IMPERIAL COLLEGE LONDON

Study of Object Recognition and Identification Based on Shape and Texture Analysis

by

Guanqi Wang

Supervisor: Dr. T. Stathaki

A thesis submitted to Imperial College London for the degree of

Doctor of Philosophy

Department of Electrical and Electronic Engineering

Imperial College London

London SW7 2AZ

October 2011

Abstract

The objective of object recognition is to enable computers to recognize image patterns without human intervention. According to its applications, it is mainly divided into two parts: recognition of object categories and detection/identification of objects.

My thesis studied the techniques of object feature analysis and identification strategies, which solve the object recognition problem by employing effective and perceptually important object features. The shape information is of particular interest and a review of the shape representation and description is presented, as well as the latest research work on object recognition. In the second chapter of the thesis, a novel content-based approach is proposed for efficient shape classification and retrieval of 2D objects.

Two object detection approaches, which are designed according to the characteristics of the shape context and SIFT descriptors, respectively, are analyzed and compared. It is found that the identification strategy constructed on a single type of object feature is only able to recognize the target object under specific conditions which the identifier is adapted to. These identifiers are usually designed to detect the target objects which are rich in the feature type captured by the identifier. In addition, this type of feature often distinguishes the target object from the complex scene.

To overcome this constraint, a novel prototyped-based object identification method is presented to detect the target object in the complex scene by employing different types of descriptors to capture the heterogeneous features. All types of descriptors are modified to meet the requirement of the detection strategy's framework. Thus this new method is able to describe and identify various kinds of objects whose dominant features are quite different. The identification system employs the cosine similarity to evaluate the resemblance between the prototype image and image windows on the complex scene. Then a 'resemblance map' is established with values on each patch representing the likelihood of the target object's presence. The simulation approved that this novel object detection strategy is efficient, robust and of scale and rotation invariance.

Acknowledgement

There are many people who have helped me in completion of this thesis and I would like to express my sincere gratitude to all of them.

Firstly, I would like to thank my supervisor, Dr. T. Stathaki, whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing of this thesis.

Secondly, I am deeply indebted to my parents, who have supported me for all these years studying abroad, mentally and financially.

I also feel grateful for the companion of my best friends at Imperial, Minhao Liu and Rex Kwow, whose kindness and friendship helped me overcome the difficulties during my time at Imperial.

Especially, I would like to give my special thanks to my wife Yinglu, whose patient love enabled me to complete this work.

Contents

List of Figures

List of Tables

Statement of Originality

Substantial parts of Chapters 2, 3, 4 and 5 of this thesis are, as far as the author is aware, original contribution to the area of object recognition. The most significant contributions are:

1. A new shape classification strategy is introduced in Section 2.1. It is a technique of learning and classifying shape contours based on a novel shape matching strategy. The Shape Context descriptor is used to represent the shape information, and during the descriptor matching procedure, the salient points (corners) are signed with larger weights than normal contour points.
2. Study of two object identifiers, each based on a single type of object feature, is presented in Chapter 3 and Chapter 4. Their simulation and performance are analyzed and compared in these two chapters. The conclusion is that the identifiers based on a single type of object features are not suitable for real life object detection in complex scene.
3. A novel object identification approach is introduced in Chapter 5. The modifications of two types of descriptors, SIFT and HOG, to capture the object's texture and shape features, respectively, are presented in Section 5.1. A new way to combine different types of descriptors in the estimation of similarity between two images is explained and justified theoretically in section 5.2. The results of the implementation of this new identifier are presented in section 5.3.

Chapter 1

Introduction & Literature Review

In computer vision and image processing, analysis of visual objects is a vital component in the tasks of object recognition, image retrieval, image registration, and more others. These tasks are often involved in many variant areas which include the applications of surveillance, video forensics, and medical image analysis for computer-aided diagnosis and etc. Among them, object recognition has received much attention recently, because of the increasing demand both from the scientific world and the industry.

The objective of object recognition is to enable machines or artificial systems to recognize image patterns or to identify an object without human intervention. According to its usage, it is mainly divided into two parts: recognition of object categories and detection/identification of objects. The goal of the former one is to classify an observed object into one of the several predefined categories. For the latter one, it is to detect whether the object of interest is in the image and then separate it from the background in a target image. The task of object recognition can also be divided into two stages, namely the 'low-level' vision and 'high-level' vision. The first stage involves the extraction of significant features from an image, such as object boundaries, and usually the segmentation of the image into separate objects. The goal

of high-level vision is to recognize these objects by finding effective and perceptually important object features. My work is to study the techniques in these two stages and focus on the latter one, which tries to solve the object recognition problem with object feature analysis and identification strategy.

The features extracted from the image are usually represented by mathematical *models*, which are different types of descriptors for the object to be recognized. Generally speaking, a model is associated with a set of parameters which represent the information regarding the shape, texture, or any other type of characteristic features of objects. A specific object of interest may be associated with a specific set of parameter values. Therefore, during the object recognition process, when these parameter values are detected a *model (descriptor) label* is attached. A model label can be interpreted as a tag pinned to an area in the image that is believed to show an instance of the corresponding object model. A model may be two- or three-dimensional, whereas labels are always after 2D model instances in the image.

In the area of object recognition, *photometry* usually refers to the processing of measuring the light intensities in terms of their perceived brightness to human vision, reflected from the surfaces in a scene and recorded on a camera film; on various occasions, data may originate from other image acquisition sources such as ultrasound or x-ray imaging instead of light. A digital computer image is often a two dimensional (2D) array of numbers called *pixels* whose values represent the scene's light intensities, that is, the strength of the brightness which is reflected from a particular point on the recording medium. When dealing with the information in a digital image and its corresponding scene, one important distinction is the term which describes the spatial dependency of a mathematical model (descriptor). The term *local* always refer to processes that only deal with the nearest neighbors of a pixel and ignore the information of the rest of the image. The term *global*, by contrast, is used to refer to the processes in which context information from the entire image or scene is considered.

1.1 Object Recognition Challenges

For humans, it is a trivial task to recognize a variety of both known and new objects in an image, despite the fact that these objects appear in the image under quite different conditions, e.g., in different scales, observed from different view-points, or subject to various types of distortion. The human visual system is “intelligent” due to the huge amount of accumulated experience which it uses to interpret visual information in the most efficient way. However, the automatic object recognition is a complicated task for the artificial systems, e.g., computers, robots etc., which is unable to compete with the human brain. There are many challenges to be faced when we are to design an object recognition system. The most important ones are described in the following sections.

1.1.1 View Point Variation

The first challenge is the variation of view-points. The objects in real life are physical 3D volumetric entities. Therefore, when an object is perceived from different view-points, it might appear completely different, as the instance illustrated in Figure 1.1. The human eyes recognize with ease an object regardless of the view-point change, but it is quite difficult for the man-made systems to identify the same object from different view-points. This is because from different view-points, the parameter values in the mathematical models (descriptors), which capture the same features of the same object, are significantly different that the corresponding object models could not be matched in the recognition procedure.

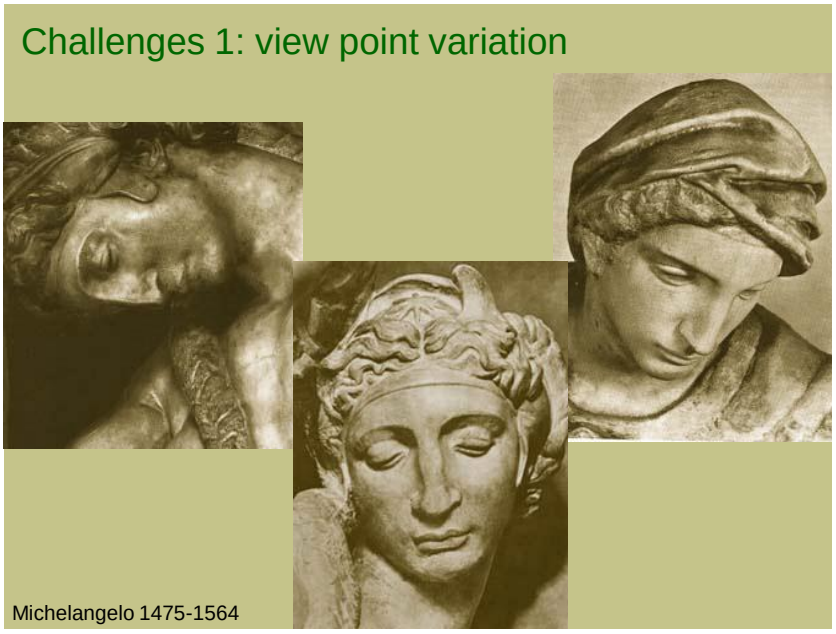


Figure 1.1 Object recognition challenges 1: View point variation

1.1.2 Illumination Changes

The next challenge in object recognition is illumination difference, which could affect the performance of the object recognition system by failing the local image models to recognize two identical image parts. For example, Figure 1.2 illustrates two images of the same person under different illumination conditions. As shown clearly, parts of the right image are significantly darker compared to the corresponding parts of the left image. Again, for a local or even global image model, this will entail essential difference in the values of the model parameters and the recognition strategy will fail this face recognition task.

Figure 1.2 Object recognition Challenges 2: Illumination

1.1.3 Occlusion

Occlusion is another challenging problem in object recognition. When the object of interest is part of a real life complex scene, its entire representation is usually not contained in the image. As shown in Figure 1.3, the target object is likely to be occluded by other objects in the scene. Therefore, there is a requirement for robust local image models (descriptors), and the global models are not suitable in this scenario.

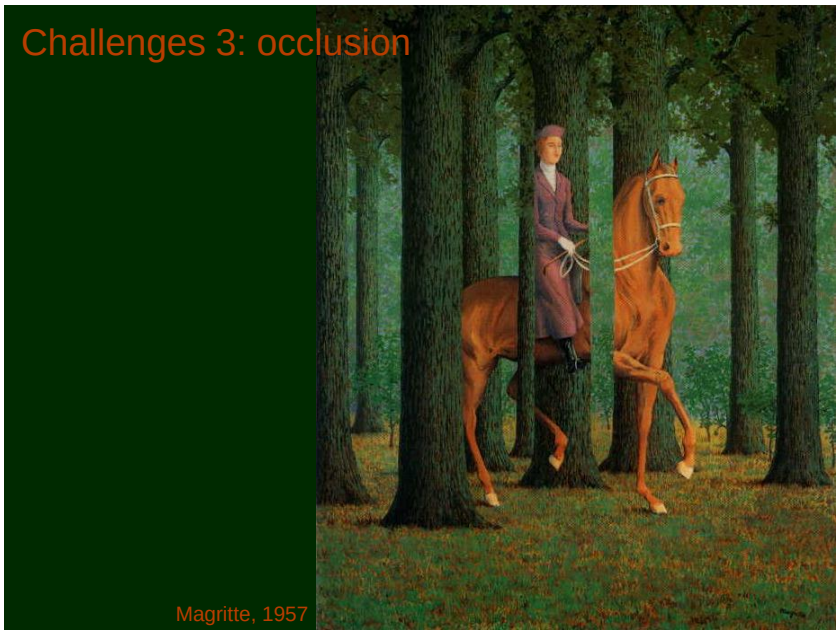


Figure 1.3 Object recognition Challenges 3: Occlusion

1.1.4 Scale Variation

Objects of the same class are often of different scales when they appear in the real life complex scene. They are in different physical sizes, or they are geometrically varied when they are far or near from the view point. In Figure 1.4, there are two classes of objects, laptops and human beings, which are naturally identified by human visual system. However, the machine algorithms, which are insensitive or unable to detect the appropriate scales of objects, have problems in recognition of the target objects. Thus, to recognize the objects with different sizes in specific images, we need to consider the scales of target objects.

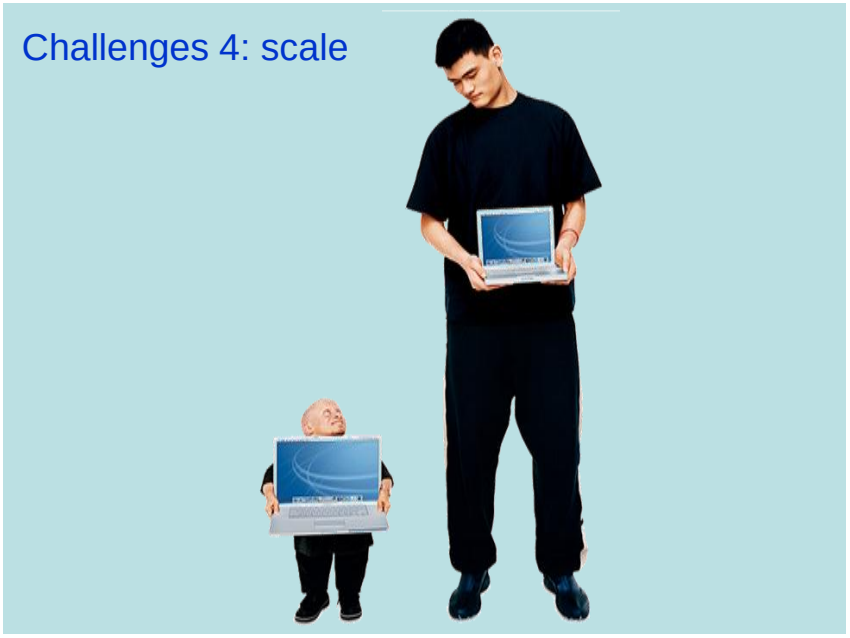


Figure 1.4 Object recognition Challenges 4: Scale

1.1.5 Deformation

Deformation is another problem, especially for the articulated objects such as the horses shown in Figure 1.5. There are lots of articulated objects in this world, including human body, animals, stringing object, and etc. Therefore, this problem must not be neglected. An expression of the articulation and deformation is in demand to describe and recognize these objects, which are perceived differently in shape, interior structure and other distortion due to the deformation.

Figure 1.5 Object recognition Challenges 5: Deformation

1.1.6 Background Clutter

To detect the faces in Figure 1.6 is not a simple task by employing any of the face recognition algorithms. This is an example of detecting the target objects in a cluttered scene. In this world, billions of objects coexist, and even some figures possess no meaning to us. These scenes, which are consisted of these objects and figures, are observed and perceived by us. Therefore how to recognize the demanded object out of this cluttered world and scene is a problem need to be solved.

Figure 1.6 Object recognition Challenges 6: Background clutter

1.1.7 Intra-class Variation

In terms of object classification, there are problems concerning the single object recognition, ranging from view-point variation and illumination to scales and etc. On top of that, the intra-class variation is another challenge need to be considered. A typical problem heard often in computer vision is: how are chairs to be recognized? As the chairs shown in Figure 1.7, even the humans are not certain to verify them. This might be an extreme example of the categorization problem. However, it demonstrates the difficulty of this problem. Therefore, a huge effort is necessitated to tackle the intra-class variability.

Figure 1.7 Object recognition Challenges 7: Intra-class variation

1.2 Previous Techniques of Shape Representation and Description

The basic features of an object are its shape and surface texture information. And I am particularly interested in the shape features. To analyze the shape information, there are two fundamental methods: shape representation and description. The shape representation approach constructs a non-numeric representation of the shape, for example a graph. On the other hand, the shape description is an approach in which a feature vector is produced to describe the shape feature uniquely and mathematically. Consider the situations in which the object shape is occluded or corrupted by noise and irrelevant objects, the task of shape description and representation faces plenty of obstructions. Not only a good shape descriptor is capable of overcoming the above difficulties, it also needs to be invariant of certain transformations of the object in the image, which are caused by the changes in the scale, location, orientation and pose of the object. A large variety of shape descriptors have been studied and evaluated in [1]. The performance and comparison of descriptors constructed on all object features can also be found in [2].

According to where the shape features are extracted, from the contour or from the whole shape region, the shape representation and description techniques are categorized into contour-based and region-based methods, respectively [1]. Then each class is further classified as global and structural methods. In global methods, the shape is represented as a whole. And in structural methods, the shape is represented by segments, named primitive. Then, according to whether the shape features are calculated in the spatial or transformed domain, the techniques are sorted as space or transform domain techniques, respectively [1].

When the boundary information is not available for shape analysis, region-based methods are preferred. In the category of these techniques, the shape representation is estimated by

employing all pixels within a shape region, instead of the boundary points used in the contour-based approaches. Moment descriptors [3-19] are the most common region-based methods. Besides the moment descriptors, there are many other methods in the region-based category, for instance, grid method [22], shape matrix [23], convex hull and media axis [24-27]. As discussed above, according to whether the methods separate shapes into sections or not, the region-based methods are classified into global and structural methods [1].

Using global methods, the descriptor covers the whole shape resulting in a numeric feature vector which can be used for shape description. The common techniques in this category use image moment invariants, which was first introduced by Hu for two-dimensional pattern recognition applications [3]. His approach is based on the theory of algebraic forms:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y), \qquad p,q = 0,1,2,\ldots . \qquad (1.2.1)$$

where x and y are the coordination of each pixel. $f(x,y)$ is the grey value of pixel (x,y). These moments, known as geometric moments, are derived from a nonlinear combination of lower order moments and they have the desirable properties of being invariant under translation, rotation and scaling. A lot of works [4-9] have been published concerning geometric moment invariants. Also this type of techniques has been used in many applications [10-13]. However, the limited number of invariants computed from the lower order moments could not offer enough discriminative power for shape representation. In addition, it is difficult to derive higher order moments [1].

Another type of moment invariants are the algebraic moments, introduced by Toubin and Cooper [14, 15]. The first *m* central moments are employed to construct matrices, whose eigenvalues are taken as the invariants of algebraic moments. Compared with geometric moment invariants, the algebraic moment invariants have the advantage of invariance to affine transformations and can be constructed up to arbitrary order. However, according to

the work [16], they only perform well when the texture feature, i.e., the distribution of pixels, is rich on the object. Based on the algebraic moment invariants, Teague in [17] introduced another moment shape descriptor, namely the orthogonal moments, as for example the Legendre moments and Zernike moments. They are calculated by replacing $x^p \ y^q$ in (1.2.1) with Legendre polynomials and Zernike polynomials, respectively. Legendre moments and Zernike moments are called orthogonal moments, because Legendre and Zernike polynomials are both complete sets of an orthogonal basis.

In the work of Teh and Chin [18], many types of orthogonal moments were studied, namely the Legedre moments, Zernike moments and pseudo-Zernike moments. Teh and Chin also compared the above orthogonal moments methods with non-orthogonal ones, e.g., geometric moments, complex moments and rotation moments. Among the moment shape descriptors, Zernike moments are found to have the best performance. Another work on survey of the moment shape descriptors was published by Liao and Pawlak [19]. Their results show that coarser quantization of image produces more accurate moments. The moment shape descriptors are successful in term of their accuracy, robustness and simple construction.

Figure 1.8 (a) An original shape in polar space; (b) polar-raster sampled image plotted in Cartesian space [20]

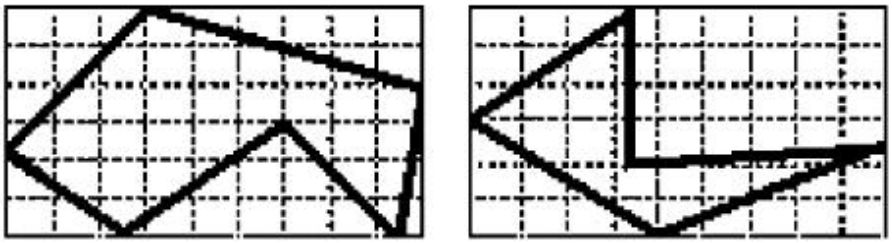


Figure 1.9 Grid representation of two contour shapes [22]

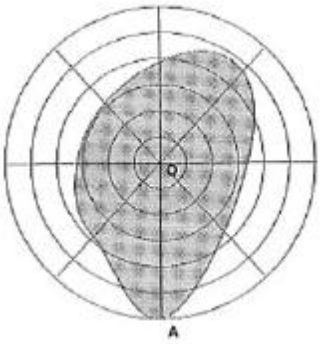


Figure 1.10 Polar raster sampling of shape [23]

Other methods which belongs to the category of global region-based shape representation techniques include generic Fourier descriptor (GFD) [20, 21], grid based method [22] and shape matrix [23]. The generic Fourier descriptor is build up by a 2-D Fourier transformation of a polar-raster sampled shape image (Figure 1.8). The basic idea of the grid based method is to overlay a grid of cells on a shape (Figure 1.9). The value of 1 is assigned to cells covered by the object shape, while the value of 0 is assigned to cells which are not covered by the shape. Afterwards, the grid is scanned from left to right and top to bottom resulting in a bitmap. Therefore, the object shape is represented as a binary feature vector. From the structure of method, it is observed that the grid descriptor is simple to build and match. However, there is problem with its rotation normalization, which relies on the major-axis of the grid. The

estimation of the major-axis of the object shape is sensitive to noise and thus unreliable. Shape matrix descriptor shares a similar idea with grid descriptor. A polar raster of concentric circles and radical lines are laid at the center of the image (Figure 1.10). At the intersections of the circles and radical lines, the shape is sampled as 1 or 0, depending on whether the shape covers the intersections or not. Afterwards a matrix of binary values is formed. The columns of the matrix represent the circles of the raster, while the rows of the matrix represent the radical lines. Before assigning binary values, the scale of the object shape is normalized by the maximum distance from the shape contour points to the center. The resulting shape matrix is invariant to translation, rotation and scale changes. However, both the grid descriptor and shape matrix are sensitive to noise because they are sparse sampling approaches.

For the structural region-based methods, the shape regions are usually split into different parts, which are used to build up separate descriptors for shape representation. There are two main methods in this category, namely the convex hull [24, 25, 26] and the medial axis [27].

Compared with region-based methods, contour-based shape techniques rely on the shape boundary (edges) of the object to represent and describe the object. This type of approach has attracted more attention than the region-based one. This is because of the assumption that humans are supposed to discriminate shapes based on their contour features and the shape interior content does not bear significant information. A representative example is the shape signature approach, which is a comprehensive way of representing the essence of the shape. Using the shape signature, the shape is represented by a one dimensional function derived from the boundary points of the shape. The common shape signatures include centroid profile, complex coordinates, centroid distance, tangent angle, cumulative angle, curvature and area [28, 29, 30]. Shape signatures are usually normalized to be translation and scale invariant. However, they also possess significant drawbacks, for example, they are sensitive to noise and

their matching is too computationally complex. In addition, large matching errors can be triggered by slight changes in the shape boundary.

The boundary moment is another type of global contour-based shape representation method. It is usually constructed based on the shape signature to reduce the dimensions of boundary representations. In [31], the author introduced the boundary moments as the *r*-th moment m_r and central moment μ_r of a shape signature, which was computed to represent the shape of the object. m_r and μ_r are estimated as follows:

$$m_r = \frac{1}{N}\sum_{i=1}^{N}[s(i)]^r \qquad \text{and} \qquad \mu_r = \frac{1}{N}\sum_{i=1}^{N}[s(i)-m_1]^r$$

where N is the number of boundary points. $s(i)$ is the shape signature. To achieve the invariance of translation, rotation and scale changes, the boundary moments are normalized as: $\bar{m}_r = m_r / (\mu_2)^{r/2}$ and $\bar{\mu}_r = \mu_r / (\mu_2)^{r/2}$. In another work [104], a histogram $h(u_i)$ is extracted from the shape signature $s(i)$, and the amplitude of $s(i)$ is considered as a random variable u. Afterwards, the *r*-th moment is calculated as follows:

$$\mu_r = \sum_{i=1}^{K}(u_i - m)^r h(u_i) \qquad \text{where} \qquad m = \sum_{i=1}^{K} u_1 h(u_i)$$

From the structure of these boundary moment descriptors, it is observed that they are simple to implement, but it is hard to interpret higher order moment with any physical meaning.

In the category of contour-based technique, stochastic models and autoregressive (AR) models [32-38] are another solution to the problem of shape description. This type of methods is constructed based on the stochastic modeling of a one dimensional function similar to the shape signature discussed above. In the linear autoregressive model, the value of a function is computed by linearly combining a limited number of preceding function values. Therefore, the

current function value in the sequence has some correlation with and is determined by the previous function values. For an instance, the AR model works as linear combination function to predict the current radius in the following way:

$$R_t = \alpha + \sum_{i=1}^{l} \xi_i R_{t-i} + \sqrt{\beta}\varepsilon_t \ , \qquad (1.2.2)$$

where ξ_i is the AR-model coefficients, and l determines the number of previous function values used in the AR model. α is a constant value , set as a proportion of the function value mean. ε_t represents the value of the current error of prediction. Afterwards, a set of function variables $(\alpha, \xi_1, ...\xi_l, \beta)$ are calculated by applying least square method [32, 34, 36] on the set of prediction functions defined as equation (1.2.2). The estimated α and β are not scale invariant, but the quotient $\alpha / \sqrt{\beta}$, which reflects signal-to-noise ratio of the boundary, is regarded as invariant to scale changes. In addition, the estimated ξ_i are invariant to translation, rotation and scale changes. As a result, the vector $[\xi_1, ...\xi_l, \alpha / \sqrt{\beta}]$ is employed as the shape descriptor in the AR model. However, there are some drawbacks concerning this method. Firstly, it is hard to associate ξ_i with any physical meaning. The value of l is usually decided empirically. Finally, if the object has complex shape boundaries, the limited number of AR parameters $[\xi_1, ...\xi_l, \alpha / \sqrt{\beta}]$ could not offer sufficient discriminative power in the shape description.

In the category of contour-based shape representation technique, spectral descriptors are the most widely used ones. The main advantage of this type of methods is that they are insensitive to noise and shape distortion. This is because that these descriptors are developed under spectral domain. Two of the most popular spectral descriptors are Fourier descriptor (FD) [39] and wavelet descriptor (WD) [40, 41, 42], which are obtained by applying spectral transform on shape boundaries represented by the shape signatures. The wavelet descriptor is of multi-

resolution nature in both spatial space and spectral space, but there is a trade-off between the spatial and frequency resolution. The wavelet descriptor has a complicated matching scheme based on a similarity measurement [42]. There is a lot of researches on Fourier descriptors [39, 43, 44]. The Fourier transformed coefficients are called Fourier descriptor and FD is backed by the well-developed and well-understood Fourier theory. Compared with other shape descriptors, FD has the following advantages, as for example, its construction is simple, each descriptor could be interpreted with a particular physical meaning, easy at shape matching because of the simplicity of normalization, able to describe both global and local features. With a small selection of coefficients, the FD is capable of capturing the overall shape features.

To overcome the problem of noise sensitivity and boundary variations in spatial domain shape methods, another approach of shape analysis called curvature scale space is proposed in [45]. To build a scale space for a shape, the shape boundary is filtered by low-pass Gaussian filters at varying widths. The curvature scale space descriptor aims at tracking the curvature zero-crossings (inflection points) on the smoothed boundaries. As the width (σ) of Gaussian filter increases, the shape boundary becomes smoother and thus less significant inflections are detected on the boundary. The inflection points that still remain represent a particular physical structure on the smoothed shape (e.g. corner, smooth joint, end and etc.). An interval tree is calculated through tracking the inflection points on different smoothed shape boundaries. This interval tree is also called curvature scale space contour image. The interval tree is interpreted by the peaks of tree branches from higher scales to lower scales. This interpretation is used as the curvature scale space descriptor to match with other shapes. The powerfulness of this approach stems from its ability to capture the location and the degree of convexity (or concavity) of curve segments on the shape boundary. These features are even very important to human perception in recognizing objects. However, the matching using the curvature scale space descriptor proves to be very complex and expensive. So there are some improved and more efficient versions of this approach, and an example is designed in [46].

