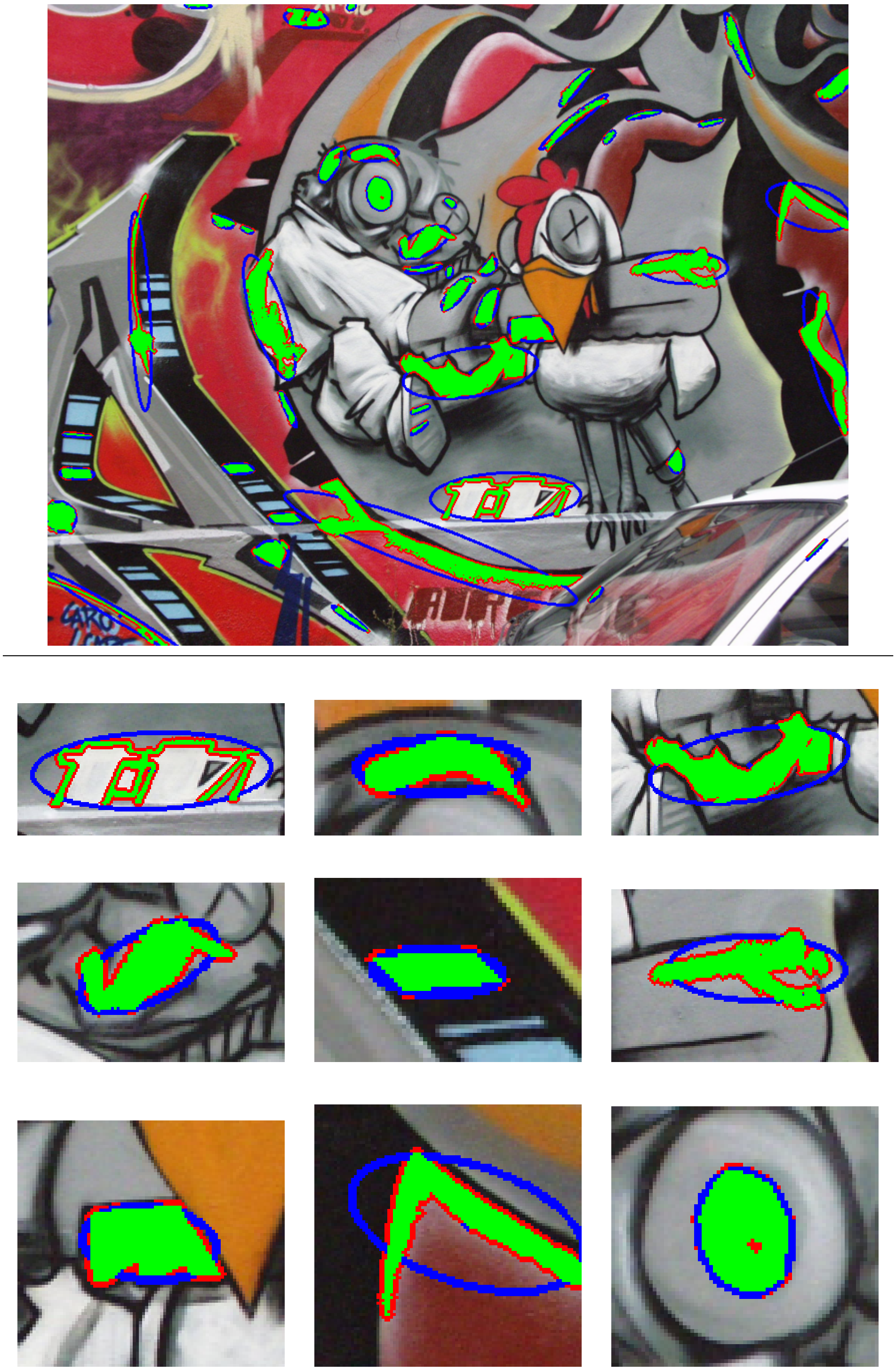


Figure 2.1. MSER are detected on the first image of Graffiti image sequence of Oxford dataset. Three representations of those regions are shown. Green region is obtained by drawing the line representation. The red boundary shows extended boundary representation of the regions. The ellipse representations are drawn with blue.

In the matching experiment, SIFT descriptor is computed for each region. SIFT which is proposed in [22] is a 128 dimensional texture based descriptor and its details are explained in Section 2.3. In this study, descriptors are computed over a circular patch instead of a rectangular patch. And the elliptical regions that are extracted by detectors are mapped into circular, then these circular regions are normalized with dominant gradient direction. Afterward, descriptors are computed by calculating the spatial distribution of gradients of these normalized circular regions. After descriptors are computed, matching score for each detector is calculated. In the calculation of matching score, matches between the regions of the reference image and the deformed images are found by the nearest neighbor approach. And Euclidean distance is used as a comparison metric for descriptors. After the match of a region is found, it is checked with the ground truth correspondence of the region. And as the ground truth, region by region correspondences that are found in the repeatability calculation are used. As a result, for a region, its match and its ground truth correspondence are the same, it is accepted as a correct match otherwise an incorrect match. And the ratio of the correct match over the number of regions on the reference image is the matching score.

2.2. Detection along Extremal Region Boundary

In the literature, there are six well known region based detectors. They detect regions which are set of consecutive pixels.

Harris-Affine region detector [27, 33] detects interest points with multi-scale Harris detector based on the Laplacian. Then an elliptical region is determined for each interest point. Its scale is selected by using the responses of the Laplacian over scales during the interest point detection. And its shape is determined by using autocorrelation matrix of intensity gradient.

Hessian-Affine region detector [27] is a similar detector with Harris-Affine. In this approach interest points are detected with a detector based on Hessian matrix instead of multi-scale Harris detector. The entries of this matrix are sensitive to blob structure especially. And to determine an elliptical region the process that is used in Harris-Affine is applied. So scale is selected based on Laplacian and shape is determined with autocorrelation matrix.

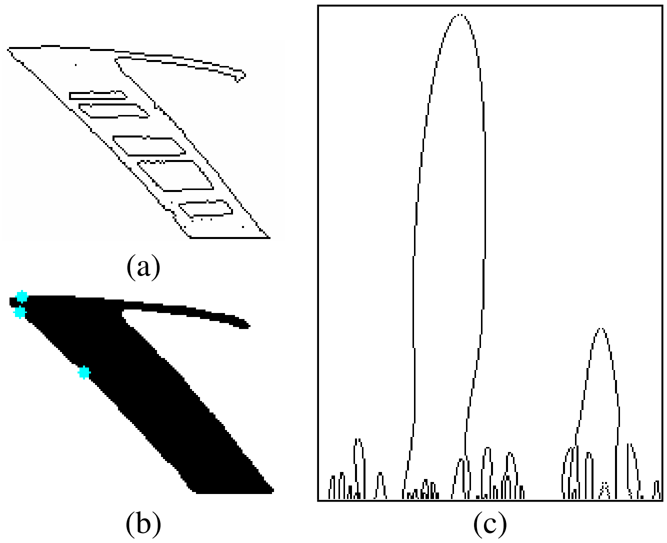
Maximally stable extremal region detector [24] detects extremal regions which is a kind of image feature. They are detected as connected components and stable extremal regions are selected by considering the stability of connected components. Details of this detector is explained in Section 2.1.

In the *edge based region detection* [42], corners and nearby edges are detected at multiple scales with Harris corner detector and Canny edge detector. Then two other point is selected by moving in both directions of the edge. Afterward, with these three points, a parallelogram region is detected by choosing the opposite of the corner as the fourth point of the parallelogram.

Intensity extrema-based region detector [43] detects an intensity extremum at multiple scale. Then from starting with this point, hypothetical rays are drawn in all directions. When moving away on the rays, a point that has high intensity change is selected as the boundary point of the region. After one boundary point on each ray is found, the arbitrary shape of the region is replaced with an ellipse.

Salient region detector [11] detects regions by using the probability density function of intensities of an image. The approach has two steps. First, at each pixel of an image, the entropy of the probability density function is constructed by considering ellipse parameters and scale. Then the set of the entropy extrema are selected as candidate salient regions. Second, the desired number of salient region is selected by ranking the candidates with respect to the derivative of the probability density function.

In the literature, there is an approach which detects interest points on the image curves such as the contour of shapes, boundary of regions that are detected by region based detectors etc. This approach is proposed by Mai et. al. [23] and it detects affine invariant points on the contour of shapes. Then contour segments are constructed from every two consecutive affine invariant points. After contour segments of two shapes are found, they are aligned by using Smith-Waterman algorithm which is a dynamic programming algorithm to find alignment by using a scoring system. After an initial contour segment alignment is provided, it is extended as long as possible to create a match between shapes. Since detection of affine invariant points can be used for the boundary of regions that are detected by region based detectors, this part is explained in detail. To detect affine invariant points, first curvature scale space (CSS) image is constructed and second, the local maxima of this image give the location of the affine invariant points on the region boundary. In order to generate the curvature scale space image, the boundary of a region is convolved with different Gaussian kernels. The kernel size is increased until the boundary becomes convex and there are no zero crossings. In Figure 2.2, there is an example region that is detected by MSER detector. And its CSS image and affine invariant points that are detected on its boundary are also indicated.


(a)

(b)

(c)

Figure 2.2. An example region that is detected by MSER detector is shown at (a). On the boundary of this region, interest points are detected with CSS approach [23] and they are shown at (b). These affine invariant points are represented as blue dots on the boundary of the region. At (c), CSS image that is computed during the affine invariant point detection is indicated. For this region, CSS approach detects three affine invariant points.

2.3. Description on Image Curves

In the literature, there are several keypoint description algorithms and in general, they can be classified into three main groups: texture based, contour based, and the others. The *texture based keypoint descriptors* such as scale-invariant feature transform (SIFT) [22], gradient location orientation histogram (GLOH) [28] etc. computes descriptors by using texture around the keypoint. The *contour based keypoint description* algorithms such as shape context (SC) [2] and curvature scale space (CSS) [30] use the contour points of the keypoint to compute descriptor. When several descriptors are proposed, besides the computation of descriptor itself, a special metric to measure the similarity and a special matching procedure is proposed as well. And in general, they are complex and sophisticated algorithms. In the following, some keypoint description algorithms and, if there is, their special metrics and special similarity matching procedures are explained briefly.

Several texture based approaches are described below.

Cross correlation (CC) [28] is computed by accumulating the pixels of the image into a vector.

Complex filters(CF) takes keypoints which are represented with ellipses such as MSER. The descriptor is computed by mapping the intensity values of the pixels that are located inside the elliptical region into a unit disk of radius one.

Gradient location orientation histogram (GLOH) [28] is similar to scalable invariant feature transform. To compute GLOH, descriptor a circular grid is fitted onto the keypoint and its surroundings. The grid has 17 bins which are obtained by dividing the grid into three in the radial direction. And outer two bins are divided into eight in the angular direction as well. Then for each of bin 16 dimension gradient histogram is computed. After applying principal component analysis (PCA), the dimension of the descriptor is reduced to 128 from 272.

Steerable filters (JLA) [8] define an image patch around the keypoint and take the derivate of this image patch up to fourth order by convolving Gaussian. The dimension of the computed descriptor is 14.

Differential invariants (KOEN) [14] is an approach like JLA. To compute KOEN descriptors, derivatives of an image patch around the keypoint are used. However, unlike JLA, the derivative is taken up to third order and the dimension of the descriptor is eight.

Moment invariants (MOM) [45] computes image moments which are the weighted average of the intensity values of the pixels inside an image patch.

Principal Component Analysis - Scalable Invariant Feature Transform (PCA-SIFT) [12] defines a patch around the keypoint and inside the patch, gradient vector in both x and y direction are accumulated in a vector. PCA is applied to the vector in order to reduce the dimension. After PCA, the dimension of the descriptors is reduced to 36.

Scalable invariant feature transform (SIFT) [22] uses a patch around the keypoint. And the descriptor computation is started by normalizing it. Onto the normalized patch, a 4 x 4 grid is fitted in order to obtain 16 bins. For each of them, an eight bin gradient histogram is computed and they are accumulated in a vector. The dimension of the vector is 128. As the last step, the descriptor is normalized by applying L2 normalization.

Spin images (SPIN) [16] is an intensity descriptor. At first, it takes an affine region and normalizes it. Rings that are centered on the normalized regions are defined and for each of them, a ten bin histogram that contains both the intensity values and location of the pixels is generated. Histograms of the bins are used to generate the descriptor.

Maximum response of filter (MR8) [46] is a texture descriptor and it uses filters. The descriptor is computed based on eight filter response which are responses of the orientation edge, the bar, the Gaussian, and the Laplacian of Gaussian filters.

Patch descriptor [47] is computed by using pixels that are located in a compact patch around the keypoints. Their intensity values are vectorized and the dimension of the descriptor is the multiplication of the dimensions of the patch.

Texture histogram rotation variant (D_xD_y) [34] is computed by generating high dimensional histograms. They are created by taking the derivative of the gray scale image at multiple scales.

Texture histogram rotation invariant (Mag-Lap) [34] has a similar descriptor computation approach with $\mathrm{D_xD_y}$. Unlike it, Mag-Lap generates rotation invariant descriptors. To do that instead of using only the first derivative, rotation invariant features which are the magnitude of the first derivative and Laplacian operator are used.

Some of the contour based keypoint description approaches are described below.

Shape context (SC) [2] is one of the most popular shape descriptors. This approach is a contour based descriptor because it uses the contour points of shapes. In the literature, there are several extensions of shape context approach such as inner distance shape context, multidimensional scaling, and shape context etc. In the original implementation, shapes are extracted and the contour points are detected by using Canny detector [6]. On the contour of a shape, a point is selected as a reference point, and the relative positions of the other points to the reference point are described. To do that, a coarse histogram in log-polar space is defined. During the histogram generation, the shape is divided into five bins in the radial direction, and twelve bins in the angular direction so the dimension of the histogram is equal to 60. For example, in [28] shape context is implemented slightly different from the original implementation. In this implementation, orientation is used besides the location information of the contour points. To integrate the orientation into the descriptor, points are weighted by their gradient magnitude. Furthermore, as distinct from the original implementation the dimension of the descriptor is 36. To obtain 36 dimensions, the shape is divided into three in the radial direction and the outer two bins are divided into four in the angular direction. This creates nine location bins, and four dimensions are caused by the orientations so the dimension of the resulting descriptor is reached to 36.

Geometric blur (GB) [3] is a similar approach to the SC. When a descriptor is computed, a reference point which is features such as edges is selected on the sparse signal. The oriented boundary points of edges are used as a sparse signal and they are blurred with different Gaussian kernels. The width of Gaussian kernel is related with the distance between the reference point and the points of the sparse signal. By counting the edge orientations, geometric blur descriptor is generated after each blur. After that, the geometric blur descriptors are concatenated in order to form the final descriptor as the last step.

When *curvature scale space (CSS)* [30] descriptor is computed, CSS image is generated for each of shape. CSS image of a shape is computed by convolving it with different the Gaussian kernels, and its zero crossings are marked. Increasing the kernel size causes smoother CSS image and at a point, the shape becomes convex and there will be no zero crossings. At that point, CSS image is generated by mapping the zero crossings. After CSS images are generated for each of shape, the maxima of contour of CSS images is used to compute the CSS descriptor.

Contour points distribution histogram (CPDH) [37] is a contour based descriptor. It is computed by using the contour points of the shape. The contour of the shape is extracted with Canny detector [6]. After contour points are detected, a circular grid is fitted onto the centroid of the shape. This circular grid has twelve bins in angular direction and four bins in radial direction so there are 36 bins in total. The descriptor is computed by counting how many contour points exist for each bin.

Curve edit [35] is another contour based descriptor computation approach. During the descriptor computation, curves are described by using their two intrinsic properties which are length and curvature. In the original approach, at first, curves are aligned. To align two curves, a segment of the curves which are high curvature points along the curve are matched and segment is extended by starting from this point. After curve alignment, descriptors are computed for both of them. Then the distance between the descriptors is calculated to classify the curves.

During *distance set* [9] descriptor computation, the relative spatial distance between points on the contour of the shape is used. During the computation, the distance between for each contour point and its certain number of near neighbor contour points is put into a set and it is called as distance set. After they are generated for each contour point, they are filtered to select significant ones among them in order to define the rich local image descriptors. The filtering process is done by thresholding the intensity values of the contour points in gray scale. After rich local descriptors are selected, they are used for several purposes such as classification, comparison etc. by measuring the similarity.

Generative models [41] is an integrated approach for recognizing shapes. In this approach, both generative models and informative features are used in recognition. Generative models are used to measure the similarity between two shapes in terms of a class of transformations. So a generative model is constructed in order to define which transformations are needed to generate one shape from the other shape. The informative features are used to construct probabilistic approximation for transformations. So they should be invariant to transforms and also representative.

Inner distance shape context (IDSC) [21] is an extension of the original SC description approach. The only difference is using inner distance instead of Euclidean distance during the histogram generation. Inner distance is a metric and it is calculated by measuring the shortest distance without going out the shape between two points. In the paper, to calculate inner distance efficiently, an approach is proposed. According to this, the contour point of a shape is mapped into a graph by assigning points as nodes and Euclidean distance between points as edges. After graph is constructed, any shortest path algorithm can be applied in order to calculate the inner distance between any pair of points. After descriptors are computed for each shape, they are matched. During this step, dynamic programming is used in order to match shapes in a more efficient and accurate way. So that, the approach is named as IDSC + DP.

During *morphological curvature scale space (MCSS)* [10] descriptor computation, five features of the shape are used. One of them contains curvature related local information, two of them contain curvature related global information, and the others contain shape related global information. To compute them, the contour of the shape is convolved with Gaussian kernel at different scales. Curvature function, curvature scalar descriptor, bending energy, eccentricity, and elongation are used in descriptor vector. Curvature function of the shape is computed at each scale, and the feature that has the maximum curvature both from top to bottom and bottom to top becomes the first feature. Curvature scalar descriptor is computed by counting the scale space entries that have the average curvature is above a threshold at both top and bottom scales. Bending energy is computed by taking sum of squared curvatures of the contour points. Eccentricity is related with how many conic sections there are on the shape and its simplest calculation is dividing the major and minor axises of the shape. Elongation as the last feature is computed by dividing the width and height of the bounding box of the shape.

Multidimensional scaling & shape context (MDS+SC) [21] is a hybrid approach that merges SC and MDS. In this approach, the signature of shapes is computed by using multidimensional scaling. During the signature generation, Euclidean distance is replaced with the inner distance which is the length of the shortest way within the shape between two points. After generating the signatures, SC approach is followed and the descriptor is computed by using the points of the signature instead of the points of the boundary. Since dynamic programming is used in the matching step, the name of the approach becomes MDS + SC + DP.

Visual parts [15] computes descriptors by using only the visual part of the shapes. To compute the descriptor for a shape, at first, the distortion is eliminated by simplifying its curve. During the simplification, discrete curves are generated by converting the shape to a set of polygonal curves. To avoid the information loss, as possible as large number of vertices are preserved. After converting the contour of the shape to a set of discrete curves, a tangent function is assigned to each of them. The distance of two shapes is measured by comparing the tangent functions of their polygonal curves. Because tangent functions are step functions, the distance between any pair of them can be calculated easily.

Principal component analysis gray (PCA Gray) [17] and *principal component analysis masks (PCA Masks)* [17] approaches describe shapes globally and they are slightly different from each other. Leonardis et al. [18] propose a framework to represent images with multiple eigenspaces. According to it, training images are represented as eigenspaces by applying principal component analysis and for each category, a separate eigenspace is constructed. The differences between PCA Masks and PCA Gray are principal component analysis is applied to which and how many eigenvectors are used for recognition. For PCA Masks, the segmentation masks are used directly and the first 30 eigenvectors are employed. For PCA Gray, principal component analysis is applied to the segmented gray value images and the first 40 eigenvectors are employed.

In the literature, some of the keypoint description algorithms cannot be classified as texture based or contour based. And a few of them are described below.

Skeletal context (SCC) [48] is computed by using skeleton of a shape which is the medial axis of its boundary. The characteristic points such as endings, junctions etc. on the skeleton are used for the descriptor computation. The descriptors are computed by following the same process of the SC which is a circular grid is fitted to the shape. The grid has five bins in radial direction and twelve bins in angular direction. So the dimension of the resulting descriptor is 60. As distinct from the SC, only the characteristic points of the skeleton of the boundary are used instead of the whole contour points of the shape.

Shock edit [36] computes shock graphs that describe shapes by putting them into a graph. To do that, at first shapes are divided into equivalence classes named as shocks by considering its skeleton which is the medial axis of its boundary. Then, for each shape, shock graphs are generated by putting equivalence classes in an order. After this point, shapes are represented by graphs. So shape matching can be done by applying any graph matching approach.

Color histogram [40] descriptor is computed by contracting histograms of the intensity values of the pixels that are located inside the shape in RGB channels. Their dimension is 16 for each color channel, so the dimension of the final descriptor is 48.

CHAPTER 3

ANALYSIS OF MSER STABILITY

3.1. Introduction

Maximally stable extremal regions (MSER) is one of most popular keypoint detection algorithm. This algorithm detects extremal regions that are a kind of image feature. These features are defined with a group of consecutive pixels as known as regions. During the detection, the algorithm uses a global approach. According to the intensities, pixels of the whole image are added from white to black and vice versa step by step and connected components are defined for each adding step. During these steps, maximally stable extremal regions are selected from connected components with respect to the area stability. So MSER detector defines features as regions, not individual points. In the literature, keypoint detection algorithms are analyzed by measuring repeatability on a few images of objects taken from different viewpoints. The repeatability is calculated based on point correspondences. Since MSER detector defines regions, a different approach is required to analyze its stability. [29] proposed a way to measure MSER repeatability. The repeatability is calculated based on the similarity between the best-fitted ellipses of MSER by computing their overlap error.

Our contributions to the literature are generating synthetic images to be able to increase the number of viewpoints. Furthermore, finding convex hulls for extremal regions which is a qualitatively better approximation to the regions than the ellipses used by [29] which is indicated in Figure 3.1.

3.2. Approach

Generation of synthetic images is the major supporting factor of analysing the stability in detail because synthetic images are generated at high frequency during the analysis process. At each generation, only one camera parameter is changed, and the amount of change is selected to be both high and low. During the generation of synthetic images, a reference image that is perpendicular to the normal vector of the object plane is necessary and sufficient. In practice, affine deformation is enough to generate images that are close to real scenes [19, 32]. This affine deformation matrix can be written as:

$$A = \lambda \begin{bmatrix} cos\psi & sin\psi \\ -sin\psi & cos\psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos\phi & sin\phi \\ -sin\phi & cos\phi \end{bmatrix} \tag{3.1}$$

Parameters of the equation are the scale parameter (λ), in-plane rotation (ψ), tilt angle (ϕ), and tilt (t). Inplane rotation (ψ) is the rotation around an axis that is perpendicular to the object plane. Tilt (t) is deformation degree and is calculated by

$$t = 1/cos(\theta) \tag{3.2}$$

where θ refers to tilt amount. A deformed image is generated by multiplying an image with an affine deformation matrix. However, during this multiplication process, dimensions of the deformed image should be calculated correctly, otherwise, some points of the reference image could be out of the deformed image. To handle that, the affine deformation matrix which is shown in Equation 3.1 is applied to the center of the reference image and that point is located to the center of the deformed image. In addition to this, each corner of the reference image is projected into deformed image and dimensions are selected in a way that those four corresponding points will be inside the deformed image.

In order to measure MSER repeatability, extremal regions are simplified by finding convex hulls, because this way is faster than to calculate the overlap between the extremal regions directly and more realistic and sensitive than overlapping ellipses. When we compare convex hull approximation with ellipse approximation, both has some issues about representing concave parts of regions, still convex hull approximation is better to represent the regions. In Figure 3.1 there are three examples of extremal regions and their convex hull and ellipse approximations. After extracting MSER on the reference image and the deformed images, there are five steps which are; convex hull conversion and projection, overlap ratio calculation, decision about which region repeats, and repeatability calculation.

Each extremal region is converted to convex hulls by using Andrew's monotone chain convex hull algorithm [1], then produced convex hulls are projected into deformed image. This projection is done by multiplying each polygonal coordinates with the corresponding affine deformation matrix from Equation 3.1. Figure 3.2 shows a projection of the convex hull of a region from the reference image to the deformed image. After

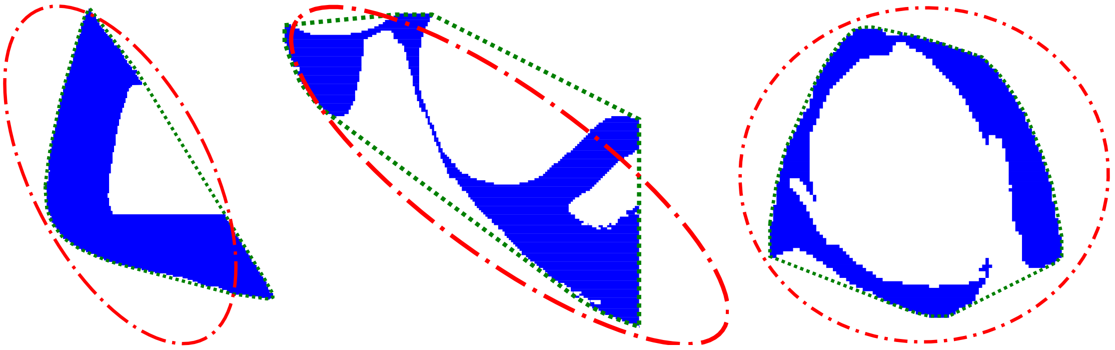


Figure 3.1. Representation of extremal regions. In the figure, blue areas show region itself, dotted-dashed lines represent the best-fitted ellipse to the regions, the convex hulls of the regions is represented by dotted lines. Both approximations have some issues about being realistic. For example, gaps, hulls etc. cannot be modeled. However, the convex hull approximation is more realistic and sensitive than the ellipse approximation.

each convex hull of the reference image is projected, overlap ratio between each of them and convex hulls of the deformed image is calculated. To calculate overlap ratio between two convex hulls, intersection area and union area are required. Polygonal coordinates of intersection are computed by using Sutherland - Hodgman algorithm [39]. By using those polygonal coordinates, the intersection area is calculated by using the method that is suggested in [4].

$$\frac{1}{2}|\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1} y_i)| \tag{3.3}$$

In Equation 3.3, N represents the number of polygon coordinates, and x_i and y_i refers to the i_{th} point coordinate. To calculate the union area, Equation 3.4 is used after area of two regions and intersection are calculated by Equation 3.3.

$$|A \cup B| = |A| + |B| - |A \cap B| \tag{3.4}$$

After the intersection and union area of the regions are computed, overlap ratio for them is calculated by the Jaccard similarity coefficient:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{3.5}$$

In the literature, threshold of overlap ratio is taken as 0.6 [29]. This means that if the overlap ratio between two regions is bigger than 0.6, those are accepted as similar regions. Figure 3.3 shows the overlap ratio calculation of region C on the reference image. It is projected into deformed image, and the overlap ratios between region C'

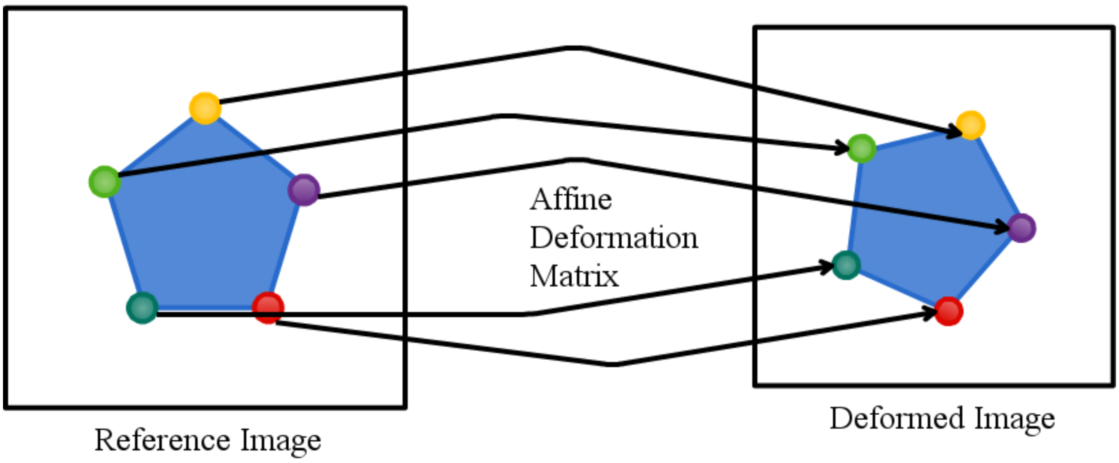


Figure 3.2. Projection of the convex hull from the reference image to the deformed image. First polygonal coordinates are projected by multiplying with the affine deformation matrix that is shown in Equation 3.1 and the transformed points are form the corresponding convex hull.

and the regions $1, 2, 3, 4, 5$ on the deformed image are calculated. For this example, C' is obviously similar to C. After that, this process is repeated for all of the other regions as well. If a region on the reference image is similar with at least one extremal region on the deformed image, the region is accepted as repeated. After deciding which regions are repeated, repeatability is calculated as

$$repeatability = 100 * \frac{\#repeatedRegion}{\#extremalRegion} \tag{3.6}$$

where $\#repeatedRegion$ stands for how many regions of the reference image are repeated on the deformed image. $\#extremalRegion$ refers the number of extremal regions on the reference image.