1.3 Recent Research of Object Recognition

As stated before, according to their applications, object recognition techniques can be roughly divided as two categories: object classification and object identification.

A lot of works has been published on object classification recently. To compare the results for various classification techniques, several challenging databases have been set up, e.g., Calthech-101 [47], Caltech-256 [48], Graz-01 [49], UIUC textures [50] and Oxford flowers [52]. The focus of the research work on image classification has led to a great improvement on classification rate for these databases. For example, on the Caltech-101 database, the classification rate has been increased from under 20% in 2004 [48] to almost 90% in 2007 [49]. The state-of-the-art object classification techniques often include an intensive object learning process, which produces a particular set of parameters only suited for a specific classification task. The most commonly used and successful learning techniques are SVM [48, 54, 55, 56] and Boosting [51]. There is also a trend of combining different types of descriptors [49, 51, 52, 53, 54], which enables the classification to incorporate heterogeneous sources of data, such as texture, shape and color of the object. In [50, 52], different types of descriptors are assigned equal weights to be combined in the classification. In [52, 53], the weights for each type of descriptor are determined by a performance optimization process tested on a validation set. In [48], an optimal descriptors' kernel is learned through an SVM training process, in which the weight for each descriptor in the kernel is modified during the whole object classification to guarantee the optimal classification result.

Although for the image classification the training/learning of the objects has become the main trend, it has been shown in [57] that a method based on nearest-neighbor distance function can also achieve relatively good performance. The author has also combined different

descriptors with fixed weight, which is determined by the Parzen Gaussian kernel of each descriptor type. An optimal Naïve-Bayes method has been derived based on Euclidean distance and justified by the theoretical formulation [57].

Besides object classification, the problem of object detection is another critical part in many vision applications, such as image retrieval, scene understanding and surveillance systems. In Chapter 3 and Chapter 4 of my thesis, I have studied and compared two object identification strategies [58, 59] based on the object shape and texture information. Both techniques employ a single type of descriptor, SIFT and shape context, respectively, to describe and identify the target object in the complex scene.

The scale invariant feature transform (SIFT) descriptor is proposed by Lowe [69]. This technique actually combines a scale invariant region detector and a descriptor based on the gradient distribution in the detected regions. The descriptor is represented by a three dimensional histogram of gradient locations and orientations while the contribution of each of these bins is weighted by the gradient magnitude. The SIFT descriptor actually captures the texture feature of the object, while the shape context descriptor [74], similar to the SIFT descriptor, extracted the shape information from the object. The shape context descriptor is a two dimensional histogram with axes the log-distance and polar angle. The descriptor is applied on the edge points only and describes the edge distribution in the surrounding region of each contour point. With the appropriate modifications, shape context becomes invariant to rotation, shift and scale changes of the target object. In general, shape context descriptor is a simple, rich descriptor that enforces good shape matching and recognition.

The prototype objects are represented by the feature descriptors extracted from them. All these descriptor are stored in the database as the prior knowledge of the objects. The features are also generated on the test complex image and matched to their counterparts in the database according to the similarity measurement. Each of the identifier employs a special

clustering strategy to group the matched features in the complex image. The locations of the clusters suggest potential existence of the desired object. Finally, a verification process is applied to determine the presence of the target object. Both of these identification approaches have been implemented and applied on variety of prototype and complex images. In Chapter 3 and Chapter 4, the analysis of the results shows us the limitations and advantages of these two local feature based identifiers.

From the study and comparison of these two detection strategies, it is demonstrated that the identification method based on a single object feature could hardly comply with the task of detecting an object with various types of features. Therefore, following the trend in the object classification, a new object identification strategy is introduced in Chapter 5, which combines different types of descriptors to extract heterogeneous information from the object. Another widely used method is the sliding window scheme, which localizes the presence of object in the region with the peak confidence value. In our work, a localization method similar to the sliding window is employed.

The intra-class variations of the objects are the obstacle of any generic detection task. To overcome this difficulty, the state-of-the-art detection methods often involve a training phase [60, 61, 62]. These detection techniques are composed of various preprocessing steps, which make them quite time consuming. As a result, these methods can hardly meet the requirement of many applications to detect the object in real time. The other disadvantages of training based detection strategies are the need of many training images and the overfitting of parameters.

To overcome above problems, efforts have been made to achieve training-free object detection [63, 64, 65, 66]. These detection systems use only one image of the prototype as the query image, and search for this object of interest in the complex scene. In [63], the author introduced a novel descriptor named ‘self-similarity’, which catches the layout of the local

similarities of the image context, including color, edges and repetitive patterns. It is modeled as a function of a simple sum of squared difference (SSD) between a center image patch and surrounding image patches. According to the characteristic of the self-similarity descriptor, the 'non-informative' descriptors are filtered out. In [63], there are two kinds of 'non-informative' descriptors, i.e., the ones that cover uniformly colored or uniformly texture regions, and the ones in which the central patch is not similar to any surrounding patches. Afterwards, a modified version of 'ensemble matching' [67] strategy is applied. The descriptors on the query image are connected and formed as 'ensemble of descriptors', which is a probabilistic 'star graph' model capturing the geometric relationships of the descriptors. The strategy, similar to sliding window, generates a dense likelihood map in the size of the complex scene. The value of each point on the map represents the similarity between the query image's 'ensemble of descriptors' and the one centered at this point on the complex scene. The large value on the map suggests the presence of the target object at the corresponding point's location. In [66], a 'resemblance map' (RM) is constructed based on the cosine similarity between two sets of descriptors.

Following the trend of training-free image analysis [57, 63, 66], in this work, a new prototype-based object identification method is proposed to detect the target object in the complex scene, using only one image as the prototype's query image. Instead of using a single type of descriptors [58, 59, 63, 66], my method incorporates different types of descriptors to capture the heterogeneous features of the target object. The descriptors are modified to meet the requirement of the framework of identification system. Thus, the strategy is able to describe and identify various kinds of objects whose dominant features are quite different from each other. Inspired by the idea of building likelihood map in [63, 66], this work introduces a novel strategy to detect and localize the target object in the complex scene based on the likelihood map of image patches, which is more efficient and faster than calculating a similarity value at each point on the complex scene [63, 66]. The patches with high likelihood value may indicate

the presence of the target object. However, it is naïve to select the patch with the largest similarity as the location of the object. This is because the complex scene may not contain the target object. Thus a verification procedure and a threshold are introduced to determine the presence and location of the target object in the complex image.

Chapter 2

Shape Classification System

2.1 2D Shape Classification System Using Shape Context

In this chapter, a novel content-based technique for efficient shape classification and retrieval of 2D objects is presented. I designed a system which is of scale and rotation invariance.

Much of the work in this area uses a finite set of points taken from the object's boundary as the shape representation. Points can be selected on the basis of maximal curvature [85], distance from the centroid [86] or any criteria considered suitable to the shapes of objects involved. More sophisticated approaches parameterize the boundary as a closed curve and slide points along the outline to minimize an objective function (e.g. [87]). These methods produce good results but generally use expensive optimization algorithms. An alternative to finding the 'correct points' is to simply place points at roughly equal intervals along the boundary. Belongie et al. [74] used this approach effectively in their work on shape contexts.

In my work, I intended to extract salient points out of the shape boundary of the objects as the shape representation. Then the distances between the descriptors of the matched correspondent points between the prototype object and training object are weighted according to the degree of their saliency. It is expected that this new strategy, the implementation of weighted salient points (the points with large Harris corner measurements), could give us a better way to categorize objects based on the shape representation.

With the implementation of a Gaussian filter as the weighting factor for all points on the object silhouette, it is shown that instead of only considering the salient points or fancied areas of the object, a better result for the object categorization could be achieved when all parts or points on the silhouette are used.

2.1.1 Shape Context

It has been shown in [10] that the shape context technique is a powerful tool for object recognition tasks, which includes the classification from binary images and images of 3D objects under various poses.

The basic idea of shape context is illustrated in Figure 2.1(a). The shape of an object is represented by a discrete set of points sampled from the internal and external contours on the shape. These can be obtained from the locations of edge pixels found by an edge detector, which provides us a set $P = \{p_1, ..., p_n\}$, where $p_i \in \mathbb{R}^2$, of n points. Consider the set of vectors originating from a point to all other sample points on a shape. These $n-1$ vectors express the configuration of the entire shape relative to the reference point. One way to

capture this information is to describe the *distribution* of the relative positions of the remaining $n-1$ points in a spatial histogram. Concretely, for a point p_i on the shape, compute a coarse histogram h_i of the relative coordinates of the $n-1$ points given by

$$h_i(k) = \#\{q \neq p_i \mid (q - p_i) \in \text{bin}(k)\} \qquad (2.1.1)$$

where the sign # indicates the number of points which satisfy the condition given in the parenthesis. This histogram is defined to the shape context of p_i. The bins used by the shape context are uniform in log-polar space, making the descriptor more sensitive to the positions of nearby sample points compared to those of points further away. Consequently, the number of bins would determine the accuracy of the descriptor. In the case of too many bins, each bin would contain very few points, especially for the bins far away from the center, which makes the descriptor inefficient. On the other hand, if there are very few bins, all the points would be distributed within the limited number of bins which makes the descriptor unable to distinguish between different shapes, as different shapes would have similar shape context. Therefore, it is important to choose a sensible number of bins which would affect the efficiency and accuracy of the descriptor. In this work, the number of inner-circles with different radius, centered at the reference point was chosen to be 4, in order to compromise the trade-off between the accuracy and the complexity. These internal circles were segmented as circular sectors with central angle equals to 30°. This results 12 equal circular sectors for each circle and therefore gives a total of 60 bins as shown in Figure 2.1 (a).

The scale of the shape context is vital in the object recognition process. Different sizes of shape contexts describing the same part of objects which are similar in shape but different in scale.

As a result, the shape contexts of corresponding points or parts between model and test objects should be of the same scale. Thus the scale needs to be determined before the process of descriptor building. The process of the scale determination will be discussed later in section 2.1.5.

As for the experiments included in this work, the shape context is used to represent single objects, thus the scale of the descriptor is as the same scale as the entire object. All radial distances are first normalized by the mean distance, α ,between the n^2 point pairs in the shape, thus ensuring that the shape context of a point on a shape is invariant under uniform scaling of the shape as a whole.

As illustrated in Figure 2.1(a), shape contexts will be different for different points on a single shape *8*; however corresponding (homologous) points on similar versions of shape *8* in Figure 2.1(a) and Figure 2.1(c) will tend to have similar shape contexts. By construction, the shape context at a given point on a shape is not invariant under arbitrary affine transforms, but the log-polar binning ensures that for small locally affine distortions due to the pose change, intra-category variation etc., the change in the shape context is correspondingly small. In addition, the richness of the shape context descriptor makes it robust to noise and light occlusion, as indicated by the experiments reported in [93].

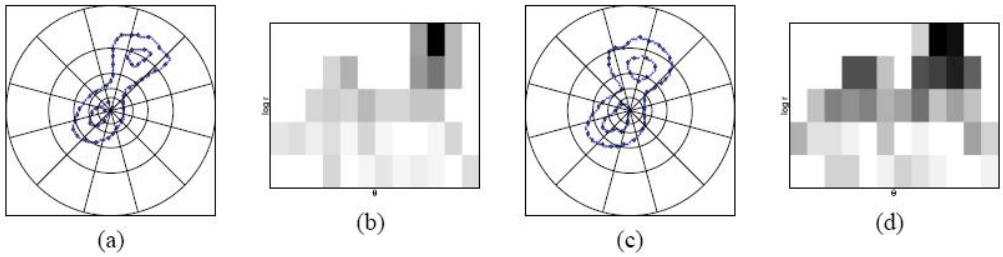


Figure 2.1 (a) On the contour of the digit eight *8*, the shape contexts are computed with respect to the circled sample points. (b) the log-polar histogram that has 5 bins for the polar direction and 12 bins for the angular direction. Each bin contains a count of the edge points falling into that bin. (c) shape context of a corresponding point on another digit *8*. (d) the histogram is similar to (b),the corresponding point on the other shape.

To measure the similarity of two shape contexts, the natural way is to use the χ^2 distance as the histogram measurement of similarity between shape contexts. It has been shown in [93] that this facilitates algorithms for solving the correspondence problems between two similar but not identical shapes such as shown in Figure 2.1(a) and (b). Consider a point p_i on the first shape and a point q_j on the second shape. Let $C_{ij} = C(p_i, q_j)$ denote the cost of matching these two points. The χ^2 distance between these two points is defined by

$$C_{i,j} = \frac{1}{2}\sum_{k=1}^{K}\frac{\left[h_i(k) - h_j(k)\right]^2}{h_i(k) + h_j(k)} \qquad (2.1.2)$$

where $h_i(k)$ and $h_j(k)$ denote the *K*-bin normalized histogram at p_i and q_j, respectively.

Given the set of costs C_{ij} between all pairs of points i on the first shape and j on the second shape, as shown in [93], minimizing the total cost of matching is subject to the constraint that the matching be one-to-one using the Hungarian method [97]. However, in this work, only the correspondent points of small costs are considered into matching process, which is faster and more realistic. To select the correspondent points between two shapes, the following strategies have been applied:

1. Nearest neighbor-based matching

The point p_i on the first shape is matched to the point q_j with the minimum cost C_{ij} on the second shape. The cost between two correctly matched points shall be very small. Therefore, to prevent the mismatches, a threshold is used here. Any two matched points with cost C_{ij} above this threshold will be disregarded. For convenience, the threshold is set as the 10% of the maximum C_{ij} between any points on two shapes. After the implementation of this matching strategy, it has been found that there are still quite a lot of mismatches in the result. Thus, there is a need to set another criteria to refine the results.

2. Cost ratio between the nearest and the second nearest neighboring points

For the point p_i on the first shape, the point q_j with minimum cost C_{ij} on the second shape is defined as the nearest neighbor. Also on the second shape, the point q_k with the second minimum cost C_{ik} is defined as the second nearest neighbor to point p_i. If the cost ratio of

C_{ij} / C_{ik} is below the certain threshold T, then the point p_i has a true match with the point q_j, i.e., if $C_{ij} / C_{ik} < T$, then p_i and q_j matched, where $T \in [0,1]$

This ratio was chosen because for a point p_i on the first object, if there is no true matched point on the second shape which would have a similar shape context as p_i, the costs of its nearest neighbor and the second nearest neighbor would not differ much. Thus the ratio value of the false match would reach 1. For simplicity and according to the results of the simulations, T is set to be 0.8.

2.1.2 Edge Detection

Canny edge detector [4] has been chosen in this work. The Canny edge detection algorithm is known to many as one of the optimal edge detectors. Canny's intentions were to enhance the many edge detectors already published at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper. In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there should be no response to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge pixels should be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the Canny edge detector first smoothes the image to eliminate the noise. Then it finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude above the second threshold, which is higher than the first one, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above the second threshold.

2.1.3 Corner Detection

Corners are used as salient points in the process of object matching in this work. Therefore it is important to choose an appropriate corner detector. Several corner detection methods [5, 6, 7] have been implemented and their performances were compared. The Harris corner detector was chosen because of its strong invariance to rotation, scale, illumination variation and image noise [94]. The Harris corner detector is based on the local auto-correlation function of an image region, where the local auto-correlation function measures the local changes of the image region with patches shifted by a small amount in different directions. Moravec [95] had first presented the discrete predecessor of the Harris detector, where the discreteness refers to the shifting of the patches.

Let I denote a 2-dimensional grayscale image. Given a point with coordinates (x, y) and a shift $(\Delta x, \Delta y)$ along x and y axis, respectively, the auto-correlation function is defined as

$$S(x,y) = \sum_x \sum_y w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2 \quad (2.1.3)$$

where $w(x, y)$ is the auto-correlation window which is a circularly weighted Gaussian window giving isotropic response.

By applying the first-order Taylor expansion to $I(x+\Delta x, y+\Delta y)$, it can be simplified as follows:

$$I(x+\Delta x, y+\Delta y) \approx I(x,y) + I_x(x,y)\Delta x + I_y(x,y)\Delta y$$

$$= I(x,y) + [I_x(x,y), I_y(x,y)]\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} \quad (2.1.4)$$

where I_x and I_y are the partial derivatives of I with respect to x and y, respectively.

Substituting equation 2.1.4 into 2.1.3, it follows that

$$S(x,y) = \sum_x \sum_y w(x,y)[I(x,y) - I(x+\Delta x, y+\Delta y)]^2$$

$$= \sum_x \sum_y w(x,y)\{I(x,y) - I(x,y) - [I_x(x,y), I_y(x,y)]\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix}\}^2$$

$$= \sum_x \sum_y w(x,y)\{[I_x(x,y), I_y(x,y)]\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix}\}^2$$

$$= [\Delta x \ \Delta y]\left\{\sum_x \sum_y \left(w(x,y)\begin{bmatrix} I_x(x,y)^2 & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y(x,y)^2 \end{bmatrix}\right)\right\}\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} \quad (2.1.5)$$

The Harris matrix, A, that captures the intensity structure of the local neighborhood, is defined by

$$A = \sum_x \sum_y \left(w(x,y) \begin{bmatrix} I_x(x,y)^2 & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y(x,y)^2 \end{bmatrix} \right)$$

$$= \begin{bmatrix} \sum_x \sum_y w(x,y) I_x(x,y)^2 & \sum_x \sum_y w(x,y) I_x(x,y) I_y(x,y) \\ \sum_x \sum_y w(x,y) I_x(x,y) I_y(x,y) & \sum_x \sum_y w(x,y) I_y(x,y)^2 \end{bmatrix}$$

$$= \begin{bmatrix} A_1 & A_{12} \\ A_{12} & A_2 \end{bmatrix} \qquad (2.1.6)$$

A corner is characterized by a large variation in S along both the directions of the vector $(\Delta x \ \Delta y)$. By analyzing the eigenvalues of A, this characterization can be expressed as: A should have two “large” eigenvalues for a corner point.

Let λ_1 and λ_2 be the eigenvalues of the auto-correlation matrix, A. Based on the argument discussed above, the following inference can be made:

1. If both λ_1 and λ_2 are small, so that the local auto-correlation function is flat (i.e., little change in S in any direction), the windowed image region has approximately constant intensity.

2. If one eigenvalue is high and the other is low, so that the local correlation function is ridge shaped, then the shifts along the ridge cause only a slight change in S and a significant change occurs in the orthogonal direction; this indicates an edge.
3. If both eigenvalues λ_1 and λ_2 are high, so that the local auto-correlation function is sharply peaked, then shifts in any direction will result a significant increase; this indicates a corner.

The calculation of the eigenvalues is computationally expensive, as it involves square root operations. Hence, Harris has suggested the following function R for the corner measurement,

$$R = \det(A) - \kappa \text{trace}^2(A) \qquad (2.1.7)$$

where κ is a tunable insensitivity parameter.

$$\det(A) = \lambda_1 \lambda_2 = A_1 A_2 - A_{12}^{\ 2} \qquad (2.1.8)$$

$$\text{trace}(A) = \lambda_1 + \lambda_2 = A_1 + A_2 \qquad (2.1.9)$$

Therefore, the algorithm does not have to actually compute the eigenvalue decomposition of matrix A. Instead it evaluates the determinant and the trace of A to detect corners. The value of κ is empirically set to be 0.04~0.15.

In this work, instead of Harris corner measurement, the measurement proposed in [96] has been used, as it is found to be more stable and efficient:

$$H = \det(A) / \text{trace}^2(A) \qquad (2.1.10)$$

After the corner measurement, a non-maximal suppression for corners has been implemented. The following parameter setups are used for this implementation and the Gaussian window.

1) The suppression patch radius is set to 3, according to the requirement of the corner detection.
2) Suppression threshold is set to 1000, considering the value of corner measurement.
3) For simplicity, the standard deviation σ of the smoothing Gaussian window is set properly as 2.
4) The size of the Gaussian window is $6\times\sigma$, which gives the best simulation results.

2.1.4 Orientation Invariance

To implement shape context, the orientation of each interesting point needs to be considered. The derivatives are used to calculate orientation of the point:

$$\theta = \tan^{-1}\left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}\right) \qquad (2.1.11)$$

Then before calculating the shape context for each point, the tangent orientation could be submitted. As the result, a specific point on the contour will have a unique edge point in its shape context, regardless of the rotation of the object. Therefore the orientation invariance problem has been solved.

2.1.5 Scale Invariance

Before building the descriptors, it is important to decide the scale of the reference part on the object or the whole single object, because it determines the property of the feature descriptor. The shape context scale of a point on the object shape would determine the number of pixels covered by the shape context. Thus, the scale will determine the histogram of the shape context.

As in the test database from this work, there is only a single object in each images, it is preferred to cover the whole object with the shape context. Hence, the scale of the shape context is chosen the same as that of the entire object. The mean distance, α , between the n^2 point pairs in the shape, is used to normalize the radial distance of shape context. Thus, it ensures that the shape context of a point on a shape is invariant uniform scaling of the shape as a whole.

In the future work, we are going to analyze the complex scene which includes multiple objects and object occlusion. Under this situation, it is more complicated to determine the scale of the object or part of object. For the scale selection method, Linderberg introduced a mature technique [8]. It has been shown that under a variety of reasonable assumptions the only possible scale-space kernel is the Gaussian function. Therefore the scale space of an image is defined as a function which is produced from the convolution of a variable-scale Gaussian. Also, in the case of detecting stable key point locations in scale space, the scale-space extrema in the difference-of-Gaussian function has been used [69].

2.1.6 Gaussian filter implementation

It has been observed that under various circumstances the objects in the image may be distorted because of occlusion or various types of transformations such as translation, rotation and scaling. As a result, some salient points or entire areas might not be detected or exist in the image. These types of problems can affect the recognition process. Therefore, by utilizing information of all of the available points of the object in the learning process, could improve the performance of the object recognition system.

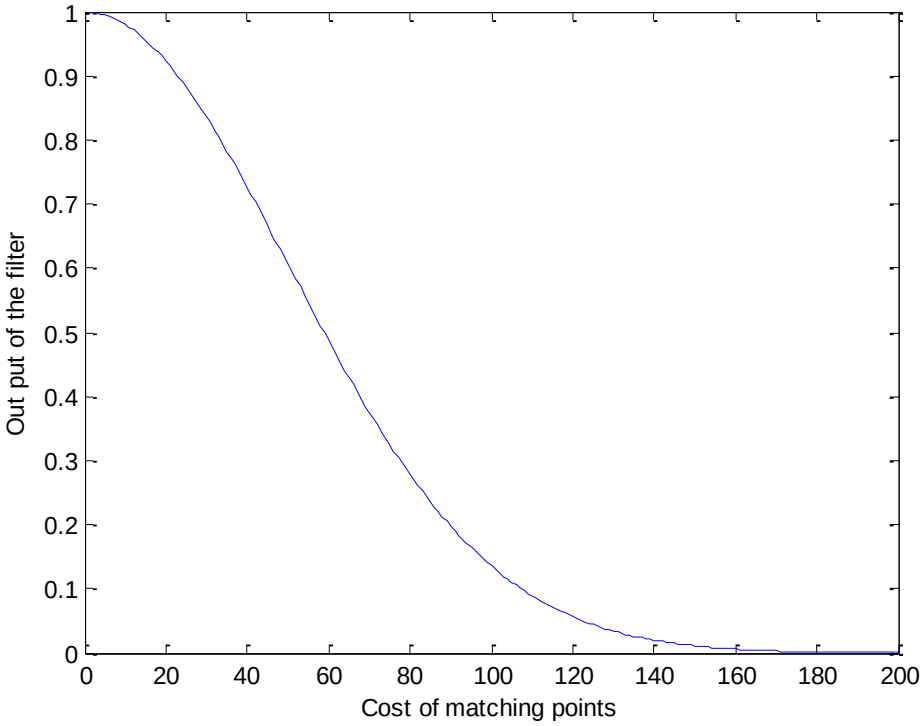


Figure 2.2 Gaussian filter for cost of matching points

The Gaussian filter is used here to modify the cost of the matching points. The output of the filter can be defined by:

$$G_{i,j} = ae^{\frac{-C_{i,j}^{\ 2}}{2\sigma^2}} \qquad (2.1.12)$$

where $C_{i,j}$ is the cost of matching points p_i and q_j, respectively. When the matching points have similar shape context descriptors, the cost $C_{i,j}$ is zero. In this case, with $a=1$, the contribution of this matching pair would be 1, as well. According to the values of $C_{i,j}$, the standard deviation of the Gaussian filter is set to 50, with which the Gaussian filter is able to produce required measurements.

As it can be seen from Figure 2.2, for the correspondent points on prototype and test objects, it is desired that the higher the cost of the matching points, the lower the output from the filter should be.

After the implementation of the Gaussian filter, it is expected to enlarge the gap between the different objects classes. This is because the sum of costs of all matching points is used to represent the cost between test object and prototype. Through this filter, the contribution of the large costs would decrease rapidly, and the more similar correspondent matching points with smaller costs would be emphasized in the learning process.

It is expected that this method would improve the object categorization performance based on the shape representation.

2.1.7 Assign Weights to Corners for Correspondence Matching

The basic motive behind the weighting strategy is to emphasize the effect of salient points in the matching process. It would improve the performance of object recognition based on shape representation. In this work, the corner points are referred as salient points.

One natural way to assign the weights is to use the corner measurement, as the corner points have much larger measurement than normal points on the shape. As discussed in section 2.1.3, the following corner measurement is used.

$$H = \det(A) / \mathrm{trace}^2(A) \qquad (2.1.13)$$

where A is the Harris matrix as mentioned before.

We also used the anistropic measurement [5] of the corner points, which is defined as follows, in order to take into account the anistropic property of each point:

$$c(x,y) = \frac{\left(\iint_{\Omega}(I_x^2 - I_y^2)dxdy\right)^2 + \left(\iint_{\Omega} 2I_xI_y dxdy\right)^2}{\left(\iint_{\Omega}(I_x^2 + I_y^2)dxdy\right)^2} \qquad (2.1.14)$$

where (x, y) is the point coordinates and I_x and I_y are the partial derivatives of I with respect to x and y, respectively. Ω is the operation window with the size of 5×5, which best suits the measurement $c(x, y)$.

In this work, weights are assigned in two different ways:

First, the measurements are assigned for the calculation of the cost of two shape contexts into equation (2.1.2):

$$C_{i,j} = \frac{1}{w_i}\frac{1}{w_j}\frac{1}{2}\sum_{k=1}^{K}\frac{(h_i(k)-h_j(k))^2}{h_i(k)+h_j(k)} \qquad (2.1.15)$$

where w_i and w_j represent the measurements of the points p_i and q_j on the prototype and test object, respectively.

The corner points would have large corner measurement. Therefore, the above equation results a smallest cost between the correspondent corner points than that between the normal points. Therefore the closest correspondent points between test and prototype objects within minimum cost could be easily detected.

The other way is to assign weights employing the process of shape matching. After matching the correspondent points between prototype and test object, a weighting factor is build for each pair of matching points according to:

$$W_{i,j} = \frac{w_i + w_j}{\frac{\sum_{k=1}^{N_T} w_k}{N_T}} \qquad (2.1.16)$$

where N_T is the number of matched points on test object, and w_i and w_j follow the definition as in equation (2.1.15). The corner measurements of each pair of matched points are normalized by the average corner measurement of the points on the test object.

Since the sum of the costs of each pair of matching points is used as the parameter to judge the shape matching, the weighting factor $W_{i,j}$ is used to build the cost of the object shapes in the following way:

$$C_{pq} = \sum_{i,j} W_{i,j} G_{i,j} \qquad (2.1.17)$$

where $G_{i,j}$ is the output of Gaussian filter as defined in equation (2.1.12). Moreover, it can be proved that when a pair of matching points are both corners, the weight of the matching points would be large, and normalized by the average corner measure of all the points on testing object. Thus, this weighting strategy would emphasize the contribution of corners, which are treated as salient points in the matching process. By emphasizing the effect of corner points in the matching process, it is believed that the performance of the matching based on shape representation will be significantly improved. Also it is noticed that due to the implementation of Gaussian filter discussed in section 2.1.6, the cost of two objects is of reverse to the similarity of two shapes, i.e., the more similar the shapes, the larger the cost.

2.2 Experiment and Result

2.2.1 Dataset

The technique proposed in Section 3 is tested on the benchmark MPEG-7 database. 5 classes of objects, each of which contains 10 observations, are chosen, as shown in Figure 2.3.

2.2.2 Correspondent Points

After the boundaries of the objects are extracted by the Canny detector, each point on the boundary is used to build a local edge point histogram descriptor to construct the Shape Context for the entire object, as discussed in Section 2.1.1. Then, for all points on the test object, their correspondences with the prototype edge points are detected, by finding the nearest Procrustes distance between Shape Contexts. The examples are illustrated in Figure 2.4 and Figure 2.5. In these examples, the points on the prototype plane shape are matched to the correspondent points on a similar plane and on a completely different object which resembles a mechanical tool, respectively.

In order to select the true correspondent points between test object and prototype, the method of cost ratio between the nearest and the second-nearest neighbouring points is implemented, as discussed in Section 2.1.1. The results are shown in Figure 2.6 and Figure 2.7. For the two planes which belong to the same object class, the number of correspondent point estimated is 92, which is four times the number of correspondent point estimated between the tool and the plane. This illustrates the effectiveness of this method to select true matched points.

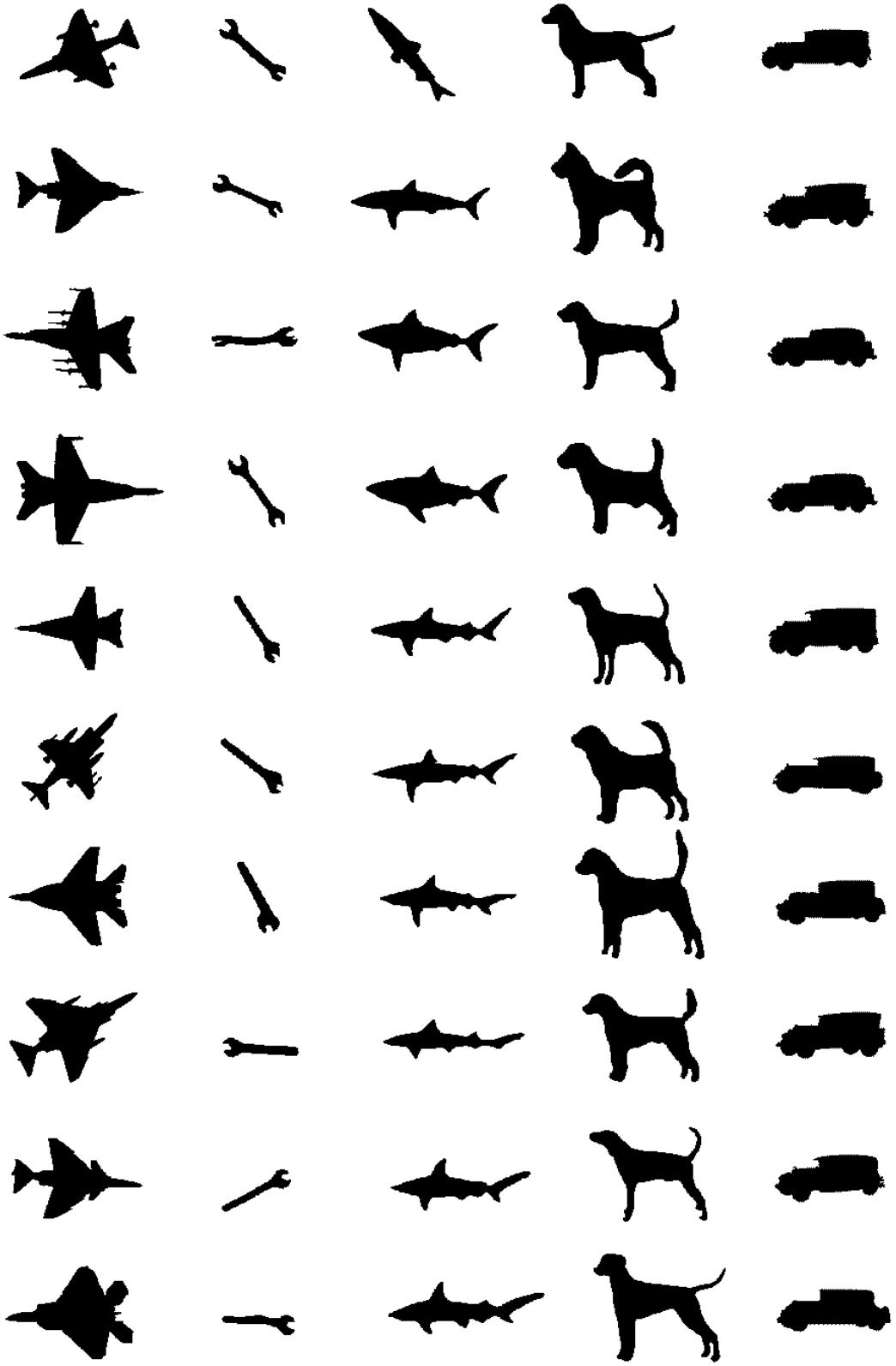


Figure 2.3 MPEG-7 Database, 5 classes with 10 observations each

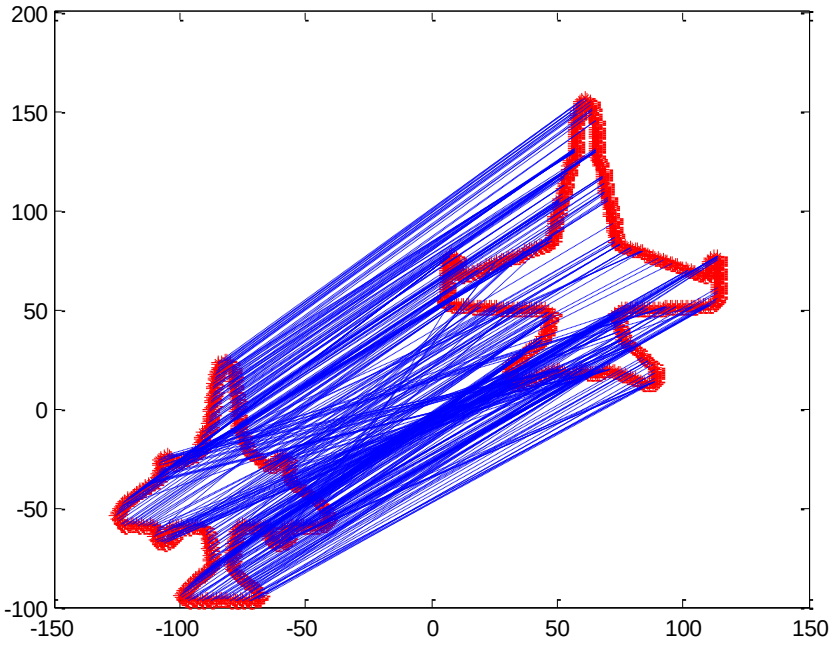


Figure 2.4 Matched points between two planes

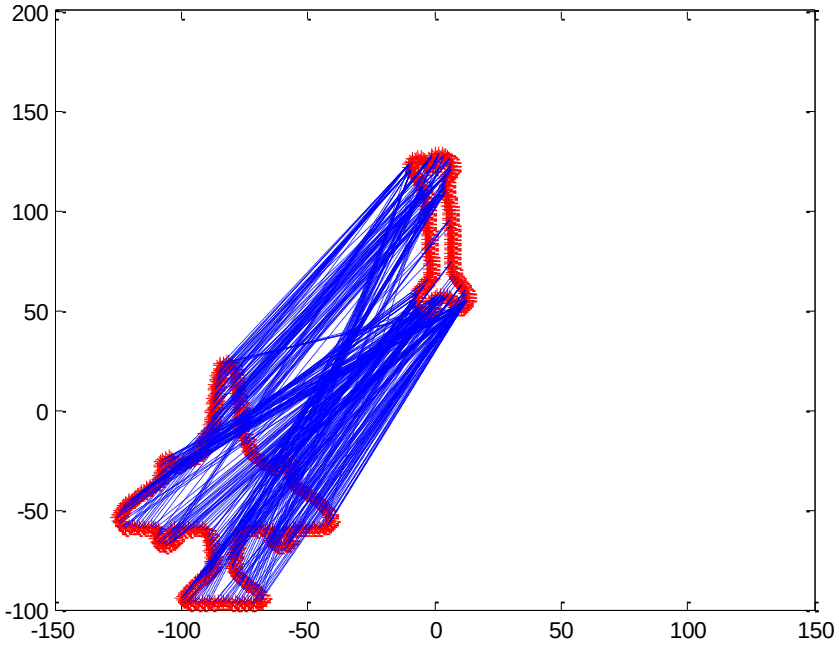


Figure 2.5 Matched points between plane and tool

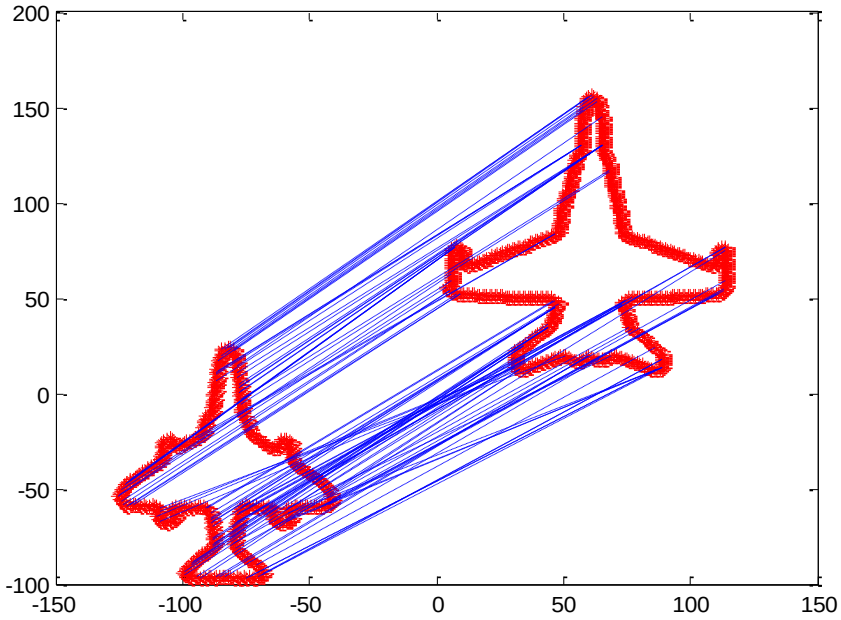


Figure 2.6 True matched points between two planes

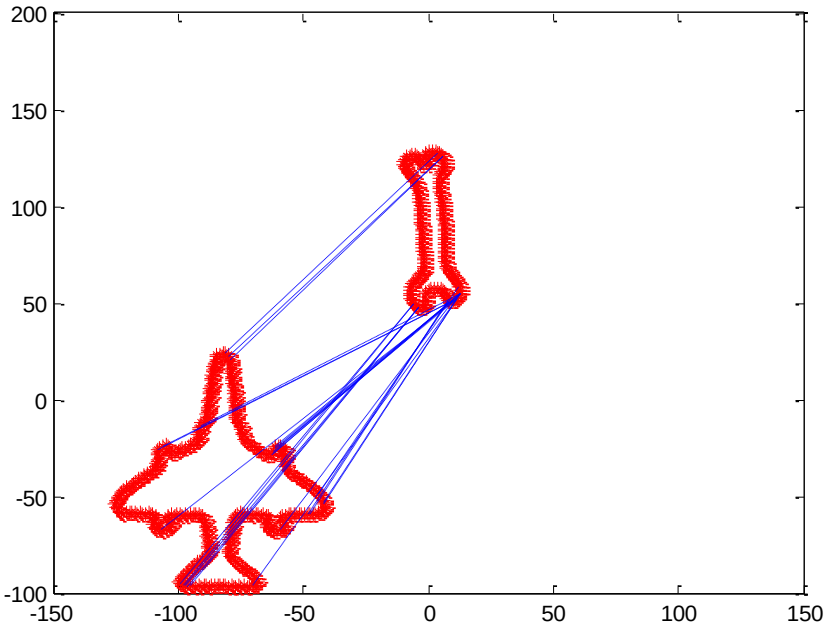


Figure 2.7 True matched points between plane and tool

2.2.3 Class Distance

After selecting the estimated matched points, their costs of the matched points are put through the Gaussian filter, and the filtered outputs are summated to formulate the cost between the test object and prototype. As shown in Table 2.1, one object is selected from each class and each entry represents the average distance between the selected object and all objects in each class. It can be observed that for the same class, the average cost is much larger than that of other classes. Therefore, this average distance could be set as the criterion for the object classification. For the 50 objects tested in my work, this strategy gives us zero error.

	Plane	Tool	Shark	Dog	Car
Plane class	99.2197	0.1594	2.7464	2.7349	1.6072
Tool class	0.0552	87.8275	23.6516	0.0006	4.5146
Shark class	3.4045	7.4174	64.1107	1.3818	3.9389
Dog class	5.0149	0.005	0.0674	92.8255	0.2347
Car class	0.9299	12.4832	12.8939	0.3225	134.3957

Table 2.1 Average distances from five objects to each class.

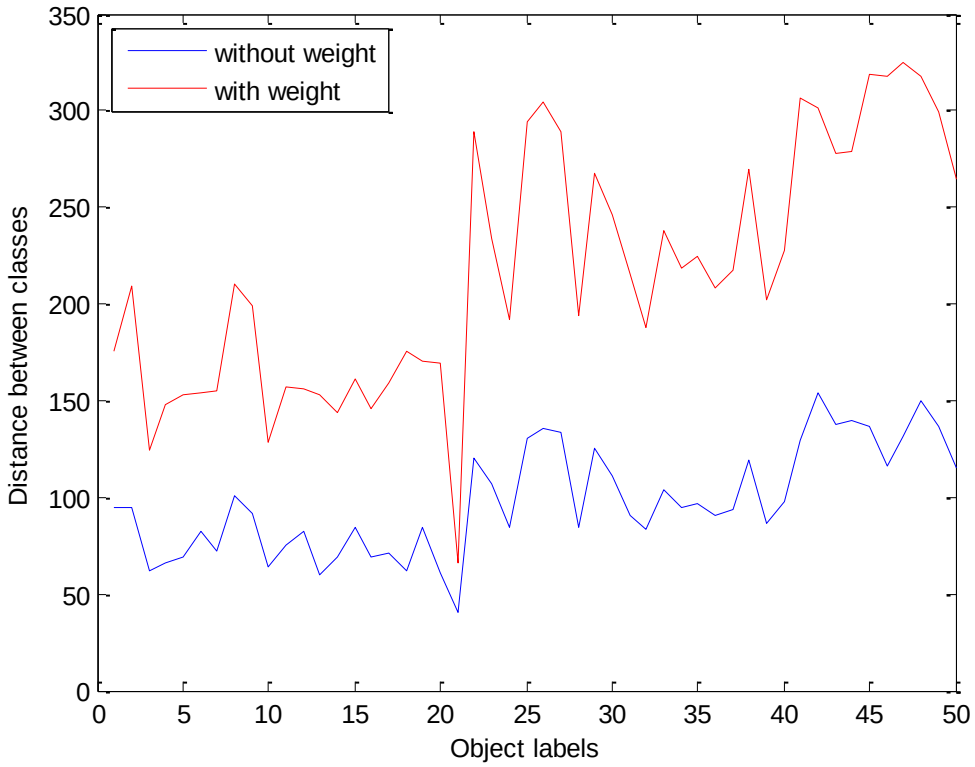


Figure 2.8 The distance between the same class and most similar one. Y axis stands for the distance between same class and most similar class, X axis refers to the object label.

After the implementation of the weighting factor, defined in equation (2.1.16), the average cost of test object and objects among the same class is compared with that of the most similar class with smallest cost. As shown in Figure 2.8, the y axis value represents the distance between the same class and the most similar class, and the x axis refers to the object labels. It is obvious that with the weighting factor, the distance between classes is larger than that without weight. Thus, this gap between these two lines gives more tolerance for object distortion or transformation in shape classification.

For the MPEG-7 database shown in Figure 2.3, my shape classification strategy has successfully classified all of the shapes. However, this strategy can only be applied on artificial object contours. For the shapes of objects in the real world, this shape classification technique hardly works. The reasons are as follows. First, it is impossible to extract the exact shape of a target object out from a complex scene. Second, the view point variance produces different shape contours for the same object. Finally, the target object may have only part of its shape represented in the image. Therefore, to solve the problem of object recognition, I have searched for more realistic and practical techniques, which are studied and analyzed in Chapter 3 and Chapter 4.

Chapter 3

Study of Object Identifier Based on Shape Information

In this chapter, an object identifier designed based on the shape feature of the object has been analyzed. It is proposed in [58], which employs the shape context descriptor to represent the object based on its shape features and then clusters the matching descriptors in the identification process. Afterwards, this technique has been performed on several real life complex images. Its efficiency and robustness have been tested. Moreover, the limitation of this method has also been discussed in the end of the chapter.

3.1 Shape Context Based Object Identifier

3.1.1 Shape Context Descriptor

Belongie [74] first proposed the idea of Shape Context to describe the characteristics of the object contour shapes. Before the Shape Context descriptor is constructed, it is required that

the edge map (collection of edge pixels) of the image is obtained by applying an edge detector. In our work, we applied Canny edge detector [84] to form the edge maps of all images.

Each pixel on the edge map is an edge point, on which a Shape Context descriptor will be build. A set of edge points $\{p_1 \ldots p_m\}$ are used to represent the object P, notated by $P \equiv \{p_1 \ldots p_m\}$. This notation will be used throughout this work for reasons of simplicity. Nevertheless, each edge point is associated with a 2-D vector of its spatial coordinates, notated by $p_i \equiv (x_i, y_i)$. The Shape Context descriptor of an edge point p_i is constructed as a two dimensional histogram with each bin representing the number of edge points in a pre-defined specific area of p_i's neighborhood. It is calculated as:

$$H_K(p_i) = (h_1(p_i), h_2(p_i), \ldots, h_K(p_i))$$

Where $h_k(p_i)$ expresses the number of contour points in the k th bin, mathematically defined as:

$$h_k(p_i) = \#\{p_j \neq p_i \mid p_j \in P, p_j \in bin(k)\}$$

The neighborhood of the Shape Context descriptor, as mentioned above, is a circle with its center at the pixel p_i. This circle's radius is represented as R. Each bin of the histogram is a uniform partition of the circle in the log-polar space (r, θ). Instead of the Cartesian space, the log-polar space is applied in forming the Shape Context descriptor. This is because that the log-polar space can offer the shape descriptor the ability to focus more on the closer neighbors of the point of interest. The sampling rates of both the polar angle θ and the logarithmic distance r from the edge point p_i determine the resolution of the circle (number of bins). It is assumed that there are N bins for the logarithmic distance and M bins for the polar angle, which defines the log-polar space (r, θ):

$$r \equiv \{r_i = \frac{R_i}{N}, i = 1,...N\}$$

$$\theta \equiv \{\theta_i = \frac{2\pi i}{M}, i = 1,...M\}$$

With these three parameters R , N , M , the Shape Context descriptor can also be expressed as $H_{R,N,M}(p_i)$. Therefore, in our work, by extracting a set of Shape Context descriptor $H_{R,N,M}(p_i)$, each of which describes the shape characteristic of an edge point's neighbor area, the shape feature of the object is represented as:

$$SC_P \equiv \{H_{R,N,M}(p_i) \mid p_i \in P, i = 1,...m\}$$

By intuition, it is noticed that the size of the neighborhood of the Shape Context descriptor will largely affect the performance of the matching and identification strategy. Consequently, this size must be set and examined carefully to obtain a good identification and detection result. According to the size of the neighborhood of the reference edge point, the descriptors can be defined as two categories: global and local shape context descriptors. A global shape context descriptor has a neighborhood which is large enough to cover the whole object. It is effective to detect the whole shape of the target object in a complex scene. The essential assumption of the application using the global descriptor is that the whole target object is present in the complex scene. This assumption can be unrealistic when deal with real world problems. When there is distortion or partial occlusion of the target object in the complex scene, the performance of the global descriptor is unreliable. Thus the second type of Shape Context descriptor, local descriptor is used to overcome this weakness. The local shape context descriptor catches the local shape features covered by the neighborhood of the reference edge point. This local shape features of the target object are identical to the prototype's correspondent shape features even when the other parts of the target object are occluded or missing in the complex scene. Thus the local shape context descriptor is used under the

situation of object distortion and occlusion. Under this circumstance, the size of the descriptor need to be necessarily determined to extract the shape features which are most distinctive in detecting the correct target object in the complex scene.

The shape context descriptor is easily constructed and at the same time provides a robust performance in the task of image shape retrieval. It catches the shape feature of the object regardless of the object's appearance in the image, including the translation, scale and orientation differences. The translation invariance is achieved because of the shape context descriptor's structure. The tangent direction for each point on the edge map is assigned as the positive x-axis of the coordinate system, based on which the edge point's shape context descriptor is constructed. In this way, the rotation invariance has been achieved for the shape context descriptor. For the scale invariance problem, we consider the situation in which the entire target object is in the complex scene and the neighborhood circle of the descriptor covers only the target object excluding other objects. Based on these assumptions, the global shape context descriptor can be applied. The radius R of the log-polar (r,θ) space of the descriptor is normalized by the mean distance of all pairs of edge points in the shape edge map. For other circumstances in which the global shape context descriptor is not appropriate, there is another strategy used to tackle the scale invariance problem. For each edge point, multiple shape context descriptors are built with different radii. The optimum scale for the descriptor is estimated according to the matching cost between the matched points on the prototype object and the detected target object in the complex scene. The radius (scale) which yields the minimum matching cost is selected. And the matching strategy will be introduced in the following section.

Assume after the edge detection, a complex scene is described as a set of edge points $S \equiv \{s_1 ... s_n\}$, same as the prototype object P. To estimate the similarity between two objects' shapes can be understood as to estimate the similarity between the two sets of shape context

descriptors. Therefore, in the work of [58], the object detection problem has been derived to finding a collection of edge points on the complex scene, whose shape contexts descriptors match the set of descriptors of prototype object with maximum similarity. For each prototype edge point in the collection, a matched point in the complex scene is decided according to the cost of matching these two points. The maximum similarity between two objects' shapes means the minimum cost of matching the correspondent sets of edge points. The matching strategy is introduced later on, while the cost of matching two edge points is represented as the χ^2 distance between their shape context descriptors. More specifically, to match a point s_i on the scene S to a point p_j on the shape P, the cost C_{ij} is calculated as:

$$C_{ij} = \frac{1}{2}\sum_{x=1}^{X}\frac{[h_x(p_j) - h_x(s_i)]^2}{h_x(p_j) + h_x(s_i)} \quad (3.1.1)$$

where $h_x(\cdot)$ stands for the number of edge points in the xth bin and X is the total number of bins estimated as $X = N \times M$. The above expression yields an $L_P \times L_S$ cost matrix C, where L_P, L_S are the number of edge points on P and S, respectively.

Once the cost $C_{i,j}$ of matching edge points is at hand, for each point p_i, the corresponding point $s_{\alpha(i)}$ on the complex scene is decided based on the point mapping strategy $\alpha : S \to P$.

3.1.2 Many-to-One Edge Point Matching

To detect the target object from the real life complex scene, the shape feature of the object has been extracted by constructing the shape context descriptors. The next step is to match the descriptors from the complex scene to the descriptors of the prototype objects, which is the same to matching the edge points of the complex scene and the prototype. Intuitively, a

one-to-one matching strategy [98] would be suggested. However, it is not realistic with this task. The complex scene often contains additional objects, noisy background and all other types of signals which provide irrelevant edge points. A matching strategy is needed to prevent these irrelative edge points from matching with the edge points on the prototype object. Moreover, part of the target object in the complex scene may be missing, or occluded by other objects. Under this situation, some of the edge points on the prototype will have no matches. At last, the complex scene may include several objects with shapes similar and/or identical to the prototype, therefore an edge point on the prototype is matched to multiple scene points. According to the discussion above, the one-to-one matching between the prototype points and the scene points needs to be abandoned, because it is restrictive to the problem and fails to detect multiple target objects in the complex scene. For the same reason, instead of the one to one matching, the authors in [58] have employed a many-to-one corresponding matching strategy.

As discussed above, a $L_P \times L_S$ non-negative cost matrix C has been calculated, whose rows and columns correspond to the edge points collection P and S, respectively. Afterwards a point pairs set E with the size $(L_P+1)\times L_S$ is introduced. The point pairs in set E are the edge point index pairs, which represent all the possible matches between point sets P and S. For example, $(i, j) \in E$ implies that the i th edge point from P is matched to the j th edge point of S. If an index pair subset A of the set E is decided, the prototype edge points are matched to the scene points according to the subset A. Therefore the task has been evolved to select the subset A satisfying an optimization criteria, which ensures the minimum accelerated costs of matches and set up later on.

It is first assumed that all edge point descriptors from the complex scene are matched to the prototype edge points, even if the points are of irrelevance. To deal with the irrelative scene points, a “dummy” point is introduced adding to the set P. This “dummy” point changes the

dimensions of the cost matrix C into $(L_P+1)\times L_S$, by adding a new row with the cost ε_d corresponding to the "dummy" point [58]. The value of ε_d is set as the threshold for matching of the points. If the cost of two points is smaller than ε_d, the scene point will be matched to the dummy point. In this case, it is concluded that for the edge point on the complex scene, there is no existing available point on the prototype edge map and this edge point is disregarded.

As discussed above, to detect the target object in the complex scene, the shape contexts descriptors have been built for edge points both on prototype and complex scene. Then the cost matrix C and points pair set E is formed. Afterwards, there is a need to select a subset A, which matches the edge points from prototype and complex scene with most similarity. The subset A is of size L_S, and represented as $A=\{(p_i, s_{\alpha(i)}, i\in[1, L_S])\}$. The essential criteria for the selection of matching points set $A\subset E$ between the prototype and complex scene is to minimize the accelerated cost of each matching of points pair, denoted as $\sum_{i=1}^{n} C_{i\alpha(i)}$. The cost of matching has a constraint that it is a many-to-one match, which states that each edge point from complex scene is matched to only one prototype edge point while each edge point of prototype is matched to at least on scene edge point. This matching strategy is formulated as estimating a function $\alpha:\{1,...,L_S\}\rightarrow\{1,...,L_P+1\}$ and each index in $\{1,...,L_P+1\}$ can be matched more than once. If a matching index pair is directed by $\alpha(i)=L_P+1$, the scene edge point s_i is not matched to any prototype edge point but the "dummy" point.