3.3. Implementation Detail

As mentioned in Chapter 2, MSER can be extracted on both grayscale and color images. Furthermore, in the literature, there are some implementations to extract MSER. One of them is Matas' implementation and the other is OpenCV implementation. In this study, extracting MSER on grayscale images is decided and Matas' implementation is selected to extract MSER. This implementation is used with default parameters which are

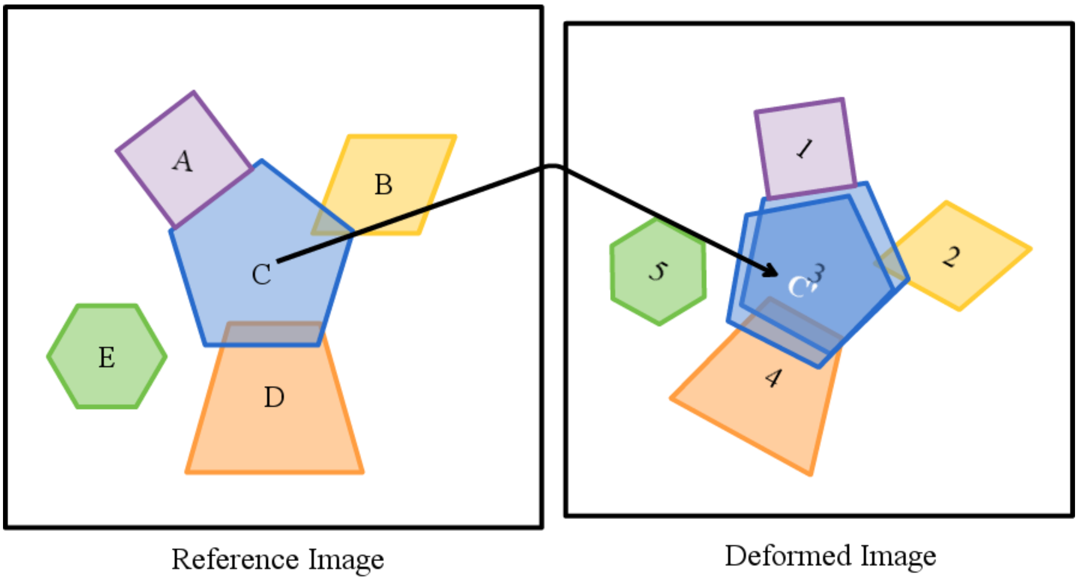


Figure 3.3. Each region on the reference image is converted to its convex hull. Those convex hulls are projected into the deformed image one by one. After projection, overlap ratio is calculated between each projected convex hull and each convex hull on the deformed image. In this figure, that process is illustrated on a sample region C. After processing regions A and B, region C is converted to convex hull. Then it is projected into the deformed image, and region C' is obtained. Overlap ratios between region C' and regions $1, 2, 3, 4, 5$ are calculated to decide whether region C repeates. After regions D and E are processed, MSER repeatability between the reference and the deformed images is calculated.

explained in Chapter 2 in detail. To perform the decisions, all images are converted to grayscale, and they are given to the Matas' executable as input.

After converting MSER to convex hulls, MSER are projected into the deformed image via the convex hull approximation. During the projection, polygonal coordinates of convex hull are multiplied with the affine deformation matrix. If any of the projected polygonal coordinates of a convex hull is located out of the deformed image, the region of the corresponding convex hull is removed and it is not regarded in the following steps since, there is no possibility that it can repeat. That means during the calculation of the repeatability, only the regions that are located in the common part of the reference image and the deformed image are considered in the following steps. To exemplify, the common part of two images is shown in Figure 3.4. The regions that are located out of the common part are removed from the calculation of the repeatability.

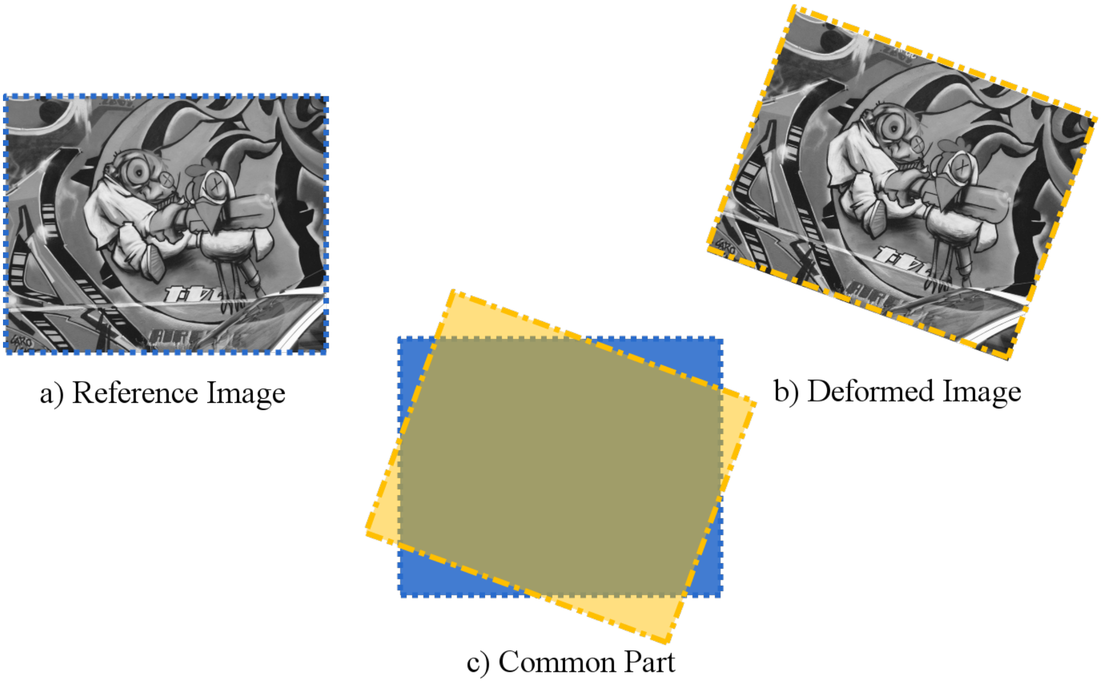


Figure 3.4. Removing the projected regions that are out of the deformed image. The dotted line shows the boundary of the reference image in a and c. The dotted-dashed line shows the boundary of the deformed image is obtained by rotating the reference image 20° in b and c. The intersection of the reference and the deformed image shows the common part of them. Only the points that are located inside the common part are considered and others are removed.

The overlap ratios are calculated between the remaining projected convex hulls and the convex hulls on the deformed image to find the similar regions. After finding the similar region pairs, repeated regions are determined. During this decision, one to one correspondence, among the similar region pairs has to be provided. In this regard, there are two cases that include ambiguity which should be handled. First, a region from the reference image cannot be accepted as repeated with more than one region from the deformed image. Second, more than one region from the reference image cannot be accepted as repeated with the same region on the deformed image. When any of the cases occurs, the similar region pair that has the highest overlap ratio is accepted as the valid pair. For example, if the overlap ratio between a region (R_A) on the reference image and a region (R_1) on the deformed image is 0.9 (R_A-R_1), and the overlap ratio between R_A and another region (R_2) on the deformed image is 0.7 (R_A-R_2), so R_A repeats both R_1 and

R_2. In that case, R_A is accepted as repeated with R_1 because the overlap ratio between R_A and R_1 is greater than the overlap ratio between R_A and R_2. To show the second case, suppose that a region (R_C) and another region (R_D) are two individual regions on the reference image and they are similar to a region (R_3) on the deformed image with 0.7 (R_C-R_3) and 0.8 (R_D-R_3) overlap ratios respectively. Since the overlap ratio between R_D and R_3 is greater than R_C and R_3, R_D and R_3 is valid pair and while R_D is accepted as repeated, R_C is not.

After providing one to one correspondences among the similar region pairs, the repeatability is calculated by using Equation 3.6. Moreover, as mentioned before, during the convex hull projection process, some regions are removed, and they are not regarded in the following steps. To do that, on Equation 3.7 $\#extremalRegion$ refers to the number of extremal regions that are located inside the common part of the reference image and the deformed image. It is calculated with

$$\#extremalRegion = \#allExtremalRegion - \#removedExtremalRegion \quad (3.7)$$

where $\#allExtremalRegion$ stands for the number of extracted extremal regions on the reference image and $\#removedExtremalRegion$ refers to the number of removed extremal regions during projection step.

3.4. Experiments

The effect of change of camera location on the repeatability of MSER is measured by evaluating experiments.

3.4.1. Setup

An image from Oxford dataset is used as the reference image to generate deformed images because in the literature Oxford dataset is mostly preferred dataset to analyse the stability of keypoints besides this, to analyse MSER they are used in [29]. In Figure 3.5, the reference image, and three deformed images are shown as an example. After the generation of deformed images, the repeatability between the reference image and each of the deformed images is measured. Only three parameters of the affine deformation matrix

Equation 3.1 is changed along the generation of deformed images which are in-plane rotation (ψ), scale (λ), and tilt amount (θ) parameters as proposed in [44]. To observe the effect of them on the repeatability separately, deformed images are generated by changing those three parameters one by one. That means experiments have three categories and they are evaluated under both high and small amount of changes. High amount of change is a way to retrieve valuable results about the object detection applications that uses keypoints. Low amount of changes is also simulated since it is a way to measure the performance of keypoint detection algorithms when they are used in the object tracking applications.

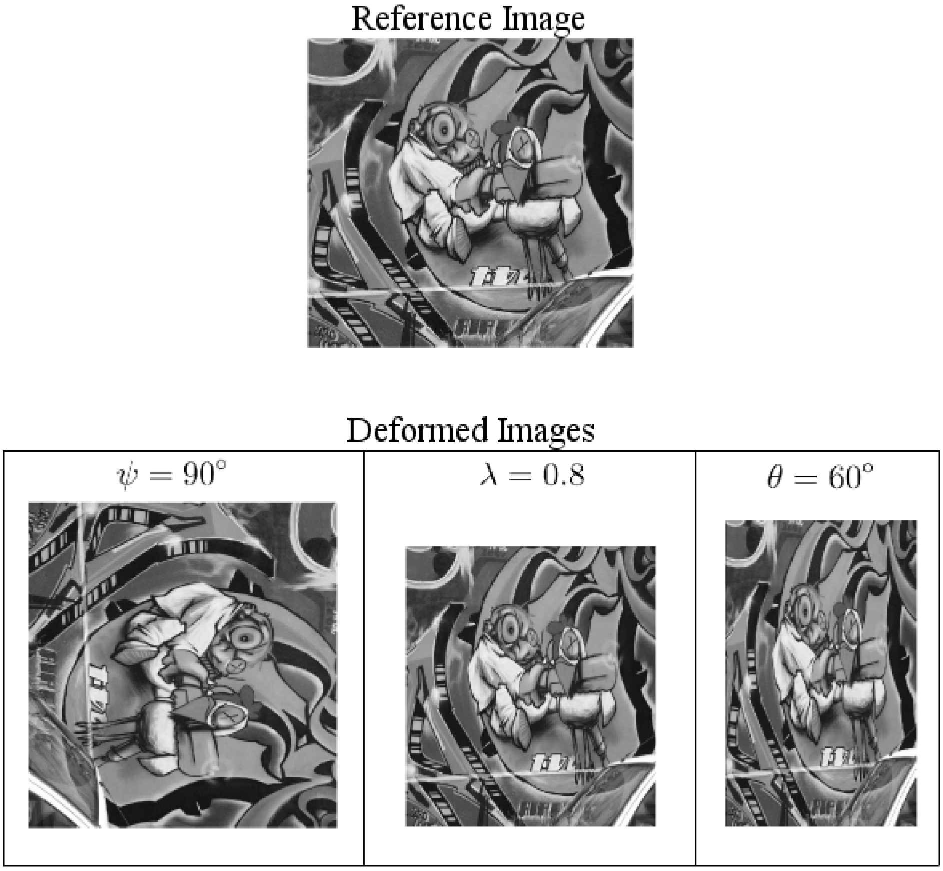


Figure 3.5. An example of deformed images by sampling three camera location parameters. From left to right, in the first image 90°in-plane rotation (ψ) is sampled. The scale (λ) is set to 0.8 in the second image. The last image is obtained by sampling tilt amount (θ) for 60°.

3.4.2. Results

Illustrated results are obtained by evaluating experiments on the only first image of the Graffiti image set from Oxford data set because results of experiments with different images are similar. The repeatability of the first Graffiti image is shown in six figures. The parameters that are listed below are changed both large and small amount.

- In-plane rotation (ψ)
- Scale (λ)
- Tilt amount (θ)

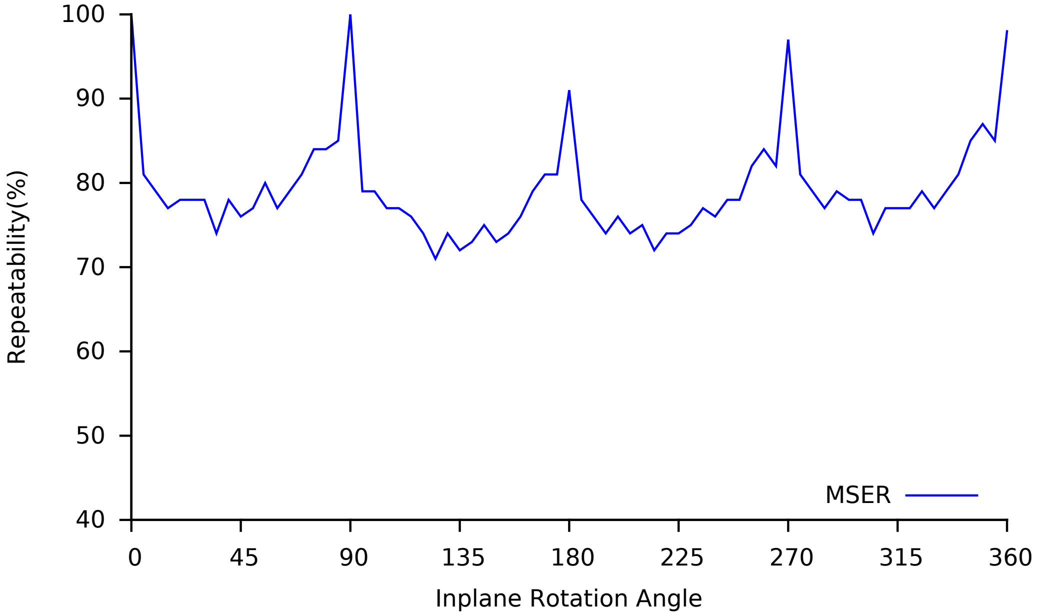


Figure 3.6. The repeatability of MSER over the large amount of change of the in-plane rotation (ψ) changes. The curve peaks at 0°, 90°, 180°, and 270°.

In the first experiment, deformed images are generated by changing the in-plane rotation (ψ) from 0° to 360° with the 5° intervals. Figure 3.6 shows its result. The repeatability peaks at 0°, 90°, 180°, and 270°and between them it is U shape curve. The lowest repeatability is 71% at 125°, and it reaches 100% at 0° and 90°.

Figure 3.7 shows the results of the second experiment. On this experiment deformed images are generated by changing the scale parameter (λ) from 0.5 to 1.5 with the

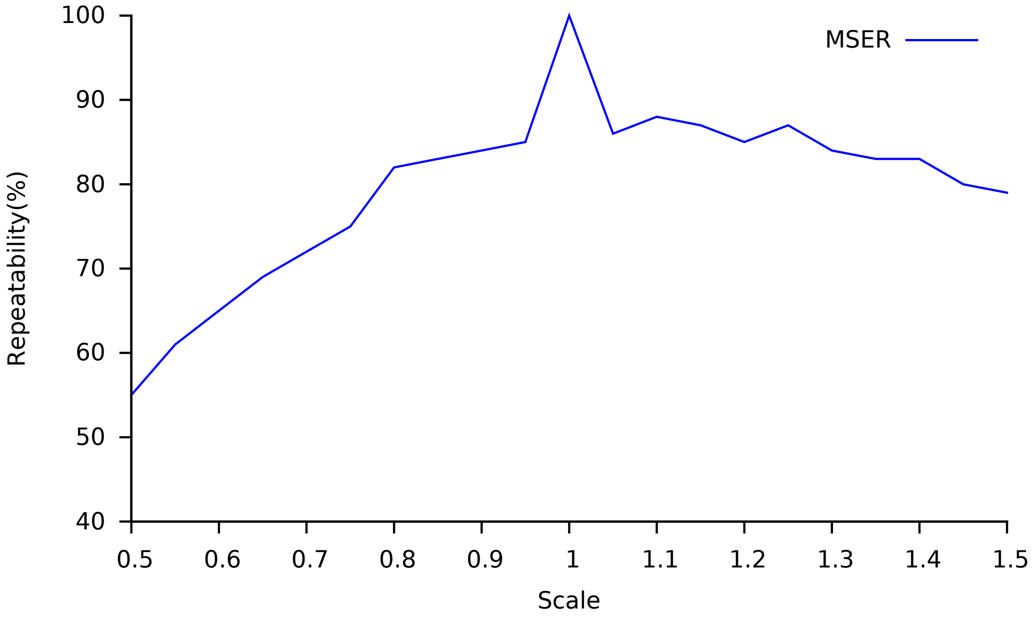


Figure 3.7. The repeatability of MSER over the large amount of change of the scale parameter (λ). It decreases when both the scale parameter is increasing and decreasing.

interval of 0.05 linearly. During the scale decreasing and increasing from 1.0, the repeatability decreases. However, the amount of decrease is different. When scale decreases, the repeatability drops faster than when scale increases. This causes asymmetry on the curve. The repeatability decreases till 55% when the scale parameter is 0.5.

Figure 3.8 shows the repeatability by changing the tilt amount (θ) in horizontal direction from 0° to 80° with 5° intervals. The repeatability decreases at 5° relatively drastic, then it remains stable. The lowest repeatability 76% is obtained at the maximum change of the tilt amount.

Figure 3.9 is obtained by repeating the in-plane rotation (ψ) experiment. However, this time, deformed images are generated with changing in-plane rotation from 0° to 1° with the 0.1° intervals. The repeatability drops from 100% to 87% at 0.1°then it continues stable at about 90% repeatability. Figure 3.10 shows the effect of the small amount of change of the scale parameter (λ). It is changed from 0.95 to 1.05 by the interval 0.01. The repeatability decreases drastically both from 1.0 to 1.01 and 1.0 to 0.99 then it remains steady at about 85%. Symmetry is observed in contrary to the result of the large amount of change of the scale parameter which is shown in Figure 3.7. In Figure 3.11,

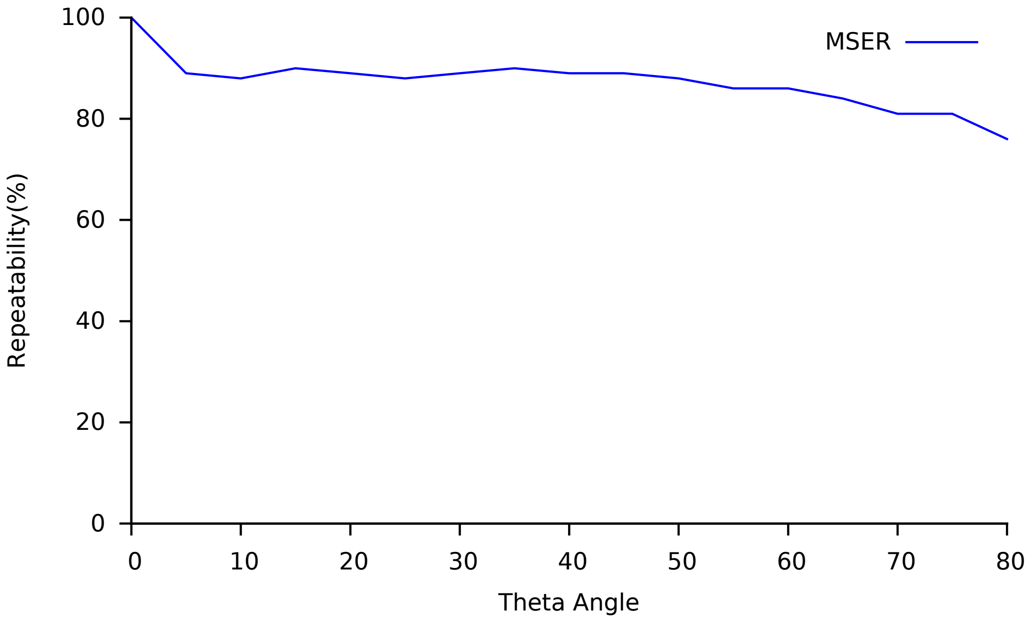


Figure 3.8. The repeatability of MSER over the large amount of change of the tilt amount (θ). It decreases slowly.

the effect of the small amount of change of the tilt amount (θ) in horizontal direction is shown. It is changed from 0° to 1° with the 0.1°intervals. The repeatability is 100% till 0.3°, then it decreases to 94% till 1°slowly.

3.5. Discussion

As shown in Figures 3.6, 3.7, 3.8, 3.9, 3.10, and 3.11, MSER is invariant to each of three camera location parameters, and especially invariance to the tilt amount is remarkable. So, MSER is well enough to be used in the various computer vision applications. In particular, it can be useful for object tracking because its repeatability under the small amount of deformations is almost 100%. In the experiments, there are two interesting observations. First, the repeatability reaches about 100% at 90° ,180° , and 270°in Figure 3.6. Second, there is an asymmetry in Figure 3.7.

In the change of the in-plane rotation (ψ), the repeatability peaks at 90° ,180° , and 270°. This is shown in Figure 3.6. The reason of that is interpolation requirement, during the generation of the deformed images. This interpolation effect causes the instability

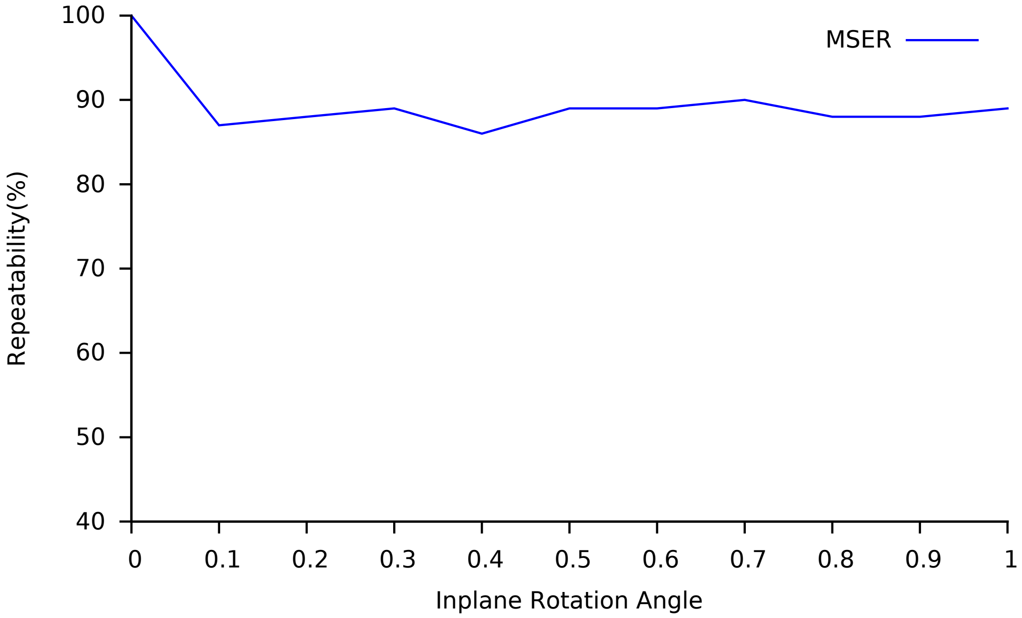


Figure 3.9. The repeatability of MSER over the small amount of change of the in-plane rotation (ψ). It drops relatively drastic at $0.1°$, then remains stable.

in the area of the candidate extremal regions which are called as connected components. The instability causes selecting different connected components as extremal regions. On the other hand, there is no interpolation effect at $90°$,$180°$, and $270°$. So neither the area of the connected components nor selecting stable ones among them change, and the repeatability reaches almost 100%.

In Figure 3.7, the performance of MSER is shown under the change of the scale parameter (λ). When the scale is getting away from 1.0, the repeatability drops, but there is a difference between the amount of decrease when the scale parameter (λ) decreases and increases, so this causes asymmetry on the graph. This is related to two parameters of MSER algorithm. As mentioned in Chapter 2, there are some parameters like maximum area, minimum area, maximum variation etc. and parameters named as maximum area and minimum area are the reason of this asymmetry. When the scale parameter (λ) is lower than 1.0, the deformed images are generated with less sample. This causes decrease in the size of images. As a result, when extremal regions are extracted, connected components are getting smaller as well. However, if the area of a connected component becomes lower than the minimum area which is determined by parameters, there will be two possible scenarios. First, this connected component cannot become an extremal

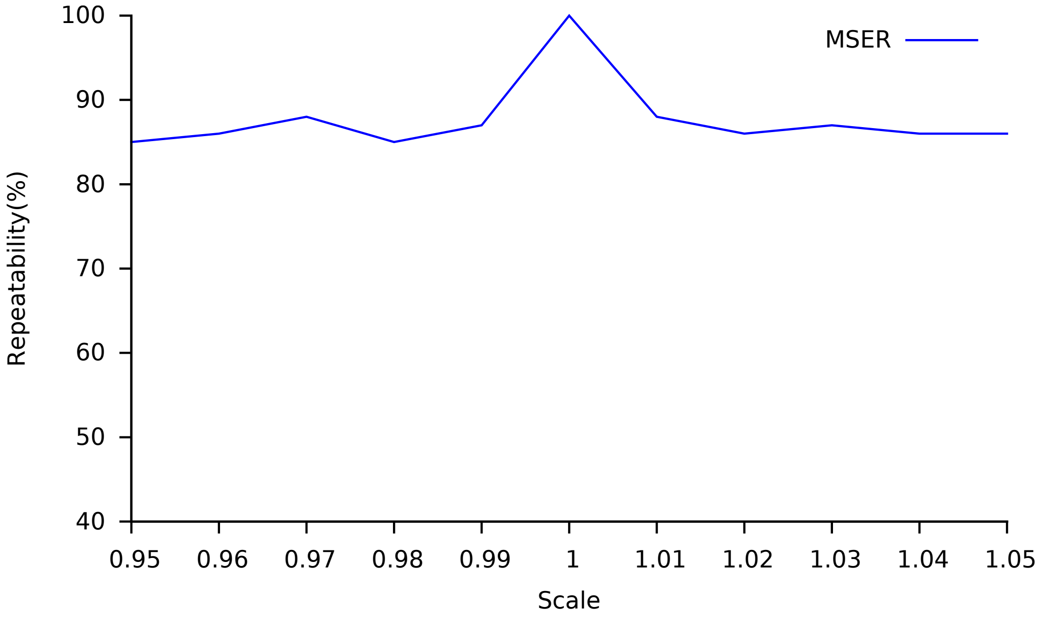


Figure 3.10. The repeatability of MSER over the small amount of change of the scale parameter (λ). It decreases drastically then remains steady when the scale decreases and increases from 1.0.

region, second if there is any possibility, it can merge with another neighbor connected component. If either of them occurs, the region of the reference image cannot match with the corresponding region. When the scale parameter (λ) is higher than 1.0, the same issue happens with the maximum area parameter. As mentioned in 3.3, MSER are extracted by using Matas' implementation with the default values of the parameters which are 60 in pixel for minimum area parameter and 14400 in pixel for maximum area parameter. With respect to the effect of the maximum area parameter, the minimum area parameter has more effect on the area of the candidate extremal regions of the images that are used in the experiments . So, the repeatability drops faster when extremal regions are getting smaller than when they are getting bigger, that appears as asymmetry in Figure 3.7.

3.6. Conclusion

In many computer vision applications such as object detection, object tracking etc., keypoints are common way to represent images as distinctive features. Their suc-

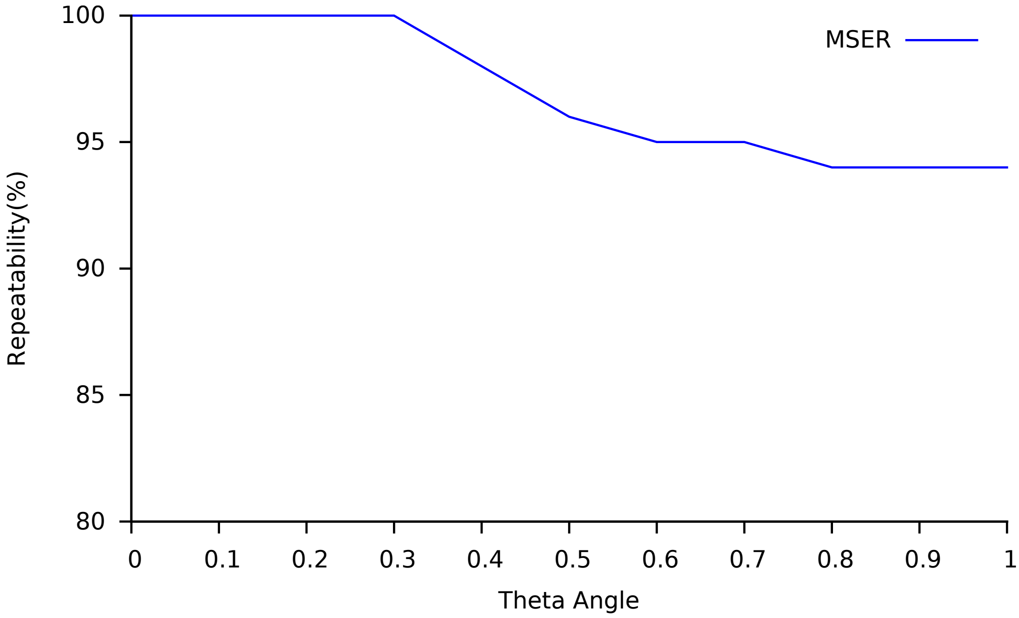


Figure 3.11. The repeatability of MSER over the small amount of change of tilt amount(θ). It is 100% till 0.3°, then drops slowly.

cess is an important point to determine whether the applications work well or not. In the literature, a way to determine the success of keypoint detection algorithms is accepted as measuring the repeatability on images of the objects taken from different viewpoints. Furthermore, there are several proposed approaches to measure the repeatability because keypoint detection algorithms have different and specific features. Among them, Maximally Stable Extremal Regions are interesting because it retrieves regions as keypoint. To analyse its stability, we propose a new approach. To measure the repeatability, extremal regions were modeled as convex hulls. Their repeatability was accepted as the repeatability of MSER and it is measured by calculating the overlap ratio. As experiments, we measured the repeatability between the reference image and the deformed images that were generated synthetically. During the generation of the deformed images, three camera location parameters were changed in high frequency. They were the in-plane rotation (ψ), the scale parameter (λ), and the tilt amount (θ). In the experiments, their effect to the repeatability of MSER is observed in detail. Since deformed images were generated by changing three parameters one by one, also the amount of change of them was selected to be both high and low.

The results of experiments show that the performance of MSER is in the accept-able range to be used in the computer vision applications. In addition to this, experiments prove that MSER is invariant to the in-plane rotation (ψ), the scale parameter (λ), and the tilt amount (θ). Especially, robustness against the change of tilt amount (θ) is significant. In brief, the results of experiments show that the repeatability of MSER is well enough to use them in order to detect and track objects.