According to the one-to-one assignment problem [98], this many-to-one matching is a modification with the extended cost matrix C as the input and matching point pairs as the output of the assignment task. It is calculated as:

$$\min \sum_{ij} C_{ij} x_{ij}$$

s.t $$\sum_{j} x_{ij} = 1, \forall i \qquad (3.1.2)$$

$$\sum_{i} x_{ij} \geq 1, \forall j \qquad (3.1.3)$$

$$x_{ij} = 0 \text{ or } 1, \forall i, j$$

In the equations, $x_{ij} = 1$ states that scene point s_i is matched to prototype edge point p_j. When $x_{ij} = 0$, the two points s_i, p_j are not matched. It is guaranteed by constraint in equation (3.1.2), that each edge point of complex scene is matched to only one prototype edge point. The constraint equation (3.1.3) makes sure that each edge point of prototype object can be matched to more than one scene edge point.

When points are matched, the above algorithm is simplified as follows. For each row i in the cost matrix C, the column $\alpha(i)$ with the minimum value of row i is selected and defined mathematically as:

$$\alpha(i) = \text{argmin}\{C_{ij} : j = 1, ..., L_P + 1\}$$

This matching strategy also solves the task of multiple target object detection in the complex scene. If more than one object in the complex scene are similar or identical to the prototype, the shape context descriptors estimated based on their shape features are also similar to the ones of the prototype. After the many-to-one matching, each edge points on the complex scene are matched to the correspondent prototype edge point with the most similar shape context. Then these matched points will be clustered according to their locations, as explained in the next section.

3.1.3 Clustering of Matched Points on Complex Scene

After the matching process discussed in the last section, for the edge points set $\{p_i \mid i = 1,...,K\}$ on the prototype, a correspondent points set $\{s_{\alpha(i)} \mid i = 1,...,J\}$ has been located on the complex scene. In this section, the locations of these matched points on complex scene are used to identify the regions of interest, which may suggest the presence of target object in the complex image.

If the target object or its transformed version is present on the complex scene, after the matching process, it is expected that regions of dense and/or sparse distributions of matched points are located on the complex scene. At the regions where target object presents, the distributions are denser than those of the regions containing irrelevant objects or background on the complex scene. Thus the dense distribution indicates the possible presence of the target object, while sparse distribution indicates the existence of less similar object. The isolated points are usually mismatched and shall be disregarded. As a result, the matched points need to be partitioned according to their distributions.

In [58] the Subtractive Clustering [99] technique has been applied to partition the data set into separate groups. Assume between points in a group, the minimum similarity is $\delta_{\min}$, and the maximum similarity with the points from other groups is $\beta_{\max}$. The basic idea of clustering is to form points groups whose $\delta_{\min}$ is larger than $\beta_{\max}$.

Consider a data set $X = \{x_1, x_2, ..., x_n\}$ in the 2-dimensional space $\mathbb{R}^2$ where x_i is a vector of two entries representing the coordination of the *i*-th point. u_i and v_i are the first and second dimensional coordinates of point x_i, respectively. Then the density function at point x_i is defined as:

$$D(x_i) = \sum_{j=1}^{n} \exp\left[-\frac{\| x_i - x_j \|^2}{(r_a / 2)^2} \right] \qquad (3.1.4)$$

where r_a is a positive constant representing a neighborhood radius and $\| x_i - x_j \|^2$ is the Euclidean distance between point x_i and x_j. It is intuitively observed that the more neighbors a data point has, the larger the density value this data point has.

An iterated modification process is applied to choose the centroid of the clusters. Each time the data point with the largest density is selected as the centroid, x_{C_k}. Then the density values of all points are updated according to the following equation:

$$D^k(x_i) = D^{k-1}(x_i) - D^{k-1}(x_{C_{k-1}}) \exp\left[-\frac{\| x_i - x_{C_{k-1}} \|^2}{(r_b / 2)^2} \right] \qquad (3.1.5)$$

where r_b is a positive constant. The purpose of the modification is to eliminate the overlap effects of the identified cluster centers. At the k th round, the point x_i is assigned the density $D^{k-1}(x_i)$ inherited from previous round $k-1$, reduced by an amount inversely proportional to the distance between x_i and the cluster center C_{k-1} selected from previous round. Therefore the points close to the cluster center will have significantly reduced density measures, which reduce the chance of being selected as new cluster center afterwards.

After being estimated, the clusters are refined using a multistage method. First the clusters contain non-informative shape features are eliminated. Then the clusters with sparse distributions of points are discarded according to the density metric explained in the section 3.1.4.

3.1.4 Object Identification

After clustering of the matched points on the complex scene, regions containing these points groups have been located on the image. The matched points with no close neighbor remain ungrouped. These regions may suggest the presence of the target object. However, they may also be mismatches clustered together. Therefore, a data mining process is applied to determine whether these regions contain the target object or part of the desired object.

The matches between the prototype and complex scene edge points is denoted as $\{(p_{i,}s_{\alpha(i)}) \mid i=1,...,J\}$, where J is the number of matched scene points. The partitioned clusters of points is represented as $\{CS \mid w=1,...,W\}$, where W is the number of clusters.

The more matched points appeared in one region, the larger the possibility that the region contains the target object or part of the object. Contrarily, when the isolated points are matched to prototype edge points, it is only suggested that they appear to be mismatches. Therefore, the isolated matched points on the scene are eliminated.

For each cluster of points on the complex scene, a measurement called slope deviation is designed in [58] to analyze the shape information carried by the clusters, calculated as following equation:

$$T = \frac{1}{n-1}\sum_{i=1}^{n}(t_i - \gamma_s)^2 \quad (3.1.6)$$

where n is the number of matched scene points in the cluster. And t_i is the tangent of slope of the line formed by the point (x_i, y_i) and the cluster center (x_c, y_c). It is denoted as:

$$t_i = \arctan\left\{\frac{y_i - y_c}{x_i - x_c}\right\} \quad (3.1.7)$$

In equation (3.1.6), γ_s is the average of t_i of all points in the cluster, defined as:

$$\gamma_s = \frac{1}{n-1}\sum_{i=1}^{n-1} t_i \qquad (3.1.8)$$

From the definition of T, it is actually the deviation of tangents of all the lines connecting the cluster points and center point. If the points in the cluster are on a straight line and the center point would consequently be the center of the line, T would be close to zero.

From the information theory, it is known that a straight line carries the least information compared with other geometric shapes. Also straight lines are the common parts of shape edges. Thus the straight lines from the irrelevant object or background noise in the complex scene are highly likely matched to straight lines on the prototype, which lead to false object detection. Therefore, the clusters in which the contained matched points form straight lines need to be disregarded. This is done by set a threshold T_thresh for the measurement T. If T is smaller than the threshold, the cluster will be eliminated as it is suggested that the cluster points are on a straight line. So the clusters remained shall satisfy the following condition:

$$T > T\_thresh$$

Apart from T, another measurement called neighborhood density D is introduced in [58] as well. It is defined as:

$$D = \frac{n}{A}$$

where n is the number of points in the cluster and A is the maximum distance between the cluster points and the center of the mass, calculated as:

$$A = \max_i \sqrt{(x_i - x_{cm})^2 + (y_i - y_{cm})^2} \qquad (3.1.9)$$

where (x_{cm}, y_{cm}) are the coordinates of the center of mass, given by the following equations:

$$x_{cm} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$y_{cm} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

The neighborhood density D measures the "crowdedness" of a cluster and describes the population of points per unit area in the cluster. The basic idea of the identification is to find a set of scene points whose shape features are similar to those of another set of points on the prototype. If two sets of points are homologous, they share the same spatial distribution. Therefore, if the scene points of a cluster are densely matched to the points of a neighborhood of the prototype edge points, the cluster represents itself as a candidate region enclosing part of the target object. The more matched points in the cluster, the highly possible that the cluster covers a part of the target object. According to above discussion, the density measure D is important in determining the candidacy of the cluster. A threshold D_thresh is set up and all the clusters need to meet the following condition:

$$D > D\_thresh$$

Both of the thresholds, T_thresh and D_thresh, are constants and defined experimentally.

3.2 Simulation of the Object Identifier based on Shape Context

3.2.1 Configuration of the Identifier

To build the Shape Context descriptor, the edge map of the image needs to be detected first. In my work, the Canny edge detection method [84] has been applied. Instead of using only the

edge points on the shape contour, all extracted edge point are used to build for the descriptor. Thus the edges at inner part of the object also contribute to the descriptor. Some of these inner edge points may be yielded from the pattern features on the surface of the object. This provides the Shape Context descriptor the ability to catch the texture information on object's surface to some extent.

The number of bins in the histogram of Shape Context descriptor is decided by the number of partitions on the log distance r and polar angle θ in the log polar space of the descriptor. We follow the settings in [74], which sets the number of bins on r to be 5 and on θ to be 12. Thus the total number of bins of the descriptor histogram is 60. The size of the Shape Context descriptor is of vital importance in the later matching and identification stage. In this work, a local shape context descriptor is used set as a circle covering the edge point's neighborhood. Its radius R is set to be 20% of the largest distance between edge points on the prototype object. In this way, when the target object in the complex scene is occluded or part of it is missing, the descriptors, which catch the shape information of the existing part of the target object, can still be matched to the correspondent points on the prototype image. To provide the shape context descriptor the ability to describe the shape information subject to rotation invariance, the tangent direction of each edge points is assigned to be the descriptor's orientation, which is regarded as the positive x-axis of the coordinates system of the log-polar space.

In the process of detection of potential candidate region on the complex scene, the subtractive clustering has been applied to partition the matched scene points into groups, whose locations indicate the appearance of the target object. During subtractive clustering, the influence range of each cluster center determines size of the candidate region. Thus the size of the cluster needs to be appropriately defined. Intuitively, the higher the scale at which the target object appears on the complex scene, the larger the radius of the cluster influence should be. In my

experiments, the objects need to be detected out form a large scene consisting of many objects, in which the target object only appears in a small part of the image. Therefore, a relatively small cluster size has been set to guarantee the accurate location of the detected target object. For each cluster center, the region of influence is an ellipse with the major and minor radii set to be 20% of the width and the height of the data space of all matched scene points.

There are two thresholds in the object identification process. T_thresh is used to determine whether clustered edge points form a straight line, and its value is set to be 0.1. For D_thresh, which stands for the threshold for cluster density, is set equal to 2.

3.2.2 Simulation Results

The object identification approach has been applied on a variety of images to detect objects with different features. It has been proved to be general and can be applied on various identification tasks. However, as test shows, this technique has its limitations.

In Figure 3.1, there is a prototype image of a red ipod and below it there is a complex scene (of size 448x299 pixels) containing several objects. The task is to detect the target object, the red ipod, in the complex image. The identification approach based on Shape Context is applied first. The edge map of the complex image is shown in Figure 3.2, and the points which are matched to their counterpart on the prototype image are highlighted in green. It is observed that the matched points are scattered over the complex scene and apart from the correct matches, they are raised from edge points of other objects and the background. The following process is to cluster these matched points to refine the results.

Figure 3.1 The prototype image is shown above and the complex image in which the target object needs to be detected is shown below.

Figure 3.2 The edge map of the complex scene, extracted by applying the Canny edge detector. The blue points in the image are the edge points, and the green points are the points which are matched to the edge points on prototype image.

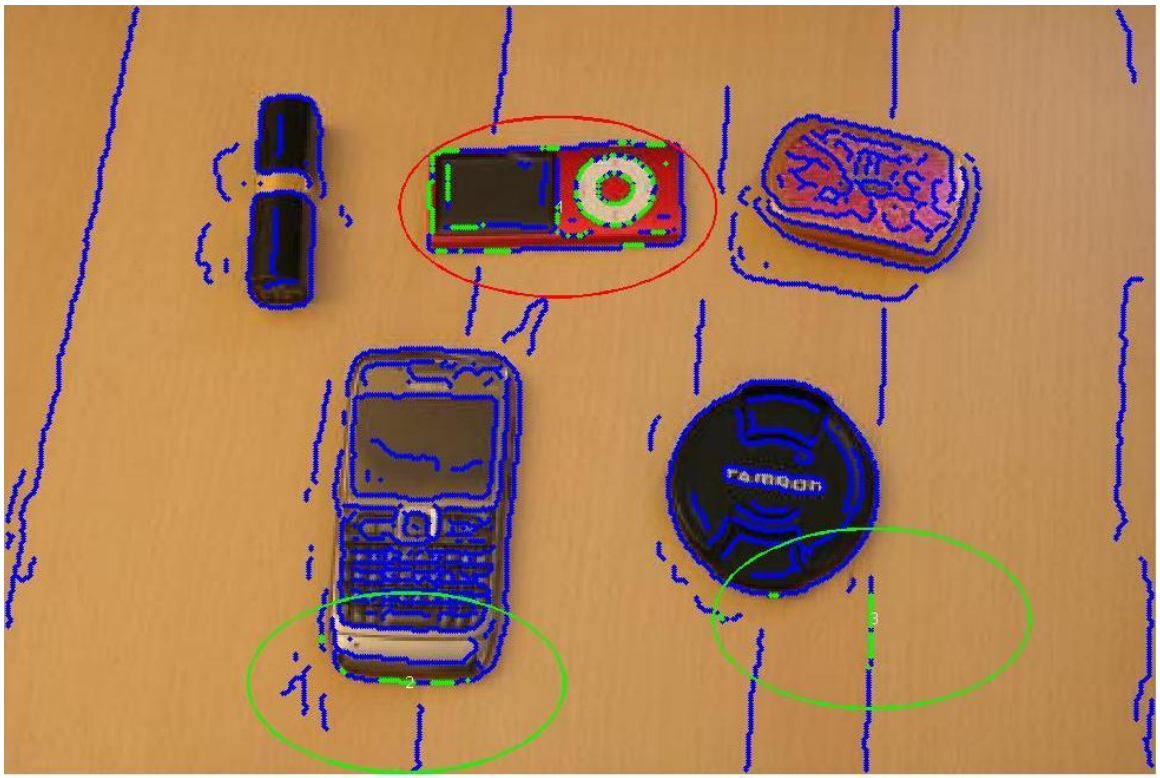


Figure 3.3 The matched points are clustered into three groups. The cluster of points in the red circle denotes the presence of the target object.

The Subtractive Clustering [58] method is applied on the matched points, resulting in three cluster groups, as illustrated in Figure 3.3. Each cluster's region of influence is marked by the ellipse circle. All the matched points not contained in these circles are discarded as incorrect matches. Each cluster suggests the presence of target object at the cluster's location and needs to be verified by the slope deviation and cluster density measurement. The slope deviations for these three clusters are $T = [0.32, 0.02, 1.99]$ and the densities are $D = [3.83, 0.83, 0.57]$. According to the predefined thresholds, only the first cluster survives both of the verifying process. Therefore, the first cluster is the only candidate region containing the target object or part of it, illustrated as the region circled by the red ellipse in

Figure 3.3. And the target object is detected at this cluster's location, which agrees with the actual scenario.

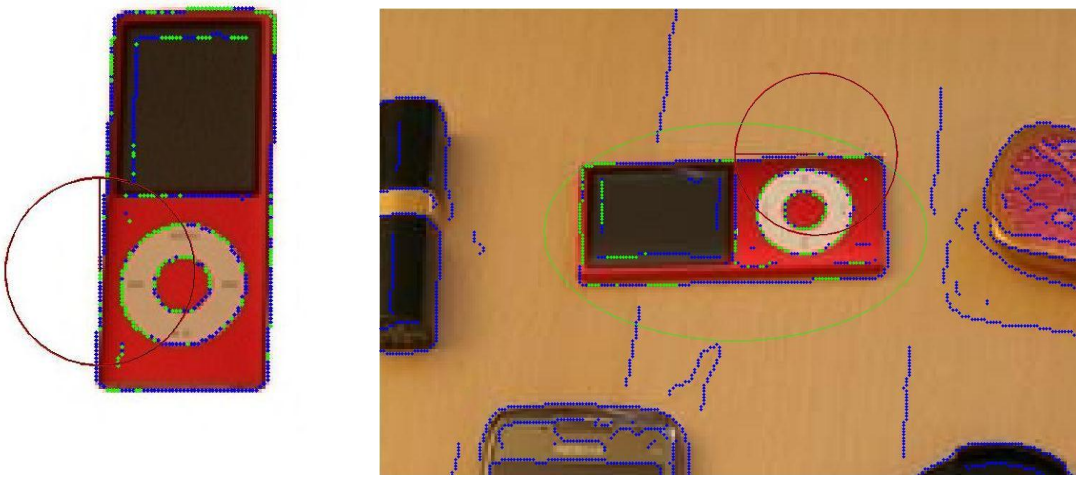


Figure 3.4 The image on the right shows the prototype edge points with the matched ones highlighted in green. The left image enlarges the cluster region where the target object is detected. Each of the red circles in both of the images represents the log-polar space used to form the shape context descriptor.

A closer view of the detected region is shown in Figure 3.4, with the prototype image on the left. The matched points in the cluster are highlighted in green, while their correspondent points on the prototype are also marked in green. An edge point's neighborhood, incorporated as the log-polar space in the construction of the Shape Context descriptor, is drawn as the red circle in both of the images. It is observed that the size of the log-polar space enable the descriptor to extract the local shape and inner edge information. The radius drawn in the circle represents the orientation of the descriptor, which enables the descriptor to describe the shape of the object rotation invariantly. As illustrated in Figure 3.4, the target object in the complex image has been rotated 90 degrees.

This example of detecting the target object in the complex scene illustrates the Shape Context's ability to catch the local shape information, robustly and rotation invariantly. Also the matching and identification strategy is shown to be simple and efficient. The identifier based on Shape Context has successfully detected the target object in Figure 3.1. However, there are limitations for this method as shown in the next simulation example.

Figure 3.5 The lizard above is a prototype and needs to be detected in the complex image below.

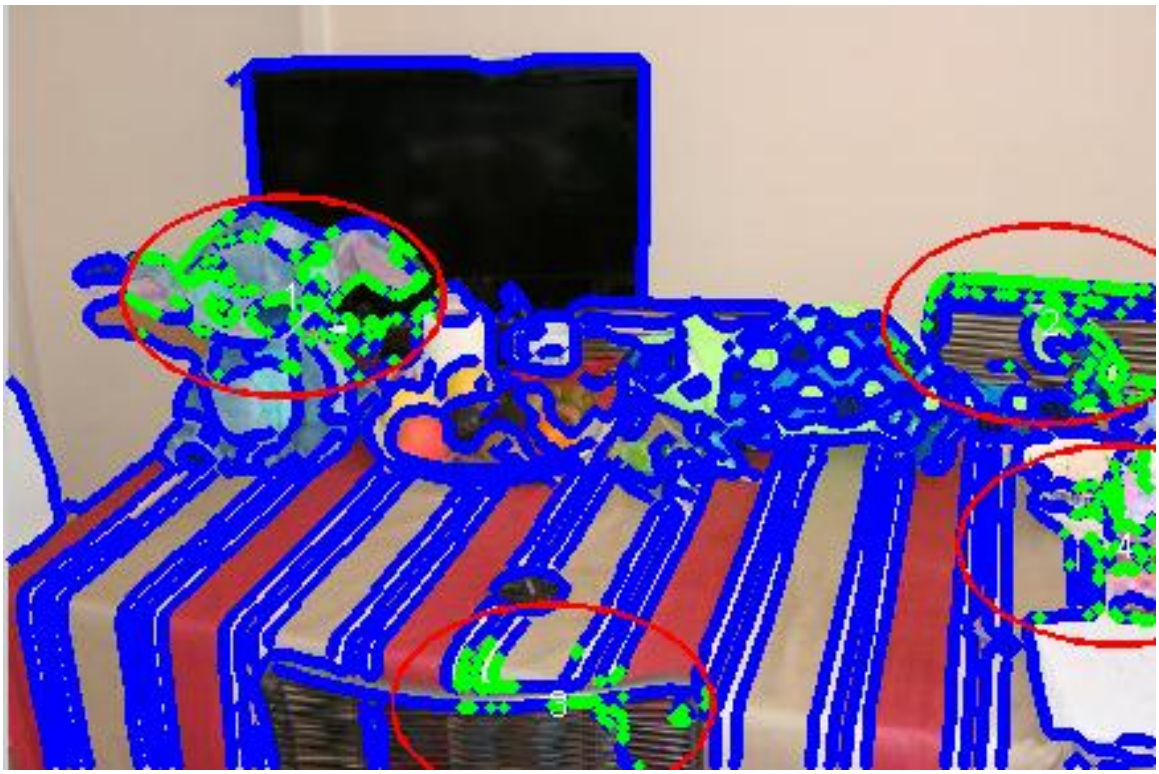


Figure 3.6 The identification result of applying shape context based identifier [58] to detect the toy lizard out from the complex scene. Each red circle contains a cluster of matched points, which are marked in green. The edge points of the scene are marked in blue.

On another image set shown in Figure 3.5, the shape context based object identifier [58] has also been applied. The prototype image contains a toy lizard and is of size 320x160 pixels. The complex image below contains a lot of objects and its size is 456x304 pixels. The result is shown in Figure 3.6, in which the edge points detected using Canny detector have been marked in blue, and there are four clusters of matched points in red circles with the matched points in green. It is observed that the four clusters detected by the identifier represent the wrong locations of the toy lizard. The approach constructed on the shape feature of the target object has failed in this task, which is to detect a texture rich lizard toy out from a complex scene. All the matched points are marked in green in Figure 3.7. Apart from the matched points located at where lizard toy actually is, all the other matched points on the other parts of the image are considered false matches. These false matches affect the Subtractive Clustering

[58] in the later stage, resulting with false clusters which suggest the wrong locations of the target object. These false matches are raised because the shape context descriptors extracted on the complex scene are sensitive to the distortion of the target object. In the complex and real life images, the target objects are usually represented themselves with many other objects and noisy background. In this situation, apart from the edge points estimated from the target object, there are many other edge points raised from other objects and noise in the complex scene. The shape context descriptor thus uses all these edge points to construct a histogram, which might be of huge difference compared with its counterpart on the prototype image. Also the descriptors calculated on other objects on the complex scene might be quite similar to the descriptors estimated on the prototype image. Therefore, the shape context descriptor lacks the discriminative power to distinguish the edge points on the target object from all the edge points on the complex scene.

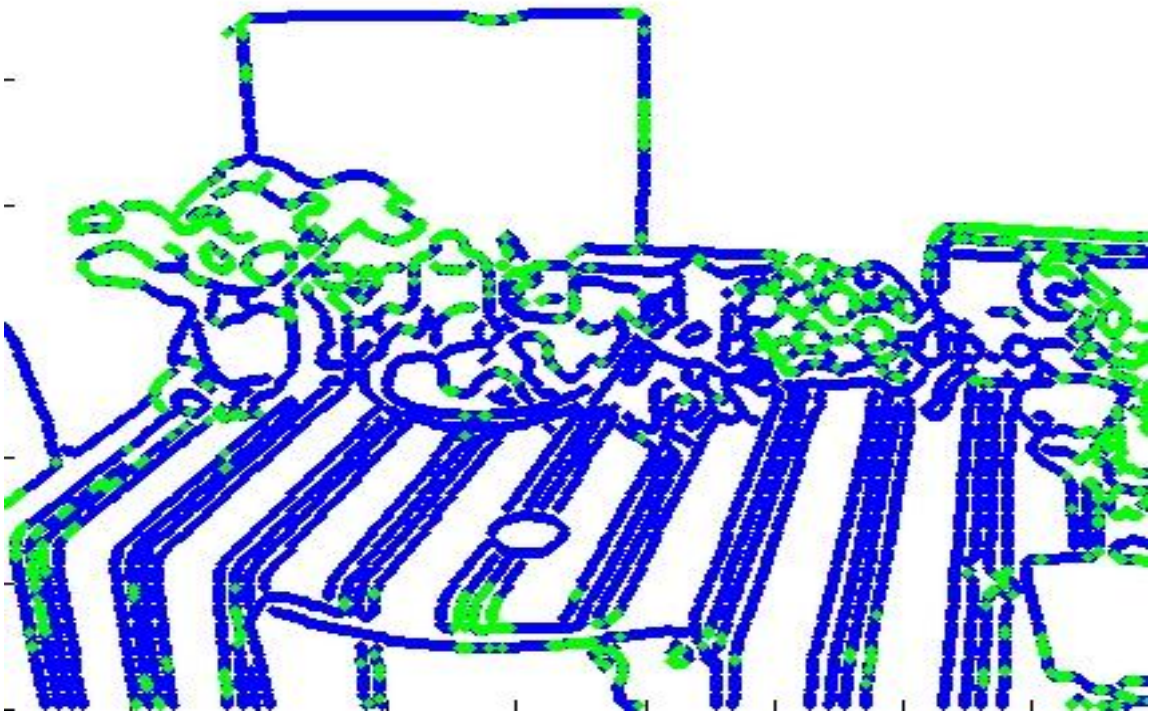


Figure 3.7 The edge map of the complex scene shown in Figure 3.8. All of the matched points are marked in green.

From the examples of simulations, it is observed that the identifier based on shape information works under certain conditions. First, the target object is required to possess a clear and distinctive shape. Second, the object is not mingled with the background in the complex scene. However, for the images of the real world, these conditions can hardly been satisfied. Therefore, the object detector studied in this chapter is not suitable for the task of identifying target objects in the real world complex images.

Because the shape information is not reliable for object detection, in Chapter 4, I have analyzed and explained another object identification strategy, which employs the texture of the object to describe and localize the target object.

Chapter 4

Study of Object Identifier Based on Texture Feature

In this chapter, I have explained the technique of an object detector which uses the texture feature to describe and localize the target in the complex scene. The object identifier employs the SIFT descriptor to catch the texture feature of the object. Afterwards a voting strategy has been applied to estimate the potential appearance of target object in the complex scene. The simulation results are also illustrated in the section 4.2, in which the limitations of the detector are also discussed.

4.1 SIFT Descriptor and Object Identification

In this section we explained the construction of the SIFT descriptor and its application of object identification.

4.1.1 SIFT Descriptor

SIFT descriptor is proposed by Lowe [100], and has been applied in many areas. It has been proved to be a robust and reliable descriptor. SIFT stands for Scale Invariant Feature Transform, as it describes image features in a scale-invariant fashion. Compared in terms of performance with other descriptors [70], SIFT and SIFT based descriptors (PCA-SIFT [71], GLOH [70]) have achieved the best results under several situations, e.g. rotation, scale changes and image blur. Also Shape Context descriptor has a high score next to SIFT in the evaluation. The aim of this work is to compare the object detection methods based on the SIFT and Shape Context descriptors, respectively.

SIFT is constructed as a process which consists of two main stages: scale-invariant region detection and key point descriptor building. One characteristic of SIFT method is that it generates a large quantity of descriptors representing stable image features at different scales and locations. As we explain later, this property of SIFT method is essential in its object identification strategy, in which at least three features are needed to determine the presence of the target object in the image.

4.1.1.1 Scale Invariant Region Detection

The first stage of SIFT model construction is to locate the key-point, each of which represents a scale-invariant region. These regions are areas that are invariant to scale change of the image and are located by detecting the stable image features across all possible scales. To search for the stable features, a scale space function [101] is constructed by convoluting the image $I(x, y)$ with Gaussian function $G(x, y, \sigma)$, which is used as the scale-space kernel, as follows:

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y)$$

where $\otimes$ represents the convolution, and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2} \right)$$

In his work, Lowe [100] introduced a difference-of-Gaussian function which is derived from the above $L(x, y, \sigma)$:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \otimes I(x, y)$$

$$= L(x, y, k\sigma) - L(x, y, \sigma)$$

where k is a constant multiplicative factor.

This function $D(x, y, \sigma)$ closely approximates to the scale-normalized Laplacian of Gaussian, $\sigma^2 \nabla^2 G$, which is required for true scale invariance as studied by Lindeberg [102].

After the scale space is constructed by applying different-of-Gaussian function, the local maxima and minima of $D(x, y, \sigma)$ are selected as the key-points. Each key-point represents a region whose scale is the key-point's located scale space level.

4.1.1.2 SIFT Descriptor Construction

After the detection of key-points with stable scale-invariant regions, the SIFT descriptor is estimated based on the scale-invariant regions. Each SIFT descriptor of a key-point has four properties attached to it: scale, orientation and coordinates in the 2D image. At the stage of scale-invariant region detection, the scale and 2D coordinates have been determined for each

key-point. Then the orientation needs to be assigned to each point. This is used to achieve the rotation invariance for the descriptor.

At the scale of the key-point, the neighborhood of the key-point on the Gaussian smoothed image L is selected to compute the orientation. For each point (x, y) in the neighborhood, the magnitude $m(x, y)$ and orientation $\theta(x, y)$ are calculated as follows:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \arctan \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}$$

Afterwards an orientation histogram is constructed based on the gradient magnitudes and orientation of each point in the neighborhood. The histogram is of 36 bins, each of which covers an angle of 10 degree. Each point contributes to the bin which corresponds to its orientation, weighted by its gradient magnitude and a circular-Gaussian window $G(x, y, 1.5\sigma)$. Then the orientation of the bin with the largest value is chosen as the orientation of the key-point.

After all the properties have been estimated for each key-point, the descriptor is easily formed. The SIFT descriptor is a 3D histogram of gradient locations and orientations. The key-point's neighborhood, whose size is determined by the key-point's scale, is divided into $n \times n$ regions. For each region, an orientation histogram of m bins is constructed with contributes from each point in the region. Each sample point adds to the histogram with the value weighted by its magnitude and the same Gaussian window $G(x, y, 1.5\sigma)$. As a result, there are $n \times n$ histograms of m bins to describe the key-point's local scale-invariant region. The SIFT descriptor is an $n \times n \times m$ vector.

From the structure of construction, it is shown that SIFT descriptor is sale-invariant and rotation-invariant. It is also designed to adapt to other changes of conditions, e.g. illumination change, affine change, as discussed in [69].

4.1.2 Object Identification Based on SIFT descriptors

Since the SIFT descriptor catches the image feature and is highly distinctive, it can be applied in many areas. One of the most general applications is object detection in complex scenes [59, 103]. In this part of my work, the identification method based on SIFT descriptors is briefly presented.

The SIFT descriptors extracted from the prototype image are stored in the database first. Then the descriptors from complex scene are calculated and their nearest neighbors are searched for in the database, based on a matching strategy. Then on the complex scene, the matched SIFT descriptors are clustered using Hough transform. After that a least-squares solution is applied to each cluster to obtain the best affine projection parameters. The mismatches are eliminated according to these parameters. At last a probability model is calculated to determine whether the complex scene contains the target object or not.

4.1.2.1 Matching of SIFT Descriptors

For each key-point on the complex scene, its matched point is selected as the nearest neighbor in the database of key-points from prototype training images. The distance between the key-point SIFT descriptor is calculated using Euclidean distance as follows:

$$D_{ij} = \frac{1}{2}\sum_{x=1}^{X}\frac{[h_x(p_j) - h_x(s_i)]^2}{h_x(p_j) + h_x(s_i)}$$

where $X = n \times n \times m$ is the number of bins of the SIFT descriptor. $h_x(\cdot)$ stands for the value of x th bin in the descriptor. p_j and s_i are the j th and i th key-points in the database and complex scene, respectively.

In the database of prototype images, there is a point with minimum Euclidean distance for each key-point. However, many of the key-points are extracted from the background noise or objects which are not desired for identification. Therefore, they should not be matched to any key-point in the database. A matching strategy is used to avoid this problem. Assume that for a key-point from the complex scene, its distance with its nearest neighbor in the database is D_1 and the distance with its second-nearest neighbor is D_2. To determine if the key-point has a match in the database, it needs to satisfy the following condition:

$$\frac{D_1}{D_2} < d$$

where d is a ratio threshold. The basic idea behind this method is that if a key-point has a match in the database, its distance D_1 with its nearest neighbor should much smaller than the distance D_2 with the second nearest neighbor. If there is no match for the key-point, both D_1 and D_2 are distances with incorrect matches and the values of them should be almost the same. The distance of correct matches should be much smaller than those of the incorrect matches, even for the second nearest neighbor.

4.1.2.2 Hough Transform and Affine Parameters

As the experiments show, a large number of key-points are extracted in a normal image. Although the distance ratio technique discussed above has been applied, there are still many undesired key-points, which are raised from the background and irrelevant objects. Therefore, the Hough transform is used to cluster the key-points and eliminate all the undesired ones.

Each matched key-point on the complex scene would suggest a candidate region in which the target object appears. This candidate region is named object hypothesis. Each object hypothesis represents the target object in a different pose compared with the prototype image. As we discussed above, for each SIFT descriptor, there are four properties attached: scale, orientation and 2D coordinates. If a key-point from the complex scene has a correct match in the database, its object hypothesis's pose, in terms of the relative location, scale and orientation to the prototype image, is predicted by the difference of these four properties of the two matched descriptors. The Hough transform is actually working as a voting system in which each match votes for its correspondent object hypothesis on the complex scene. Each match contributes its vote to its object hypothesis and the matches voting for the same hypothesis are clustered as a group. Since the matches are grouped according to their 2D locations, orientations and scales, a four dimensional voting system is formed to cluster the key-points. Each match votes for 2 closest bins in each dimension to avoid the boundary effect in bin assignment. In this way, 16 entries contribute to the Hough transform system for one match. Only object hypothesis with votes larger than three remain. This is because three sets of parameters are the minimum requirement to solve the affine transformation problem. As only four parameters are used to describe the 3D objects in the scene, there are object transformations which cannot be approximated by these four parameters. Thus, the bin size for each property needs to be large enough to tolerate some distortions.

The affine transformation is applied to approximate the 3D rotation of the target object in the image. Then the least square solution is used to estimate the best affine parameters. Assume a prototype point (x, y) is matched to a key-point (u, v) on the complex scene, their affine transformation can be modeled as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} = \begin{bmatrix} \sigma\cos\theta & -\sigma\sin\theta \\ \sigma\sin\theta & \sigma\cos\theta \end{bmatrix}$$

where $[t_x \ t_y]^T$ is the model translation parameter. σ stands for the scale change and θ is the change of orientation for the object in the scene.

To solve the affine transformation, the equation is rewritten into:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

The above linear equation represents a single transform. Any new matches can be added into this equation. To solve it, at least 3 matches are required.

The above equation can also be represented in a matrix form as follows:

$$\mathrm{Ax} = \mathrm{b}$$

where A represents the matrix formed by the key-points on prototype image. x is the column vector of the affine parameters. b is formed with the coordinates of the scene key-points.

The least square solution is applied in the above linear system, for the purpose of estimating the affine parameter vector x:

$$\mathrm{x} = [\mathrm{A}^{\mathrm{T}}\mathrm{A}]^{-1}\mathrm{A}^{\mathrm{T}}\mathrm{b}$$

This solution provides the best affine projection parameter for mapping the cluster of scene key-points to the matched data points. For each matched pair of key-points, their changes of

parameters are checked with that of x. If they do not agree, this match is removed from the cluster. Then the least solution is applied again on the cluster. This iteration terminates when every match in the cluster agrees with the affine parameters or there are less than three matches in the cluster. The clusters with less than three matches will be disregarded. After the above operation, the remaining clusters will have a set of affine parameters defining the candidate object hypothesis's location, scale and orientation.

4.1.2.3 Probability Decision Model

After elimination of the incorrect matches and clusters, each of the remaining groups of matches present a candidate region on the image, with their affine parameters suggesting the presence of the target object in the complex image. To determine whether the cluster represents the target object, a probability model is used to make the decision [59].

Assume that C is a candidate cluster of matches connecting a scene region and a prototype object O. The cluster C also predicts the target object's location, scale and orientation in the image. Then the probability $P(O|C)$ describes the likelihood of the O's presence at the region determined by C on the complex image. According to Bayes' theorem, $P(O|C)$ is derived as:

$$P(O|C) = \frac{P(C|O)P(O)}{P(C)}$$

$$= \frac{P(C|O)P(O)}{P(C|O)P(O) + P(C|\neg O)P(\neg O)}$$

The value of $P(C|O)$ is set to 1, because it is predicted that when O is in the image, all the key-points of C are observed in the complex image. $P(\neg O)$ is also set to 1. This is because it is very rare for the target object to exist in the complex image at a specific pose determined by C.

Then the conditional probability model is simplified as:

$$P(O|C) \approx \frac{P(O)}{P(O)+P(C|\neg O)}$$

$P(C|\neg O)$ is the probability of the key-points of C matching incorrectly to the points of O, while O is not in the image. $P(O)$ is the prior probability that the target object is in the complex scene. This simplified model can be interpreted that if $P(C|\neg O)$ is much smaller than the prior probability $P(O)$, the chance for O existing at the pose determined by C is very high.

The value of $P(O)$ is approximately estimated as the ratio of the key-points in C to all the matched points on the complex scene.

The number of key-points in cluster C is denoted as k, and n is the number of all key-points extracted in the projected object hypothesis region. If the probability of accidently matching a key point to the candidate object pose is p, $P(C|\neg O)$ can be defined as:

$$P(C|\neg O) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

This is an accumulative binomial distribution estimating the probability of incorrectly matching k points out of n candidates on the complex scene on which the object O is not present. The matching probability p is defined as:

$$p = dlrs$$

where l, r and s is the likelihood of the matches' agreement on location, orientation and scale with the cluster C. d is the probability of selecting a random match into the cluster C, calculated as the ratio of the matches in C to all the matches on the image.

Finally, the existence of target object O in the complex image can be verified by the cluster C, if the value of $P(O|C)$ is larger than 0.95.

4.2 Simulation of the Identifier based on Texture Feature

In this part, we have tested and compared the performances of the two object identification strategies based on SIFT and Shape Context descriptor, respectively. Their results are described in details and analyzed.

4.2.1 Configuration of SIFT Descriptor and Detection Strategy

Because of the implementation of difference-of-Gaussian space, the key-points and their correspondent neighborhoods on the image are extracted subject to scale-invariance. Afterwards the SIFT descriptor is estimated using the gradient magnitudes and orientations of each point covered by the key-point's neighborhood. It is a histogram vector of size $n \times n \times m$, where $n \times n$ implies that the key-point's neighborhood is divided into n^2 regions and m stands for the number of orientation bins in each region. In our work, n is set equal to 4 and m to 8, resulting in a descriptor vector of 128 bins.

After the detection of key-points and construction of SIFT descriptors, the key-points from prototype object and complex scene are to be matched based on the ratio of nearest and second nearest distance. The threshold verifying the matches is set equal to 0.8. After the matching process, all the matches are going to be clustered using Hough transform. The voting space of Hough transform is of four dimensions, which represent clusters' variations upon location, scale and orientation. For orientation, the bin size is of 30 degree. A factor of 2 is

used for scale bin size. For the locations, a quarter of the maximum size of the projected prototype image is used.

After solving the least square solution and discarding the irrelevant clusters of matches based on affine parameters, the remained cluster are verified by a probability model $P(O|C)$. Before calculating $P(O|C)$, the probability p needs to be decided, which tells how likely a key-point is accidentally matched to the candidate object pose. It is defined as $p = dlrs$. According to [59], the difference of location should be within 20% of the projected region size. The orientation change has a constraint of 30 degrees. Therefore, $l = 0.2 \times 0.2 = 0.004$ and $r = 30/360 = 0.085$. For scale difference, s is set to be 0.5, a relatively relax constraint.

4.2.2 Simulation Results of SIFT Based Object Identification Strategy

Firstly, the SIFT based identifier has been applied on image set shown in Figure 3.5. The prototype image contains a toy lizard and is of size 320x160 pixels. The complex image below contains a lot of objects and its size is 456x304 pixels. After the detection and extraction of SIFT features, 159 key-points have been generated on prototype and 819 on complex image. Afterwards 52 matches have been formed and put into the voting system of Hough transform to be clustered into groups. 9 clusters are detected and after the operation of least square solution to eliminate the incorrect matches and clusters, only one cluster of 18 matches remains. The target object has been correctly detected by this cluster, with the probability $P(O|C)$ equals 0.99. The results have been shown in Figure 4.1, in the complex image a parallelogram is drawn as the boundary of the prototype image after the affine transform. The circles in the prototype and complex image represent the matched SIFT features of the last key-points cluster, which is used to calculate the parameters of the affine transform and verify the presence of the target object.

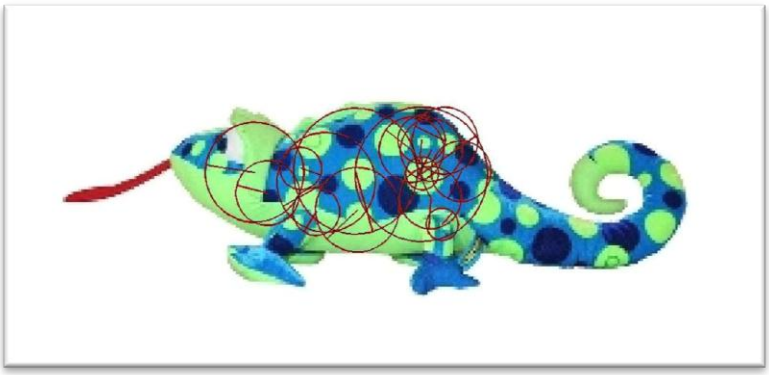


Figure 4.1 The circles in the images indicate the SIFT features used to identify the presence of the target object. The outer parallelogram shows the boundary of the target image under the affine transform used for recognition.

On the image set shown in Figure 3.1, the object detector based on SIFT descriptor is also applied for the identification of the target object. In Figure 4.2, SIFT features extracted from the complex and prototype image in Figure 3.1 are shown. On the complex image (size of 448x299 pixels), 342 key-points have been constructed scale invariantly. On the prototype

image (size of 115x171 pixels), 21 key-points have been constructed. As a fair amount of key-points have been located, it is expected that the target object in the complex scene could be identified easily. However, after matching these key-points and clustering the matches using the Hough transform method, there is only one set of matches left, which is shown in Figure 4.3. It can be observed that three SIFT features are grouped together, all of which are matched to the same key-point on the prototype. It is obvious that the cluster of key-points indicates the wrong location of the target object and the scales of the SIFT features on the complex image are quite different from the ones on the prototype. In this task, the SIFT descriptor based identifier fails in detecting the target object.

During the identification, the matching of the SIFT features plays a significant role as it provides the potential candidates for the next stage of Hough transform. If there were not enough matches, it is difficult to form clusters in which the properties (location, orientation and scale) of the matches agree. To find out the reason of the strategy's failure, all matches of key-points from prototype and complex image have been drawn and illustrated in Figure 4.4. On the complex scene, there are 26 matches and only four SIFT features extracted at the location of the target object have been matched to the prototype image. All of the four matches have been regarded as incorrect matches as they do not form a cluster in the Hough transform. Because there is no match indicating the presence of the target object, the object could not be detected in the complex image.

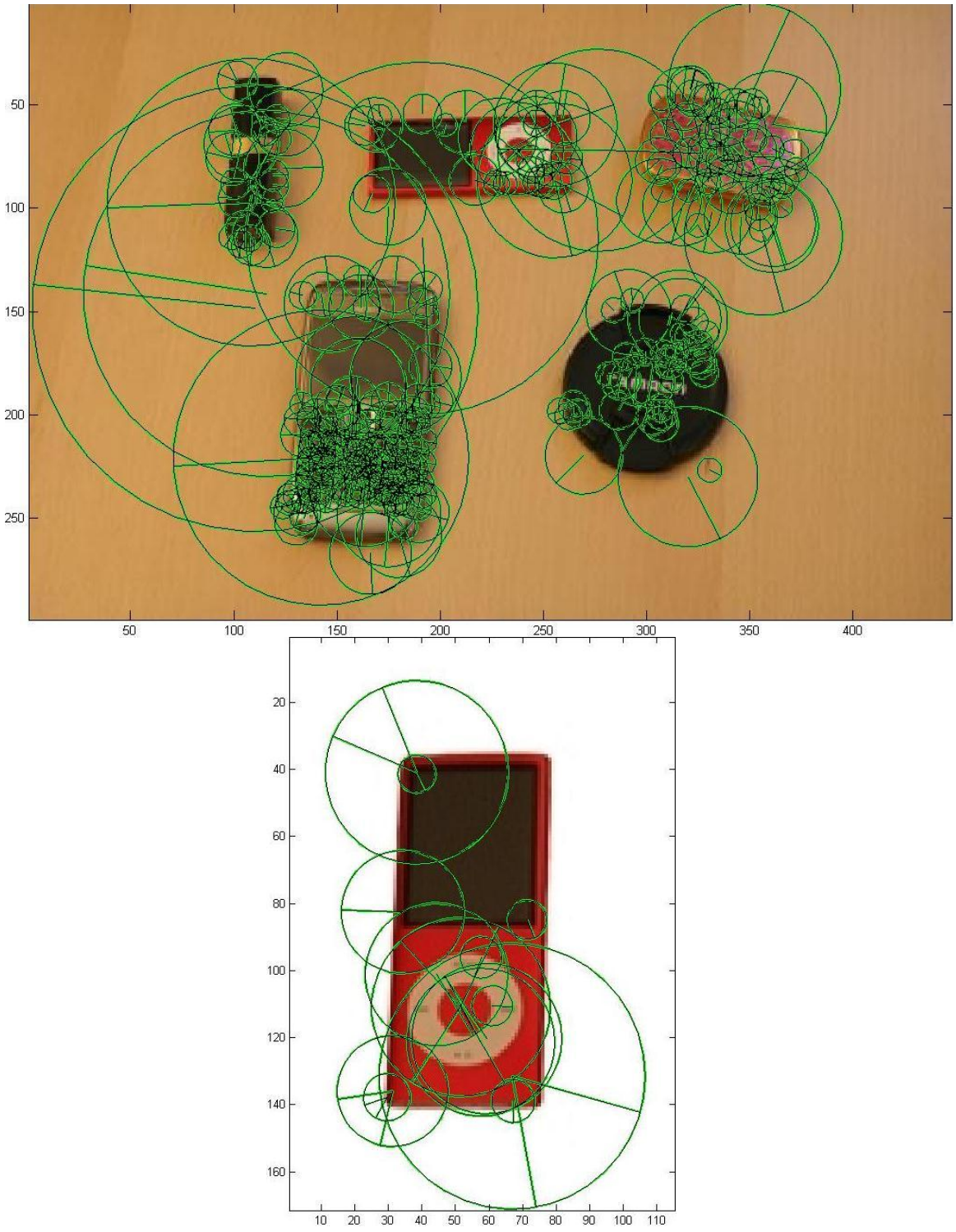


Figure 4.2 The SIFT features extracted from the complex and prototype image. Each circle represents one SIFT descriptor with the key-point as the circle center. The descriptor's scale is denoted by the size of the circle and orientation is the direction of the radius.

Figure 4.3 In the left image, clustered SIFT shape features have been located, with their correspondences on the prototype object marked in the right image.

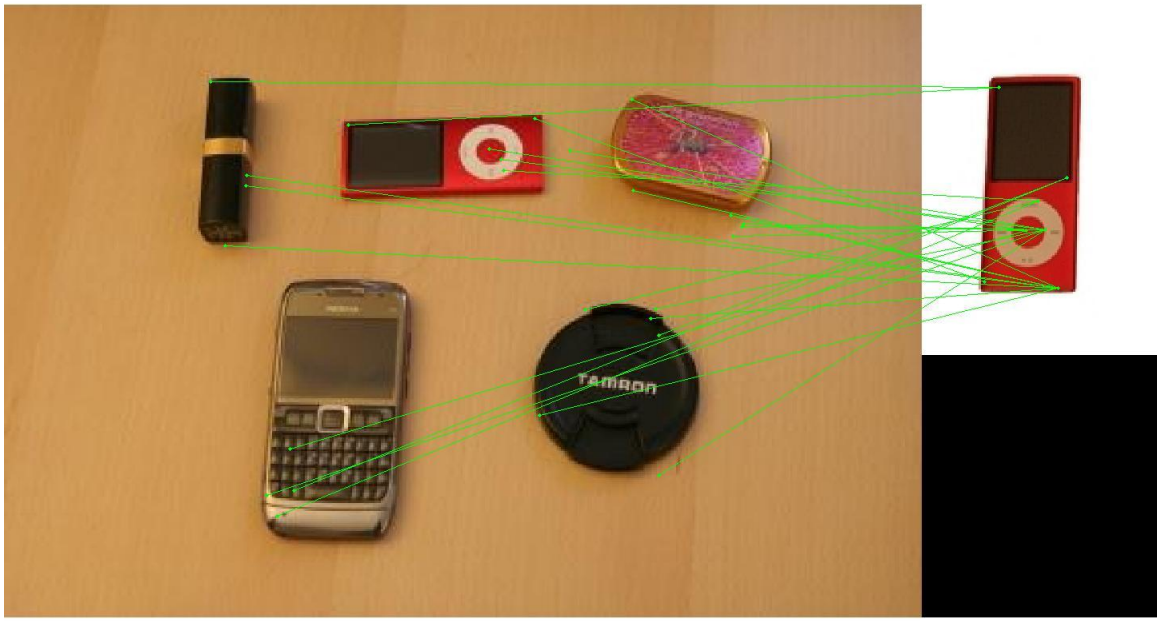


Figure 4.4 The matches of key-points of SIFT features from prototype and complex image have been shown. Each green line connects a pair of matched key-points. The green points indicate the locations of the key-points.

As observed from the examples of simulations, the number of SIFT features extracted from the lizard are eight times of those extracted from the ipod. Therefore there are much more matches for the lizard than the ipod in the complex images. The existence of the target object is identified by the clusters formed by these matches. Because there are clusters of enough matches for the lizard, it is detected with ease. For the ipod, under the condition of rare matches, the identifier does not function well.

From the above two examples, it is demonstrated that the quantity of the SIFT features is essential for the object identification based on SIFT descriptors. However, for the objects with less texture feature on the surface, e.g. ipod, they could not generate sufficient quantity of SIFT features for their identification in the complex scene. Therefore, it is concluded that the SIFT descriptor based identifier is not suitable for the task of identifying the texture-less objects.

4.3 Conclusion

The performances of the identifier have been evaluated and analyzed in the experiment. For the texture rich prototype object, a large number of SIFT features are extracted. The large quantity of descriptors ensures that there are sufficient matches to vote for a potential pose of the target object in the complex image. Under this situation, the SIFT based object detection method has been proved to be robust and reliable. However, if the task is to detect a texture-less object out from the complex scene, this strategy usually fails. Only a small number of SIFT features are extracted as a result of the texture-less surface of the object. These limited quantity of SIFT features leads to insufficient matches to vote for the presence of the target

object in the complex image. Therefore, the texture-less objects are not detected by the SIFT features.

While for the Shape Context based identifier explained in Chapter 3, the object with distinctive shape and interior edges are detected with confidence. Nevertheless, under the situation where the target object's contour shape have been occluded or mixed with other objects or background noise in the complex image, the strategy has failed to identify the target object. This is because the Shape Context descriptor only considers the shape information. If there was no reliable shape to be caught, the Shape Context descriptor is constructed based on the wrong shape information. Consequently, incorrect matches were formed from these inaccurate descriptors on the complex image. The target object can hardly be detected and localized by these incorrect matches. The Shape Context based identifier is not suitable in detecting the target object with distorted or missing shape information.

An object in the real world bears all types of information, for example, shape, texture surface and spatial layout of each part of the objects. Utilizing only one aspect of the information is neither realistic nor reliable in describing the object. The identification strategy incorporating only one type of object feature performs well only in the specific tasks which the identifier is designed for. However, when the object carries different features, these identifiers fail to cope with the changes. To solve the problem of detecting a general object in the real life images, an identification strategy needs to consider and incorporate all types of object information.

Chapter 5

Object identification System Employing Multiple Types of Image Features

In this chapter, a novel object identification strategy is introduced. The aim of the identifier is to detect and localize the target object in a complex scene with one prototype image. Recently there is a trend of image learning and parameter training for the object recognition techniques. However, the training process always requires large set of example images and the learning are often quite time consuming. Moreover, the training procedure results in excessive number of parameters. Many applications require the object detection in a complex scene with only one prototype image, like the high speed action recognition, automatic passport control and image retrieval from the web. Also the requirement for real time object recognition indicates that the object identifier needs to be simple and efficient. Therefore, in this part of my thesis, I introduce a novel object detector, which identifies the target object in the complex scene with only one prototype image by the capture and employment of multi-type image features. The feature descriptors are simplified and can be estimated from the image instantly to meet the

requirement of real time recognition. In addition, the identification process utilizing these feature descriptors are also easy and costs little time to estimate.

The objects usually represent themselves in the image with multi-type image features, and two of the most important features are shape and texture. The first stage of the proposed identification strategy employs these two types of image features to describe the objects. Each type of the feature descriptors extracted from the images is then employed to estimate a similarity measure. It is illustrated that by simply summing these two kinds of similarity measures, each of which corresponds to a specific type of image feature, the target object has a larger probability to be detected. At last, a likelihood map is constructed based on the similarity measurement, and the target object is identified according on this map.

5.1 Capture of Object Features

To design a successful and efficient object identification system, the first step is to extract and describe the object's features, which are later used to measure the similarities between different objects. Many successful descriptors have been constructed and their implementation and evaluation are shown in the literature [70]. Two of the main objectives in designing a successful descriptor are to obtain high discriminative power and invariance to transformations. However, as stated in the work of [48], there is a trade-off between the discriminative power and invariance. The author in [48] built a system to learn this trade-off and selected a combination of descriptors which best suits the specific object classification problem. Following the trend of descriptor combination [48, 53, 56, 57], different types of image descriptors are extracted and employed in the proposed object identifier.

An object in an image usually contains various heterogeneous types of information, such as texture, shape and color. Two of the most significant features of the object are its shape and texture information. This section explains the implementation and modification of two types of descriptors which focus on shape and texture features, respectively.

5.1.1 Texture Extraction

This part of my thesis explains the capture of the texture feature by the modified SIFT descriptor. Its performance, which has been demonstrated in various pieces of work [70], shows that it is a robust and successful texture descriptor. Many other types of descriptors (e.g. PCA-SIFT [71], GLOH [70]) are also built based on it. The construction of a SIFT descriptor involves two steps, namely the localization of the scale-invariant key-points and the descriptor histogram building.

In the first step, the points which indicate the scale invariant regions are localized. A scale space is formed by applying the different-of-Gaussian function. Then the local maxima and minima of the scale space are selected as the key-points, which represent regions with scale assigned as the key-points' located scale space level. After the first stage, the SIFT descriptors are built on these scale-invariant regions. A region is divided into 4×4 parts, and for each part a histogram of gradient orientations is formed, where the angles of the gradients are quantized into 8 orientations. Thus, the SIFT descriptor is a histogram with 128 bins.

In my work, instead of detecting the scale-invariant regions, the image is divided into rectangular patches of equal size. The histograms of SIFT descriptor are constructed for all image patches across the whole prototype and complex image. Therefore, all these SIFT descriptors are of the same scale. These histograms of SIFT are the simplified version of the original SIFT descriptor. Also for convenience, the orientations of all descriptors are set as zero.