CHAPTER 4

DETECTION ALONG EXTREMAL REGION BOUNDARY

4.1. Introduction

After analyzing the stability of MSER in Chapter 3, its vulnerability was determined. When deformation like rotation, zoom, occlusion, illumination etc. occurs, there are three possibilities. First, the MSER area might be found as the same as before deformation. Second, it might be corrupted drastically because two separate regions might be merged or one region is separated into subregions. The last possibility is the corruption on the MSER area because some part of the region drops out of the image. The second and third are the causes of the observed vulnerability. And when they happened while the MSER area changes drastically, the boundary remains close to the boundary before deformation or at least some part of the boundary remains stable. In Figure 4.1 some regions are indicated in order to exemplify the regions that has instable region area but stable region boundary under partial occlusion and various deformations.

By taking advantage of the stability of the MSER boundaries, we propose a novel local analysis on the boundary of each region in addition to the global MSER approach. The proposed algorithm locates discriminative points on the boundaries by using its curvature. It also provides to prevent the information loss. In the literature, MSER are often used with the best-fitted ellipse representation . This causes ignorance of the content of the boundary like intrusions, extrusions, gaps etc. In addition to this, the algorithm provides another acquisition. It is, in contrary to the MSER, the proposed algorithm retrieves individual points instead of the regions. This provides benefits in some fields of computer vision. For example, especially in 3D operations, individual points are required instead of regions.

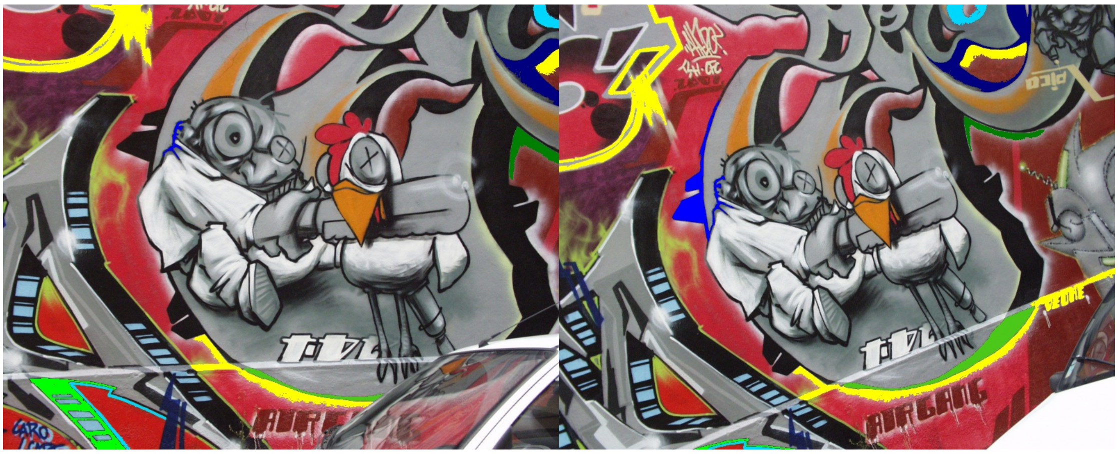


Figure 4.1. Some MSER that exemplify the vulnerabilities of MSER detector are shown. For example, the area of yellow region that is located on the top left corner of images is changed drastically under deformation. However, some parts of it remain stable so the boundaries of those parts remain stable as well. Moreover, there is a yellow and a green region that are laying next to each other on the bottom part of images. Although the area of them is changed a lot under partial occlusion, their left parts and also the boundary of those parts remain the same.

4.2. Approach

The proposed algorithm has three main stage which are the boundary traversal, tangent direction estimation, and curvature computation.

After MSER are extracted, for the boundary of each region, traversal is required because the tangent direction estimation algorithm needs the boundary points as a chain and the direction of traversal effects the behavior of the algorithm. During the traversal process, the boundary of holes are ignored and outer boundary is retrieved only. This retrieved boundary has all outer boundary points, that means any of the boundary points is not omitted. As a result of this process, the outer boundary points are obtained as a chain. Figure 4.2 shows the regions itself and their traversed boundary. In this figure the below region has a hole, and as seen it is ignored during the traversal. Furthermore, as direction we choose counter clockwise for each region. The direction issue is about the sign of curvature values and the important thing is not choosing clockwise or counter clockwise, it is applying the same direction for each region.

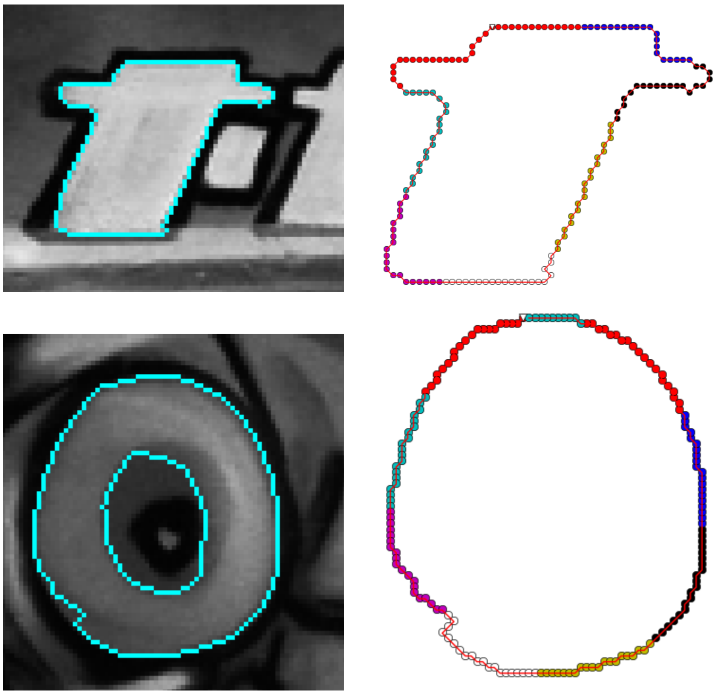


Figure 4.2. For each region traversed boundaries are retrieved. The first column shows the regions on the patch of the image, and its points are drawn with light blue. The second column shows their retrieved boundary that is obtained by traversal process. It starts with the white triangle and continues by following the outer boundary in counter clockwise direction till it reaches the starting point. The color of the traversed boundary changes for each 20 consecutive points.

After the outer boundary of regions is traversed as a chain, the tangent direction is estimated by applying the median filtered differencing algorithm [25]. It is a way to estimate the tangent direction of discrete curves and compute its curvature. In the algorithm, for each point of the curve, the tangent direction is computed. To do that vectors are defined that are between corresponding point and its neighbors. How many neighbors will be used is determined by the parameter m. When the tangent direction is computed for one point, $2m$ vectors are defined. Half of them are next vectors and the others are previous vectors. For next vectors, the current point is taken as the starting point, and next m neighbor is the ending points of the vectors. The previous vectors are defined between the previous neighbors as the starting points and the current point as the ending point. In figure 4.3, the difference vectors of a boundary point is shown. After $2m$ difference vector is calculated, they are represented in polar coordinates. The

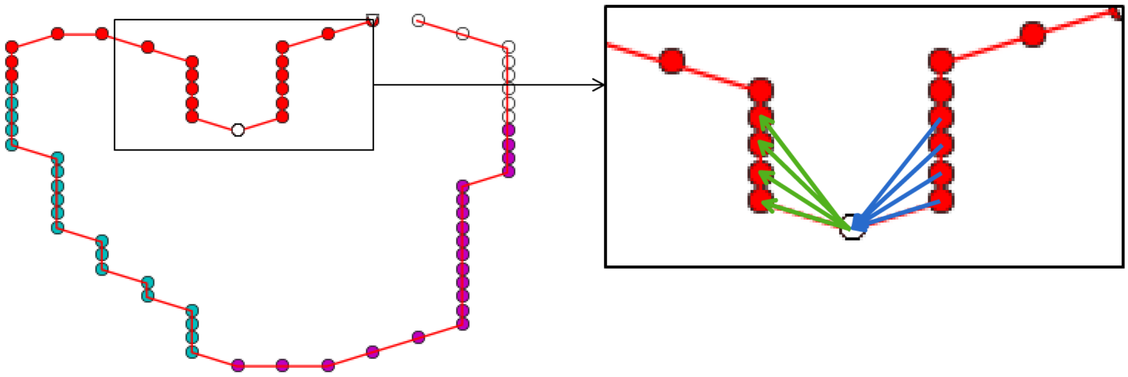


Figure 4.3. For each point on the boundary, the tangent direction is computed. Left sub figure shows the whole region. The representation of the region is the same as Figure 4.2. Right sub figure is zoomed in the part of the boundary, and it shows the $2m$ difference vectors of the white point when m is equal to 4. Green vectors represent its next vectors and blue vectors are its previous vectors.

median of $2m$ polar angle is accepted as its tangent direction. After the calculation of the tangent direction of each point on the outer boundary, they are stored and this collection is called as the tangent direction curve of the boundary. The direction of traversal effects the sign of the tangent directions. In this work, we choose counter clockwise as the traversal direction and it is guaranteed by traversal process. The parameter m is related with the smoothness of the tangent direction curve. To decide m, common geometric shapes like square, and parallelogram are generated and their tangent direction curves are computed with different m. After testing them, we observed that smoothness is enough and the tangent direction curve is enough definitive to represent the characteristic of the shape of the boundaries when the parameter m is taken as four.

After the tangent direction curve is calculated, to find characteristic and discriminative points on the boundary, its derivative is taken and this process is called as curvature computation. Before taking the derivative, it is smoothed to remove ripples. To do that the tangent direction curve is convolved with the Gaussian kernel when its sigma is equal to 1.5. To compute the curvature of the tangent direction curve the derivative is taken by using below Equation

$$difference = td_{i+1} - td_{i-1} \tag{4.1}$$

It is a way to differentiate the discrete curves and called as the center difference. On the curvature computation, local extremes are accepted as the interest points. During the finding local extremes 0.35 radian is taken as the threshold. While for local maxima, it is applied as 0.35, for local minima, it is taken as -0.35. After drawing the curvature and marking interest points, we observed that, if the boundary turns left, local minima is observed, and right turns are observed as local maxima.

When keypoints are located, some thresholds, parameters have to be decided. To do that, we tried some different values for them to choose the most suitable values with our expectations. In addition to this, we observed the behavior of the algorithm when synthetically generated a square, a parallelogram, and a trapezoid was given as input regions and we try to make results close to each other. Figure 4.4 shows their shapes, curvature and derivative graphs, and keypoints. On the first row, the synthetically generated region square is shown and changes on its tangent direction curve and pits of the derivative graph are exactly the same. On the second row, the algorithm is tested with a parallelogram as the input, and there are two types of changes at both graphs. At the corners that have the acute angles, the change of the tangent direction curve is higher than other two corners. Likewise, the rate of decrease at the pits of the acute angles is higher than the obtuse angles. At the third row, a trapezoid is given to the algorithm, the same difference between the corners with the acute angles and the obtuse angles is observed. Furthermore, all keypoints are located at the corners of each the synthetically generated regions. So, this shows, the proposed algorithm has robustness against various deformation because a square can become a parallelogram and trapezoid under various deformations.

In Figure 4.5, there is an example of a real region and its tangent direction and the derivative curves. And also, keypoints are indicated. The proposed algorithm detects all turns of this region and marks them as interest points. Turns of all this region are left except for one turn. This exceptional turn is observed as a peak and others are as pits on the derivative curve.

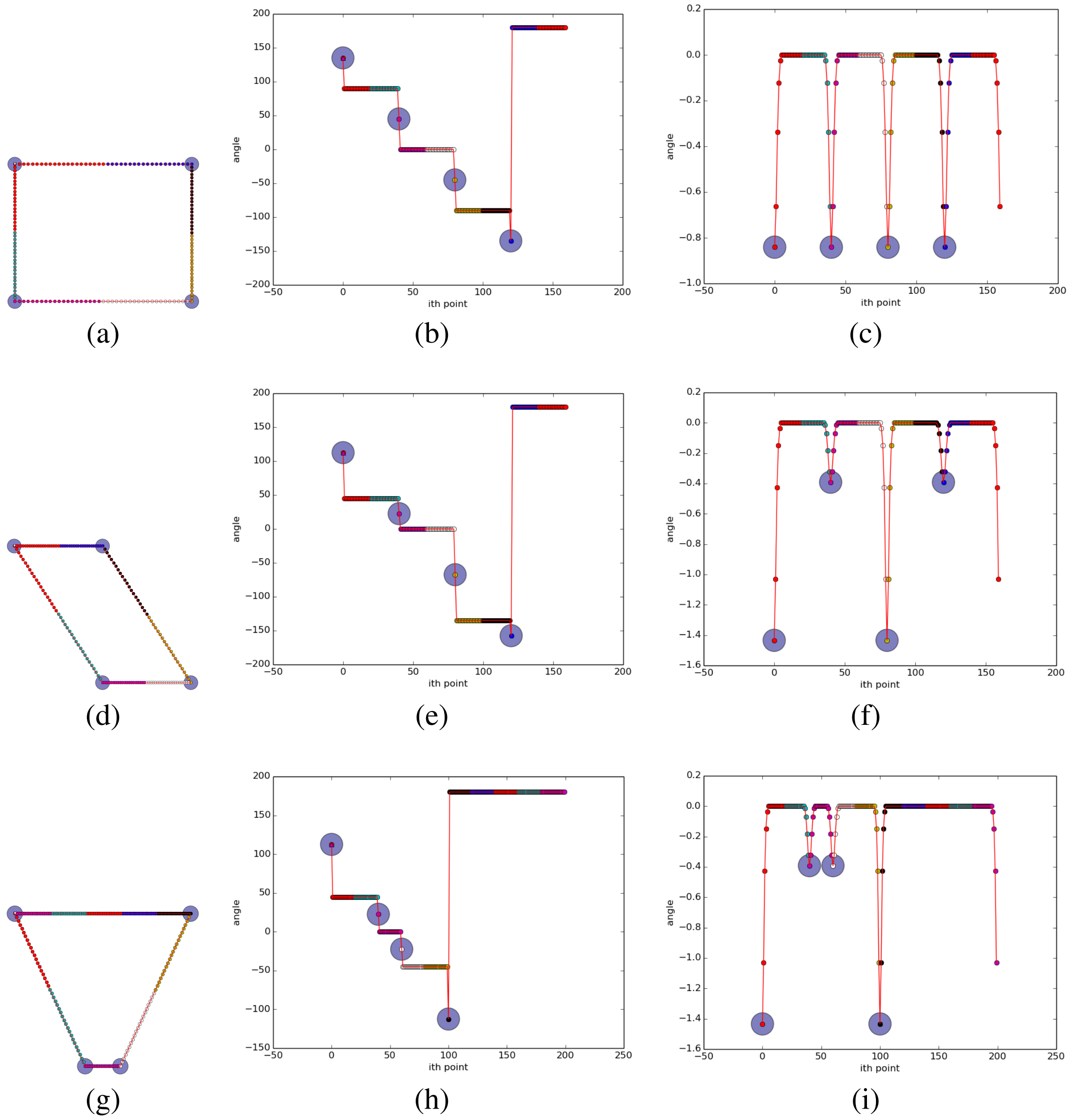
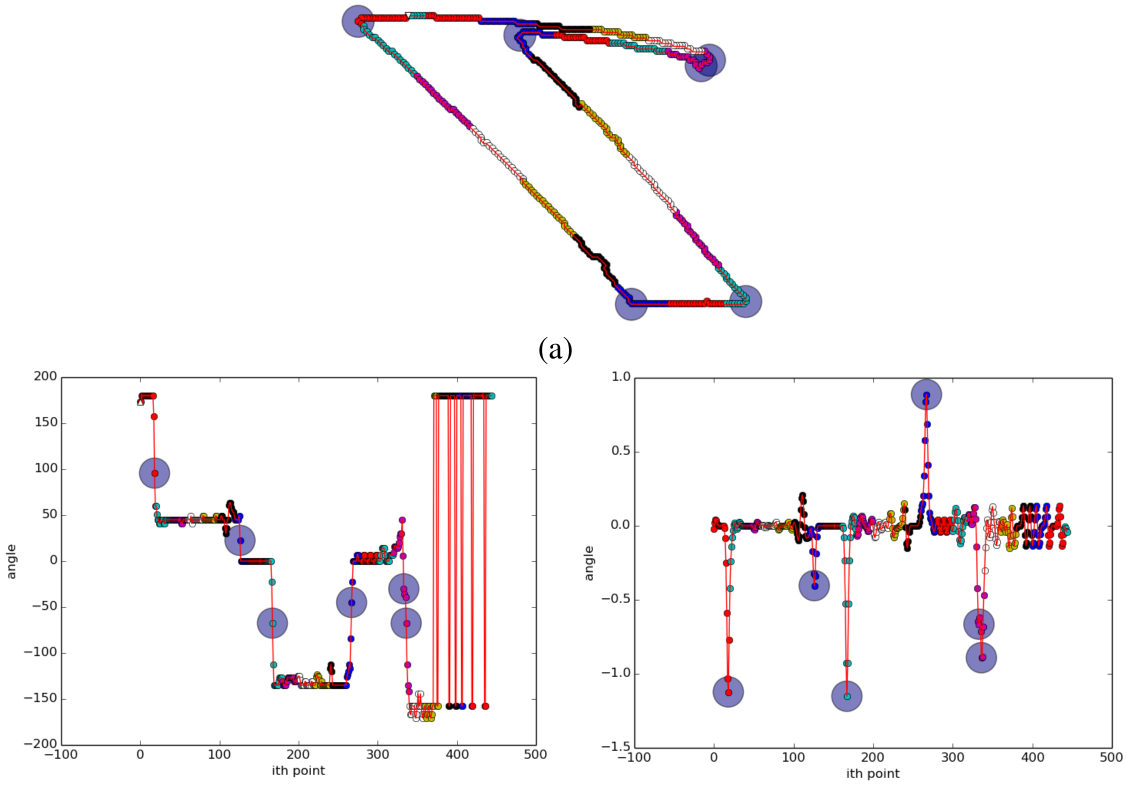


Figure 4.4. Synthetically generated test regions, their tangent direction curve and the curvature graphs and keypoints. On the figure, blue, relatively big circles show the keypoints, others show the points of the region boundary at (a), (d), and (g), they represent the tangent directions of the points at (b), (e), and (h), and at (c), (f), and (i) they indicates the curvature. The color of the small circles changes for every 20 points to make it more clear.



(a)

(b)

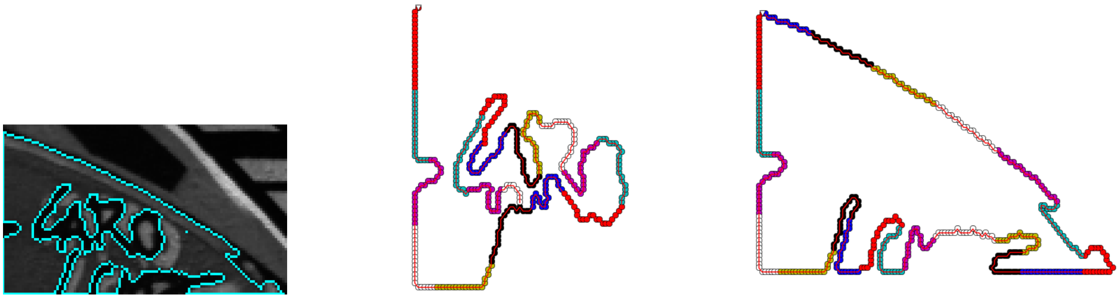
(c)

Figure 4.5. A real MSER and its tangent direction and curvature curves. All bending points of the boundary are detected by the proposed algorithm.

4.3. Implementation Detail

As mentioned earlier, MSER can be extracted both grayscale and color images.And there are possible implementations in the literature. In this thesis, we use Matas' executable with the default parameters on grayscale images. Moreover, Matas' executable has three different output type as mentioned in Chapter 2. We use second output type which is extended boundary because we try to locate keypoints on the boundary of the regions.

During the boundary traversal, first, the point that has minimum y coordinate is selected as the starting point. Then its eight neighbor pixels are checked there exists any



(a) (b) (c)

Figure 4.6. When the region at (a) is traversed, traversal starts with the starting point which is indicated by the white triangle it goes through the outer boundary until reaches to the point that the outer boundary point is one pixel close to the boundary of the hole inside the region. At that point, it continues with the boundary of the hole at (b), then it returns the outer boundary. Afterwards, it continues till reaches the starting point through the outer boundary at (c).

boundary point. If there is, it will be selected the second point of traversal and the algorithm continues checking its eight neighbor pixels. If there are more than one boundary point of eight neighbor pixels of any point, when going down relatively right neighbor is selected and when coming back relatively left neighbor is selected. It means from starting point to the bottom turning point right dominates, for the other half left dominates. By doing this, counter clockwise traversal is guaranteed. This algorithm defines closed curves in ideal cases however, there exist some exceptional cases like regions that have holes inside it. This type of regions causes failure sometimes and traversal process cannot define a closed curve. Because in some cases the boundary of a hole is one pixel close to the outer boundary and the traversal continues with the boundary of the hole. To overcome these cases, if the traversed boundaries cannot reach the starting point and there is no other unvisited boundary point at neighbors of the last visited point, the algorithm returns back till finding the boundary point that traversal can continue. Until the traversed boundary reaches the starting point, the process will continue and if necessary, it will return back. If the traversed boundary reaches the starting point that means traversal process finishes successfully. Figure 4.6 exemplifies that traversal continues with the boundary of a hole and returns back to the outer boundary and continues with it till reaches the starting point.

After the outer boundary is collected as a chain, it becomes an input for the second stage to calculate its tangent direction curve. During this process, there is a periodicity issue in horizontal. The boundary of extremal regions define closed curves and at the traversal, it is preserved. So when its tangent direction curve is calculated, the previous vectors of the first m points are defined by accepting last points of the curve as the starting point. For example, when the tangent direction of the first point is calculated, m previous vector is defined between the last m points and the first point itself. For the second point, the last $m - 1$ points are used and it goes like this till m^{th} point. Figure 4.7 shows the definition of the difference vectors of the first and the second points of a region. The same periodicity issue is valid for last m points as well. By taking advantage of the closed curve, horizontal periodicity is provided and $2m$ vectors can be defined for even the first and the last points of the boundary. After vectors are defined, they are converted to polar coordinates by taking $\arctan$ and the range is between $-\pi$ and π. Preserving the range of angles causes vertical periodicity. After sorting angles, the median of them is calculated

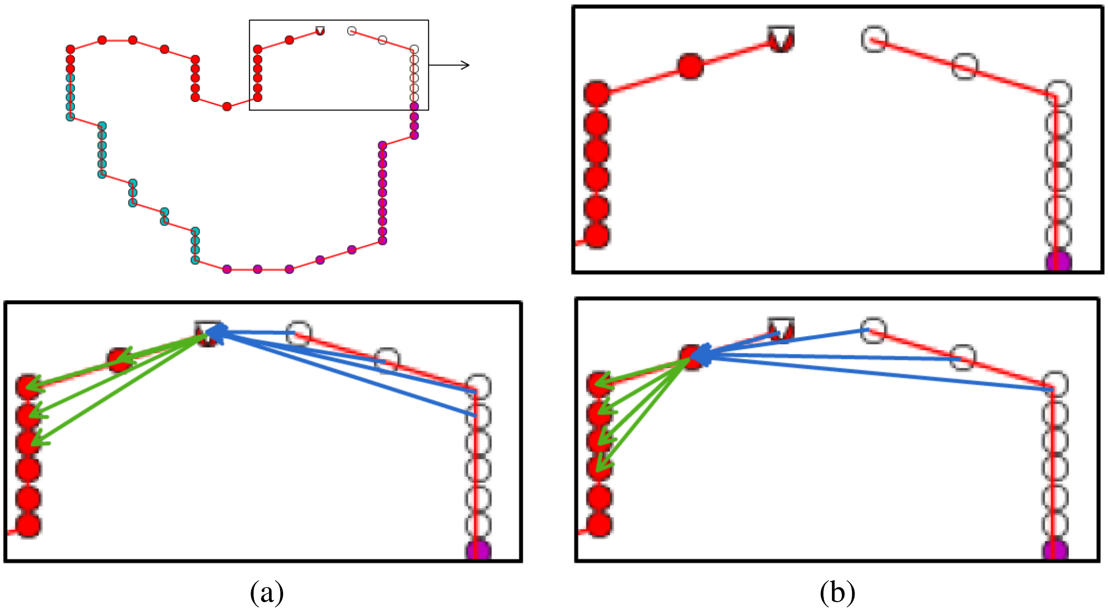


Figure 4.7. The horizontal periodicity is provided by taking advantage of the closed curve. For first m points during the definition of their difference vectors, the last points of the boundary is used. The same issue is valid for last m points as well. The region that is shown in the left top corner and at the right top the first points of its boundary are zoomed. The difference vectors of the first point and the second point are shown at (a) and (b) respectively.

by using below equation

$$tangent\ direction = \frac{\theta_m + \theta_{m+1}}{2} \tag{4.2}$$

because there are even number of vectors for each point. When applying Equation 4.2 vertical periodicity has to be handled. To do that, first $\cos$ and $\sin$ of angles are calculated and their average is calculated separately then taking $\arctan$ of them again gives the tangent direction without corrupting its the vertical periodicity. After the tangent direction curve of regions is computed, they are convolved and then their derivative is taken to compute their curvature. During both of them, the horizontal periodicity is provided by using closed curve advantage like tangent direction computation. And to handle the vertical periodicity at convolution part, the same procedure that is used at the finding median of $2m$ angle is applied. On the other hand, when Equation 4.1 is applied to differentiate, another procedure is followed to handle vertical periodicity. It is during the subtraction two angle if one of them is between $\frac{\pi}{2}$ and π and the other is between $-\frac{\pi}{2}$ and $-\pi$, they are subtracted then 2π added to the result. For other ranges, subtraction is done without adding 2π. After applying the additional controls to provide the vertical and horizontal periodicity, the tangent direction estimation and the curvature computation is stable enough to find the appropriate discriminative points. After the derivative curve is computed, local extrema are found to locate interest points as mentioned in section 4.2 and their locations are retrieved as the location of keypoint in the format of (x, y) coordinates.

4.4. Experiments

After detecting keypoints on the MSER boundaries, the stability of the proposed approach is analyzed. To do that its repeatability is measured on images of 2D objects and 3D objects. On the experiments, different experimental setups are followed for each object type. In addition to the proposed approach, the performance of two other approaches is analyzed in order to make comparison. The others are MSER itself [24] and Curvature Scale Space (CSS) [23]. The MSER stability is analyzed because comparing MSER and the proposed approach provides the answer of the question of how much the proposed approach improves the performance of MSER. To analyse the stability of MSER, its center points of the best-fitted ellipse of regions are used because in the literature, their best-fitted ellipses are preferred frequently instead of regions themselves. Details of

MSER algorithm and its ellipse approximation are mentioned in Chapter 2. The stability of affine invariant points that are detected by CSS is measured because its purpose and the purpose of the proposed approach are approximately the same. And details of CSS are mentioned in Chapter 2. The stability of MSER and CSS is measured by following the same experimental setups that are mentioned in next chapter in order to provide a fair comparison.

4.4.1. Setup

For experiments of 2D data, Oxford data set is used. In this dataset, there are 8 different image sequence [29], each of them has a different type of deformation.

- In Bikes and Trees sequence, the type of deformation is blur.
- In Graffiti and Wall sequence, the type of deformation is viewpoint change.
- In Bark and Boat sequence, the type of deformation is zoom and rotation change.
- In Leuven sequence, the type of deformation is light change.
- In UBC sequence, the type of deformation is JPEG compression.

In the image sequences, the first image is the reference image and the others are the deformed images. Figure 4.8 shows the reference images of the first four image sequence of Oxford dataset. The rate of deformation is increased from the first to the last image of the deformed images. Figure 4.9 shows the Graffiti image sequence and Trees image sequence. The dataset also has the ground truth homography between the reference image and the deformed images. Homography is a way to define the relation between two images of the same planar scene.

The stability analysis performs by measuring the repeatability between the reference and the deformed images. After the proposed keypoints are extracted from each image of a sequence, the keypoints of the reference image are projected into the deformed images. To project them, homogeneous coordinates of each keypoint is multiplied with the ground truth homography. After projection, their correspondence is found by checking is there any keypoint at that point with at most 3 pixel distance. If there is not any, the keypoint of the reference image is accepted as not repeat on the deformed images. If

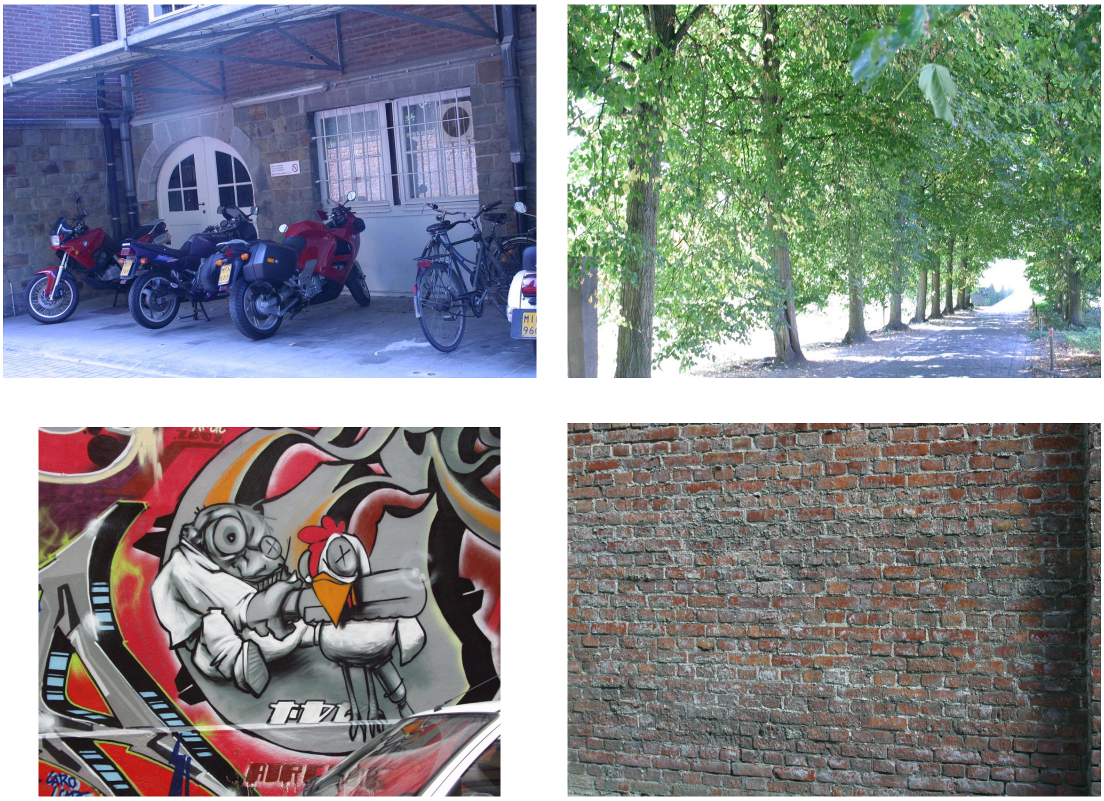


Figure 4.8. The reference images from Oxford dataset of Bikes, Trees, Graffiti, and Wall image sequences from right to left and top to bottom.

there exists only one point in 3 pixel distance, they are accepted as the potential correspondence. If there exists more than one keypoint, the closest keypoint among them and the projected keypoints are accepted as the potential correspondences. Figure 4.10 shows the projection and potential correspondences determination. After the potential correspondences are determined, two post process is applied. First, one to one correspondence is provided to obtain real correspondences. Second, the keypoints that are located out of the common part of the image pair is removed from the repeatability calculation. For both of the post processes, the same procedure that is mention in Chapter 3 is followed. After measuring the repeatability of the proposed keypoints, the experiment is repeated with the center coordinates of the ellipse of MSER and coordinates of affine invariant points that are detected by CSS. Due to the performing the same experiment both the proposed approach, MSER, and CSS, we can compare them fairly.

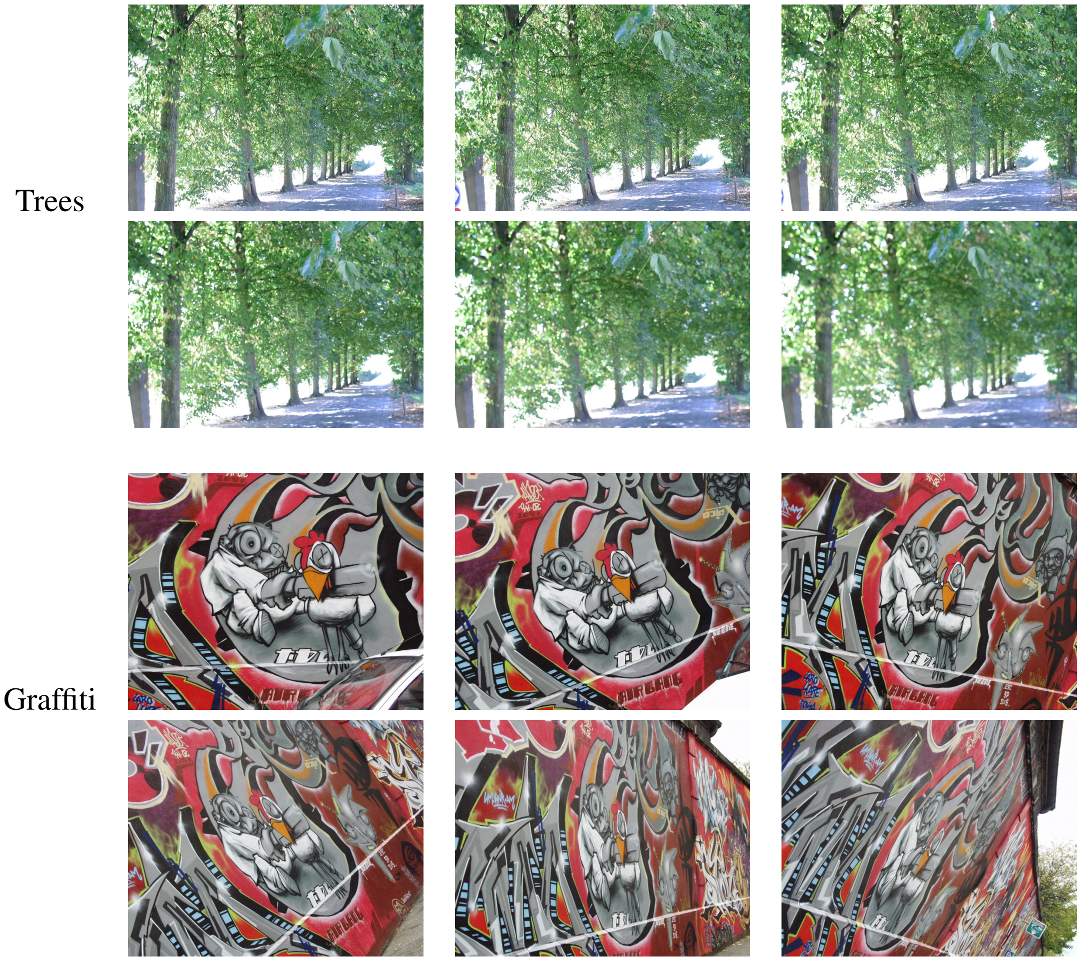


Figure 4.9. Trees and Graffiti image sequences. The reference images and five deformed images of the Trees and Graffiti image sequences. The amount of deformation increases from first to last.

For the experiment with 3D data, Moreels et al. dataset is used [31]. In this dataset, there are photos of 100 objects. 3D objects are dominated in the dataset but besides 3D object, there are some flat objects as well. While some of them have textured surface, some of them have homogeneous surface. Figure 4.11 shows some example objects. When the dataset is created, they were put on the turntable and their images are taken by two cameras which are bottom and top. The range of taking images is 5° when the turntable is turning around. And the objects were turned three times to take their images under three different lighting conditions. In addition to images of objects, there are also images of calibration pattern which is chessboard. And they are taken from -55° to $+55^{\circ}$ for each 5°. So there are 23 images for each camera, and the calibration

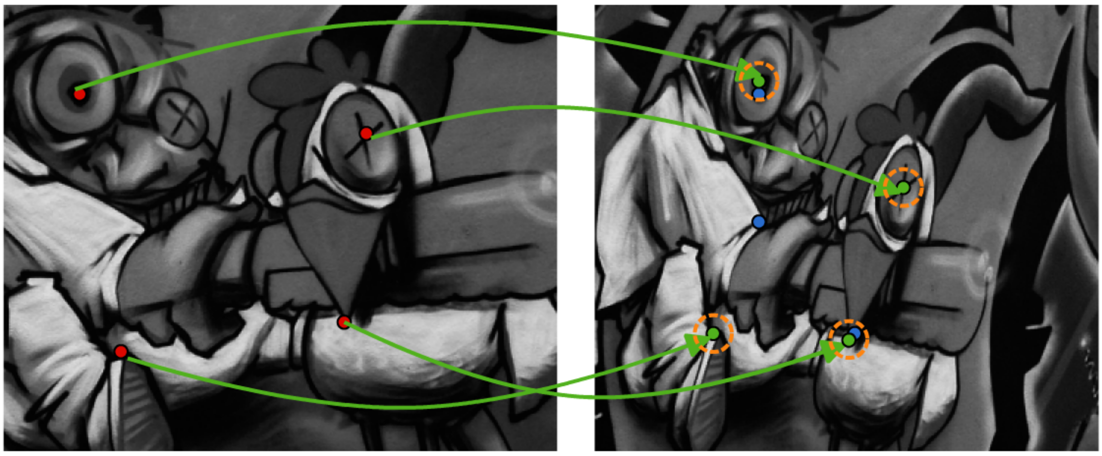


Figure 4.10. To calculate repeatability, keypoints of the reference image are projected into the deformed image, and searching there exists any keypoint close to them at most 3 pixel distance. The left image shows a part of the reference image of the Graffiti image sequence and the right image shows a part of the second deformed image. Red dots represent the keypoint of the reference image, green dots are obtained by multiplying the location of the red keypoints with the ground truth homography and blue dots are the keypoints of the deformed image. The yellow dotted circle shows the searching area of the projected keypoints. In this simplified example, two of the keypoints of the reference image is repeated. And the repeatability is 50% that is calculated by dividing the number of the repeated keypoints to the number of the reference keypoints.

pattern put three different positions, so the number of the image of the calibration pattern is 69 for each of camera. To calibrate cameras, corners of the chessboard is marked and calibration and distortion matrices and for each angle translation and rotation matrices are obtained by using OpenCV methods. After calculating them, for each possible image pair, fundamental matrix can be calculated by using them. The fundamental matrix defines the relation between two images of the same 3D object.

$$\begin{bmatrix} F \end{bmatrix} = [e_2]_\times \begin{bmatrix} P_2 \end{bmatrix} \begin{bmatrix} P_1 \end{bmatrix}^+ \tag{4.3}$$

To calculate it Equation 4.3 is used. In this equation, e_2 is the epipole of the camera of the second image and calculated by using Equation 4.4.

$$\begin{bmatrix} e_2 \end{bmatrix} = \begin{bmatrix} P_2 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} -R_1 \end{bmatrix}^T \begin{bmatrix} t_1 \end{bmatrix} \\ 1 \end{bmatrix} \tag{4.4}$$

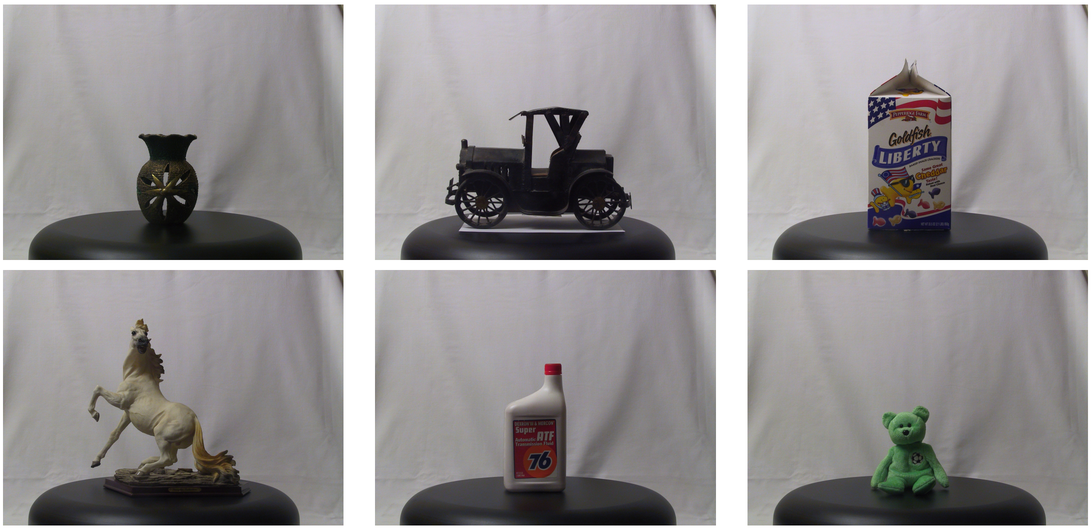


Figure 4.11. Example objects from Moreels et al. dataset [31]. These images are taken from the bottom camera with the angle of the turntable is $0°$. The objects are called as Base, Car2, Carton, Horse, Oil, Teddy Bear from left to right and top to bottom.

In the fundamental matrix (F) and epipole (e) equations, P_1 and P_2 stand for projection matrices of the first and second camera and they are calculated by using Equation 4.5.

$$\begin{bmatrix}P\end{bmatrix} = \begin{bmatrix}K\end{bmatrix}\begin{bmatrix}R|t\end{bmatrix} \tag{4.5}$$

In the epipole (e) and projection matrix (P) equation, K stands for calibration, R rotation, t translation matrices and they are computed in the calibration process.

The stability is measured by using geometric constraints at the triplets which are $0°$ bottom, $0°$ top, and $\theta°$ bottom images [31]. θ is selected from $-50°$ to $+50°$ with steps of $10°$ and only the first lightning condition of images is used in our experiments. After each image of the triplet is undistorted, keypoints are extracted. Then keypoints of $0°$ bottom image is projected into $0°$ top image and $\theta°$ bottom image as epipolar line. During this process, Equation 4.6 is used to project points as epipolar line. Assume that, there are two images, and F_{12} in the equation represents the fundamental matrix that is calculated by using Equation 4.3. It is used to project any point x_1 from the first image into the second image as an epipolar line l_2.

$$\begin{bmatrix}l_2\end{bmatrix} = \begin{bmatrix}F_{12}\end{bmatrix}\begin{bmatrix}x_1\end{bmatrix} \tag{4.6}$$

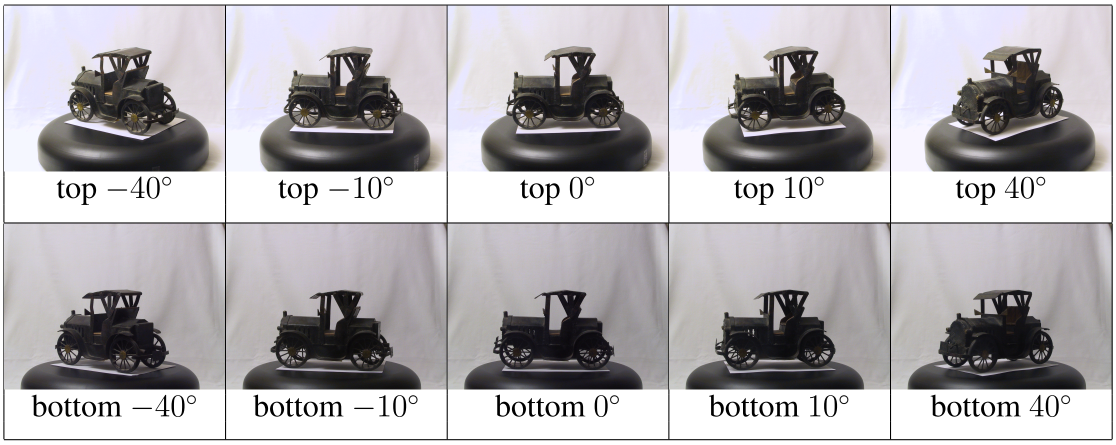


Figure 4.12. Car2 image sequence. After this object put on the turntable, its images are taken for each $5°$ from both top and bottom cameras. In the first row, only $-40°$, $-10°$, $0°$, $10°$, and $40°$ images that are taken from the top camera are shown. In the other row, images from the bottom camera are shown.

Then if there are any keypoints close to the epipolar line correspondence at most 3 pixel distance on the $0°$ top image, they are also projected into the $\theta°$ bottom image by using Equation 4.6. After intersection points of the epipolar lines that are coming from $0°$ bottom and top images are defined, whether is there any keypoint close to the intersection points at most 3 pixel distance is checked. If there is no, the keypoint of the $0°$ bottom image is accepted as unstable keypoint. If there exists only one keypoint that provides the constraints, it is marked as potential correspondence. If there are more than one keypoint, the closest keypoint is accepted as potential correspondence. Figure 4.13 shows an example of determining the potential correspondences step of the stability analysis process. A keypoint of the $0°$ bottom image that is indicated as red dot projected into $0°$ top image and $30°$ bottom image. Then if there are any keypoint close to the epipolar line at most 3 pixel distance on the $0°$ top image, they are projected into the $30°$ bottom image. After that intersection points of epipolar lines are determined. If there are any keypoint close to the intersection points, the keypoint at the $0°$ bottom image and $30°$ bottom image are accepted as potential correspondence. According to this process, because of using $0°$ top image as the intermediate step, stability between $0°$ bottom and $0°$ top image effects the stability between $0°$ bottom and $30°$ bottom image.

After potential correspondences are determined, two post process is applied. They are providing one to one correspondences and removing keypoints that are located out of

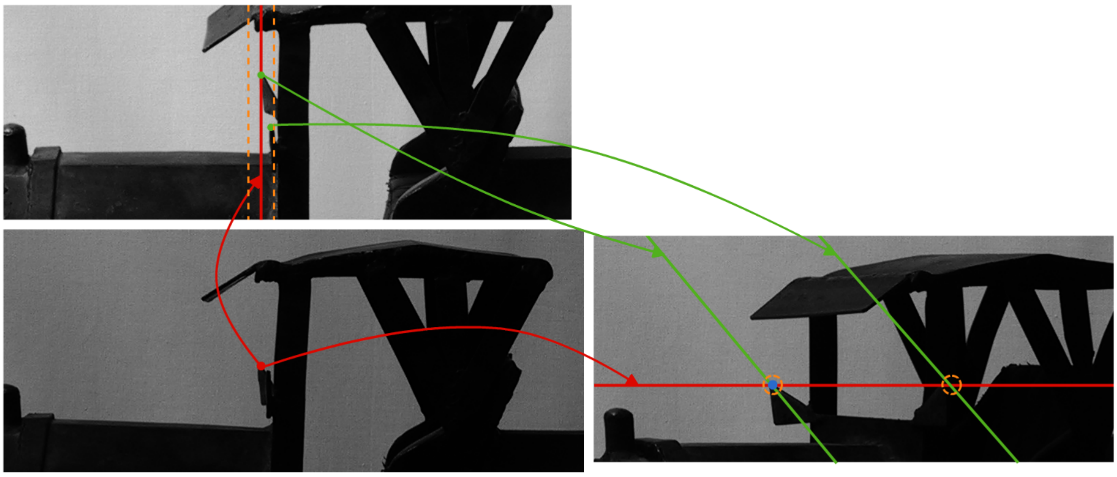


Figure 4.13. The stability analysis of 3D data. The triplet is obtained by zooming to Car2 image. Sub image at the left top shows 0° top image, left bottom is 0° bottom image and right bottom is 30° bottom image. Keypoint that is indicated as a red dot at the left bottom sub image projected into the 0° top and 30° bottom images as epipolar lines. They are drawn as red lines. The yellow dotted lines at the left top image show the searching area and their distance to the red line are 3 pixels. There are two green keypoints inside the area and they are projected into the right bottom image as epipolar lines. They are indicated as green lines. After keypoints are projected, intersection points of the epipolar lines are determined. And then the other searching process is begun to find there exist any keypoints inside the yellow dotted circles. The half radius of the circles is 3 pixel. There is only one keypoint on the searching area at the right bottom image and it is shown as a blue dot. So blue keypoint is the potential correspondence of the red keypoint.

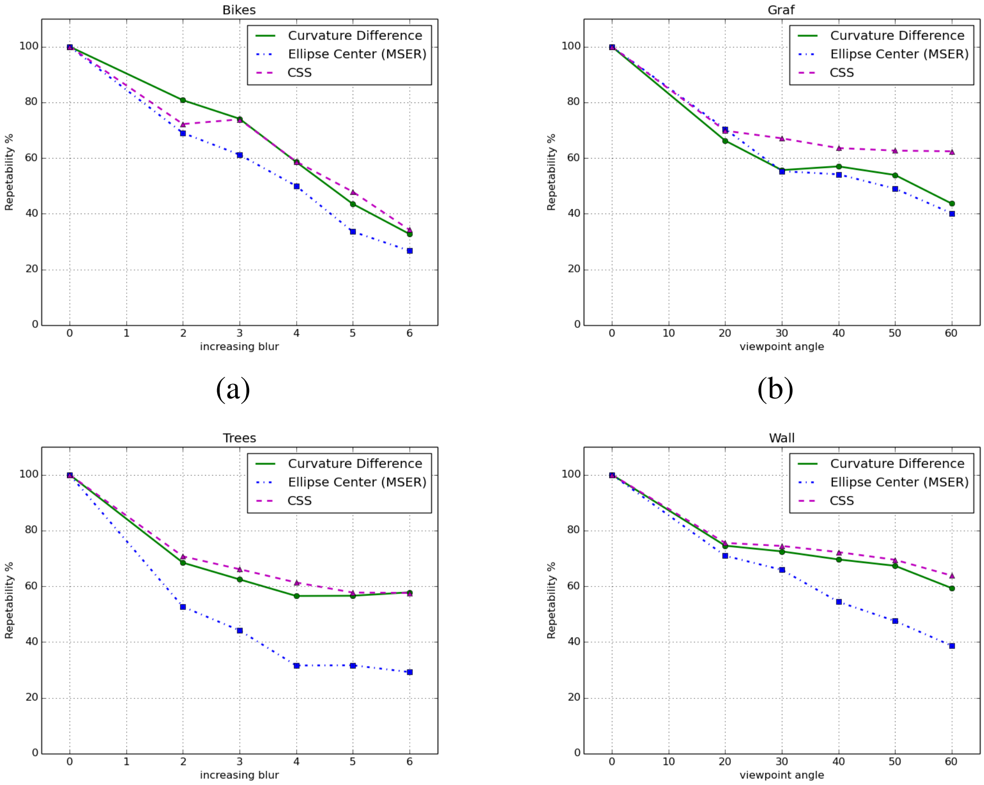
the common part of triplet images from stability analyses. For the first post process, the procedure that is applied in experiments with 2D data is applied exactly. For removing keypoints that are located out of common part of a pair of images, two similar procedures are applied. During the epipolar line projection steps from 0° bottom image to the 0° top image and from 0° bottom image to the θ° bottom image, if epipolar lines are out of the projected images, they are removed from the stability analysis. After the keypoints are projected from 0° top image into the θ° bottom image, intersection points are calculated between the epipolar line pairs. If all intersection points are found out of the θ° bottom image, this keypoint is removed the stability calculation as well. This stability analysis is applied keypoints that are detected by proposed approach, center points of the best-fitted ellipses of MSER [24], and the affine invariant points that are detected by CSS [23] for a fair comparison.

4.4.2. Results

After experiments are finished, the results are represented as graphs. On the graphs, the repeatability of the approaches are shown. The results of the proposed approach, the ellipse centers, and CSS are marked with green circles, blue squares, and purple triangles respectively. And there are three different lines in addition to the marked points on each graph and they are listed below:

- Green line represents the repeatability of points that are detected by the proposed approach.
- Purple dashed line shows the repeatability of center points of the best-fitted ellipses of MSER.
- Blue dotted dashed line demonstrates the repeatability of the affine invariant points that are detected by CSS.

Figure 4.14 shows the results of experiments with 2D data. The repeatability of the first four image sequence of the Oxford dataset is indicated in this Figure. The repeatability of the other four object is in Figure A.1 in Appendix A. When the stability of three approaches are sorted, the CSS is ranked as the first. And it is close to curvature difference. Moreover, for almost each image, the repeatability of curvature difference is higher than the repeatability of MSER. So when they are sorted, the sequence is CSS, curvature difference, and then MSER. By comparing curvature difference and the CSS, their success is close to each other and their behavior are almost the same. Moreover, the repeatability score of the CSS is higher than curvature difference except for first images of Bikes image sequence. And except for the Graffiti image sequence, the performances of curvature difference and CSS reach almost the same repeatability value at the high amount of deformation. When curvature difference is compared with MSER, curvature difference has better repeatability values than MSER. Especially, Trees and Wall image sequences, the gap between them is considerable. In Trees image sequence, while the repeatability of curvature difference is going down from 68% to 58% through the deformed

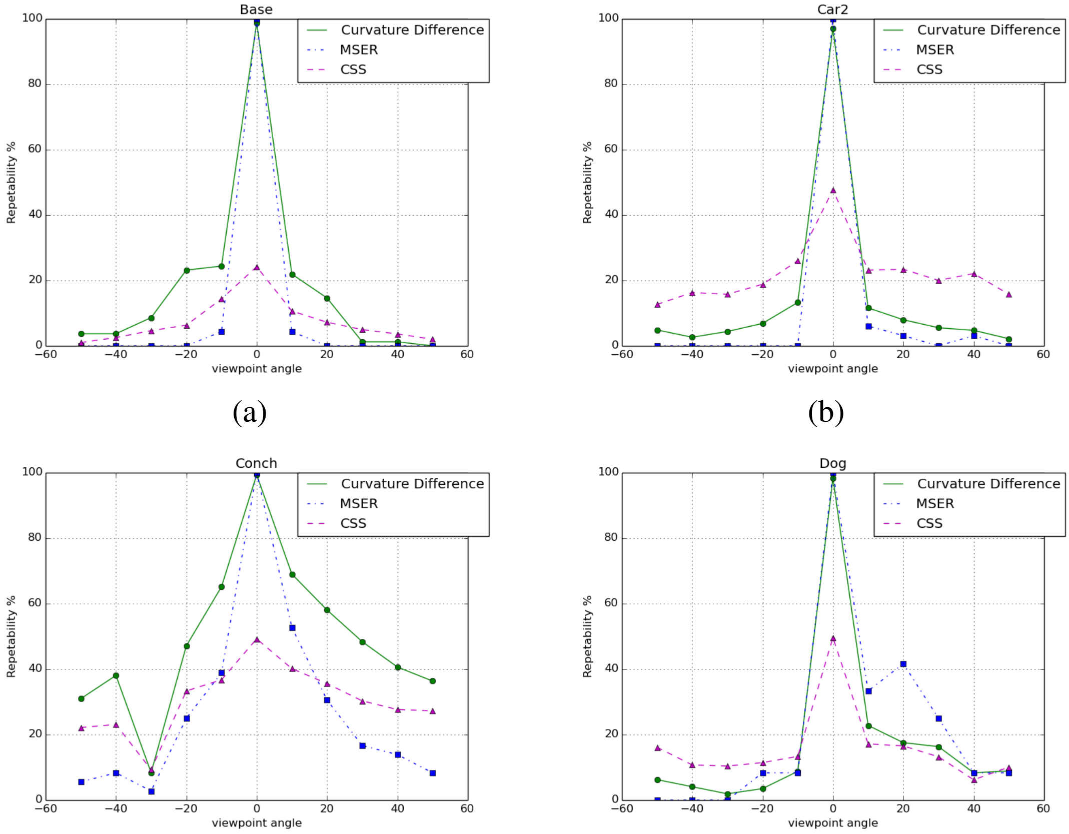


(a) (b)

(c) (d)

Figure 4.14. The repeatability of first four image sequences. The results of Bikes, Trees, Graffiti, and Wall image sequences are shown from top to bottom and left to right. The performance of the proposed method is better than MSER and close to CSS on the Bikes, Trees, and Wall image sequences. For Graffiti image sequence, the performances of the proposed method and MSER are close to each other and they are worse than CSS.

images, MSER's is going down from 53% to 29%. So the worst repeatability result of curvature difference is 5% better than the best of the MSER. For Wall image sequence trend of both curves is the same with the Trees image sequence. The results of curvature difference are between almost 75% and 59% and the results of the MSER are between almost 71% and 39%. The decrease of curvature difference is half of the decrease of the MSER repeatability. For Bikes, although their performance is close to each other and shows the similar behavior, the curvature difference shows better performance than the MSER and the difference between their repeatabilities is about 10% for each deformed image. Only for Graffiti image sequence, the repeatability of MSER is higher than curvature difference repeatability, but it is short time precedence. On the third image, their repeatability scores are equal to each other and almost 55%. For the last three deformed images, their performances are close to each other and curvature difference has slightly better performance than the MSER.



(a)

(b)

(c)

(d)

Figure 4.15. The repeatability of first four objects of a subset that has 3D objects. The results of Base, Car2, Conch, and Dog objects are shown from top to bottom and left to right. For the object Base and Conch, the repeatability values of curvature difference has the highest values. For Car2 object CSS performs better than curvature difference and MSER. For object Dog, their performances are close to each other except for at 0°and 20°.

Figure 4.15 and Figure 4.16 show the results of experiments with 3D data. The first figure shows the repeatability of only first four objects in alphabetic order from a subset of Moreels et al. dataset and their names are Base, Car2, Conch, and Dog. The repeatability of the other objects which named FlowerLamp, GrandfatherClock, Horse, Motorcycle, Robot, TeddyBear, and Tricycle is indicated in Figures A.2 and A.3 in Appendix A. The second figure shows the repeatability of first four objects in alphabetic order from a subset of the same dataset. The subset has flat objects which are Carton, Clamp, Eggplant, and Lamp and the repeatability of the other objects which named Mouse and Oil is shown in Figure A.4 in Appendix A. For almost each of the eight object, the repeatability at 0° is the highest and it decreases while going away from 0°. Another common feature of the curves, almost all of them have symmetry according to 0°.

According to Figure 4.15, for the object Base and Conch, the repeatability values of curvature difference has the highest values. For Car2 object, CSS performs better than curvature difference and MSER. For object Dog, their performances are close to each other except for at 0°and 20°. Especially, for Base and Car2 objects, the curves of the approaches has symmetry according to 0°. While, in general, the repeatability of MSER drops 0% immediately, the repeatability of curvature difference and CSS drops steady and they do not reach 0% at even the highest amount of deformation that is 50°and -50°. For object Conch, except for the pit on -30°, curves of the approaches decrease steadily in both directions. However, the decrease of MSER and CSS is more than the curvature difference. The performance of curvature difference is different from MSER and CSS is 25% and 15%, on the average respectively. Among these four objects, the MSER's performance is better than others for only Dog object in positive direction, but at the high amount of deformations, the repeatability of curvature difference and CSS reach to the MSER's and it is shown at 40°and 50°.

Figure 4.16 shows the result of the stability analysis when flat 3D object subset is used in the experiment. For Carton and Lamp objects, the curvature difference has the highest repeatability at each angle. And for the object Carton, the gap between curvature difference and MSER is in the range between about 40% and about 15%. The highest gap is at -10° and the smallest gap is at -50° and 50°. The gap between curvature difference and CSS is higher in the negative direction than the positive direction and it is shown as asymmetry in the graph. So in the negative direction, their performance are better than the performance under the positive direction changes. Its maximum value is 47% at 0°and its minimum value is 5% at 40°. For Lamp, their performances are closer to each other than the object Carton's. MSER repeatability for the same object drops to 0% and it remains 0% between 20° and 50°. For the object Clamp, their stabilities are close to each other and except for -20° and 0°. The result of the experiment with Eggplant object shows instability for the MSER. It is about 17% at 0° then it increases 33% at -10° and 50%

at 10°, then it drops to 0% drastically for both direction. The repeatability of curvature difference for Eggplant object is 76% at 0°, then it drops suddenly, then it increases a little bit, then drops slowly in both directions. On the other hand, the performance of CSS starts with 42% and drops steadily.