In this way the simplified SIFT descriptors are built across the whole image with the same standard. The simplified SIFT descriptors computed on the prototype and complex image are represented mathematically by the following functions:

$$S_P = [s_P^1, s_P^2, ..., s_P^n] \in \mathbb{R}^{l_{SIFT} \times n_P} \quad (5.1.1)$$

$$S_C = [s_C^1, s_C^2, ..., s_C^n] \in \mathbb{R}^{l_{SIFT} \times n_C} \quad (5.1.2)$$

where n_P and n_C is the number of image patches on the prototype and complex image, on which the SIFT descriptors are constructed. l_{SIFT} is the number of bins in simplified SIFT descriptor's histogram. s_P^i and s_C^i are column vectors of length l_{SIFT} and stand for the modified SIFT descriptor computed on the i-th image patch of the prototype and complex image, respectively.

In the original SIFT descriptor, only the texture features around the scale-invariant key-point is captured, while the texture features of the pixels outside this region are missing. After the alternation, although all the texture information on the image is captured, the scale invariant property of the SIFT descriptor is lost. This is an example of the trade-off between the discriminative power and invariance ability of the descriptor [48]. However, the detected scale-invariant regions are not of any use in the identification strategy detailed in the next section. Therefore, the invariance has been traded for the discriminative power here. Also, as shown by many studies [57, 75, 76], the methods which capture image features densely outperform the ones only utilizing key-point based features in the task of object classification. As explained in [57, 75, 76], if the densely computed features were to put into a feature space of fine bins, the distribution of descriptors would follow a power-law (i.e., long-tail or heavy-tail distribution). This implies that the extracted features are mostly located in the low density area (the descriptors are infrequent) in the feature space and are quite isolated from each

other. Therefore the techniques only employing key-point based features entail huge information loss. In [57], the author found that the unison of all low-discriminative descriptors offers a significant discriminative power. Following this observation, in my work the SIFT descriptors are computed densely across the image and employed collectively in the similarity measurement explained in section 5.2.

5.1.2 Shape Representation

How to catch the shape information of an object is explained in this section. In contrast to the simplified SIFT descriptor constructed to capture the texture features of the entire image, local shape descriptors are built for each image patch. To extract the shape information, a popular and practical idea is to employ the distribution of edge orientation [68, 70, 72, 73], without the precise knowledge of the corresponding edge positions. Another idea is to describe the local shape around a particular edge point using the sparse shape features, e.g. edges of the shape. The best example of this idea is Shape Context [74] descriptor, which captures the geographic distribution of the edge points around a key-point's divided neighborhood. This technique has been employed in many applications [58]. Similar to the original SIFT descriptor, Shape Context is formed around a particular key edge point, together with all of its neighboring edge points, which may be raised from the background noise, contributing to the construction of the descriptor. This makes Shape Context extremely sensitive to the changes in the background noise. Furthermore, in the complex scene, another common distraction arises from the features of other objects and the background noise. Thus, Shape Context is not suitable for the object detection in the complex image.

Based on the above argument, I have used a histogram of edge orientations to represent the local shape features on an image patch. It is named Histogram of Orientated Gradients (HOG)

in [68]. The orientations of the edge points are quantized into M bins. Each edge point votes to the bin whose angular range covers its gradient orientation, weighted according to its gradient magnitude. Therefore, each bin in the histogram then represents the number of edge points whose orientations are located within a certain angular range. The HOG descriptors extracted from the prototype and complex image are noted as follows:

$$H_P = [h_P^1, h_P^2, ..., h_P^n] \in \mathbb{R}^{l_{HOG} \times n_P} \quad (5.1.3)$$

$$H_C = [h_C^1, h_C^2, ..., h_C^n] \in \mathbb{R}^{l_{HOG} \times n_C} \quad (5.1.4)$$

where n_P and n_C is the number of image patches on the prototype and complex image. l_{HOG} is the number of bins in HOG's histogram. h_P^i and h_C^i are column vectors of the length l_{HOG} and stand for the HOG descriptor computed on the i-th image patch of the prototype and complex image, respectively.

Compared with other shape representation techniques, HOG has several advantages. It captures the local shape information efficiently. Based on the structure of its construction, HOG achieves easily a certain degree of invariance to local geometric and photometric transformations. It is tolerant to minor translation, rotation or scale changes if they are within the size of local spatial or orientation bin size. To calculate the similarity between two set of image patches, this property is of vital importance. Because there are minor scale, translation and rotation changes of the target object in the complex scene, which makes it impossible to have the grid cover the target object in the same manner as the prototype. Thus for one image patch from the prototype image, even the most similar image patch on the complex image can hardly cover exactly the same part of target object. This requires the descriptor to be able to tolerate the minor geometric transformations. Therefore, HOG descriptor has been selected to represent the shape feature on the images.

5.2 Identification Strategy

In the previous section, on the prototype and complex image, the dense shape and texture features have been extracted by HOG and SIFT descriptors, respectively. These feature descriptors form a density space with a distribution of long-tail [57] and contain high discriminative power collectively. The next stage is to utilize the collection of features to detect the target object in the complex image.

In most work of object recognition based on feature-extraction, Euclidean distance or L_1 distance are often employed, either for nearest-neighbor finding or similarity measuring. Recently, attention has been paid to techniques measuring similarity based on correlation metrics [66]. In the studies of object classification [77-79] and subspace learning [80], this technique has been shown to outperform the conventional Euclidean distance or L_1 distance. Therefore, the similarity measurement based on correlation metric is applied in my work, which is explained and verified in this section.

5.2.1 Correlation Metric

Because the image features are represented by vectors of feature descriptors, as explained in section 5.1, the prototype image P and complex scene C can be considered as two random variables. In this way, to measure the similarity between the two images is the same to estimate the correlation metric between the two variables. There are many ways to calculate the correlation metric, and the cosine similarity metric is employed in my work.

Correlation refers to the quantity measuring the extent of interdependence between variables. In statistics, it refers to the strength and direction of a linear relationship between two random variables. The most common and standard correlation coefficient is Pearson's correlation coefficient. Another popular correlation coefficient is cosine similarity [80].

Assume that for two random variables X and Y, there are n available measurements, denoted as $x_1, x_2, ..., x_n$ and $y_1, y_2, ..., y_n$. Then the Pearson's correlation coefficient is estimated as:

$$\rho(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \qquad (5.2.1)$$

where $\bar{x}$ and $\bar{y}$ are the means of measurements of X and Y.

Based on the measurements, the cosine similarity of X and Y is denoted as:

$$\gamma(X,Y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^{\,2} \sum_{i=1}^{n} y_i^{\,2}}} = \cos\theta \in [-1,1] \qquad (5.2.2)$$

The cosine similarity value $\gamma(X,Y)$ ranges from -1 to 1. $\gamma(X,Y)$ equals -1 when X and Y are exactly opposite. The result value 1 means X and Y are exactly the same. When $\gamma(X,Y) = 0$, the two variables are independent.

It is observed that when the measurements of variables have zero means, cosine similarity equals Pearson's correlation coefficient. However, the cosine similarity is more discriminative than Pearson's correlation. This is because the centered measurements carry less information than the original ones and the zero or small measurements affect the computation of

measurement means [66]. Therefore, to fulfill the requirement of more discriminative power in object identification, cosine similarity is applied.

The prototype image P and complex image C are considered to be random variables, represented by the matrices of descriptors, S_P, S_C, H_P, and H_C, which are sets the SIFT and HOG descriptors generated from the prototype and complex image. Each descriptor vector in the matrices represents a measurement. The similarity between P and C is measured based on their shape or texture feature, each of which forms a separate similarity measure for the two images.

HOG descriptors have been constructed to densely capture the local shape information of the images. Thus, the shape similarity between two images is computed by calculating the cosine similarity between H_P and H_C:

$$\gamma_H = \frac{\sum_{i=1}^{n_H} h_P^i \bullet h_C^i}{\sqrt{\sum_{i=1}^{n_H} \left\| h_P^i \right\|^2 \sum_{i=1}^{n_H} \left\| h_C^i \right\|^2}} \in [0,1] \qquad (5.2.3)$$

where h_P^i and h_C^i are the i-th descriptor in H_P and H_C, respectively. $\langle \bullet \rangle$ represents the inner product of two vectors. n_H is the number of HOG descriptors computed on two images, which are of the same size. The value of γ_H ranges from 0 to 1, because the values of the vector histograms of h_P^i and h_C^i are all positive. When γ_H equals 0, H_P and H_C are independent. If the value of γ_H is 1, H_P and H_C are the same.

By connecting all vectors of the matrices to form two single vectors, the equation (5.2.3) can be rewritten in the same way as equation (5.2.2). The cosine similarity actually measures the angle difference between two vectors, and ignores the scale change on magnitude. Then this metric has the advantage of being insensitive to outliers, which the traditional Euclidean

distance lacks. In [80], the author builds two new feature-extraction algorithms based on the cosine similarity metric. While in [66], the author introduced a similarity measure called 'Matrix Cosine Similarity' (MSC), which is exactly the same as cosine similarity by just converting the matrix into a single vector.

The texture similarity measure is calculated in the same way by applying SIFT descriptor sets, S_P and S_C:

$$\gamma_S = \frac{\sum_{i=1}^{n_S} s_P^i \bullet s_C^i}{\sqrt{\sum_{i=1}^{n_S} \left\| s_P^i \right\|^2 \sum_{i=1}^{n_S} \left\| s_C^i \right\|^2}} \in [0,1] \qquad (5.2.4)$$

where s_P^i and s_C^i are the i-th descriptor in S_P and S_C, respectively. n_S is the number of SIFT descriptors. The value range of γ_S is the same as that of γ_H.

Finally, to combines shape and texture features together, the total similarity between prototype image and complex scene are estimated as follows:

$$\gamma = \frac{1}{2}(\gamma_H + \gamma_S)$$

$$= \frac{1}{2} \times \left(\frac{\sum_{i=1}^{n_H} h_P^i \bullet h_C^i}{\sqrt{\sum_{i=1}^{n_H} \left\| h_P^i \right\|^2 \sum_{i=1}^{n_H} \left\| h_C^i \right\|^2}} + \frac{\sum_{i=1}^{n_S} s_P^i \bullet s_C^i}{\sqrt{\sum_{i=1}^{n_S} \left\| s_P^i \right\|^2 \sum_{i=1}^{n_S} \left\| s_C^i \right\|^2}} \right) \in [0,1] \qquad (5.2.5)$$

The value of γ ranges from 0, meaning two images are completely independent, to 1, meaning two images are exactly the same. The summation in equation (5.2.5) enables the similarity measurement to estimate the similarity from two types of image features, namely the shape and texture information. In addition, apart from shape and texture, this summation can also include other types of features which are significant for the target objects, e.g., color,

spatial lay-out. If one type of the object feature has been distorted by the background noise or irrelevant objects in the complex scene, the similarity measure could still distinguish the target object by other types of object features. For example, when the target object is occluded or part of its shape contour is missing, the similarity measure could still rely on the texture to identify the target on the complex scene. Consider another situation in which the target object does not bear much texture feature, or the texture could not offer sufficient discriminative power for the target object, the similarity measure could still separate the target object according to its shape feature. The logical reasoning of the implementation of summation is also justified theoretically in section 5.2.2. Additionally, the simulation results in section 5.3 approve the effectiveness of this approach.

After the calculation of the similarity measure, a 'resemblance map' (RM) [66] is to be generated for the complex image. The examples of the resemblance map are shown in the next section. It is an image map consisting of image patches, whose values indicate the likelihoods of presence of target object in the complex scene. On the complex scene, each image patch, together with its neighboring patches, forms an image window, which is of the same size of the prototype image. Therefore, a set of overlapping image windows are generated by shifting the window to the right down corner with single column image patch each time. For the j-th image patch, a measure value γ_j is assigned, representing the similarity measurement between the image window C_j centered at the j-th image patch and the prototype image P. To construct the 'resemblance map', instead of simply using the similarity measure γ_j, a mapping function is used:

$$\psi(\gamma_j) = \frac{\gamma_j^2}{1-\gamma_j^2} \qquad (5.2.6)$$

It is shown in [66] that the 'resemblance map' based on $\psi(\gamma_j)$ offers more contrast, this is because the results of $\psi(\gamma_j)$ are of a more dynamic range ($\psi(\gamma_j) \in [0,\infty]$) than the original γ_j ($\gamma_j \in [0,1]$). The impact of applying $\psi(\gamma_j)$ is illustrated in the examples of Figure 5.1 and Figure 5.2, which provide the PDF histograms of image patches' values of the 'resemblance map', constructed with γ_j and $\psi(\gamma_j)$, respectively.

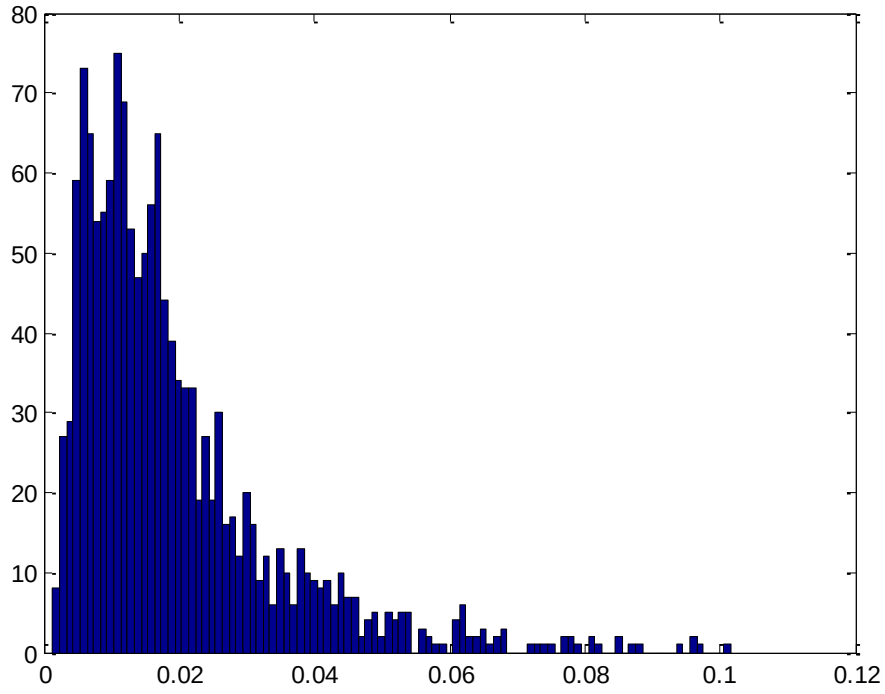


Figure 5.1 The PDF histogram of similarity measures on the 'resemblance map'

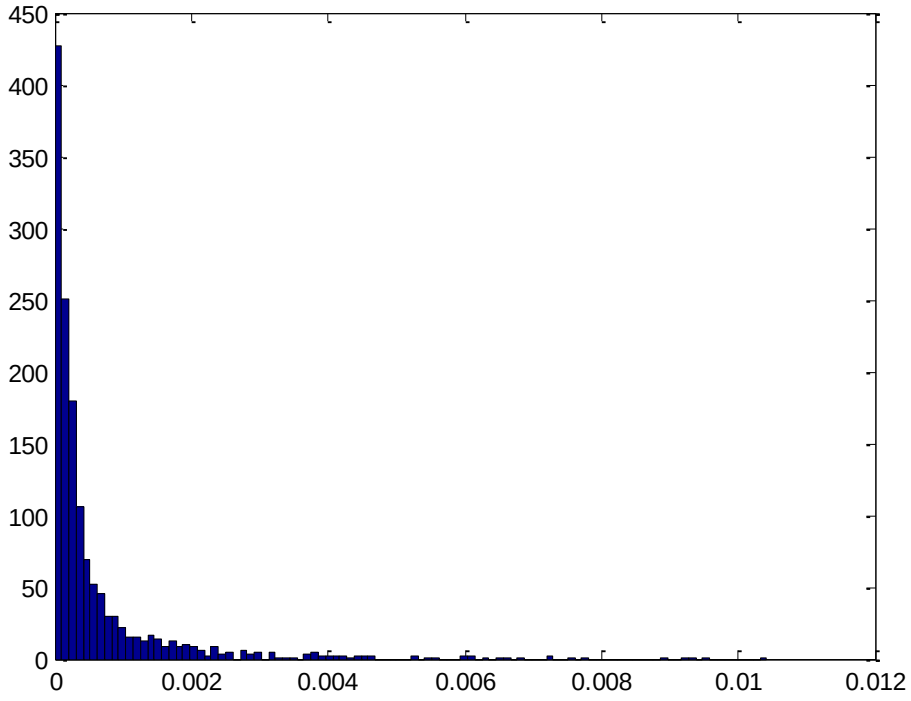


Figure 5.2 The PDF histograms of values of $\psi(\gamma_j)$ on the 'resemblance map'.

This 'resemblance map' only provides us with an image containing patches with values indicating how much each area is similar to the prototype image. Intuitively, people may choose the patch with largest value as the location of the target object. However, this is not practical when the target object is not in the complex image at all. Thus a strategy of two stages is developed to decide and locate the presence of the target object. First, a threshold γ_T is set empirically to be 0.1 for the total similarity measure γ_j. For γ_T, the choice of such a small value is due to the transformation of the target object in the complex image, as well as the shifts of patch locations when estimating the cosine similarity between two images. If there exists an image patch with values larger than $\psi(\gamma_T) = \frac{\gamma_T^2}{1-\gamma_T^2} \approx 0.01$, the target object is decided to be contained in the complex image.

5.2.2 Theoretical Justification

The object detection strategy has been explained in the previous sections. It is a simple approach compared with other identifiers which may employ a sophisticated training stage or complicated feature matching strategy. However, it is an efficient and robust identifier, and a theoretical justification derived in this section verifies the logic reasoning of this simple method.

In the works of [57, 66, 80], the optimal Bayes decision rule has been employed to justify the proposed techniques approximately. Inspired by these studies, it is shown that our method can be derived from the naïve-Bayes approach. Given a prototype image P, the task is to find the most similar image among a set of overlapping image windows C_i generated from complex scene. It is known [81] that estimating the maximum-a-posteriori (MAP) probability minimizes the average classification error. Because the image windows C_i are uniformly distributed, the MAP is reduced to maximum likelihood (ML) decision rule. Thus the task is noted as:

$$C_i = \arg\max_i p(C_i \mid P) = \arg\max_i p(P \mid C_i) \qquad (5.2.7)$$

The prototype image P and complex image window C_i are represented by the sets of feature descriptors, $\overline{S_P}$, $\overline{H_P}$, $\overline{S_C}$ and $\overline{H_C}$, which are normalized descriptors and represented as:

$$\overline{S_P} = [\overline{s_P^1}, \overline{s_P^2}, ..., \overline{s_P^m}], \qquad \overline{S_C} = [\overline{s_{C_i}^1}, \overline{s_{C_i}^2}, ..., \overline{s_{C_i}^m}]$$

$$\overline{H_P} = [\overline{h_P^1}, \overline{h_P^2}, ..., \overline{h_P^m}], \qquad \overline{H_{C_i}} = [\overline{h_{C_i}^1}, \overline{h_{C_i}^2}, ..., \overline{h_{C_i}^m}]$$

where m is the number of image patches generated on P and C_i. Each descriptor is normalized as:

$$\overline{s_P^m} = \frac{s_P^m}{\sqrt{\sum_{j=1}^{m} \left\| s_P^j \right\|^2}}, \qquad \overline{s_{C_i}^m} = \frac{s_{C_i}^m}{\sqrt{\sum_{j=1}^{m} \left\| s_{C_i}^j \right\|^2}}$$

$$\overline{h_P^m} = \frac{h_P^m}{\sqrt{\sum_{j=1}^{m} \left\| h_P^j \right\|^2}}, \qquad \overline{h_{C_i}^m} = \frac{h_{C_i}^m}{\sqrt{\sum_{j=1}^{m} \left\| h_{C_i}^j \right\|^2}}$$

Though occasionally satisfied in real life, for convenience, it is assumed that different types of features sets are independent from each other. Therefore, the ML decision rule is rewritten as:

$$C_i = \arg\max_i p(\overline{S_P}, \overline{H_P} \mid \overline{S_{C_i}}, \overline{H_{C_i}}) = \arg\max_i p(\overline{S_P} \mid \overline{S_{C_i}}) p(\overline{H_P} \mid \overline{H_{C_i}}) \quad (5.2.8)$$

In order to obtain useful result, the descriptors are assumed to be independent from each other. Then equation (5.2.8) is transformed as:

$$C_i = \arg\max_i p(\overline{S_P} \mid \overline{S_{C_i}}) p(\overline{H_P} \mid \overline{H_{C_i}})$$

$$= \arg\max_i p(\overline{s_P^1}, \overline{s_P^2}, ..., \overline{s_P^m} \mid \overline{S_{C_i}}) p(\overline{h_P^1}, \overline{h_P^2}, ..., \overline{h_P^m} \mid \overline{H_{C_i}})$$

$$= \arg\max_i \prod_{j=1}^{m} p(\overline{s_P^j} \mid \overline{S_{C_i}}) p(\overline{h_P^j} \mid \overline{H_{C_i}}) \quad (5.2.9)$$

To solve the equation (5.2.9), it is needed to compute the probability density $p(\overline{s_P^j} \mid \overline{S_{C_i}})$ and $p(\overline{h_P^j} \mid \overline{H_{C_i}})$. Because the descriptors are generated densely on the image, a Parzen density [57, 83] estimation is applied to approximate the local probability density of individual SIFT and HOG descriptors. For SIFT descriptor, its individual probability density is estimated as:

$$p(\overline{s_P^j} \mid \overline{S_{C_i}}) = \frac{1}{m} \sum_{k=1}^{m} K(\overline{s_P^j} - \overline{s_{C_i}^k}) \quad (5.2.10)$$

where $K(.)$ is the Parzen kernel function, typically a Gaussian:

$$K(\overline{s_P^j} - \overline{s_{C_i}^k}) = \exp(-\frac{1}{2\sigma^2}\left\|\overline{s_P^j} - \overline{s_{C_i}^k}\right\|^2) \quad (5.2.11)$$

In [57], it is shown that employing only the nearest neighbor is sufficient for the approximation of estimation equation (5.2.10). This is due to the long-tail property of the feature space. The author in [46] proposed to use the spatially nearest neighbor and qualitatively, the resulting estimate is quite similar to the original one. Therefore in my work, the same theory applies and the spatially corresponding descriptors are utilized to compute $p(\overline{s_P^j} \mid \overline{S_{C_i}})$:

$$p(\overline{s_P^j} \mid \overline{S_{C_i}}) \approx \exp(-\frac{1}{2\sigma^2}\left\|\overline{s_P^j} - \overline{s_{C_i}^j}\right\|^2), \quad j = 1,...,m \qquad (5.2.12)$$

As in the work of [46], the $\log p(\overline{S_P} \mid \overline{S_{C_i}})$ can be approximated as:

$$\log p(\overline{S_P} \mid \overline{S_{C_i}}) = \sum_{j=1}^{m} \log\left(p(\overline{s_P^j} \mid \overline{S_{C_i}}) \right)$$

$$\approx -\frac{1}{2\sigma^2}\sum_{j=1}^{m}\left\|\overline{s_P^j} - \overline{s_{C_i}^j}\right\|^2$$

$$= -\frac{1}{2\sigma^2}\sum_{j=1}^{m}\left(\left\|\overline{s_P^j}\right\|^2 + \left\|\overline{s_{C_i}^j}\right\|^2 - 2\overline{s_P^j} \bullet \overline{s_{C_i}^j} \right)$$

$$= -\frac{1}{2\sigma^2}\sum_{j=1}^{m}\left(\frac{\left\|s_P^j\right\|^2}{\sum_{j=1}^{m}\left\|s_P^j\right\|^2} + \frac{\left\|s_{C_i}^j\right\|^2}{\sum_{j=1}^{m}\left\|s_{C_i}^j\right\|^2} - 2\frac{s_P^j \bullet s_{C_i}^j}{\sqrt{\sum_{j=1}^{m}\left\|s_P^j\right\|^2}\sqrt{\sum_{j=1}^{m}\left\|s_{C_i}^j\right\|^2}} \right)$$

$$= -\frac{1}{2\sigma^2}\left(2 - 2\frac{\sum_{j=1}^{m} s_P^j \bullet s_{C_i}^j}{\sqrt{\sum_{j=1}^{m}\left\|s_P^j\right\|^2}\sqrt{\sum_{j=1}^{m}\left\|s_{C_i}^j\right\|^2}}\right) \qquad (5.2.13)$$

For $\log p(\overline{H_P} \mid \overline{H_{C_i}})$, it is derived in the same way and expressed as:

$$\log p(\overline{H_P} \mid \overline{H_{C_i}}) = -\frac{1}{2\sigma^2}\left(2 - 2\frac{\sum_{j=1}^{m} h_P^j \bullet h_{C_i}^j}{\sqrt{\sum_{j=1}^{m}\left\|h_P^j\right\|^2}\sqrt{\sum_{j=1}^{m}\left\|h_{C_i}^j\right\|^2}}\right) \qquad (5.2.14)$$

Then equation (5.2.9) is boiled down as:

$$\begin{aligned}
C_i &= \arg\max_i p(\overline{S_P} \mid \overline{S_{C_i}})\, p(\overline{H_P} \mid \overline{H_{C_i}}) \\
&= \arg\max_i \left(\log\left(p(\overline{S_P} \mid \overline{S_{C_i}})\right) + \log\left(p(\overline{H_P} \mid \overline{H_{C_i}})\right)\right) \\
&= \arg\max_i \left\{\frac{1}{2\sigma^2}\left(\frac{2\sum_{j=1}^{m} s_P^j \bullet s_{C_i}^j}{\sqrt{\sum_{j=1}^{m}\left\|s_P^j\right\|^2}\sqrt{\sum_{j=1}^{m}\left\|s_{C_i}^j\right\|^2}} + \frac{2\sum_{j=1}^{m} h_P^j \bullet h_{C_i}^j}{\sqrt{\sum_{j=1}^{m}\left\|h_P^j\right\|^2}\sqrt{\sum_{j=1}^{m}\left\|h_{C_i}^j\right\|^2}} - 4\right)\right\} \\
&= \arg\max_i \frac{1}{2} \times \left(\frac{\sum_{j=1}^{m} s_P^j \bullet s_{C_i}^j}{\sqrt{\sum_{j=1}^{m}\left\|s_P^j\right\|^2}\sqrt{\sum_{j=1}^{m}\left\|s_{C_i}^j\right\|^2}} + \frac{\sum_{j=1}^{m} h_P^j \bullet h_{C_i}^j}{\sqrt{\sum_{j=1}^{m}\left\|h_P^j\right\|^2}\sqrt{\sum_{j=1}^{m}\left\|h_{C_i}^j\right\|^2}}\right) \\
&= \arg\max_i \frac{1}{2}(\gamma_S^i + \gamma_H^i) \qquad (5.2.15)
\end{aligned}$$

where γ_S^i and γ_H^i are the similarity measure between P and C_i computed with SIFT and HOG descriptors, respectively.

The above equations clearly demonstrate that to estimate the ML decision rule in equation (5.2.7), it is equivalent to compute the cosine similarity measures using equation (5.2.5), which combines the shape and texture features.

5.2.3 Scale and Rotation Invariance

Because the simplified SIFT and HOG descriptors are tolerate to minor image transformation, the proposed object identifier is able to handle modest scale and rotation variation. However, it is desirable for the detection strategy to possess the capability of recognizing the target object under large scale and orientation transforms. In this part of my work, a method is designed to enable the proposed detection system to detect the target objects representing themselves with different scales and orientations in the complex image.

To cope with the large orientation change of the target object in the complex image, a set of images are constructed by rotating the prototype image by 30 degrees each time. Afterwards, this set of 12 rotated prototype images are used to build a set of 'resemblance maps' with the complex scene. Each of them represents the likelihood map of the target object's appearance at a particular angle in the complex image. Afterwards the map which contains the image patch of the largest value of $\psi(\gamma_j)$ is selected and used to decide the existence of the target object, as explained above.