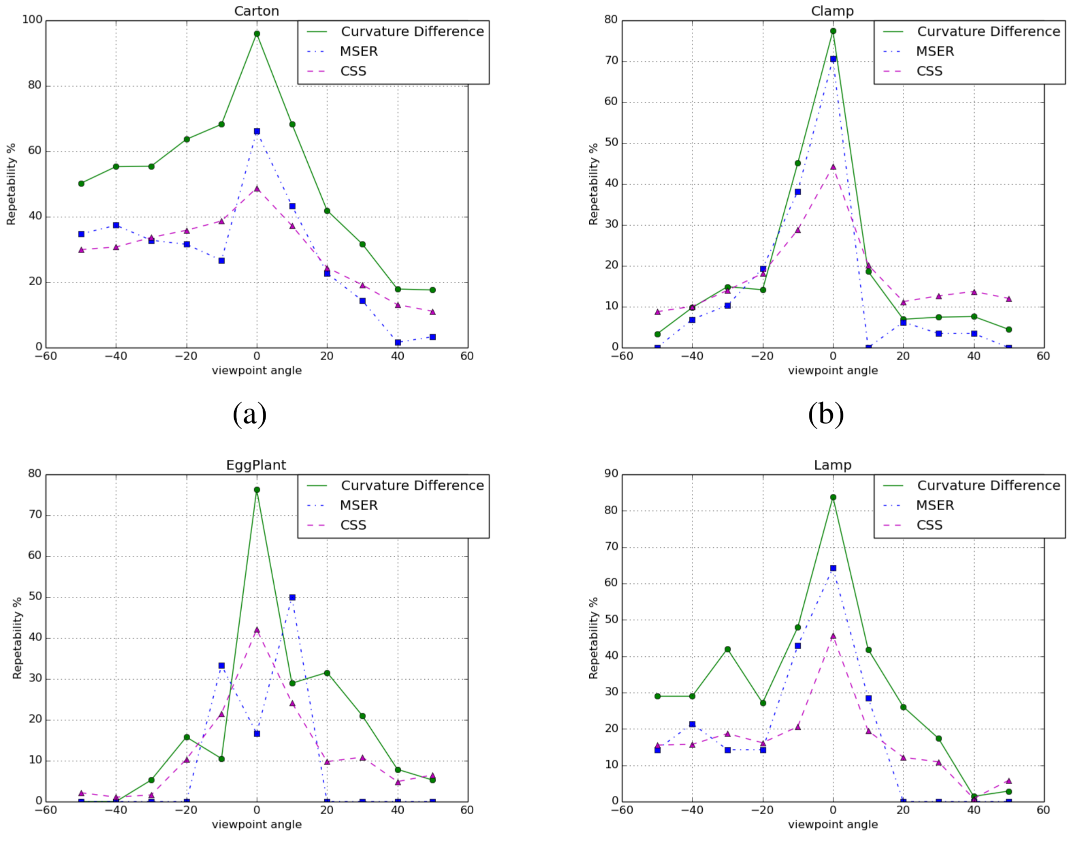
4.5. Discussion

As shown in Figures 4.14, 4.15, and 4.16, the performance is the proposed approach is better than the center of the best-fitted ellipses of the regions. When we compare the repeatability of the points that are detected by the proposed approach with affine points that are detected by CSS, although, in the experiments with 2D data the performance of CSS is slightly better than curvature difference, in the experiments with 3D data the curvature difference performance is better than the CSS performance. So the proposed approach is useful when MSER algorithm will be used in the works that require individual points like 3D reconstruction. That means using the proposed approach increases the performance of MSER in especially 3D operations which supports our initial expectation. In the experiments, there are some significant observations that are explained below.

The first observation is that the gap between the curvature difference and ellipse centers is getting bigger when the amount of the deformation is increasing. In other word, the curvature difference is more robust to the high amount of deformation by comparing the ellipse center approach. This can be seen in Figures 4.14(c), 4.14(d), 4.15(c), and 4.16(a) particularly.

The second observation is the trend of the graph of the proposed approach and CSS is the same. Figures 4.14(c), 4.14(d), 4.15(c), and 4.16(a) are examples of this observation. This behavior occurs because the proposed approach and CSS are an algorithm to detect interest points on the region boundary. Furthermore, although both approaches detect interest points at different locations, they are triggered by the same segment of a boundary. So the stability or instability of the segment affects their performance in the same manner.

Another significant observation is that zero repeatability value in a short time and the instability of the performance of the ellipse center approach. The repeatability of the ellipse centers with 3D data reaches 0% suddenly even under small rotation changes. This can be seen in especially Figures 4.15(a), 4.15(b), and 4.16(c). On the other hand, in the experiments with 2D data this cannot be observed and also the lowest repeatability value is almost 30%. Furthermore, while the repeatability results with 2D data of the ellipse center have a specific behavior, it has instability in the experiments with 3D data. For example, its performance for the object Eggplant shown in Figure 4.16(c) has a pit at 0°and two high peaks at -10°and 10°. These two observations that occur in experiments with 3D data causes a doubt to use MSER in individual points required works.



(a)

(b)

(c)

(d)

Figure 4.16. The results of Carton, Clamp, Eggplant, and Lamp objects are shown from top to bottom and left to right. The proposed method has better performance on the Carton and Lamp objects and the performances of approaches are close to each other for the object Clamp. For Eggplant, there is instability in the MSER performance and while at 10°and -10°it has better performance than the others, in general, the repeatability of others are better than MSER. Furthermore, except for 10°and -10°, curvature difference has the highest performance.

The last observation occurs in the experiments with 3D data only. While the repeatability of the reference image is 100% in the experiments with 2D data, it does not reach 100% for some 3D objects. For example, the repeatability of the curvature difference for the reference images reaches 96% for the Carton object, 77% for the Clamp object, 76% for the Eggplant object, and 84% for the Lamp object. Likewise, the repeatability of the ellipse center of MSER also cannot reach to 100% and they are 66%, 70%, 17%, and 64% for the same objects respectively. Furthermore this observation is more obvious in the repeatability of the affine points that are detected by CSS also cannot reach to 100% and they are 48%, 44%, 42%, and 45% for the same objects respectively and this can be seen in Figure 4.16. This is caused by the experimental setup, because according to it, the repeatability between any image pair is calculated by using an auxiliary image. So the repeatability between the reference image and the auxiliary image affects the repeatability of any image pair.

4.6. Conclusion

MSER keypoint detection algorithm detects extremal regions which is a kind of image feature and it retrieves regions as a group of consecutive pixels. However in the Literature, they are used with the best-fitted ellipse approximation and because of this, the information on the boundary is ignored. For example, they are used in 3D reconstruction with the center of their best-fitted ellipses because individual points are required instead of regions in 3D operations. Furthermore the outcome of the analysis of the stability of MSER shows the stability of the boundaries are better than the stability of the regions themselves. In order to utilise the stability of the boundary, to prevent the information loss, and to be able to use them in 3D operations with better performance, we propose a local approach. According to the proposed approach, after regions are detected, tangent direction for each boundary point is calculated and by taking their derivative the curvature of the boundary is computed. Then discriminative points are selected by comparing the curvature values of the boundary points.

The proposed approach is evaluated by comparing its performance with affine invariant points that are detected by CSS and the center of the best-fitted ellipse approximation. In the evaluation, 2D and 3D data are used by following the proper experimental setups. The results of the evaluation show the performance of the proposed approach is better than using the center of their best-fitted ellipses. Although the success of the proposed approach is close to the success of CSS in the experiments with 2D data, it is double of the success of CSS in the experiments with 3D data. So especially results that are measured in the experiments with 3D data shows the significant performance improvement and this improvement fulfills our initial expectations. In a short, MSER can be used in 3D operations efficiently by following the proposed approach.

Although the proposed approach detects interest points on the boundary of regions and their stability is good enough to use them in computer vision applications, scale for them cannot be estimated. And this causes two consequences. First, the proposed keypoints cannot be used in scale invariant matching. Second descriptors for them cannot be computed because patch around them cannot be specified accurately.

CHAPTER 5

DESCRIPTION ON IMAGE CURVES

5.1. Introduction

In computer vision, description of the features makes them usable in applications. To compute descriptors, first an image is given to a feature detection algorithm to extract the features. Then they are given to a feature description algorithm in order to compute descriptors. Descriptors contain discriminative information of the keypoint and it can be computed from the texture, shape etc. In the literature, there are several algorithms to compute descriptors. When they are classified according to the input, there are two main groups which are texture based descriptors and shape based descriptors. For the first group, individual points are required as an input and the algorithms define a patch around the keypoint and use the texture inside the patch. For the second group, a shape is required as an input and according to their description principle, they are classified as contour based, region based, and skeleton based. Contour based algorithms such as curvature scale space [30], shape context [2] etc., ignore the texture and concentrate on only the contour of the shape. Region based shape descriptors [13, 20] are computed by using the texture inside the region. The last type of shape descriptor such as shock graph [38] first computes the skeleton of the shape then by using it the descriptor is computed.

In this chapter, we design an algorithm to compute a descriptor in order to describe objects by using only the contour of shapes. So as an input for this algorithm, boundary points of a shape, like contour of a silhouette or contour of a region that is detected by region detectors like MSER, are required. So it can be classified as shape based descriptor according to the input and contour based descriptor according to the description principle. Moreover, the main target of this descriptor is classifying characters with high success rate. Characters are selected because we believe that their boundary has enough information to classify them.

5.2. Approach

The proposed approach has two main part to compute descriptor. They are orientation estimation and descriptor computation. Before the main parts, the boundary points of the contour of a shape should be traversed. During the traversal, only the outer boundary is retrieved as a chain in the counter-clockwise direction and no point that is located on the outer boundary is omitted. And it is mentioned in Chapter 4 in detail.

To estimate the orientation of a shape, tangent direction for each individual point of traversed boundary is computed by applying the median filtered differencing algorithm [25] that is also mentioned in detail in Chapter 4. This algorithm takes m as a parameter. It defines the half number of the difference vectors and it has an influence on the smoothness. Since the tangent directions of the points are computed by taking the median of the polar angle of the vectors. So, by choosing a high m, the effect of the noise vectors are minimized. During the process m is taken as three, so six difference vectors are generated because we observe that six is enough to handle the noise. After the tangent directions are computed, its histogram is generated with 64 bins. To obtain a stable histogram, linear interpolation is applied. Then, its peaks are found and labeled as orientations. However, before finding the peaks, the histogram is smoothed to eliminate peaks that are caused by ripples. To smooth, the mean filter is passed two times on the histogram. It is one of the spatial filters also the convolution filter. Its kernel that is shown in Equation 5.1 is selected one dimensional in order to smooth the histogram in one direction.

$$\frac{1}{3} * \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{5.1}$$

During the finding the peaks of the smoothed histogram, local maxima points are found, then the global maxima among them is labeled as the first orientation. For others, if their number of the elements is higher than or equal to 60% of the number of the elements of the first estimated orientation, they are accepted as secondary estimated orientations. Both the first estimated orientation and secondary orientations are accepted as the estimated orientations and they have no precedence over each other. Figure 5.1 shows the orientation estimation process of a region that is shown in Figure 5.2(a). In this figure, there are 64 blue bars that are the histogram bars of the tangent direction of the boundary points. Red line is obtained by smoothing the curve that is formed by combining the top of histogram bars. Black dotted line shows the limit of selecting the secondary orientations, so local maximas below it are not marked as orientation. Green circles represent the selected local maxima that are estimated orientations. So, for this shape estimated orientations are 167.5°, 317.5°, and 27.5°which are given in the descending order with respect to the number of samples at those angles. They are 30.09, 29.90, and 20.96 respectively. So 167.5°is selected as the primary orientation and the others are the secondary. The limit of selecting the secondary orientation is calculated as 18.05 and it is indicated as black dotted line.

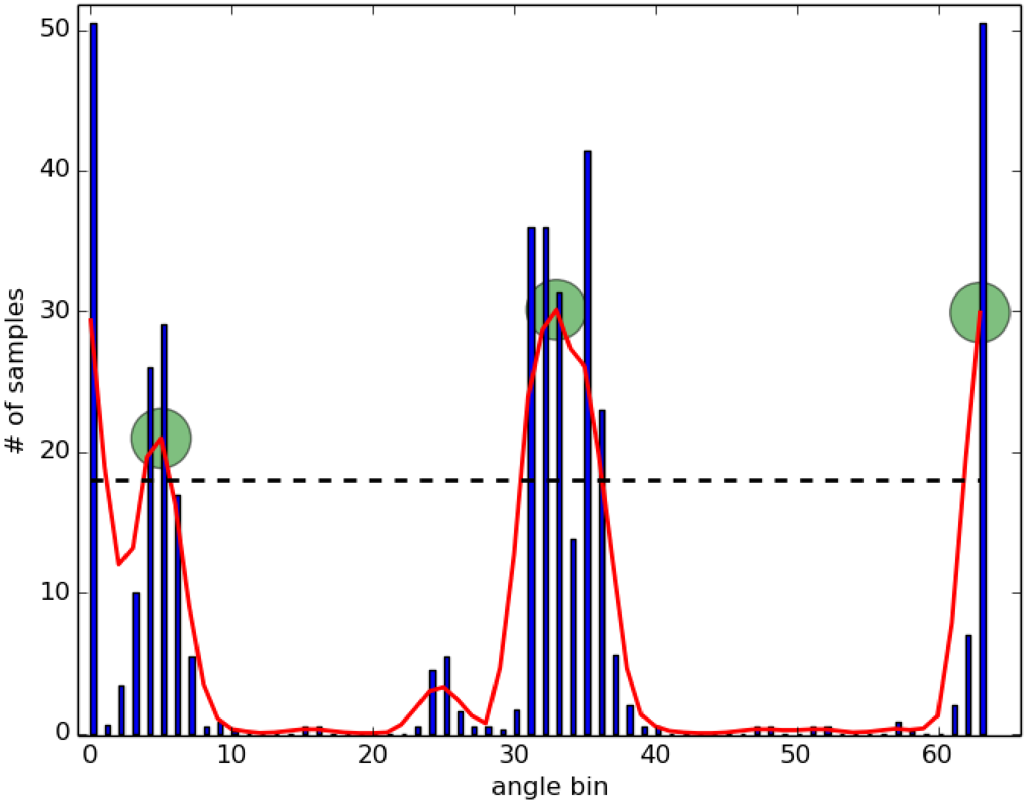


Figure 5.1. To estimate orientation, the histogram of the tangent directions of the points of the shape is generated and it is represented with blue bars. After it is smoothed which is shown as red line, its peaks are marked as the estimated orientations and indicated as the green circles. Black dotted horizontal line shows the limit of selecting the secondary orientation. For this sample shape which can be shown in Figure 5.2 (a), there are three estimated orientations which are 167.5°, 317.5°, and 27.5°.

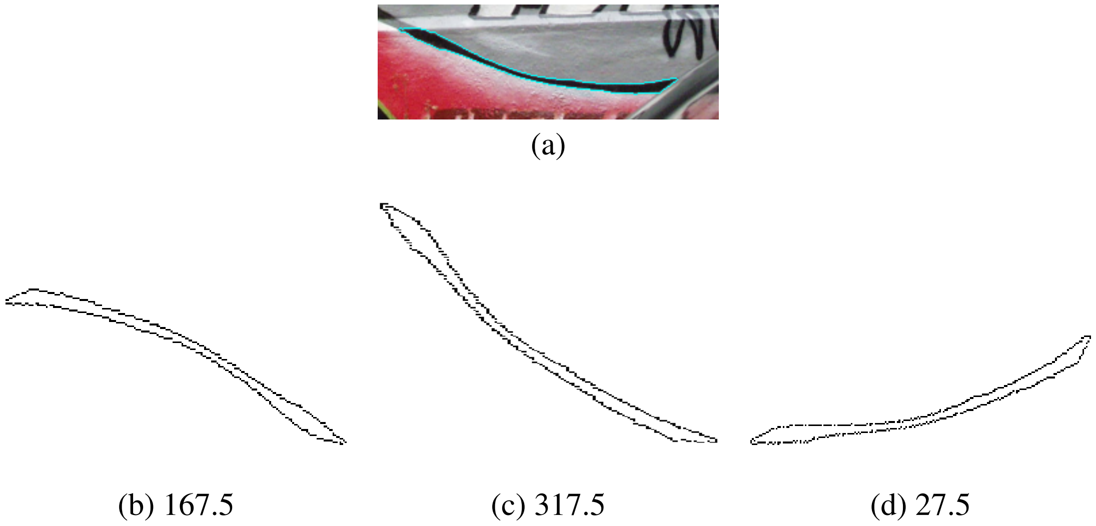
After orientations are estimated for a shape, for each orientation descriptor is computed by following the below processes. So actually, for each shape, the descriptor is computer and the below processes are applied as many as the number of estimated orientations.

1. Shape normalization
2. Fitting a grid onto the normalized shape
3. Computing how many boundary points exist inside each cell of the grid
4. Computing the tangent directions of points of the normalized shape
5. Descriptor computation

In the first process, the shape is normalized by rotating. The locations of the contour points are multiplied with the rotation matrix. Equation 5.2 is constructed when ψ is equal to the negative estimated orientation.

$$R = \begin{bmatrix} cos\psi & sin\psi \\ -sin\psi & cos\psi \end{bmatrix} \tag{5.2}$$

In Figure 5.2, an example shape and its normalized versions according to estimated orientation are shown. For this sample region, number of the estimated orientations is three and they are 167.5°, 317.5°, and 27.5°. After normalization, a grid is fitted to the normal-



(a)

(b) 167.5

(c) 317.5

(d) 27.5

Figure 5.2. (a) shows the region and its boundary which is drawn with light blue. (b), (c), and (d) shows its normalized versions according to the estimated orientations which are 167.5°, 317.5°, and 27.5°.

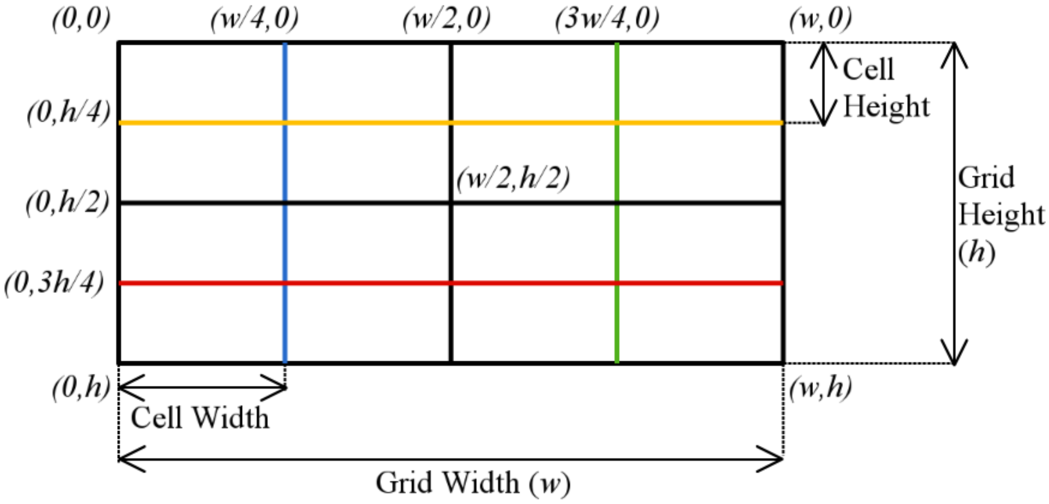


Figure 5.3. Fitting the grid. The grid width (w) and the grid height (h) is accepted as the width and height of the bounding box that is extracted from the normalized shape. The cell width (c_w) and cell height (c_h) are the dimensions of cells. They are computed by dividing w and h into four to equalize the dimensions of cells.

ized shapes. To do that, the scale is required, so before fitting the grid, a bounding box is extracted from the normalized shape, and its width (w) and height (h) are accepted as width and height of the grid. Figure 5.3 shows the fitting grid to the bounding box. The top left corner of the bounding box and grid are overlapped at $(0, 0)$, their top right is at $(w, 0)$, the bottom left is at $(0, h)$, and the last corner of them are overlapped at (w, h). Another certain point is intersection of the middle vertical and the middle horizontal lines which is at $(\frac{w}{2}, \frac{h}{2})$. The coordinates of the other intersection points are related to the ratio of the cell height (c_h) over grid height (h) and the ratio of the cell width (c_w) over grid width (w). To make the dimensions of each cell equal to each other, y coordinate of the top horizontal line drawn with yellow is $0.25 * h$, y coordinate of the bottom horizontal line drawn with red is determined by $0.75 * h$. The left most and right most lines have the same relation to the w.

After the grid is fitted to the bounding box of the normalized shape, its boundary points are assigned to cells according to their spatial values. During this, weight is distributed among the cell that contains the point, and its neighbors according to bilinear interpolation to obtain a stable distribution. After the weights of the points are calculated, the tangent directions of the points of the normalized shape are computed by using median

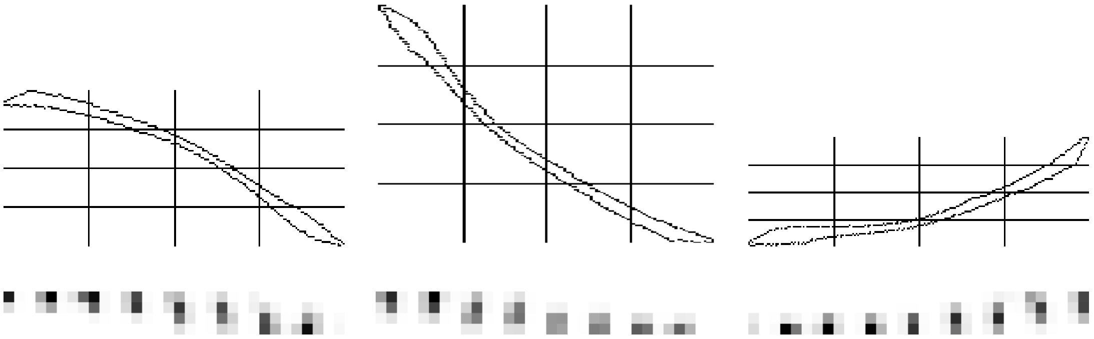


Figure 5.4. After grid is fitted the normalized versions of the shape, descriptor is computed by generating eight bin histogram for each cell. The final descriptor is obtained by concatenating those eight bin histograms. In the upper row, grid fitting process is shown. In the bottom row, the computed descriptors for each normalized version are shown. The descriptor representation is obtained by drawing white when the value of the corresponding element is zero and it is getting darker with respect to the increase of the value. In this representation, there are four rows which are corresponded to the rows of the grid. Moreover, there are 32 columns which are obtained by multiplying four and eight which are the number of cells of the grid in the horizontal direction and the number of bins that is generated for each cell of the grid respectively. So, this representation has 128 cells in total which is the dimension of the proposed descriptor.

differencing algorithm [25]. During the tangent direction computation, the parameter m is taken as two. And for each cell, an eight bin histogram is generated by combining the tangent directions and the weights of the points. After the histograms are generated, they are concatenated by starting from the histogram of the top left cell, to the bottom right cell to generate 128 dimensional descriptor. Then it is normalized with L2 normalization, and it becomes the final descriptor. In Figure 5.4, the normalized versions for the sample shape which is indicated 5.2 (a) and the computed descriptors for each normalized shape is shown. As is seen, the spatial information of the boundary points appears in the representation of the final descriptors. In the first two normalized versions, the shape is going from the top left cell to the bottom right cell and their descriptors is also going from the top left corner to the bottom right corner. On the other hand, in the last version, the shape points are lain from the top right to the bottom left corner so its descriptor shows up as black in the same diagonal of the descriptor representation.

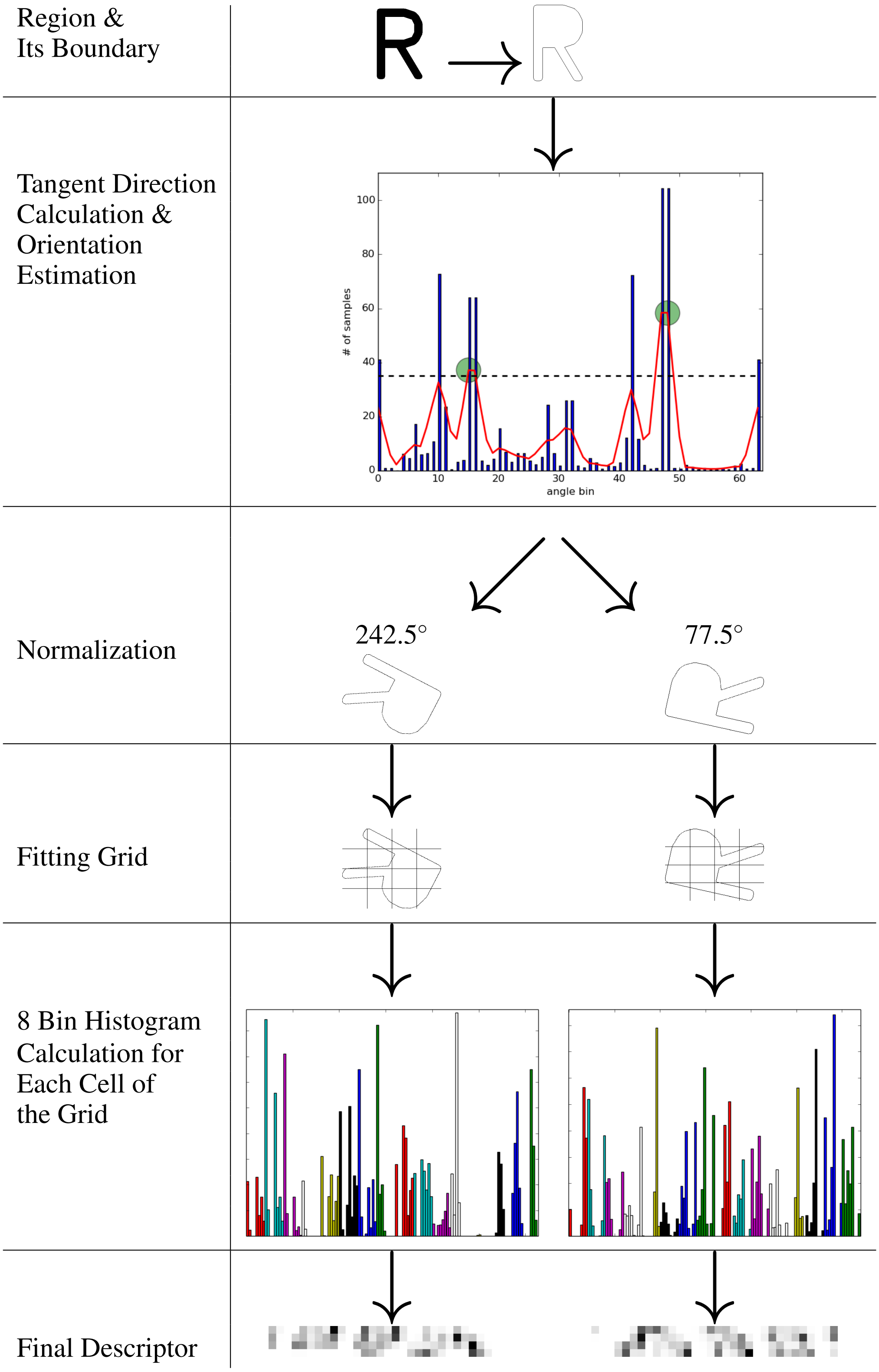


Figure 5.5. Descriptor Computation Summary

Figure 5.5 summarizes the whole descriptor computation process. After boundary is extracted, tangent direction for each boundary point is computed and a 64 bin histogram is generated from them. Then the curve that is formed by combining the top of the histogram bins is smoothed a few times. After smoothing, local maxima of the curve is marked as estimated orientations. Then according to the orientations, the shape is normalized and then a four by four grid fitted onto them. Afterwards, for each cell of the grid, an eight bin histogram is generated and it becomes to the final descriptor by concatenating them. In the figure, in order to exemplify, the character ”R” in upper case which is synthetically generated is used. In the second row, the histogram that is generated to estimate orientation is indicated. With the histogram, the smoothed curve and the local maxima is indicated as well. For the character, estimated orientations are 242.5°and 77.5°which are ordered with respect to the height of the peaks. Because there are two estimated orientations, the following processes perform two times. In the third row normalized versions of the character ”R” is indicated then a grid is fitted onto them. In the next row, histogram is obtained by generating an eight bin histogram for each cell and concatenating them. So in those histograms there are 128 bar and each color represents the histogram of a cell of the grid. For example first eight red bar is generated from the left top corner cell and next light blue eight bar is obtained from the one right cell of the top left corner. In the last row, histograms that are indicated in the previous step are represented like 2D code.

In order to reveal the effect of the tangent direction, another approach is designed. In this approach, descriptors are computed in the same way but gradient direction is used instead of tangent direction. Gradient direction for an image point i is

$$\theta_i = \arctan(\frac{g_{y_i}}{g_{x_i}}) \tag{5.3}$$

where g_{y_i} and g_{x_i} are the gradient of the same point in the y direction and in the x direction respectively. And they are calculated as

$$\begin{aligned} g_{x_i} &= I[x_i + 1, y_i] - I[x_i - 1, y_i] \\ g_{y_i} &= I[x_i, y_i + 1] - I[x_i, y_i - 1] \end{aligned} \tag{5.4}$$

So the same procedure is applied with an alteration in two steps. The alteration is using gradient direction in the calculation of the direction. And it is applied during the direction calculation for the shape and its normalized version. During the orientation estimation, gradient direction is calculated for each boundary point and the dominant direction is selected by making their histogram. In addition to orientation estimation, after the shape is normalized, the gradient direction of its points is calculated. And the descriptor of the shape is computed by combining the locations of the boundary points and their gradient direction. The performance of this approach is measured in addition to the proposed approach that uses the tangent direction.

In Figures 5.6 and 5.7, the orientation estimation and descriptor computation processes are indicated. In these figures, there are several examples and the proposed approach with tangent direction and gradient direction is compared. The samples are selected as three digits, four upper case characters, and three lower case characters because as mentioned in Section 5.1, characters are accepted as the main target. Sample characters that are shown in both figures are generated synthetically. They are binary images that their background are white and the characters are black. After they are generated, their boundary points are extracted by using find contour method of OpenCV.

Figure 5.6 shows sample characters, the estimated orientations, and normalized shapes for each sample. In the figure, orientation is estimated and shapes are normalized according to tangent direction and gradient direction separately. The estimated orientation with gradient direction is the same in general because the characters are generated as synthetic images that have white background and black characters. So mostly the gradient of the boundary points is the same value for different characters and it dominates and found as estimated orientation.

In Figure 5.7, the shape of sample characters, the fitting grid onto them, and their computed descriptors are shown. For visual simplicity, estimated orientation for each character is assumed that there is one estimated orientation for each shape and it is equal to zero, so orientation estimation step is skipped. Because the characters are generated synthetically, they reflect ideal cases. So the only difference between the character ”E and F” in upper case is the last horizontal line and it can be seen in the descriptors representation as well. While in the bottom part of the descriptor of the character ”E” there are non-zero values, the values of the descriptors of the character ”F” are zero at the same part. By comparing the descriptors of the characters, it is observed that while the non-zero values are distributed on the 2D code in tangent direction version, they are clustered into four in gradient direction version. The reason of this, the gradient direction of the boundary points are not distributed evenly because of the lack of texture on the sample characters.

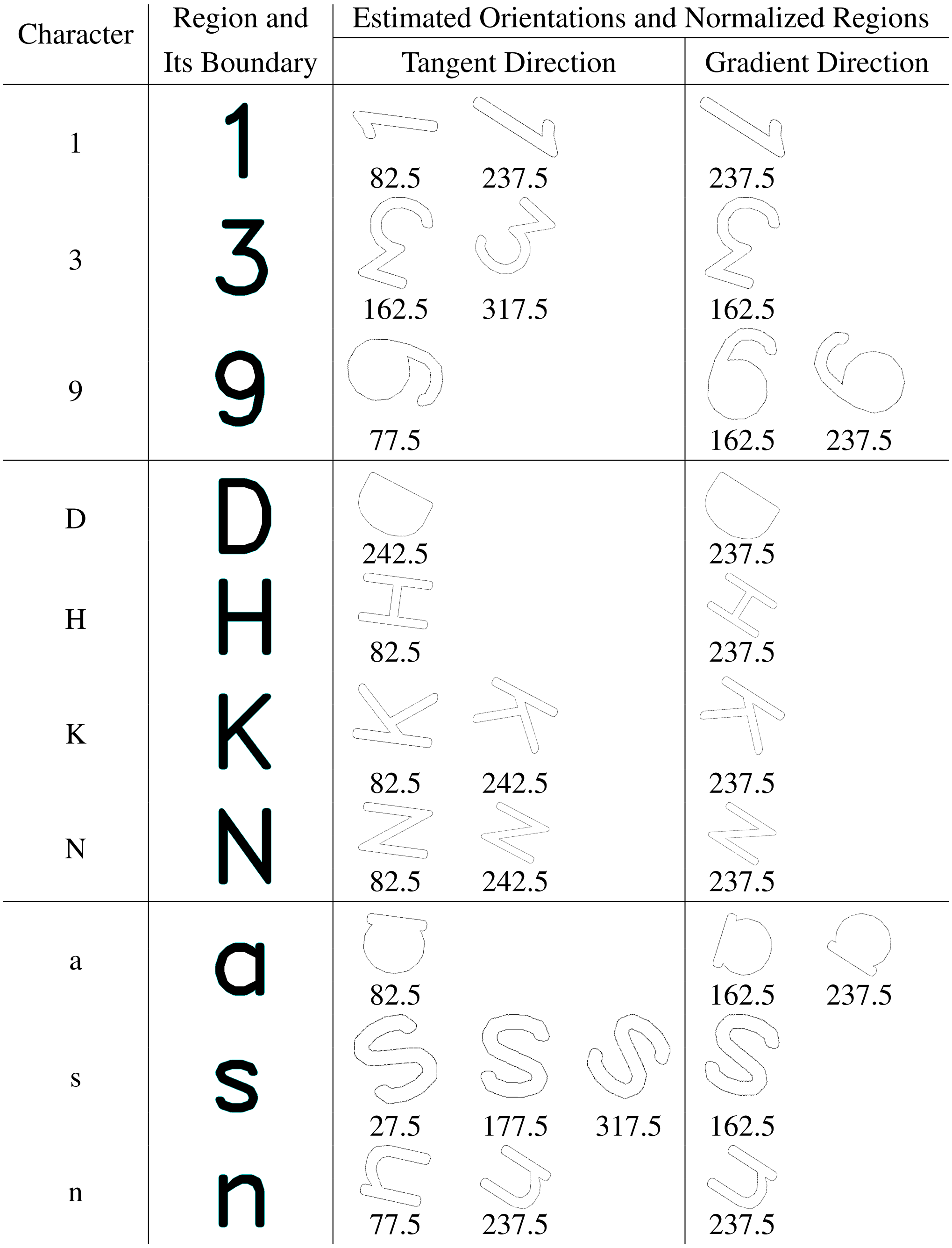


Figure 5.6. Estimated orientations are shown for each sample character. In the first column their ground truth information is given. In the second column, sample regions are indicated as binary image and their boundary is drawn with light blue. In the next column, estimated orientations and normalized shapes that are calculated by tangent direction indicated. And the last column shows the same but in the calculation gradient direction is used.

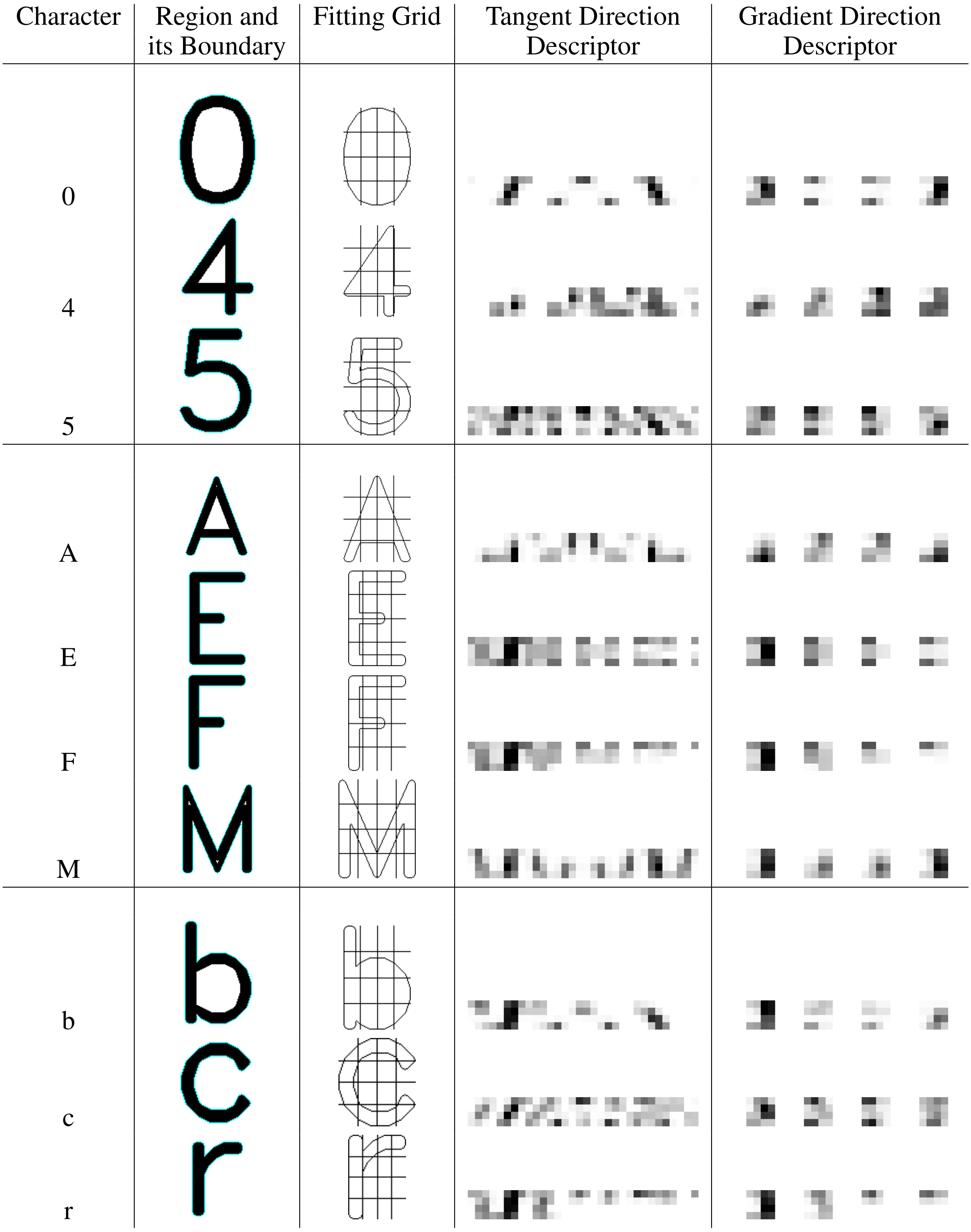


Figure 5.7. Ten different sample characters, the descriptor computation process, and their 128 dimensional descriptors are indicated. In the first column their ground truth information is given. In the second column, sample regions are indicated as binary image and their boundary is drawn with light blue. In the next column, fitting 4 x 4 grid is shown. In this figure orientation estimation is skipped in order to provide visual simplicity. In the last two columns, there are their final descriptors that are represented like 2D code.

5.3. Implementation Detail

In both orientation estimation and descriptor computation, the median differencing algorithm [25] is used to compute the tangent direction of each point located on the shape boundary. The algorithm requires a parameter m. It is related to the smoothness of the tangent direction and it defines how many point will be used to calculate the tangent direction of a point. In the orientation estimation, it is taken as three, so the median of six difference vector becomes the tangent direction. On the other hand, in the descriptor computation, it is taken as two, so the median of four difference vector is calculated to compute tangent direction. In other word, computed tangent directions in the descriptor computation part are less smoothed than tangent directions that are computed in the orientation estimation part. Because in the descriptor computation the details of the contour is more important and they are necessary to describe the shape well. On the other hand, in the orientation estimation, the details of the boundary are less important and necessary to find the dominant tangent direction. So ignoring these details is feasible during the orientation estimation.

As mentioned earlier, more than one orientation can be estimated for a shape. So more than one descriptor are computed because the shape is normalized according to the estimated orientation and descriptor is extracted from the normalized shapes. So for each shape, the number of computed descriptors is equal to the number of estimated orientations. In the matching to handle that when the distance is calculated between descriptors of a shape and the other shape, all combinations of them is calculated then minimum among them is accepted as the final distance between the descriptors. In the experiments, to match descriptors Euclidean distance is used as a metric and its calculation between two descriptor is shown in Figure 5.8.

Descriptor of a shape $d_1 : \{ d_{11}, d_{12}, \ldots , d_{1N} \}$

Descriptor of the other shape $d_2 : \{ d_{21}, d_{22}, \ldots , d_{2M} \}$

$$\text{euc\_dist}\,(d1 , d2) = \min \begin{pmatrix} \text{euc\_dist}\,(d_{11},d_{21}) \\ \text{euc\_dist}\,(d_{11},d_{22}) \\ \ldots \\ \text{euc\_dist}\,(d_{11},d_{2M}) \\ \text{euc\_dist}\,(d_{12},d_{21}) \\ \text{euc\_dist}\,(d_{12},d_{22}) \\ \ldots \\ \text{euc\_dist}\,(d_{12},d_{2M}) \\ \ldots \\ \ldots \\ \text{euc\_dist}\,(d_{1N},d_{21}) \\ \text{euc\_dist}\,(d_{1N},d_{22}) \\ \ldots \\ \text{euc\_dist}\,(d_{1N},d_{2M}) \end{pmatrix}$$

Figure 5.8. Euclidean distance calculation between two descriptors. N and M are the numbers of estimated orientation for the first and the second shape. To compute the Euclidean distance between them, NxM times Euclidean distance should be calculated and the minimum of them is the final Euclidean distance between the descriptors of the shapes.

5.4. Experiments

The performance of the proposed approach and its second version is evaluated on five datasets that are listed below:

- Modified ICDAR 2013 dataset
- De Campos et al. Chars74k dataset [7]
- Mpeg - 7 dataset
- Kimia's - 99 dataset
- ETH - 80 dataset

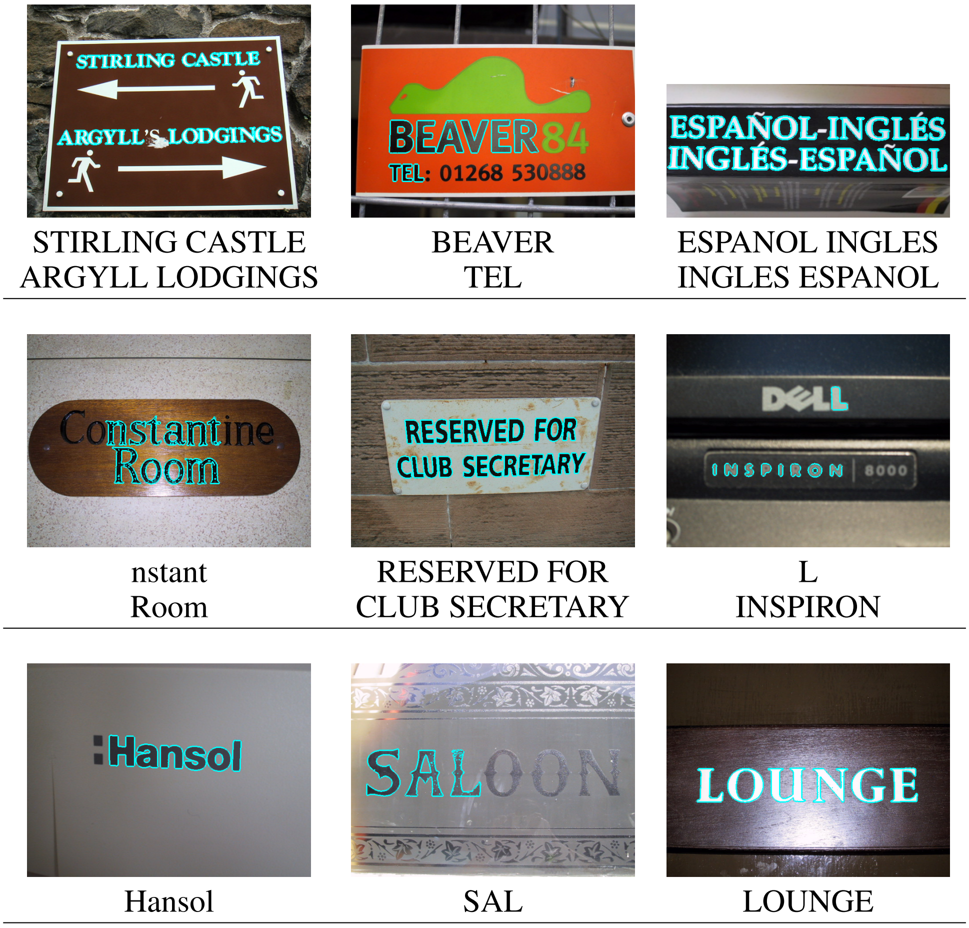
In the first two datasets there are characters and the others are shape datasets. In order to perform experiments on these datasets, the contour of each shape should be found because the proposed approach focuses on the shape classification from their contour particularly, so the algorithm requires the contour points of each shape. For each dataset, different processes are applied to detect objects and to find their contours.

Modified ICDAR 2013 dataset is created by detecting the characters of the training images of ICDAR 2013 which is one of the standard character datasets. In the training group, there are 4419 characters and their bounding boxes are given as ground truth. When the modified ICDAR 2013 dataset is created, MSER [24] are detected on each training word images, but the algorithm detects regions that are non characters as well. So they are eliminated according to the bounding boxes of the characters. After detection and elimination, 3141 characters of 4419 are found as extremal regions. So modified ICDAR 2013 is a subset of the original ICDAR 2013. For each detected character, the extended boundary and the ellipse representation of the extremal regions are retrieved. The reason of the using MSER when characters are detected, we want to compare the performance of the proposed method with the other descriptors. Because the dataset is created by us, there are no evaluation results with the other state-of-art descriptors, so we need to measure their performance besides the proposed approaches. To do that easily, characters are detected through extremal regions and their descriptors are computed by using the executable of Mikolajczyk et al. [28] The executable takes the extremal regions as input and it computes ten different local descriptors on MSER. And they are listed below:

- Scalable Invariant Feature Transform (SIFT) [22]
- Gradient Location-orientation Histogram (GLOH) [28]
- Shape Context (SC) [2]
- Principal Component Analysis SIFT (PCA-SIFT) [12]
- Spin Images (SPIN) [16]
- Steerable Filters (JLA) [8]
- Differential Invariants (KOEN) [14]
- Complex Filters (CF) [33]
- Moment Invariants (MOM) [45]
- Cross-correlation (CC) [28]

After characters are detected by extracting the extremal regions, the total number of detected individual characters is 3141. This dataset contains only English characters in both upper and lower case and digits and punctuations are not included. Among 3141 characters there is no homogeneous distribution. While there is no sample of the character ”J and Z” in lower case and the character ”Q” in upper case, the number of sample of the character ”E” in both cases are more than 180 and also there are more than 120 sample of the character ”A,O and R” in both cases. In Figure 5.9, selected characters are exemplified. Under each image, characters that are detected are written. As it is seen, while sometimes the every character of the word is detected, sometimes some characters are missed.

Chars74k character dataset is used to compare the performance of the proposed algorithms. In the dataset, there are English and Kannada characters. Only English characters are used in the experiment and it includes characters in upper and lower cases and digits. So there are 62 classes. The dataset has three group which are called as fonts, handwritten, and natural images. In the first group, there are 62992 images of characters that are created synthetically and there are 1016 different sample for each class. All of them are binary images with white background and black character. In Figure 5.10, sample characters from the font group are indicated. In the other group, handwritten characters exist. For each class, there are 55 samples and they are also binary images like the first



STIRLING CASTLE ARGYLL LODGINGS

BEAVER TEL

ESPANOL INGLES INGLES ESPANOL

nstant Room

RESERVED FOR CLUB SECRETARY

L INSPIRON

Hansol

SAL

LOUNGE

Figure 5.9. Modified ICDAR 2013 dataset. MSER are detected and they are eliminated according to the ground truth character locations. Cyan regions shows the boundary of remaining MSER. Below charters shows the only detected characters from images, so the remaining characters of words are not involved in Modified ICDAR 2013 dataset.

group. Figure 5.11 shows the example handwritten characters. In the last group, there are 7705 samples and they are a patch of images and inside the patch, there is only one character mostly. Unlike the first two group, the last group does not have a homogeneous distribution between the classes. In Figure 5.12, there are some examples from this group.

For this dataset, only the performance of the proposed methods is measured. Because it is compared with the performance results of the other state-of-art descriptors that are stated on published works. So instead of using a region detector like MSER to detect characters, their contours are found and extracted directly. For the first two group, the contour of characters is found and extracted by using find contour method of OpenCV.

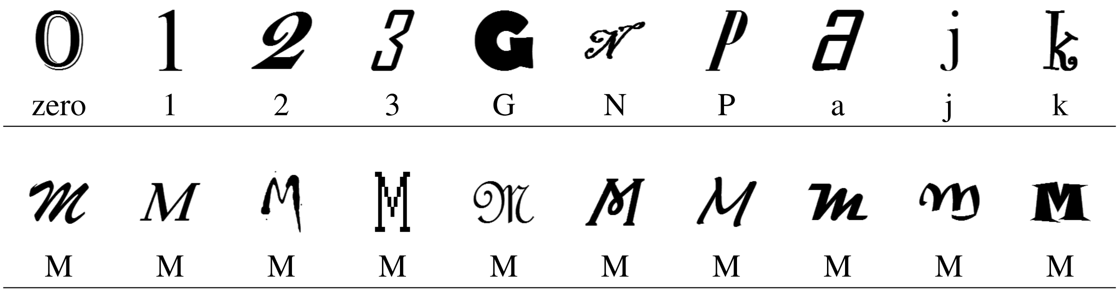


Figure 5.10. Chars74k dataset group font. The first row shows 10 samples from different classes and the second row shows the 10 different sample from a class.

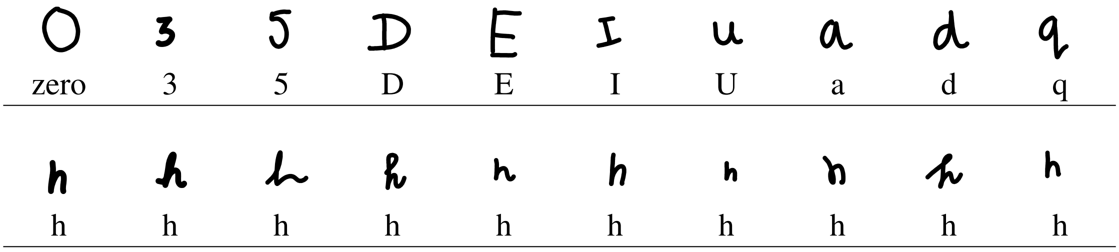


Figure 5.11. Chars74k dataset group handwritten. The first row shows 10 samples from different classes and the second row shows the 10 different sample from a class.

This method finds the contour of silhouettes of characters and boundary points of the contours are retrieved as a chain in counter clockwise direction and without elimination any point that is located on the contour. So its output is suitable for the proposed approach. Except for eliminating the boundary points of holes inside the region, the other features that are required for the descriptor computation are provided by this method. For the last group to detect characters and retrieve their contours, more sophisticated preprocess is required. First, the intensity histogram of the grayscale image patches that contains individual characters is computed, then the highest and the second highest peaks of the histogram are found and according to their middle point single thresholding is applied. Next closing operator which is one of the morphological operators is applied to eliminate holes that are caused by the thresholding. The boundaries of some characters in the dataset are touching the frame of the patch, and this causes a difficulty during the char-

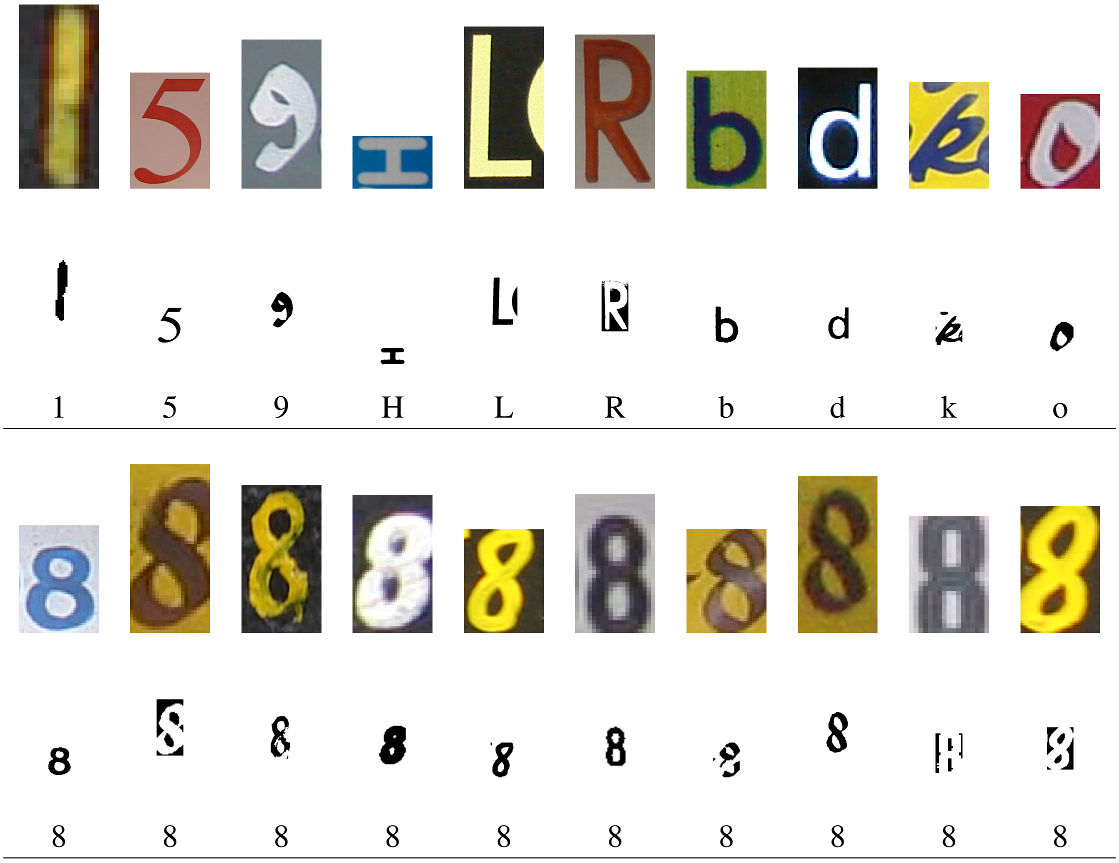


Figure 5.12. Chars74k dataset group natural images. The first row of upper part shows the natural images of 10 samples from different classes and the second row shows their binary images that are created after preprocess. The first row of the lower part shows the natural images of 10 different sample from a class and the second row shows their binary images.

acter detection. To avoid that, all images are enlarged to create a margin between the character and the patch frame. After these are performed, natural images became like the image patches of the first and the second group of the dataset. So the find contour method of OpenCV is invoked to find and extract the boundary of the contour. In Figure 5.12, example characters from the natural images group and their binary images are shown.

Mpeg - 7 dataset has 70 objects and 20 different images for each of them so there are 1400 images. All of them are silhouettes of the corresponding object. Most of them are binary images with black background and white silhouette. In Figure 5.13, there are one sample image of the different objects. Figure 5.14 shows all 20 samples of an b-object called ”camel”. Kimia’s - 99 dataset has 99 images from 9 different classes with 11

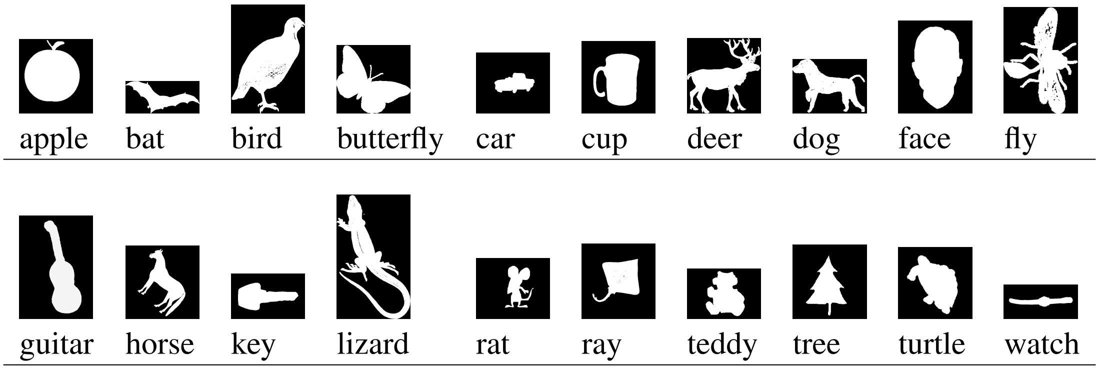


Figure 5.13. Mpeg - 7 dataset. Figure shows 20 samples from different classes.

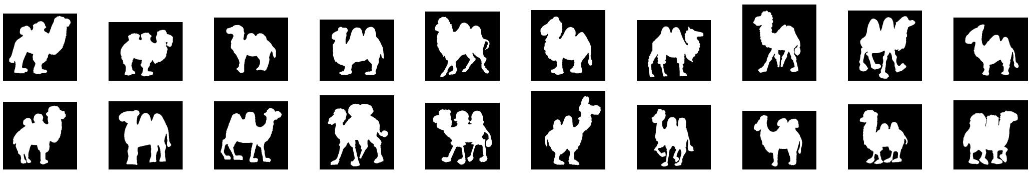


Figure 5.14. Mpeg - 7 dataset. Figure shows 20 samples from the same class which is ”camel” object.

samples. Images contain only silhouette of objects like the Mpeg -7 dataset. Figure 5.15 shows one sample image for each object in the first row and all images of an object in the second row. For images of both datasets find contour method of OpenCV is sufficient to find and extract the boundaries. ETH - 80 dataset has 8 categories and 10 objects for each category with 41 different images so there are 3280 images. In Figure 5.16, one sample images for each category is indicated in the top row. Furthermore, the first four of them is zoomed and their contour images are indicated in the second part. Figure 5.17 shows one sample image for each class of a category called ”dog” in the top row. In the second part, there are the zoomed versions of the first four object and their contours. Unlike Mpeg - 7 and Kimia’s - 99 images of ETH - 80 dataset contains only the contour of the objects. So there is no need to find the boundary and only black collecting the points is enough to retrieve the boundary.

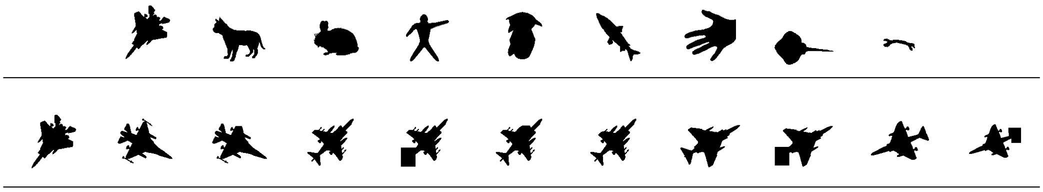


Figure 5.15. Kimia's - 99 dataset. For each class a sample is shown in the first row. In the second row every sample of a class is shown.

5.4.1. Setup

In order to perform experiments, matching score is measured and it is called as recognition rate as well. Protocols that are used to measure matching score for each dataset have some variety. Using nearest neighbor classification method is used to find the matching between descriptors for every dataset. The similarity between descriptors is measured by Euclidean distance for the proposed approaches. Calculating the Euclidean distance between two descriptor is mentioned in Section 5.3.

For the Modified ICDAR 2013 dataset, 80% of the whole data is separated as train set and the other shapes become test set. Euclidean distance between descriptors of each shape in the test set and each shape of the train set is calculated. To classify each test shape, its neighbors are found for k = 1. Then the number of correct classified test shapes are counted and matching score is calculated by dividing the number of the correct classified with the number of the test shapes. When the data is separated as train and test, they are selected randomly so the whole process is repeated 20 times. For a fair comparison, this experimental setup is applied for the state-of-art descriptors in addition to the proposed approaches and they are listed below:

- Cross-correlation (CC) [28]
- Complex Filters (CF) [33]
- Gradient Location-orientation Histogram (GLOH) [28]
- Steerable Filters (JLA) [8]
- Differential Invariants (KOEN) [14]

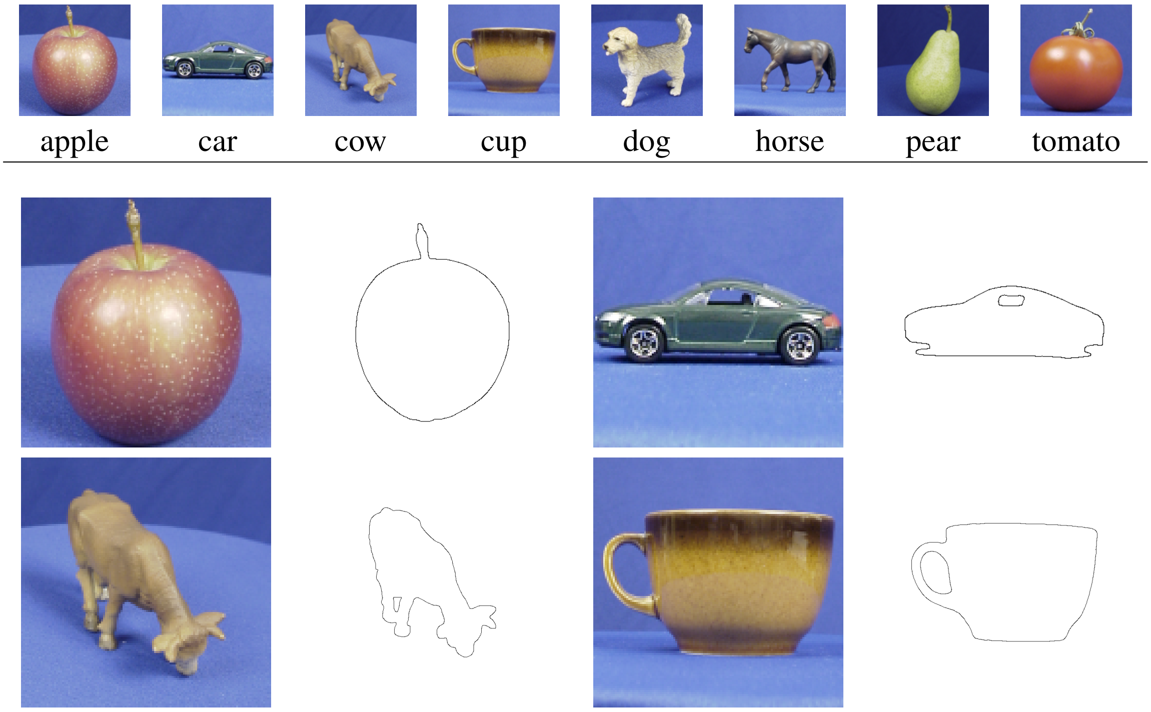


Figure 5.16. ETH - 80 dataset. The top row shows a sample of each category in color and first four object's contour representations are shown in the second part.