For the scale variation of the target object, a similar strategy is applied. A pyramid of query images is built by scaling the prototype image. Then each query image is employed in the detection system to compute a 'resemblance map' with the complex image. Therefore, a pyramid of 'resemblance maps' is constructed with each level representing a different scale. The level holding the largest value of $\psi(\gamma_j)$ is considered as the potential scale at which the target object appears.

5.3 Experiments and Results

In this part of my work, the performance of the proposed approach is tested on various types of images under different conditions. The aim of the method is to identify and localize the target object in the complex image. If the target object is detected in the complex scene, a bounding box is drawn around the object of interest. Compared with the two object detection strategies [46, 57] described in the second part of my thesis, it is shown that the technique proposed in this chapter is more robust and efficient than these two techniques which identify the appearance of the target object based on the matching of corresponding key-points. It is also illustrated that when the target object is under large translation, rotation and scale changes, the detector is capable of localizing the prototype in the complex image.

5.3.1 Implementation of Shape and Texture Descriptors

The images tested by the proposed method are transformed into gray scale images. This is because that the SIFT and HOG descriptors are generated using gray scale images. For the image patch where the local descriptors are generated, its radius is set to be 9 pixels. The HOG descriptor is estimated on the edges extracted by Canny edge detector [84]. In the construction of the HOG, its orientation range is set to be [0, 360], which is more discriminative than using the range [0, 180] (where the contrast sign of the gradient is ignored). Regarding the histogram of the HOG descriptor, the number of orientation bins is set to be 8. As mentioned in section 5.1.1, the simplified SIFT descriptor has 128 bins. All of the SIFT descriptors' orientations are set to be 0.

5.3.2 Representative Simulations

Firstly, the detection method is applied to the same simulation in [58], which is a military application of detecting a helicopter in a complex scene. The prototype of a helicopter is shown in Figure 5.3(a), and the same helicopter with part of it being occluded is presented in Figure 5.3(b). Figure 5.3(c) is the complex image in which the same size target object needs to be identified.

The descriptors are generated densely on the image patches of the prototype images, which are illustrated in the Figure 5.4 and Figure 5.5. Each circle on the image represents the image patch on which SIFT and HOG descriptors are generated. The lines connecting the circle centers and circles represent the orientations of the simplified SIFT descriptors. The descriptors are extracted only from the image patches which cover the prototype helicopter. This is because the blank image patches around prototype object affect the calculation of similarity measurement by introducing irrelevant feature descriptors. After the estimation of the similarity measures between the prototype and image windows, the identification results are shown in Figure 5.6, in which the target objects are localized by the green bounding boxes. Compared with the approach proposed by [58], in which the matched key-points are grouped to form a set of candidate clusters suggesting the appearance of the target object, only one image patch is selected in my approach indicating the location of the target object.

To demonstrate the accuracy and reliability of the detection method, the ‘resemblance map’ has been drawn to show the similarity measures between the helicopter and complex image. The likelihood map which only employs the texture information is illustrated in Figure 5.9. It is observed that besides the image patch, where the helicopter actually is, there are several other image patches with large similarity values in the lower part of the scene, indicating the possible appearances of the helicopter. Judging their relative locations on the complex image, they are from the trees which possess similar texture feature as the helicopter. Thus the

texture features of the helicopter are not distinguished from the background scene. While in Figure 5.7, on the map employing the shape information, only the image patch where helicopter appears holds a similarity value much larger than the rest of the patches. The difference between the two similarity map's distributions is also displayed in Figure 5.8 and Figure 5.10, which clearly demonstrate that for the helicopter in the complex image, its shape features are more discriminative than its texture features. Because both shape and texture features are incorporated, the object identification approach is still capable of detecting the prototype helicopter. Finally the likelihood map combining both shape and texture features are shown in Figure 5.11, where the location of the target helicopter is successfully indicated by the image patch with the largest similarity measure.

(a) (b)

(c)

Figure 5.3 (a) The prototype image of a helicopter. (b) The prototype image which contains part of a helicopter. (c) The complex image in which the target object of helicopter needs to be identified.

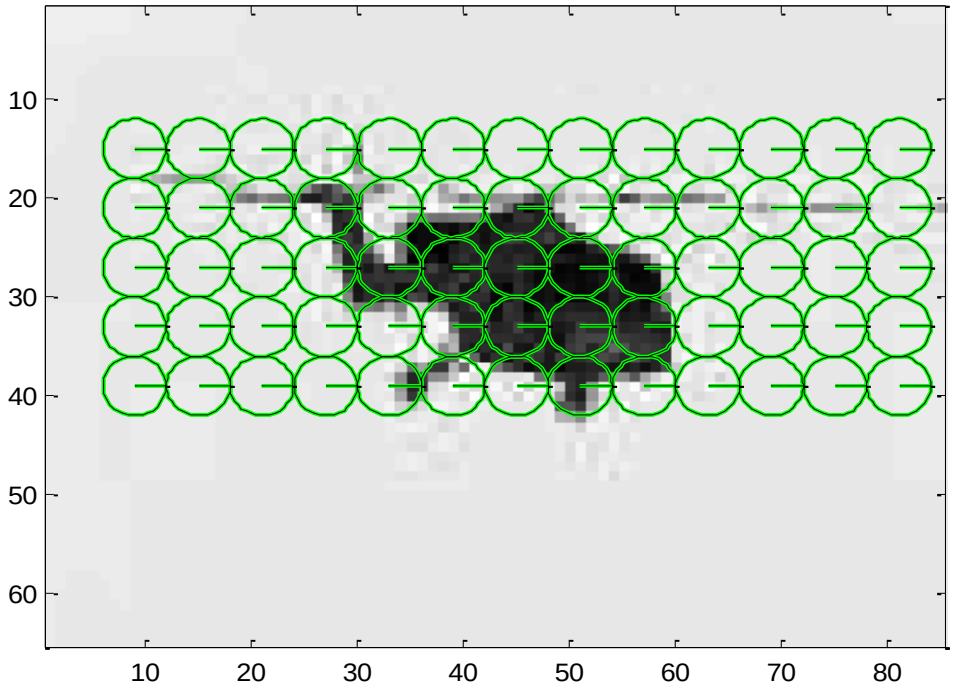


Figure 5.4 The image patches which cover the prototype helicopter. The descriptors are generated densely on these image patches. Each circle is an image patch on with the SIFT and HOG are build. The lines connecting the circle centers and circles represent the orientations of each SIFT descriptor.

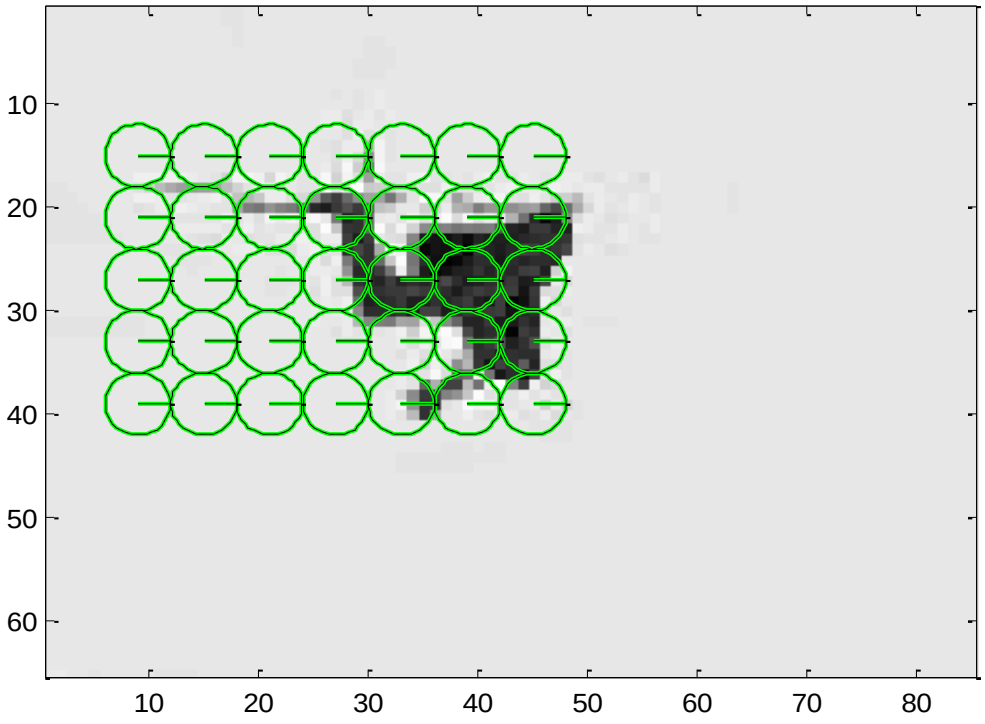
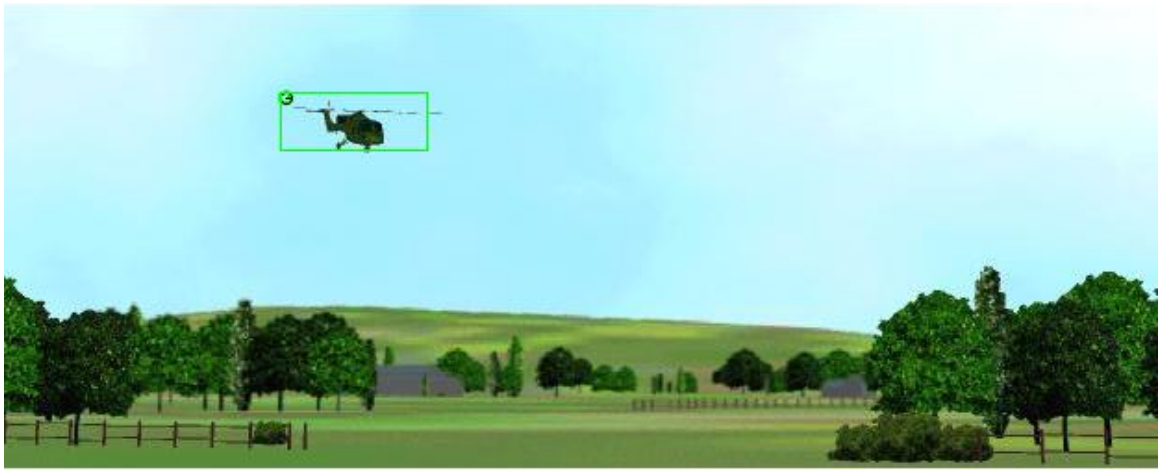


Figure 5.5 The image patches which cover the prototype partially-occluded helicopter. The descriptors are generated densely on these image patches. Each circle is an image patch on with the SIFT and HOG are build. The lines connecting the circle centers and circles represent the orientations of each SIFT descriptor.



(a)

(b)

Figure 5.6 The appearance of the target object is detected and localized by the green bounding box. (a) The prototype helicopter is identified by the bounding box. (b) The helicopter partially occluded is also identified.

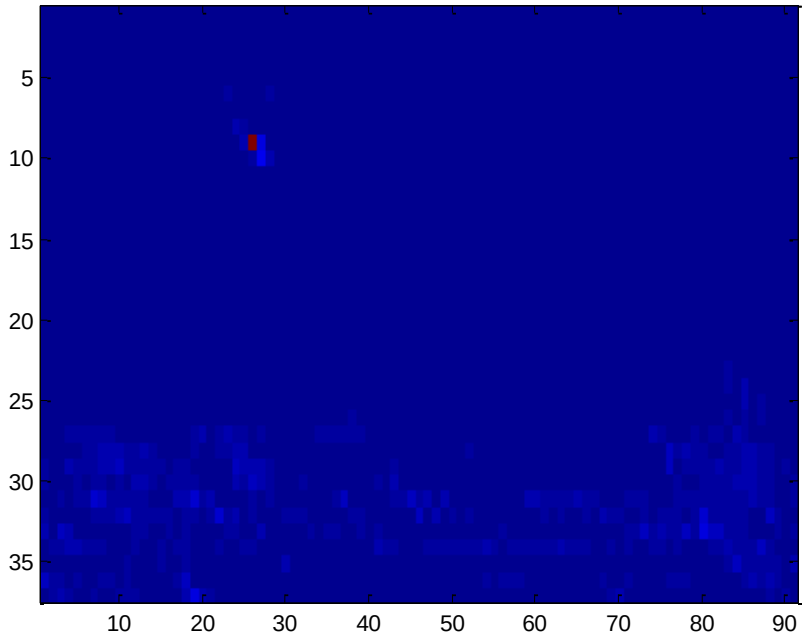


Figure 5.7 The 'resemblance map' of similarity measurements between the prototype of helicopter (Figure 5.3(a)) and image windows on complex scene (Figure 5.3(c)), constructed with only shape features (HOG descriptor).

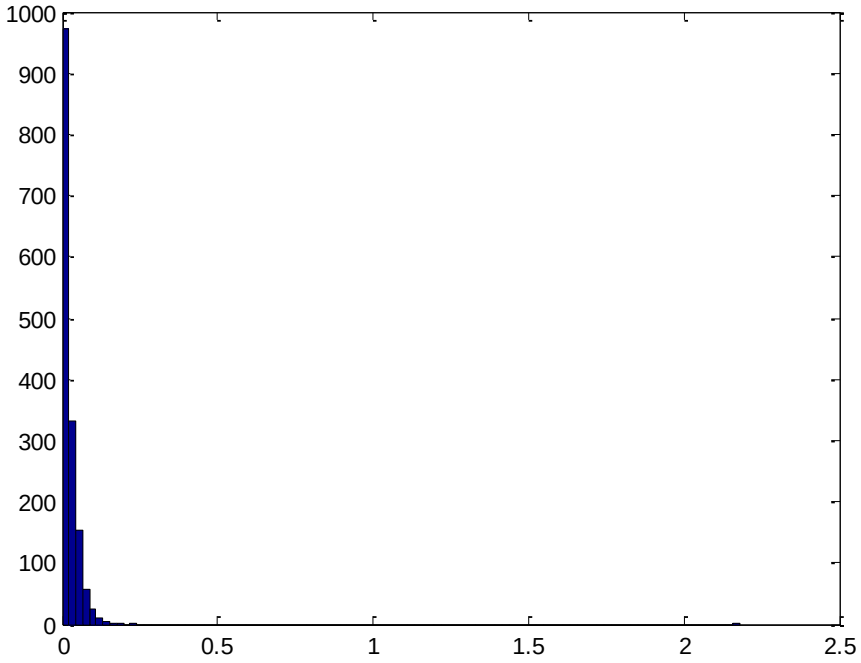


Figure 5.8 The distribution of similarity measures on the likelihood map employing only shape information (HOG descriptor).

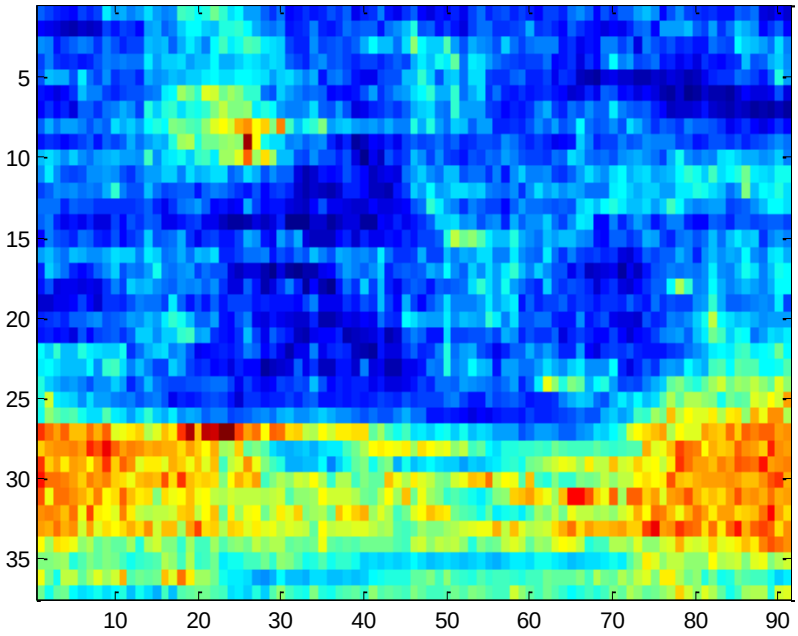


Figure 5.9 The 'resemblance map' of similarity measurements between the prototype of helicopter (Figure 5.3(a)) and image windows on complex scene (Figure 5.3(c)), constructed with only text features (SIFT descriptor).

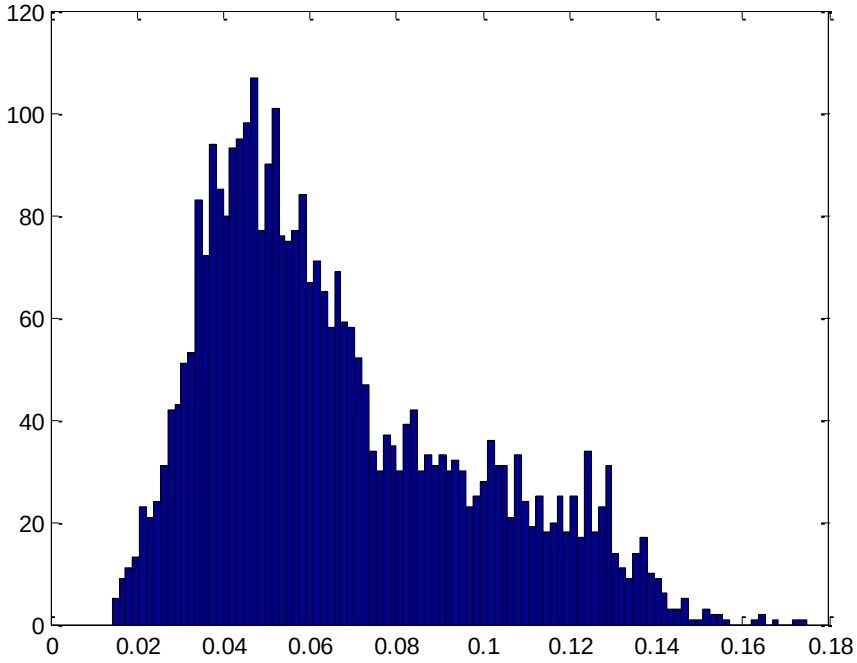


Figure 5.10 The distribution of similarity measures on the likelihood map employing only texture information (SIFT).

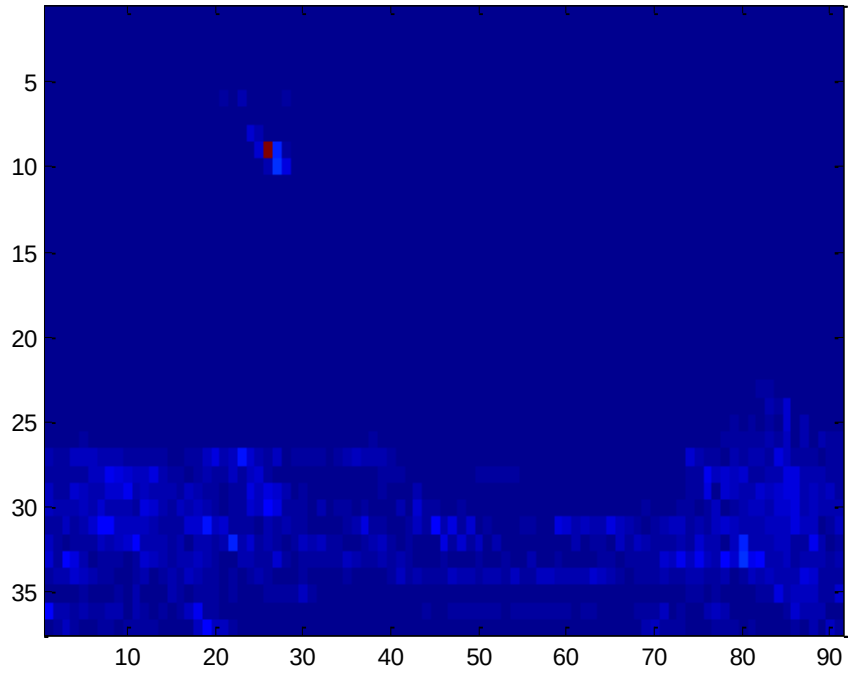


Figure 5.11 The 'resemblance map' of similarity measurements between the prototype of helicopter (Figure 5.3(a)) and image windows on complex scene (Figure 5.3(c)), estimated using both shape (HOG) and texture features (SIFT).

All the tests were simulated in Matlab running on a Dell laptop with an 8G RAM and Intel Core i7 CPU working at 2.67 GHz. For the picture set shown in Figure 5.3, after the estimation of the descriptors, it costs 0.71 second for the multi-type feature identifier to detect the object, while the algorithm in Chapter 3, which is designed on Shape Context descriptor, takes 21.383 seconds to finish the task. And the SIFT based detector explained in Chapter 4 takes 1.021 second to find the correct location of target object in the complex scene. The method introduced in this Chapter is less time consuming than the other techniques discussed in the previous two chapters. This has also been observed in all the other simulations.

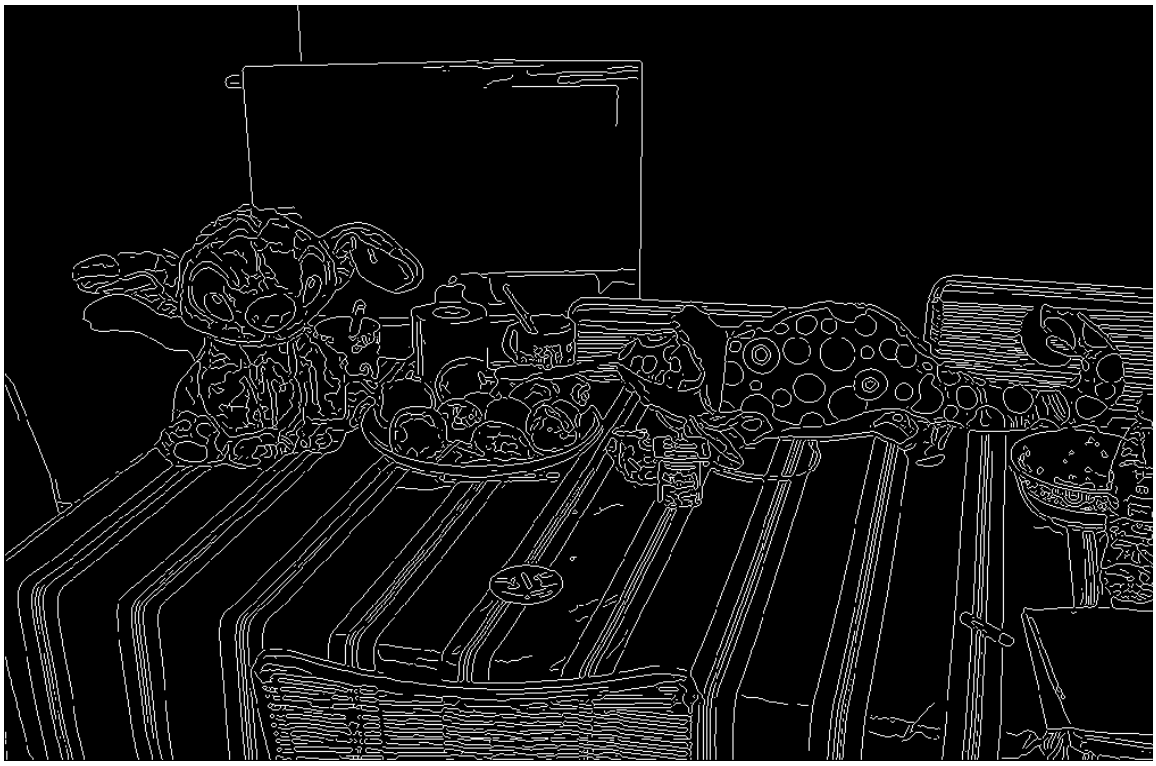
After illustrating the result of detecting the helicopter on the synthetic complex image, the identification approach is applied on searching for the target objects in the real practical pictures. An example is shown in Figure 5.12, in which the prototype image of bear toy's face (Figure 5.12(a)) needs to be identified and localized in the complex image (Figure 5.12(b)). It is observed that the complex image contains many objects and large background noise. The shape information of the prototype image is distorted by the edges which are raised from the objects but the target object and the noise of the background, as shown in Figure 5.13. The result of this distortion is illustrated in Figure 5.14, where the image patch with the largest similarity measure is not at the location of the target object. However, the texture features of the prototype object offer high discriminative power in the detection process, as shown in the 'resemblance map' estimated based on SIFT descriptor (Figure 5.15), where target object is correctly localized by the image patch with largest similarity. The likelihood map using both shape and texture features are illustrated in Figure 5.16 and the final recognition result is shown in Figure 5.17, with the bounding box suggesting the existence of the bear.

(a)

(b)

Figure 5.12 (a) The prototype image of the face of a toy bear. (b) The complex scene in which the prototype needs to be detected.

(a)



(b)

Figure 5.13 (a) The edge map of target toy bear. (b) The edge map of complex image.

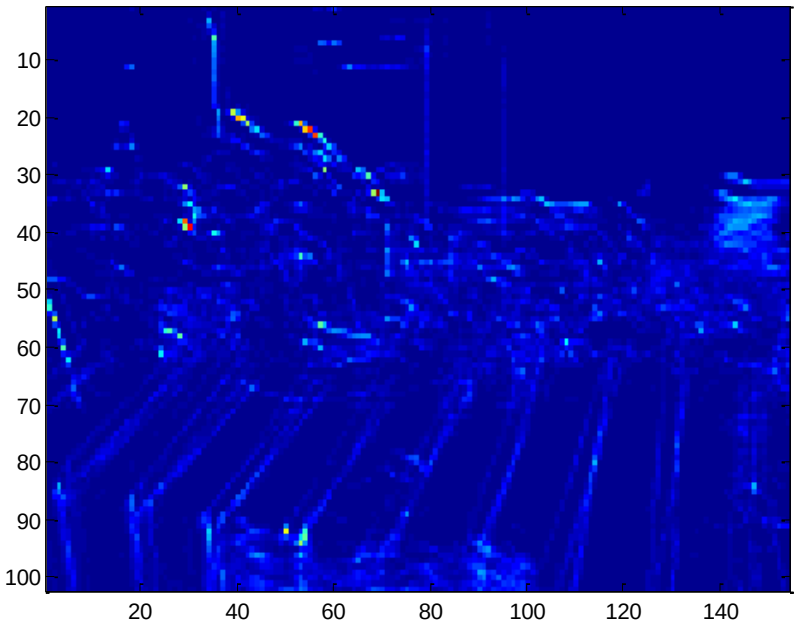


Figure 5.14 The likelihood map of similarity values between the prototype image (Figure 5.12(a)) and image windows of complex scene (Figure 5.12(b)), measured only with shape features (HOG descriptor).