- Moment Invariants (MOM) [45]
- Proposed Approach - Gradient Direction (Ours-Gradient)
- Proposed Approach - Tangent Direction (Ours-Tangent)
- Pricipal Component Analysis Scalable Invariant Feature Transform (PCA-SIFT) [12]
- Shape Context (SC) [2]
- Scalable Invariant Feature Transform (SIFT) [22]
- Spin Images (SPIN) [16]

In the Chars74k the experimental setup that is proposed by [7] is followed. As mentioned earlier the dataset has three group: font, handwritten, and natural images. For all of them, 15 randomly selected samples from each class are separated as the test set and train set has 1001 samples in the first group, 40 samples in the second group and for the last group the number of sample of the train set changes because the natural images

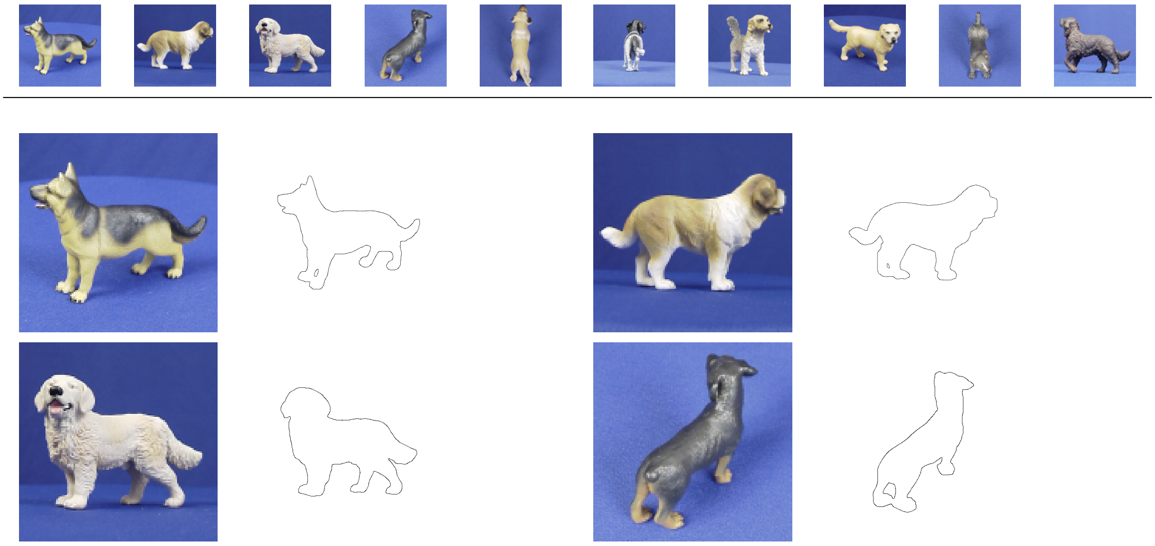


Figure 5.17. ETH - 80 dataset. The top row shows every classes of a category in color and first four object's contour representations are shown in the second part.

group has heterogeneous number of sample for each class unlike the first and the second group. After test and train sets are defined, each shape is classified according to its nearest neighbor which is found by comparing the Euclidean distance between the descriptor of the test shape and the descriptor of each shape of the train set. After classification, the number of correctly classified sample over the number of test set gives the matching score. Because the train and test sets are separated randomly, this process is repeated many times like the previous experimental setup. Only the performance of the proposed methods is measured and the proposed methods are compared the stated results of the state-of-art descriptors in [7] directly. They are listed below:

- Geometric Blur (GB) [3]
- Maximum Response of Filter (MR8) [46]
- Patch Descriptor [47]
- Shape Context (SC) [2]
- Scalable Invariant Feature Transform (SIFT) [22]
- Spin Images (SPIN) [16]

In the literature, Bullseye test [26] is mostly preferred experimental setup to analyze the performance of a shape descriptor by using the Mpeg - 7 dataset. In Bullseye test, every shape is defined as an element of the test set and they are matched one by one with the each shape of the whole original dataset. So in the matching Euclidean distance between a test shape and each shape of the dataset which includes test shape as well. Classification is done by nearest neighbor and for each test shape, its top 40 near neighbors are found. Among them, correct classes are counted and for a shape, the number of correct class among its top 40 near neighbors can be 20 at most. After counting the correct neighbors for each shape, the matching score is calculated by dividing this number with the best matching score. It is 28000 and calculated by $1400x20 = 28000$ where 1400 is the number of samples in the dataset and for each of them there can be at most 20 neighbors from the corresponding class. Like Chars74k, only the performance of the proposed methods is measured and their performance is compared with the stated results of descriptors given below:

- Curvature Scale Space (CSS) [30]
- Contour Points Distribution Histogram & Earth Mover's Distance (CPDH+EMD) [37]
- Curve Edit [35]
- Distance Set [9]
- Generative Models [41]
- Inner Distance Shape Context & DP (IDSC+DP) [21]
- Morphological CSS (MCSS) [10]
- Multidimensional Scaling & SC & Dynamic Programming (MDS+SC+DP) [21]
- Shape Context (SC) [2]
- Skeletal Context (SCC) [48]
- Visual Parts [15]

For the Kimia's - 99 dataset, another well-known classification setup is used which is leave one out nearest neighbor for k = 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10 [21]. k is selected as 10 at most because for each test shape in the train set there are at most 10 samples from the same class. So each shape descriptor is matched by calculating the distance between the shape descriptor and the descriptor of the other shapes. Then correctly matched shapes are counted. So for each k the number of matched shape can be reached to the 99. For example, during k = 10, a shape is matched with the other shapes and if among its nearest 10 neighbors correct class dominates the other classes, it will accept as correct match and the number of correct matches is increased. For this dataset, like Mpeg - 7 only the performance of the proposed approaches is measured and they are compared with the stated results of the descriptors from the literature directly and they are given below:

- Contour Points Distribution Histogram & Earth Mover's Distance (CPDH+EMD) [37]
- Generative Models [41]
- Inner Distance Shape Context & DP (IDSC+DP) [21]
- Multidimensional Scaling & SC & Dynamic Programming (MDS+SC+DP) [21]
- Shape Context (SC) [2]
- Shock Edit [36]

In the ETH - 80 dataset, there are 8 different classes which represent categories, for each of them there are 10 subclasses which are different objects from the same category and for each class, there are 41 images so in total there are 3280 images. Like Kimia's - 99 leave one out nearest neighbor is selected as the experimental setup [21]. For each time, a subclass is determined as a test object, for its each image the nearest neighbor is found by calculating the distance with the images of the other subclasses(79 class). So to classify a subclass minimum $41x79x41 = 132799$ distance should be calculated. After their nearest neighbor is found for each image of the subclass, the number of correct nearest neighbor is counted and it is divided by the number of images of the subclass. This gives the recognition rate of the subclass. After it is computed for each subclass, the recognition rate of a class is calculated by taking their average. After the recognition rate of each class is calculated, their average gives the overall recognition rate. Like Kimia's - 99, only the experimental setup is applied only to the proposed approaches and their recognition rates are compared with the results which are stated in published works directly and the descriptors that are used in comparison are listed below:

- Color Histogram [40]
- Texture Histogram Rotation Variant (D_xD_y) [34]
- IDSC+DP [21]
- Texture Histogram Rotation Invariant (Mag-Lap) [34]
- MDS+SC+DP [21]
- Principal Component Analysis (PCA) Gray [18]
- PCA Masks [18]
- SC Greedy [2]
- SC+DP [2]

Table 5.1. Summary of datasets and experimental setups.

	Modified ICDAR 2013	**Chars74k Font**	**Chars74k Handwritten**	**Chars74k Natural Images**
# of samples	3141	62992	3410	930
# of classes	49	62	62	62
# of test samples	629	930	930	930
# of train samples	2512	62062	2480	6775
classification method	kNN (1)	kNN (1)	kNN (1)	kNN (1)

	Mpeg - 7	**Kimia's - 99**	**ETH - 80**
# of samples	1400	99	3280
# of classes	70	9	80
# of test samples	1400	99	3280
# of train samples	1400	99	3280
classification method	kNN (top 40)	kNN (1, ..., 10)	kNN (1)

Table 5.1 shows the summary of datasets that are used in the experiments and their experimental setups. Each dataset is explained with its size of samples, classes, test

samples, and train samples. Moreover, classification method is indicated for each of them. Except for Mpeg - 7 and Kimia's - 99 the nearest neighbor for k = 1 method is used for classification.

5.4.2. Results

Table 5.2. The matching score of the descriptors that is computed on characters of Modified ICDAR 2013 dataset. Characters are matched according to the nearest neighbor classifier. The matching score is measured 20 times and the average and standard deviation of the matching scores are calculated and stated.

Algorithm	**k = 1 (%)**
CC [28]	79.44 ± 1.21
CF [33]	76.05 ± 2.24
GLOH [28]	81.53 ± 1.51
JLA [8]	74.28 ± 2.70
KOEN [14]	66.82 ± 2.44
MOM [45]	76.26 ± 2.22
Ours-Gradient	75.04 ± 1.43
Ours-Tangent	$\mathbf{88.75 \pm 1.21}$
PCA-SIFT [12]	77.30 ± 2.23
SC [2]	78.81 ± 2.23
SIFT [22]	81.16 ± 2.57
SPIN [16]	56.15 ± 3.23

Table 5.2 shows the matching score of the characters of Modified ICDAR 2013 dataset. When the matching score of descriptors except for the proposed approaches is calculated, the descriptors are computed with different scales that are 0.25, 0.3, 0.35, 0.375, 0.5, 0.625, 0.75, 0.875, and 1.0. And the results for those scales is indicated in Table B.1 in Appendix B. And the best matching score for each descriptor is selected and stated in Table 5.2. According to the results, the matching score of the proposed approach with tangent direction is the best score and it is 88.75%. This approach is followed by GLOH whose matching score is 81.53%. Moreover, the score of the other proposed

approach is 75.04%. And the difference between the score of the proposed approaches is 14%.

Table 5.3. The matching score of the descriptors that is computed on images of characters from Chars74k dataset. The descriptors of characters are classified by the nearest neighbor. And for the font group the measurement is repeated ten times, for the handwritten characters it is measured five times. The average and standard deviation of measurements are stated. For the characters that are extracted from the natural images, the experiment is performed once.

Algorithm	**Fonts (%)**	**Handwritten (%)**	**Natural Images (%)**
GB [3]	69.71 ± 0.64	65.40 ± 0.58	47.09
MR8 [46]	30.71 ± 0.67	25.33 ± 0.63	10.43
Ours-Gradient	85.78 ± 0.73	75.55 ± 1.08	49.28
Ours-Tangent	$\mathbf{87.83 \pm 1.37}$	$\mathbf{77.18 \pm 0.50}$	**56.86**
Patches [47]	44.93 ± 0.65	69.41 ± 0.72	21.40
SC [2]	64.83 ± 0.60	67.57 ± 1.40	34.41
SIFT [22]	46.94 ± 0.71	44.16 ± 0.79	20.75
SPIN [16]	28.75 ± 0.76	26.32 ± 0.42	11.83
ABBYY	66.05 ± 0.00	-	30.77

In Table 5.3, the matching score of the descriptors that is computed on the images of the characters from Chars74k dataset. In the first result column, the scores for font group of the dataset is indicated. And they are calculated by averaging the results that are measured by repeating the experiment ten times. The next column shows the results of the handwritten group. And they are calculated by taking the average of the results that are obtained by repeating the experiment five times. In the last column, the results of natural images indicated. And the experiment is performed once for the natural images. The results except for the proposed approaches are taken from [7]. The table shows that the performance of the proposed approaches are close to each other. Moreover, the approach with tangent direction is the best in each group and its scores are 87.83%, 77.18%, and 56.86%. The scores of the approach with gradient direction are 85.78%, 75.55%, and 49.28%. Furthermore, the followed approach is GB for the font characters with 69.71% score, Patches for the handwritten characters with 69.41%, and GB for the natural images with 47.09% score. And the gap between the followed approaches and the proposed approach with tangent direction is almost 18% for the first group, almost 8% for the second group, and almost 10% for the last group.

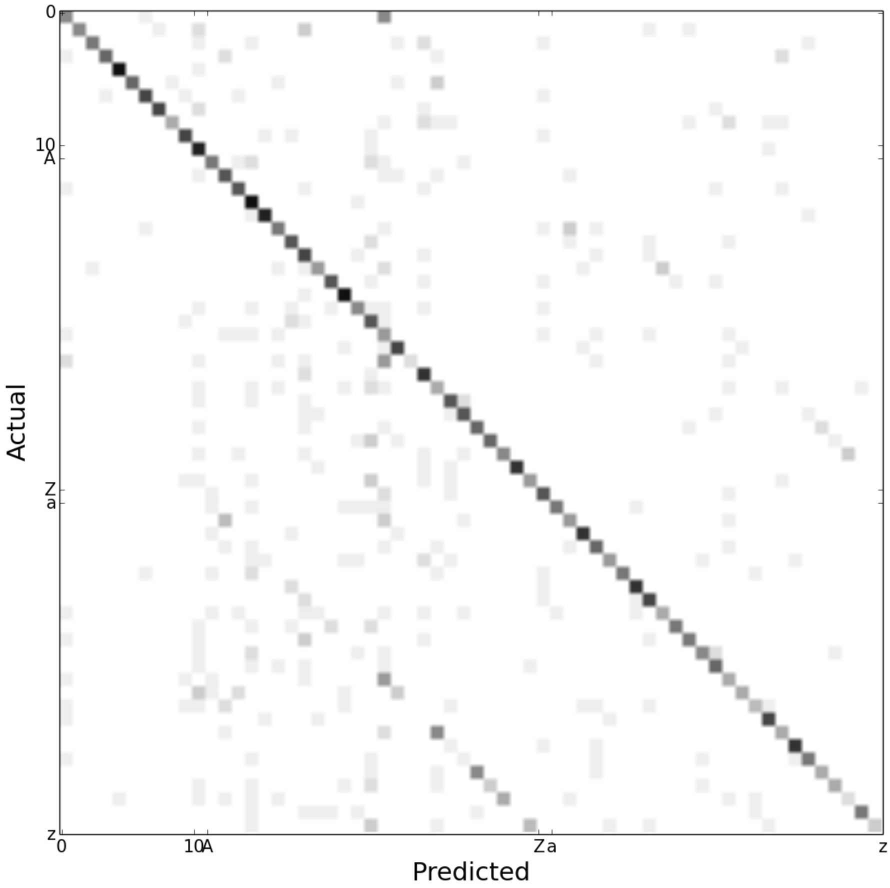


Figure 5.18. Confusion matrix for natural images group of the Chars74k dataset. Descriptors are computed by the proposed approach when their direction are **tangent direction**. In the figure, y-axis shows the actual values of the samples and x-axis shows the predicted values for the samples. So samples that are shown in the diagonal axis of the matrix are classified correctly.

In Figures 5.18 and 5.19, confusion matrix for the natural images group of Chars74k dataset is shown. Y-axis of the figures shows the actual values for each sample that is known as ground truth. And x-axis shows the predicted values for each sample. So for a sample, if it is seen in the diagonal, its actual value is the same as its predicted. That means this sample classified correctly. A matrix entry can be 15 as maximum because it is the size of test samples of each character. And if an entry is equal to the 15, it is

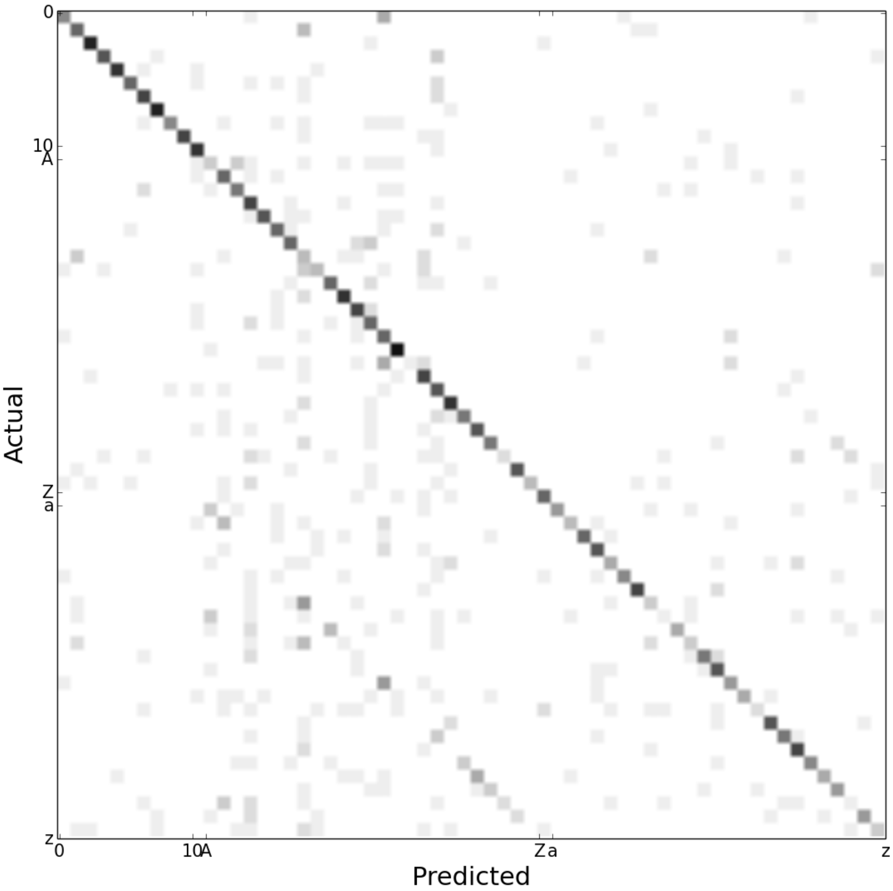


Figure 5.19. Confusion matrix for natural images group of the Chars74k dataset. Descriptors are computed by the proposed approach when their direction are **gradient direction**. In the figure, y-axis shows the actual values of the samples and x-axis shows the predicted values for the samples. So samples that are shown in the diagonal axis of the matrix are classified correctly.

drawn with black and the brightness is increased with respect to the value of entry. When the entry value reaches to zero, its brightness reaches to top and it is seen as white in the figures. Besides confusion matrices, top 10 misclassified test samples are also indicated in Figures 5.20 and 5.21. Those misclassified samples correspond to the top 10 darkest points except for the main diagonal in the confusion matrices.

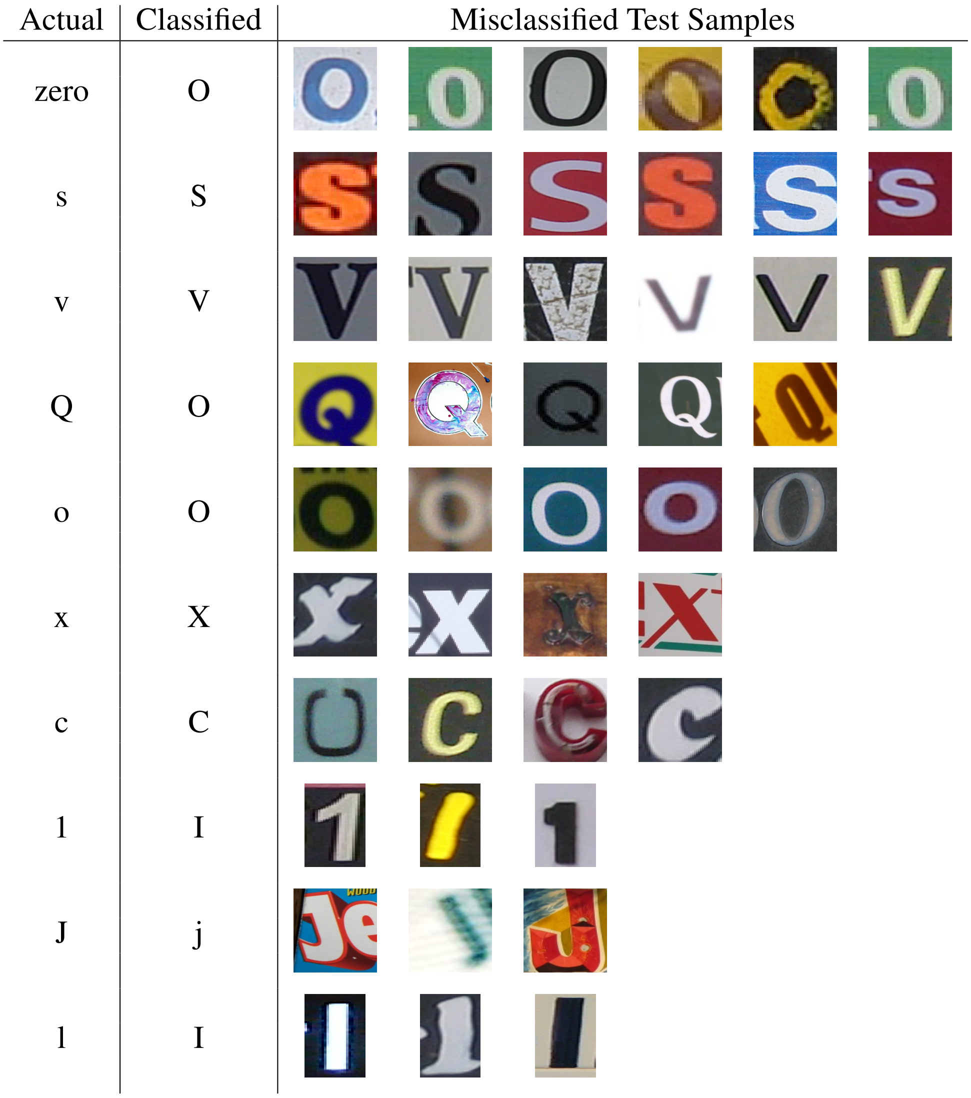


Figure 5.20. This figure shows the top 10 misclassified test samples in the classification with the proposed approach with **tangent direction**. In this classification, six zero, ”S” in lower case, and ”V” in lower case are classified as ”O” in upper case, ”S” in upper case, and ”V” in upper case respectively. Furthermore, misclassified samples for ”Q” in upper case, ”O” in lower case, ”X” in lower case, ”C” in lower case, ”1”, ”J” in uppercase, and ”L” in lower case are indicated.

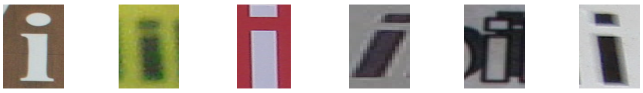
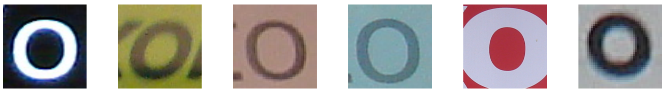
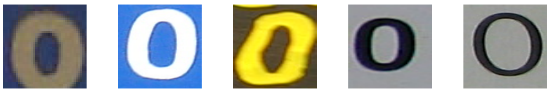
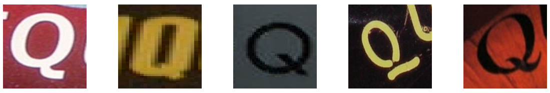
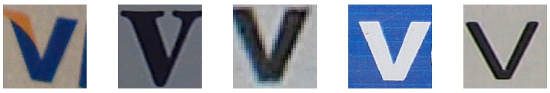
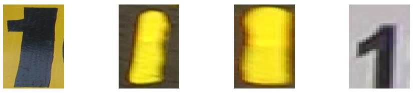
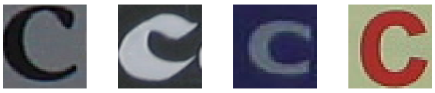
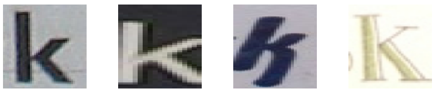
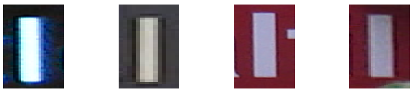
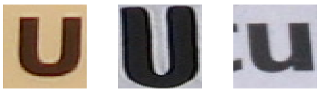
Actual	Classified	Misclassified Test Samples
i	I	
o	O	
zero	O	
Q	O	
v	V	
1	I	
c	C	
k	K	
l	I	
u	U	

Figure 5.21. This figure shows the top 10 misclassified test samples in the classification with the proposed approach with **gradient direction**. In this classification, six ”I” in lower case and ”O” in lower case are classified as ”I” in upper case and ”O” in upper case respectively. Furthermore, misclassified samples for zero, ”Q” in upper case, ”V” in lower case, ”1”, ”C” in lower case, ”K” in lower case, ”L” in lower case, and ”U” in lower case are indicated.

Table 5.4. The matching score of the descriptors that is computed on images of objects of Mpeg - 7 dataset. Bullseye test is applied to measure the score.

Algorithm	**Retrieval Rate (%)**
CSS [30]	75.44
CPDH + EMD [37]	76.56
Curve Edit [35]	78.16
Distance Set [9]	78.38
Generative Models [41]	80.03
IDSC + DP [21]	**85.40**
MCSS [10]	78.80
MDS + SC + DP [21]	84.35
Ours-Gradient	41.85
Ours-Tangent	72.55
SC + TPS [2]	76.51
SCC [48]	79.92
Visual Parts [15]	76.45

Table 5.4 shows the matching score of the descriptors that is computed on images of objects of Mpeg - 7 dataset. These scores are measured by applying the bullseye test which is mentioned in subsection 5.4.1. In the experiments we measure the bullseye score for only the proposed approaches and the other scores are used from the published works. IDSC + DP achieves the best performance with 85.40. The score of the proposed approach with tangent direction is 72.55% which is 13% lower than the best score. The score of the other proposed approach is 41.85% which is lower than the half of the best score.

In Table 5.5, the matching score of the descriptors that is computed on images of the objects of Kimia's - 99. In this experiment, leave one out nearest neighbor classifier is applied and like the Mpeg -7 the performance of the proposed approaches is measured only. The approach with tangent direction is the best except for when k = 6. With respect to the overall performance, its score is 981 that means only 9 images of 990 are classified incorrect. So the percentage of this score is equal to 99.09%. It is followed by MDS + SC + DP and the difference between their performance is 17. So the ratio of the incorrectly classified samples of the followed approach is three times higher than the same ratio of the proposed approach. And when the approach is applied with gradient direction, its

Table 5.5. The matching score of the descriptors that is computed on images of objects of Kimia's - 99 dataset. In this experiment leave one out nearest neighbor is applied. Classification is done for k = 1, ..., 10.

Algorithm	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**	**Total**
CPDH + EMD [37]	96	94	94	87	88	82	80	70	62	55	808
Generative Models [41]	**99**	97	**99**	**98**	96	96	94	83	75	48	885
IDSC + DP [21]	**99**	**99**	**99**	**98**	**98**	97	**97**	**98**	94	79	958
MDS + SC + DP [21]	**99**	98	98	**98**	97	**99**	**97**	96	**97**	85	964
Ours-Gradient	91	91	83	78	75	75	73	75	74	72	787
Ours-Tangent	**99**	**99**	**99**	**98**	**98**	98	**97**	**98**	**97**	**98**	**981**
Shape Context [2]	97	91	88	85	84	77	75	66	56	37	756
Shock Edit [36]	**99**	**99**	**99**	**98**	**98**	97	96	95	93	82	956

score decreases to 787. So 200 more images are classified incorrectly.

Table 5.6 shows the matching score of the descriptors that is computed on images of ETH - 80 dataset. The same experimental setup as Kimia's - 99 is applied for only the proposed approach and the score of the other approaches are used from the published works. In the publication that proposes IDSC + DP and MDS + SC + DP, the matching scores for each object are not stated separately so the corresponding entries of the table are remained empty. The best score is achieved by PCA Gray for apple object and the scores of the proposed approaches are almost 10% lower than the best score. For car object, PCA Masks classify all objects correctly and the proposed approaches are also close to 100%. For object cow, the best score is achieved by Mag - Lap and it is 94.4% and almost 20% higher than the proposed approach with gradient direction and almost 5% higher than the proposed approach with tangent direction. For the object cup, SC Greedy is the best-performed descriptor and its score is close to the proposed approach with tangent direction and almost 10% higher than the proposed approach with gradient direction. In the images of the object dog, SC + DP reaches the highest score which is 82.9%. The score of the proposed approach with tangent direction is almost 5% lower than the highest and the score of the proposed approach with gradient direction is half of the highest score. For horse object, the proposed approach with tangent direction has the highest score which is 86.3%. The score of the followed approaches is SC Greedy and SC + DP with the same score which is 84.6%. The score of the other proposed approach is almost 60%. For the object pear, PCA Gray has the highest score like the object apple

Table 5.6. The matching score of the descriptors that is computed on images of objects of ETH - 80 dataset. Images of objects are classified with their nearest neighbor. During the matching, for each time samples of a subclass is assigned as test samples and the others are assigned as train samples. And each test sample is classified as the class of the nearest neighbor. The matching score is the ratio of the correct classified sample over the number of test sample.

Algorithm	**Recognition Rate (%)**								
Categories:	**apple**	**car**	**cow**	**cup**	**dog**	**horse**	**pear**	**tomato**	**Avg**
Color Histogram [40]	57.6	62.9	86.6	79.8	34.6	32.7	66.1	**98.5**	64.9
D_xD_y [34]	85.4	98.3	82.7	66.1	62.4	58.8	90.0	94.6	79.8
IDSC + DP [21]	-	-	-	-	-	-	-	-	**88.1**
Mag - Lap [34]	80.2	77.6	**94.4**	77.8	74.4	71.0	85.4	97.1	82.2
MDS + SC + DP [21]	-	-	-	-	-	-	-	-	86.8
Ours-Gradient	78.5	99.0	73.9	90.5	43.7	59.8	93.2	90.2	78.6
Ours-Tangent	76.1	99.8	88.5	99.3	76.3	**86.3**	90.2	64.1	85.1
PCA Gray [18]	**88.3**	97.1	62.4	96.1	66.3	77.3	**99.8**	76.6	83.0
PCA Masks [18]	78.8	**100.**	75.1	96.1	72.2	77.8	99.5	67.8	83.4
SC Greedy [2]	77.1	99.5	86.8	**99.8**	82.0	84.6	90.7	70.7	86.4
SC + DP [2]	76.3	100.	86.3	99.0	**82.9**	84.6	91.7	70.2	86.4

and it is almost 10% higher than the proposed approaches. For the last object called tomato, the highest score is performed by Color Histogram with 98.5% and the score of the proposed approach with gradient direction is 93.2% which is almost 10% lower than the highest. The score of the proposed approach is the worst score which is 64.1%. In the last column of the table, the average matching scores of the descriptors are stated. According to it, the highest score is reached by IDSC + DP and it is 88.1%. The average score of the proposed approach with tangent direction is 85.1% and it is slightly lower than the highest score. And the average score of the proposed approach with gradient direction is 78.6% and it is one of the lowest overall performance on the table.

In Tables 5.7 and 5.8, the confusion matrix of the proposed approaches for images of ETH - 80 dataset is indicated. Vertical direction shows the actual values of the samples and horizontal direction shows their predicted values. For each row the summation of the

Table 5.7. Confusion matrix for images of the ETH - 80 dataset. Descriptors are computed by the proposed approach when their directions are **tangent direction**. In the table, y-axis shows the actual values of the samples and x-axis shows the predicted values for the samples. So samples that are shown in the diagonal axis of the matrix are classified correctly.

		Predicted							
		apple	car	cow	cup	dog	horse	pear	tomato
Actual	apple	**312**	0	0	3	0	0	6	89
	car	0	**409**	0	1	0	0	0	0
	cow	0	11	**363**	5	14	15	1	1
	cup	1	0	0	**407**	0	0	0	2
	dog	0	3	35	0	**313**	59	0	0
	horse	0	1	24	0	31	**354**	0	0
	pear	14	0	0	12	0	0	**370**	14
	tomato	131	2	0	6	2	2	4	**263**

entries reaches 410 which is the number of test samples. While entries in the diagonal of the matrix shows the number of samples that are classified correctly, the others shows the number of incorrectly classified samples.

5.5. Discussion

Pure SC that is proposed by Belongie et al. [2] is well known shape descriptor and relatively simple approach. It is explained in Chapter 2 in detail. Comparing the proposed approach with SC is a fair comparison because both of them are in the same class of descriptors and their simplicity are close to each other. In experiments with the images of Modified ICDAR 2013, each group of Chars74k, and Kimia's - 99 datasets, the proposed approach with tangent direction has higher performance than SC. And the differences between them are almost 10%, 23%, 10%, 23%, and 23% respectively. For Mpeg - 7 and ETH - 80 datasets, experiments shows its success is lower than the SC's. And the differences between them are almost 9% and 1%. So for ETH - 80 dataset their overall scores are close to each other. In brief, when SC and the proposed approach are compared, the proposed approach beats SC in experiments with three datasets, is beaten

Table 5.8. Confusion matrix for images of the ETH - 80 dataset.Descriptors are computed by the proposed approach when their directions are **gradient direction**. In the table, y-axis shows the actual values of the samples and x-axis shows the predicted values for the samples. So samples that are shown in the diagonal axis of the matrix are classified correctly.

		Predicted							
		apple	car	cow	cup	dog	horse	pear	tomato
Actual	apple	**322**	0	0	1	0	0	6	81
	car	0	**406**	0	3	0	0	0	1
	cow	0	16	**303**	1	35	55	0	0
	cup	12	10	0	**371**	0	0	1	16
	dog	0	7	63	0	**179**	161	0	0
	horse	0	0	58	0	107	**245**	0	0
	pear	19	1	0	0	0	0	**382**	8
	tomato	25	3	0	4	5	1	2	**370**

in experiments with one dataset, and tie with it in one dataset.

In character classification, the weakness of the proposed approaches is caused by ambiguous characters which have either exactly the same shape or close shapes in upper and lower cases. For example, the character ”O” in upper case and lower case has exactly the same shape and the shape of the digit zero is close to them. Another example is the character ”S” in both cases has the same shape. In addition to this, some samples for the ambiguous characters are indicated in Figures 5.20 and 5.21. This ambiguity causes incorrect classification and this appears as a line parallel to the main diagonal and below of it in Figures 5.18 and 5.19.

The weakness of the proposed approach with tangent direction in the experiments with images of ETH - 80 dataset is confusion between the objects ”apple” and ”tomato”. Its matching score is 76.1% for the ”apple” object and 64.1% for the ”tomato” object that is indicated in Table 5.6. Furthermore, this confusion can be observed in confusion matrix as well and it is shown in Table 5.7. According to the confusion matrix, 89 ”apple” samples are classified as ”tomato”. Likewise, 131 ”tomato” samples are classified as ”apple”. The reason for the confusion between ”apple” and ”tomato” is the contour of their samples looks like each other. So classifying them is a challenge by considering only their contours. When the overall matching score is calculated by excluding the object

”apple” and ”tomato”, the score of the proposed approach reaches 90.1%. Moreover, there are only two approaches, that have higher matching score, which are named SC Greedy and SC + DP. They perform slightly better than the proposed approach and their scores are 90.6% and 90.8%. Note that IDSC + DP and MDSC + SC + DP cannot be used in this compassion because their scores are not stated with respect to the objects in [21]. While classifying objects named ”apple” and ”tomato” causes a weakness for the proposed approach with tangent direction, the success of classification those objects, especially ”tomato”, is relatively high for the proposed object with gradient direction. In Table 5.6 shows that the matching score of this approach is 78.5% and 90.2%. Moreover, in the confusion matrices, the number of objects that are actually ”apple” but classified as ”tomato” decreases to 81 from 89 and the number of objects that are actually ”tomato” but classified as ”apple” decreases to 25 from 131. So the improvement of the score is more obvious in the classification of the ”tomato” object. However, the success of the proposed approach with gradient direction is lower than the approach with tangent direction in the classification of the objects named ”dog” and ”horse” especially. So the overall score of the approach with tangent direction is more than 5% higher than the proposed approach with gradient direction.

5.6. Conclusion

Many computer vision applications use image features via their descriptors. So in the literature, there are several feature description algorithms. In this part of this study, a novel shape based descriptor is proposed in order to classify the shapes especially characters. There are two motivation of selecting characters as the main target. First is ignoring texture and shape around the charter and second, the boundary of characters has enough information to describe them. So this approach uses only the contour of shapes to provide the motivations and it has two main parts which are orientation estimation and descriptor computation. Orientation is estimated by finding the dominant direction of the boundary points and it is used to normalize the shapes. During the descriptor computation, the location and direction of the boundary points of the normalized shape are combined. So the computed descriptor contains two information which is spatial and directional.

The proposed approach is evaluated by comparing its matching score with 28 states of the art approaches. In the evaluation, five datasets are used. Two of them are character datasets and the others are shape datasets. The results of the experiments shows that the proposed approach with tangent direction is a convenient choice for especially character recognition applications. It has the highest matching score in the experiments with characters and it is shown in Tables 5.2 and 5.3. Moreover, the approach is competitive for shape recognition applications. In three experiments with shapes of objects, it has the lowest score for Mpeg - 7 dataset, the highest score for Kimia's - 99 dataset, and a score in the middle for ETH - 80 dataset. However, by considering its simplicity the proposed approach can be a wise choice in order to use in shape classification and recognition applications.

The proposed descriptor can be used to match keypoints that are detected by region based keypoint detector. To do that, the descriptor can be computed by using the boundary points of regions instead of the contour points of objects. After descriptors are computed for regions that are detected on different viewpoint of objects, keypoints that have the closest descriptors are accepted as a match. Unfortunately, although the success of the proposed descriptor is good enough to classify shapes, it is not proper for keypoint matching. Because in keypoint matching the distinctive feature between regions that are detected on different viewpoint of objects is texture. Furthermore, most of them have a circular shape and this causes ambiguity for the proposed descriptor.

CHAPTER 6

CONCLUSION & FUTURE WORK

In this thesis, we focused on image curves and performed three distinct studies. In the first part, a detailed and realistic stability analysis was designed for image curves like regions that are detected by region based keypoint detectors. Second, we proposed an approach that is used to detect interest points on the image curves like the extremal region boundary. As a third step, we designed a novel shape based descriptor based on tangent directions by gathering information only from image curve structure.

We developed a stability analysis for region based detectors. In the literature, their stability is analyzed by measuring repeatability. Repeatability is a metric in terms of percentage and it determines the success of detection of the corresponding scenes in different viewpoints of objects. In the literature, there are several keypoint detectors and they have different aspects so measuring repeatability is adapted with respect to the aspects of detectors. The proposed stability analysis is designed to measure the repeatability of region based detectors. In the stability analysis, repeatability of regions is computed. At the beginning of the analysis, synthetic images are generated by deforming a reference image with three camera position parameters. After deformed images are generated, regions on the reference image are transformed into the deformed images. Then the overlap ratio is computed between the transformed region and each region on the deformed image. According to the overlap ratio, the correspondences are found and the fit ratio of these correspondences and the ground truth gives the repeatability. When regions are transformed into the deformed images, an approximation is required for the sake of computational simplicity. And in this study, convex hull approximation is used because it is faster than the transforming regions directly. And it is more sensitive and realistic than the ellipse approximation which is mostly used in the literature. After the repeatability of regions is computed, we observed that the repeatability of MSER is high enough for computer vision application such as object detection and tracking. In addition to this, its robustness against three camera position parameters is shown.

In the second part, we designed a detector in order to locate interest points on the boundary of regions. Regions are detected by region based keypoint detectors and in this study, MSER detector is selected. In the proposed approach, the interest points on the boundary are detected by considering their curvature value. During the curvature calculation, first, tangent direction for each boundary point is calculated, second, the derivative of tangent directions is taken to compute the curvature. And the local extrema of curvature are found and marked as interest points. After the approach is designed, it is evaluated by comparing the repeatability of the points that are detected by the proposed approach, the center points of the best-fitted ellipse of regions, and affine invariant points that are detected by curvature scale space [23]. In the evaluation, two datasets are used. One of them is 2D dataset named Oxford dataset and the other is 3D dataset which is created by Moreels et. al. [31]. The results of experiments with 2D dataset show the success of the proposed approach and CSS is close to each other and they are better than the success of the ellipse center. In the experiments with the 3D dataset, the results show the repeatability of the proposed approach is the highest. And the followings are affine invariant points that are detected by CSS and the center points of the best-fitted ellipse of MSER in descending order. In addition to the improvement, the proposed approach makes MSER usable in 3D reconstruction because 3D operations require individual points instead of regions. So the proposed approach is suitable for computer vision applications that use 3D data.

In the last part of this thesis, a shape based descriptor was proposed. The purpose of the proposed approach is recognizing objects with high recognition rate. And its main target is objects that have significant shapes such as characters. In the approach, there are two main steps which are orientation estimation and descriptor computation. In the first step, orientation is estimated by finding the dominant direction of tangent directions of contour points. After the orientation is estimated, the shape is normalized. And descriptor is computed by using the normalized shape by combining the spatial and directional information of the shape contour. The proposed approach is evaluated by comparing its success with the success of several state-of-art descriptors. And the experiments are evaluated in five datasets. Two of them is character datasets and the others are shape datasets. The results of the experiments with character datasets show the success of the proposed approach is the highest. And in the experiments with shape datasets, the proposed approach is competitive. So choosing to use the proposed approach in character recognition applications is convenient.

6.1. Future Work

In the second part of this study, interest points are detected on only the outer boundary of the extremal regions. So holes inside regions are ignored. With some adaptations in the boundary traversal, the boundary of holes can be added to the detection process thus information loss that is caused by ignoring the boundary of holes can be prevented. In the same approach, the boundary of the regions is guaranteed to be closed curve in traversal process. And this creates a vertical periodicity during the tangent direction calculation. So with some adaptations in the tangent direction calculation, the closed curve restriction can be removed and this gives an opportunity to apply the proposed approach to any type of curves like edges. This means that removing the closed curve restriction increase the usability of the proposed approach. In addition to these possible extensions, after keypoints on the boundary is detected, scale for them can be estimated. To do that a post process might be added to the approach and it can be based on Laplacian responses over scales.

In the description of shapes, one of the observed vulnerabilities is caused by ambiguous characters that have the same shape in upper and lower case such as the characters ”O, S, P ...”. To overcome this vulnerability, the proposed approach can be adapted. And this adaptation can be comparing the height of the target character with the height of the other characters in the word. Moreover, the proposed approach focuses on only describing characters. So it is supposed that there is an object detector which detects objects and prepares the contour of their shape for the description process. In the evaluation, some mechanisms are suggested to detect objects and collect the contour points of their shape. However, there is no general solution that is proposed in this study. For this reason, proposing a detector which is compatible with the requirement of the descriptor can be useful in order to increase the usability of the proposed descriptor in the real world. In addition to them, the proposed descriptor can be used to classify characters in different alphabets from the English alphabet. For example, in the Turkish alphabet there are some extra characters which are ”ğ, Ğ, ç, Ç, ş, Ş, ü, Ü, ö, Ö, ı, and İ”. During their classification, they can be classified as the closest characters in the English alphabet. Namely ”ç” can be classified as ”c”. After that, a postprocess which is responsible for searching dot below the character can be applied in order to separate ”ç” from ”c”. Although this solution can be applied to classify characters in closest alphabets to English, in the classification of the totally different characters such as Korean, Chinese alphabets cannot perform well. For those characters, the number of grid cells of the proposed descriptor can be increased in order to improve the level of descriptiveness of the descriptor. For this part of the study, another future work can be combining the proposed descriptor with neural network based approaches. To combine them, the descriptor can be given as input to a neural network based classifier in numeric or as an image like 2D code.

REFERENCES

[1] A. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219, 1979.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[3] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 26–33. IEEE, 2005.

[4] P. Bourke. Calculating the area and centroid of a polygon, 1988.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[7] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *VISAPP (2)*, pages 273–280, 2009.

[8] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[9] C. Grigorescu and N. Petkov. Distance sets for shape filters and shape recognition. *IEEE Transactions on Image Processing*, 12(10):1274–1286, 2003.

[10] A. C. Jalba, M. H. Wilkinson, and J. B. Roerdink. Shape representation and recognition through morphological curvature scale spaces. *IEEE Transactions on Image Processing*, 15(2):331–341, 2006.

[11] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *European conference on computer vision*, pages 228–241. Springer, 2004.

[12] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, pages 506–513, Washington, DC, USA, 2004. IEEE Computer Society.

[13] W.-Y. Kim and Y.-S. Kim. A region-based shape descriptor using zernike moments. *Signal processing: Image communication*, 16(1):95–102, 2000.

[14] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological cybernetics*, 55(6):367–375, 1987.

[15] L. J. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[16] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–319. IEEE, 2003.

[17] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–409. IEEE, 2003.

[18] A. Leonardis, H. Bischof, and J. Maver. Multiple eigenspaces. *Pattern Recognition*, 35(11):2613–2627, 2002.

[19] V. Lepetit, P. Lagger, and P. Fua. Randomized Trees for real-time keypoint recognition. San Diego, CA, June 2005.

[20] S. X. Liao and M. Pawlak. On image analysis by moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):254–266, 1996.

[21] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE transactions on pattern analysis and machine intelligence*, 29(2):286–299, 2007.

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[23] F. Mai, C. Chang, and Y. Hung. Affine-invariant shape matching and recognition under partial occlusion. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 4605–4608. IEEE, 2010.

[24] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, pages 384–393, September 2002.

[25] J. Matas, Z. Shao, and J. Kitter. Estimation of curvature and tangent direction by median filtered differencing. In *8th Int. Conf. on Image Analysis and Processing*, San Remo, 1995.

[26] G. McNeill and S. Vijayakumar. 2d shape classification and retrieval. 2005.

[27] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *Computer Vision - ECCV 2002*, pages 128–142, 2002.

[28] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005.

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

[30] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. pages 35–42, 1996.

[31] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D

objects. *International Journal of Computer Vision*, 2006.

[32] J.-M. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[33] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or ”how do i organize my holiday snaps?”. In *European conference on computer vision*, pages 414–431. Springer, 2002.

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

[35] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE transactions on pattern analysis and machine intelligence*, 25(1):116–125, 2003.

[36] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on pattern analysis and machine intelligence*, 26(5):550–571, 2004.

[37] X. Shu and X.-J. Wu. A novel contour descriptor for 2d shape matching and its application to image retrieval. *Image and vision Computing*, 29(4):286–294, 2011.

[38] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.

[39] I. E. Sutherland and G. W. Hodgman. Reentrant polygon clipping. *Commun. ACM*, 17(1):32–42, 1974.

[40] M. J. Swain and D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[41] Z. Tu and A. L. Yuille. Shape matching and recognition–using generative models and informative features. In *European Conference on Computer Vision*, pages 195–209. Springer, 2004.

[42] T. Tuytelaars, L. Van Gool, L. D'haene, and R. Koch. Matching of affinely invariant regions for visual servoing. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1601–1606. IEEE, 1999.

[43] T. Tuytelaars and L. J. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, volume 412, 2000.

[44] F. E. Uzyıldırım, A. Köksal, and M. Özuysal. A detailed analysis of mser and fast repeatibility. In *Signal Processing and Communications Applications Conference (SIU), 2015 23th*, pages 2098–2101. IEEE, 2015.

[45] L. Van Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. In *European Conference on Computer Vision*, pages 642–651. Springer, 1996.

[46] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *European Conference on Computer Vision*, pages 255–271. Springer, 2002.

[47] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, volume 2, pages II–691. IEEE, 2003.

[48] J. Xie, P.-A. Heng, and M. Shah. Shape matching and modeling using skeletal context. *Pattern Recognition*, 41(5):1756–1767, 2008.

APPENDIX A

REPEATABILITY OF THE REMAINING SETS

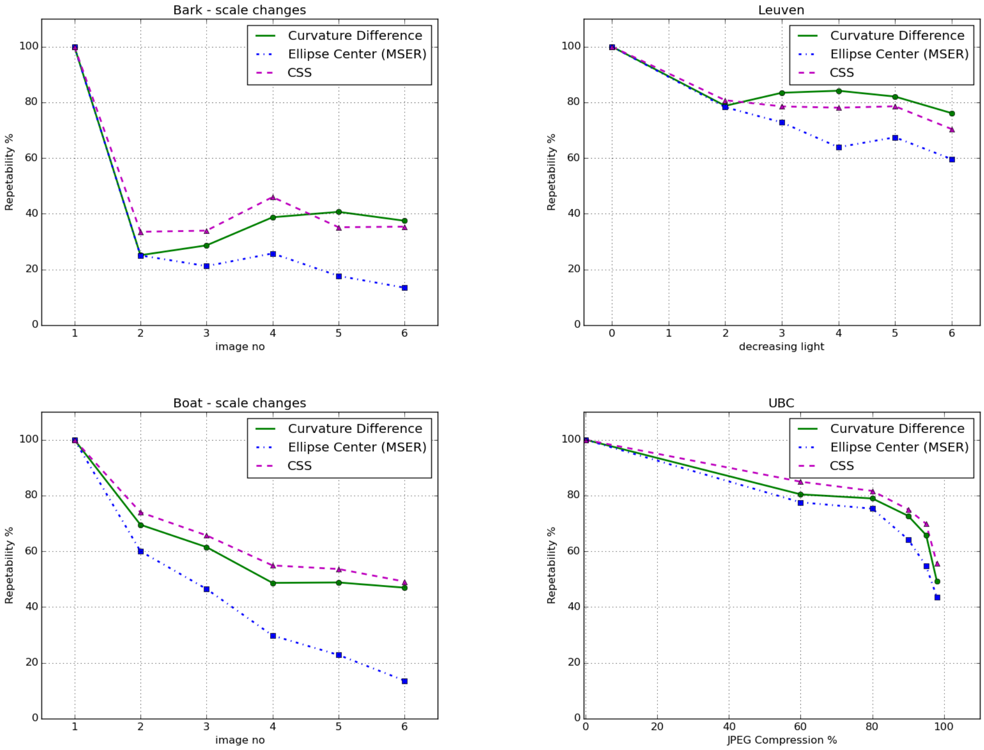


Figure A.1. The repeatability of remaining four image sequences. The results of Bark, Boat, Leuven, and UBC image sequences are shown from top to bottom and left to right. The performance of the proposed method is better than MSER and close to CSS on all image sequences.

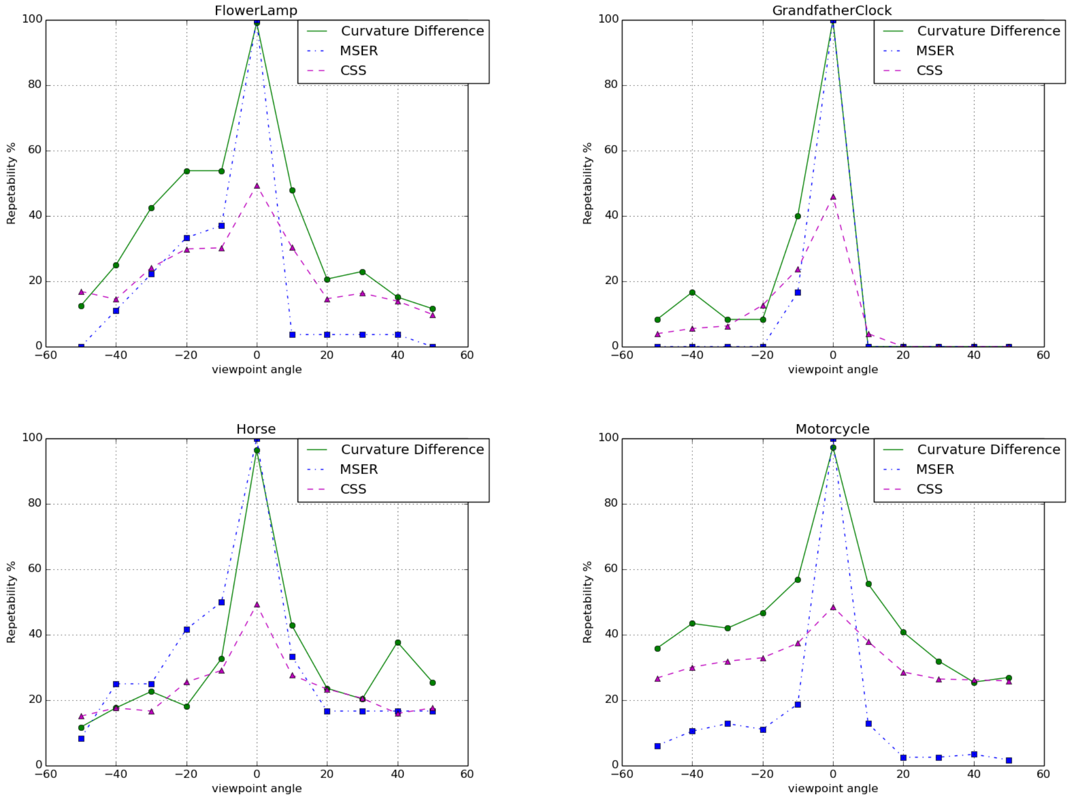


Figure A.2. The repeatability of FlowerLamp, GranfatherClock, Horse, and Motorcycle objects are shown from top to bottom and left to right. The proposed method has better performance on images of the FlowerLamp, GranfatherClock, and Motorcycle objects. For object Horse, performances of methods are close to each other and MSER has the highest repeatability values.

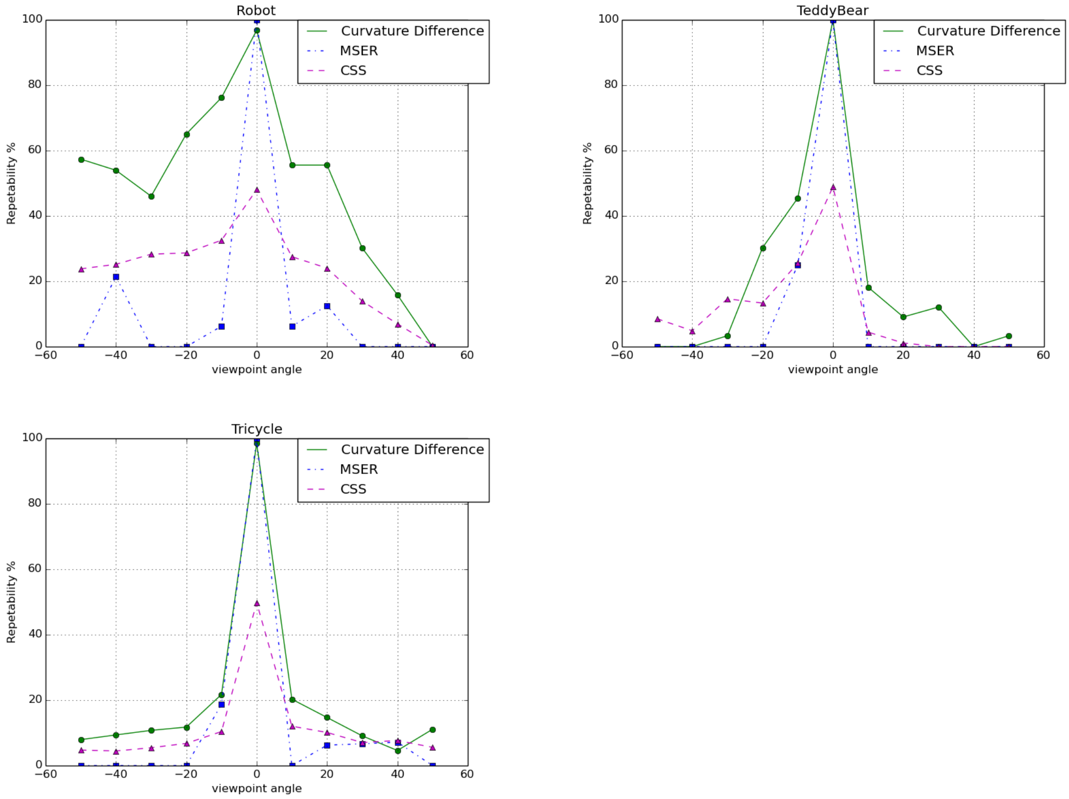


Figure A.3. The repeatability of Robot, TeddyBear, and Tricycle objects are shown from top to bottom and left to right. For all of them, the proposed method has the best performance. Furthermore, while for the object Robot, the gap between curvature difference and other methods is considerable, for the object Tricycle, the performance of curvature difference is slightly better.

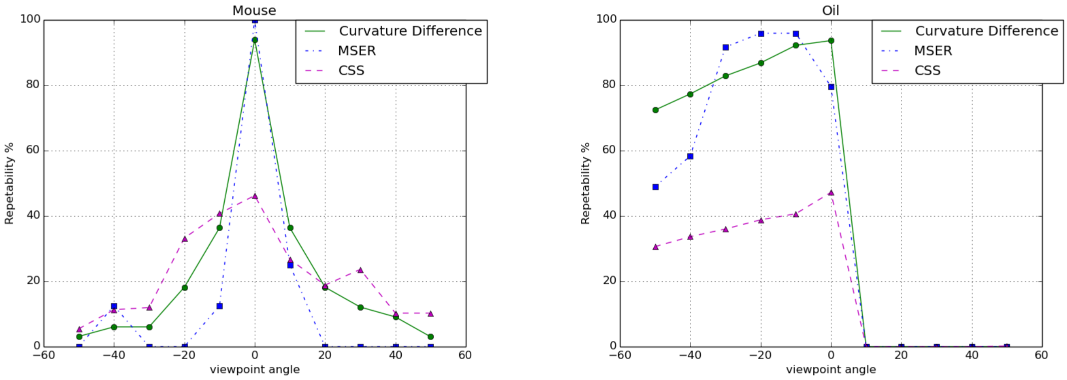


Figure A.4. The results of Mouse and Oil objects are shown from top to bottom and left to right. The repeatability of all approaches drops to 0% at 10°and remains 0% till 50°. So there is no keypoint that repeats in positive direction turn. For object Mouse except for 0°and 10°, CSS is slightly better than the proposed method. The performance of MSER reaches 0% at -20°and 20°. For object Oil, the proposed method and CSS has exactly the same behavior and the repeatability values of the proposed method are double up CSS. Furthermore, MSER has the best performance at -10°, -20°, and -30°.

APPENDIX B

MATCHING SCORE UNDER VARIOUS SCALE

Table B.1. Under various scale factor, the matching score of the descriptors that is computed on characters of Modified ICDAR 2013 dataset. Characters are matched according to the nearest neighbor classifier. The matching score is measured 20 times and the average and standard deviation of the matching scores are calculated and stated.

	Scale Factor								
Algorithm	**0.25**	**0.3**	**0.35**	**0.375**	**0.5**	**0.625**	**0.75**	**0.875**	**1.0**
CC	74.56	**80.45**	77.74	79.17	71.54	61.53	41.02	27.82	21.62
CF	58.82	64.86	**72.34**	72.02	66.77	58.98	52.31	42.93	28.46
GLOH	76.31	80.29	79.81	**81.08**	78.7	74.72	66.93	58.98	42.61
JLA	69.0	**73.29**	72.97	73.13	69.0	57.07	38.31	24.32	17.49
KOEN	57.07	**64.23**	61.69	59.3	51.51	32.59	21.3	12.72	10.17
MOM	**72.02**	71.38	64.55	59.14	39.11	25.6	18.6	12.88	10.49
PCA-SIFT	76.31	**79.01**	78.86	78.06	67.57	56.6	39.9	27.82	19.4
SC	73.29	76.47	77.11	**78.06**	76.79	70.59	62.32	47.38	34.5
SIFT	77.27	80.29	80.29	**80.6**	80.13	74.4	67.41	54.85	40.7
SPIN	36.72	42.77	47.54	**49.92**	45.63	43.72	37.84	31.16	27.34