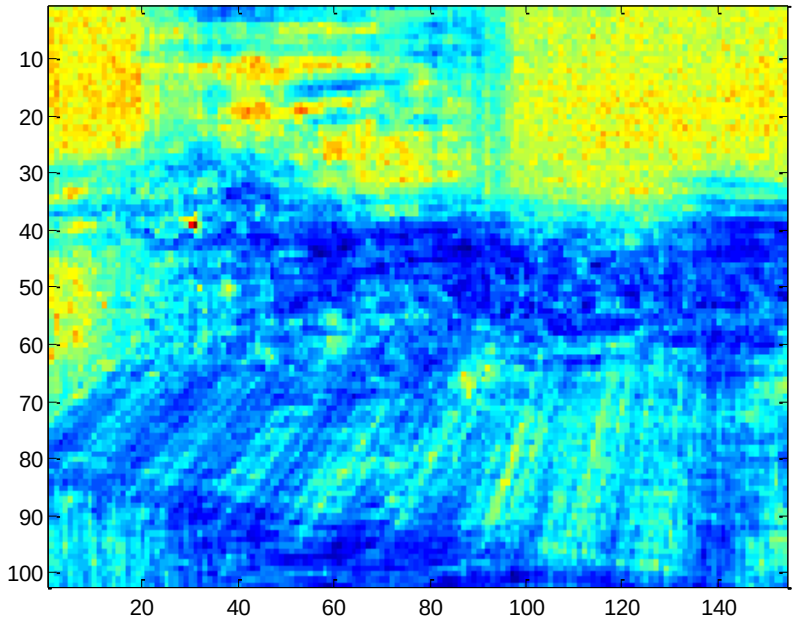


Figure 5.15 The likelihood map of similarity values between the prototype image (Figure 5.12(a)) and image windows of complex scene (Figure 5.12(b)), measured only with texture features (SIFT descriptor).

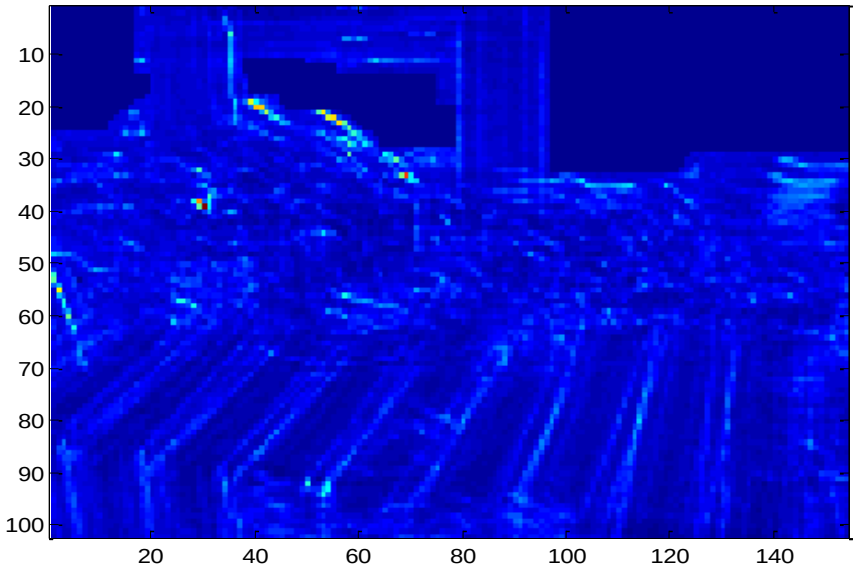


Figure 5.16 The likelihood map of similarity values between the prototype image (Figure 5.12(a)) and image windows of complex scene (Figure 5.12(b)), estimated based on both shape (HOG descriptor) and texture features (SIFT descriptor).

Figure 5.17 The target image has been successfully identified by the bounding box in the complex scene.

The identification strategy has been designed to detect target object under large rotation and scale changes. So a simulation of detecting objects under large rotation is presented below. The prototype images of different objects (lens cover, ipod and mobile phone) are shown in Figure 5.18. The complex scene in which these objects need to be identified and localized is illustrated in Figure 5.19. It is observed that the ipod and mobile phone are rotated by about 30 and 60 degrees, respectively. To detect the rotated target object, the proposed method rotates the prototype images by a certain degree and creates a pile of 'resemblance map' for each orientation. For instance, the rotated images of ipod are shown in Figure 5.20, where the object is rotated anti-clockwise by 30 degrees each. Afterwards, 12 'resemblance maps' are constructed, each of which represents the likelihood map of the similarities between the complex image and a prototype image of ipod rotated with a certain angle. According to the decision criteria, the target ipod is detected with a rotation of 240 degree anti-clockwise, as shown on its likelihood map (Figure 5.21), on which the ipod's location has a much larger value than the surrounding area. The detection results of all the prototype objects are demonstrated in Figure 5.22, where location of each target object is marked with bounding box of different colors.

(a)

(b)

(c)

Figure 5.18 (a) Prototype image of a lens cover. (b) Prototype image of an ipod. (c) Prototype image of a mobile phone.

Figure 5.19 The complex image in which the prototypes in Figure 5.18 need to be detected.

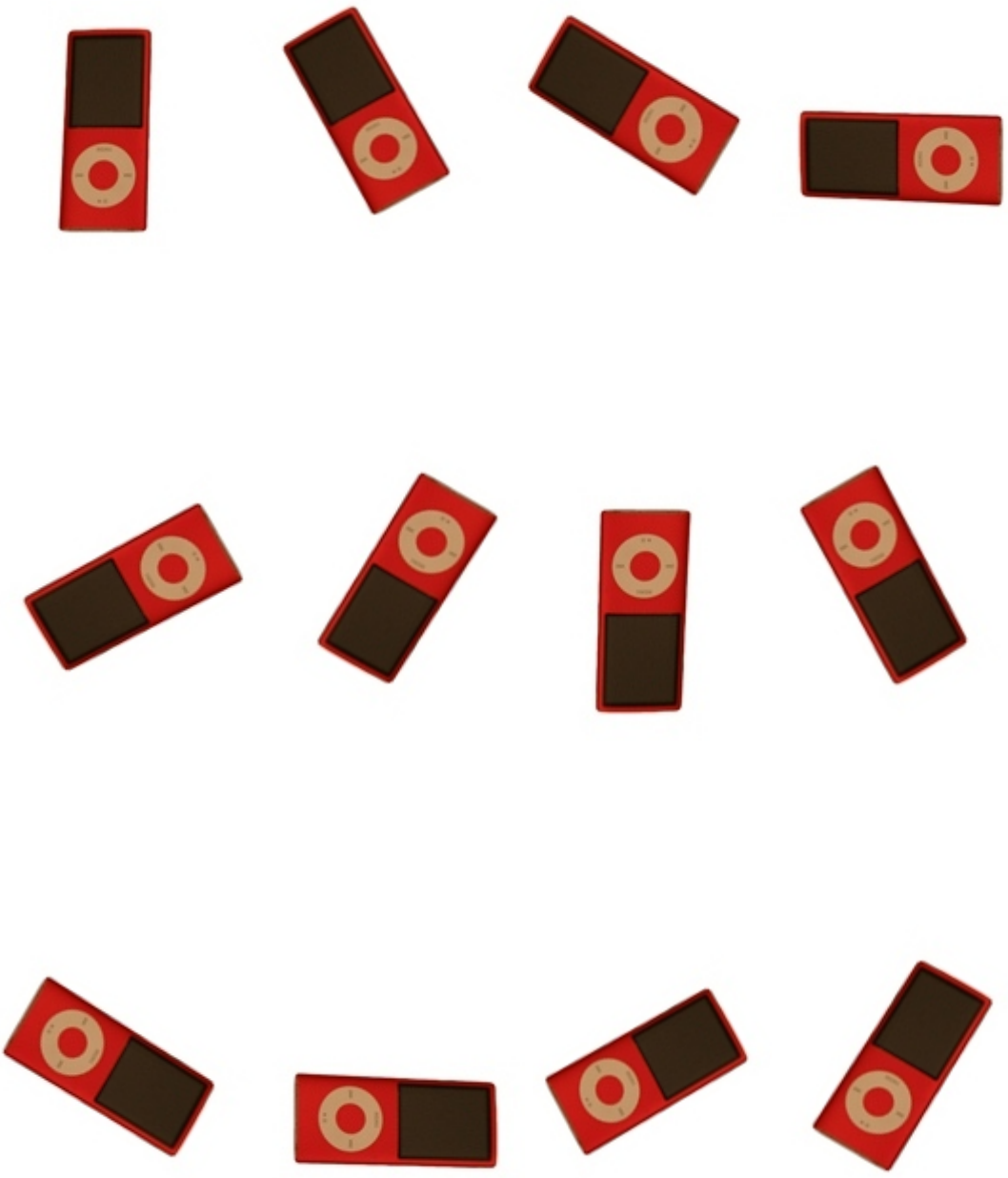


Figure 5.20 A set of images of ipod rotated by 30 degrees each.

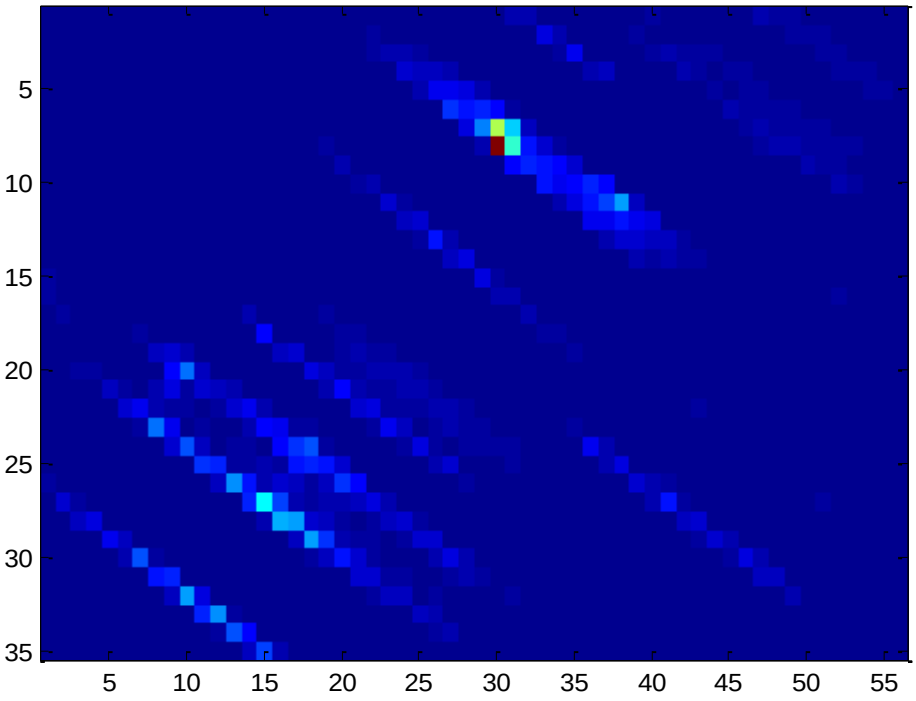


Figure 5.21 The 'resemblance map' of similarity measures between complex image (Figure 5.19) and the ipod rotated by 240 degrees anti-clockwise.

Figure 5.22 The detection results of the prototype objects (Figure 5.18). Each objec is located with a bounding box of different color.

Chapter 6

Discussion and Conclusion

My thesis focus on the study and investigation of object recognition techniques based on the representation of local image features. These techniques often involve roughly three stages of operations. The first is to extract stable and reliable local object features from the database images. The second stage is to find correspondent image features via certain strategies, e.g., training of the identifier parameters and similarity estimation. The final stage is to verify the existence of the target object by certain procedure using the measurements of the feature similarities.

According to the generalization of the object recognition techniques, the extraction and description of the image features are of vital importance. Of all the heterogeneous features of an object, the shape information is an essential one. A literature review of the shape representation and description techniques is presented in Chapter 1.

Chapter 2 of this thesis addressed the problem of object recognition and classification based on shape representation. The research has focused on the process of shape matching and resulted in an efficient, correspondence-based shape classification algorithm for 2D boundaries. In this method, shapes are described globally using the Shape Context, which is a shape feature descriptor. Then matched points are detected by strategy of cost ratio between

the nearest and the second nearest neighboring points, followed by a Gaussian filter applied to filter the cost of matched points. After the summation of the output of the filter, the similarity value between each class is calculated to classify the object. In addition, the performance of the classification algorithm has been improved by the weighting strategy using the Harris corner measure. The algorithm was tested on the MPEG-7 shape database and had a very good performance. The ideas presented are more relevant to generic shape classification and retrieval problems which have insufficient knowledge to construct a priori models. So there is a need to investigate a learning strategy which could automatically build a model for each class with training objects.

Apart from the shape information, another important image feature is texture of the object, which also represents the object's characteristics. A rich literature of descriptors has been published, which either extract the shape or texture feature of the image. The state-of-the-art object recognition approaches based on these descriptors have been discussed in Chapter 1.

From Chapter 3, the focus of this thesis is narrowed to the object identification techniques. To investigate and analyze the efficiency and reliability of the object detector based on local image features, two identifiers have been studied and compared. Both of the identifiers rely on a single type of feature descriptors to represent and store the characteristics of the prototype objects. In order to solve the problem of detecting the object out from a complex scene, in which plenty of irrelevant objects and noise may occlude and distort the target object, not only the feature descriptors are required to be able to catch the properties of the object precisely and robustly, they are also demanded to be discriminative enough with each other to enable an accurate retrieval in the vast database. Considering the changes of the target object in the complex scene, including scale, orientation, perceived point and location etc., the descriptors need to be constructed in a way which is invariant to all these transformations. Two descriptors, shape context and SIFT, both fulfill the above requirements and have been

successfully applied in many areas in these days. The shape context descriptor describes the shape features of the object, while the SIFT descriptor captures the texture information. The results of the simulation illustrate the limitation of both methods. Neither of the identifier achieves the goal of detecting a general object in a complex scene. For SIFT descriptor based identifier, the texture-less object could hardly be recognized, while the shape feature based identifier performs poor when the target object is messing with other objects and noise in the complex scene. Both of the identifiers only operate under specific conditions. The object identification strategies employing single type of image features could only capture the correspondent features on the objects. If the target object is rich with this feature and this type of feature distinguishes the target object in the complex scene, the identifier could detect the target object. However, objects in the real life image represent themselves with heterogeneous features and their main features vary from each other. Therefore, the object identifier constructed on a single type of image features could not detect the general objects.

Consequently, it is thought that by combining different types of descriptors, the target object could be easily detected. However, the identifiers are always designed according to characteristics of their corresponding feature descriptors, as the ones studied in Chapter 3 and Chapter 4. This results in quite different identification strategies and raises the question of how to combine the descriptors in a way which could take full advantage of each type of image features. To tackle this problem, in Chapter 5, I have introduced a novel object identification approach, which employs the cosine similarity measure to build a ‘resemblance map’. The prototype and complex images are divided into small image patches, on which the histogram-based descriptors, SIFT and HOG, are modified and utilized to capture the texture and shape features of the objects, respectively. In this way, the features of the images are densely extracted and the employment of these features collectively offers far more significant discriminative power than the key-point based detector. For each image patch on the complex image, an image window of the same size as the prototype image is established with its

neighboring patches. The similarity between each image window and prototype image is computed by the proposed resemblance measure which calculates the cosine similarity of their SIFT and HOG descriptors. Then by assigning each image patch its similarity value, a likelihood map is formed based on the shape and texture features of the images. Besides these two features, other types of features can also be incorporated by adding their cosine similarities into the formation of the likelihood map. After the estimation of the 'resemblance map', the appearance of the target object is decided and localized by the image patch whose similarity value passing a certain threshold. The method is also designed to be scale and rotation invariant. The usage of cosine similarity has been justified theoretically and derived from a naïve Bayes decision rule. The identification method is applied on various images and objects of different characters, e.g. shape or texture rich. The results have verified the robustness and efficiency of the detector, even under large rotation and scale changes.

Bibliography

[1] Dengsheng Zhang, Guojun Lu, Review of shape representation and description techniques, *Pattern Recognition*, Volume 37, Issue 1, January 2004, Pages 1-19

[2] Krystian Mikolajczyk, Cordelia Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1615-1630, October, 2005

[3] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT(8):179–187, 1962.

[4] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis and Machine Vision, Chapman & Hall, London, UK, NJ, 1993, pp. 193–242.

[5] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1992, pp. 502–503.

[6] B. Jahne, Digital Image Processing—Concepts, Algorithms andScienti1c Applications, Springer, Berlin, Heidelberg, 1997, pp. 509–512.

[7] S.X. Liao, M. Pawlak, On image analysis by moments, IEEE Trans. Pattern Anal. Mach. Intell. 18 (3) (1996) 254–266.

[8] M.R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Am. 70 (8) (1980) 920–930.

[9] C.-H. Teh, R.T. Chin, On image analysis by the methods of moments, *IEEE Trans. Pattern Anal. Mach. Intell.* 10 (4) (1988) 496–513.

[10] B.M. Mehtre, M.S. Kankanhalli, W.F. Lee, Shape measures for content based image retrieval: a comparison, *Inf. Process. Manage.* 33 (3) (1997) 319–337.

[11] A.S. Dudani, K.J. Breeding, R.B. McGhee, Aircraft identi1cation by moment invariants, *IEEE Trans. Comput*. C-26 (1) (1977) 39–46.

[12] S.O. Belkasim, M. Shridhar, M. Ahmadi, Pattern recognition with moment invariants: a comparative study and new results, *Pattern Recognition* 24 (12) (1991) 1117–1138.

[13] R.J. Prokop, A.P. Reeves, A survey of moment-based techniques for unoccluded object representation recognition, *Graph. Models Image Process*. 54 (1992) 438–460.

[14] G. Taubin, D.B. Cooper, Recognition and positioning of rigid objects using algebraic moment invariants, *SPIE Conference on Geometric Methods in Computer Vision*, Vol. 1570, 1991, pp. 175–186.

[15] G. Taubin, D.B. Cooper, Object recognition based on moment (or Algebraic, in: J. Mundy, A. Zisserman (Eds.), Geometric Invariance in Computer Vision, MIT Press, Cambridge, MA, 1992, pp. 375–397.

[16] B. Scassellati, S. Slexopoulos, M. Flickner, Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgments, *SPIE Proceedings on Storage and Retrieval for Image and Video Databases II*, Vol. 2185, San Jose, CA, USA 1994, pp. 2–14.

[17] M.R. Teague, Image analysis via the general theory of moments, *J. Opt. Soc. Am*. 70 (8) (1980) 920–930.

[18] C.-H. Teh, R.T. Chin, On image analysis by the methods of moments, *IEEE Trans. Pattern Anal. Mach.* Intell. 10 (4) (1988) 496–513.

[19] S.X. Liao, M. Pawlak, On image analysis by moments, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (3) (1996) 254–266.

[20] D.S. Zhang, G. Lu, Generic Fourier descriptor for shape-based image retrieval, *Proceedings of IEEE International Conference on Multimedia and Expo (ICME2002),* Vol. 1, Lausanne, Switzerland, August 26–29, 2002, pp. 425–428.

[21] D.S. Zhang, G. Lu, Enhanced generic Fourier descriptor for object-based image retrieval, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2002)*, Vol. 4, Orlando, FL, USA, May 13–17, 2002, pp. 3668–3671.

[22] G.J. Lu, A. Sajjanhar, Region-based shape representation and similarity measure suitable for content-based image retrieval, *Multimedia Syst.* 7 (2) (1999) 165–174.

[23] A. Goshtasby, Description and discrimination of planar shapes using shape matrices, *IEEE Trans. Pattern Anal. Mach. Intell.* 7 (1985) 738–743.

[24] E.R. Davies, Machine Vision: Theory, Algorithms, Practicalities, Academic Press, New York, 1997, pp. 171–191.

[25] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis and Machine Vision, Chapman & Hall, London, UK, NJ, 1993, pp. 193–242.

[26] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1992, pp. 502–503.

[27] H. Blum, A transformation for extracting new descriptors of shape, *W. Whaten-Dunn (Ed.), Models for the Perception of Speech and Visual Forms*, MIT Press, Cambridge, MA, 1967, pp. 362–380.

[28] E.R. Davies, Machine Vision: Theory, Algorithms, Practicalities, Academic Press, New York, 1997, pp. 171–191.

[29] P.J. van Otterloo, A Contour-Oriented Approach to Shape Analysis, Prentice-Hall International (UK) Ltd, NJ, 1991, pp. 90–108.

[30] D.S. Zhang, G. Lu, A comparative study of Fourier descriptors for shape representation and retrieval, *Proceedings of the Fifth Asian Conference on Computer Vision (ACCV02)*, Melbourne, Australia, January 22–25, 2002, pp. 646–651.

[31] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis and Machine Vision, Chapman & Hall, London, UK, NJ, 1993, pp. 193–242.

[32] R. Chellappa, R. Bagdazian, Fourier coding of image boundaries, *IEEE Trans. Pattern Anal. Mach. Intell.* 6 (1) (1984) 102–105.

[33] M. Das, M.J. Paulik, N.K. Loh, A bivariate autoregressive modeling technique for analysis and classi1cation of planar shapes*, IEEE Trans. Pattern Anal. Mach. Intell.* 12 (1) (1990) 97–103.

[34] S.R. Dubois, F.H. Glanz, An autoregressive model approach to two-dimensional shape classi1cation*, IEEE Trans. Pattern Anal. Mach. Intell. 8* (1986) 627–637.

[35] K. EOM, J. Park, Recognition of shape by statistical modeling of centroidal profile, *Proceedings of the Tenth International Conference on Pattern Recognition*, Vol. 1, Atlantic City, NJ, 1990, pp. 860–864.

[36] K.L. Kashyap, R. Chellappa, Stochastic models for closed boundary analysis: representation and reconstruction*, IEEE Trans. Inform. Theory* 27 (1981) 627–637.

[37] Y. He, A. Kundu, 2-D shape classi1cation using hidden Markov model, *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (11) (1991) 1172–1184.

[38] I. Sekita, T. Kurita, N. Otsu, Complex autoregressive model for shape recognition*, IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992) 489–496.

[39] R. Chellappa, R. Bagdazian, Fourier coding of image boundaries, *IEEE Trans. Pattern Anal. Mach. Intell.* 6(1) (1984) 102–105.

[40] J.R. Ohm, F.B. Bunjamin, W. Liebsch, B. Makai, K. Muller, A. Somlic, D. Zier, A set of visual feature descriptors and their combination in a low-level description scheme, *Signal Process. Image Commun.* 16 (2000) 157–179.

[41] Q.M. Tieng, W.W. Boles, Recognition of 2D object contours using the wavelet transform zero-crossing representation*, IEEE Trans. Pattern Anal. Mach. Intell.* 19 (8) (1997) 910–916.

[42] H.S. Yang, S.U. Lee, K.M. Lee, Recognition of 2D object contours using starting-point-independent wavelet coefficient matching, *J. Visual Commun. Image Represent*. 9 (2) (1998) 171–181.

[43] D.S. Zhang, G. Lu, A comparison of shape retrieval using Fourier descriptors and short-time Fourier descriptors, *Proceedings of the Second IEEE Paci1c-Rim Conference on Multimedia (PCM01)*, Beijing, China, October 24–26, 2001, pp. 855–860.

[44] F.J.S. Marine, Automatic recognition of biological shapes with and without representation of shape*, Artif. Intell*. Med. 18 (2000) 173–186.

[45] F. Mokhtarian, A. Mackworth, Scale-based description and recognition of planar curves and two-dimensional shapes*, IEEE Pattern Anal. Mach. Intell*. 8 (1) (1986) 34–43.

[46] M. Daoudi, S. Matusiak, Visual image retrieval by multi-scale description of user sketches, *J. Visual Lang. Comput*. 11 (2000) 287–301.

[47]Li Fei-Fei, Rob Fergus, Pietro Perona, Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshop Generative-Model Based Vision*, 2004.

[48] Manik Varma, Debajyoti Ray, "Learning The Discriminative Power-Invariance Trade-Off," Computer Vision, IEEE International Conference on, pp. 1-8, 2007 IEEE 11th International Conference on Computer Vision, 2007.

[49]G. Griffin, A.Holub and P.Perona. Caltech-256 object category dataset. Technical report, CalTech, 2007.

[50]Svetlana Lazebnik, Cordelia Schmid, Jean Ponce, "A Sparse Texture Representation Using Local Affine Regions," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1265-1278, August, 2005

[51]A.Opelt, M.Fussenegger, A. Pinz, and P.Auer. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV*, 2004

[52] M. E. Nilsback and A. Zisserman, A visual vocabulary for flower classification. In CVPR, volume 2, pages 1447-1454, New York, 2006

[53] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with spatial pyramid kernel. In *Proc. CIVR*, 2007.

[54] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 2007.

[55] H.Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.

[56] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition,* 2006.

[57] O. Boiman, E. Shechtman, and M. Irani, In defense of nearest-neighbor based image classification, *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 947-954, 2005.

[58] S. Giannarou and T. Stathaki, Object identification in complex scenes using shape context descriptor and multi-stage clustering, *Proceedings of the 15th international conference on digital signal processing* (2007), pp. 244–247.

[59] D.G. Lowe. Local features view clustering for 3D object recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, pp. 682-688

[60] S. Agarwal, A. Awan, D. Roth, Learning to detect objects in images via a sparse, part-based representation, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, No. 11. (2004), pp. 1475-1490.

[61] Bo Wu, R. Nevatia, Simultaneous object detection and segmentation by boosting local shape feature based classifier, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, June 2007.

[62] Ashish Kapoor, John Winn, Located hidden random fields: learning discriminative parts for object detection, *Proc. European Conf. Computer Vision,* vol. 3954, pp. 302-315, May 2006

[63] E. Shechtman and M. Irani, “Matching Local Self-Similarities across Images and Videos,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, June 2007.

[64]E. Shechtman and M. Irani, “Space-Time Behavior-Based Correlation—or—How to Tell If Two Underlying Motion Fields Are Similar without Computing Them?” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 2045-2056, Nov. 2007.

[65]C. Yeo, P. Ahammad, K. Ramchandran, and S.S. Sastry, “High-Speed Action Recognition and Localization in Compressed Domain Videos,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1006-1015, Aug. 2008.

[66] Hae Jong Seo, Peyman Milanfar, "Training-Free, Generic Object Detection Using Locally Adaptive Regression Kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1688-1704, September, 2010

[67] O. Boiman and M. Irani. Detecting irregularities in images and video. In *ICCV*, Beijing, October 2005.

[68] N. Dalal and B. Triggs, Histogram of oriented gradients for human detection, *IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 1, pp. 886-893, 2005.

[69] D. Lowe, Distinctive Image Features from Scale-Invariant Key-points, *Int’l J. Computer Vision*, vol. 20, pp. 91-110, 2004.

[70] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. IEEE PAMI, 27(10):1615–1630, 2005.

[71] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 511-517, 2004.

[72] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. Intl. Workshop on Automatic Face-and Gesture- Recognition, IEEE Computer Society, pp. 296-301, June 1995.

[73]A Bissacco, M. H. Yang, and S. Soatto. Detecting humans via their pose. In NIPS, 2006.

[74]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.

[75]T. Tuytelaar and C. Schmid, “Vector Quantizing Feature Space with a Regular Lattice,” Proc. IEEE Int’l Conf. Computer Vision, pp. 1-8, Oct. 2007.

[76]F. Jurie and B. Triggs, “Creating Efficient Codebooks for Visual Recognition,” Proc. IEEE Int’l Conf. Computer Vision, vol. 1, pp. 604-610, 2005.

[77]D. Lin, S. Yan, and X. Tang, “Comparative Study: Face Recognition on Unspecific Persons Using Linear Subspace Methods,” Proc. IEEE Int’l Conf. Image Processing, vol. 3, pp. 764-767, 2005.

[78]Y. Ma, S. Lao, E. Takikawa, and M. Kawade, “Discriminant Analysis in Correlation Similarity Measure Space,” Proc. 24th Int’l Conf. Machine Learning, vol. 227, pp. 577-584, 2007.

[79]Y. Fu, M. Liu, and T.S. Huang, “Conformal Embedding Analysis with Local Graph Modeling on the Unit Hypersphere,” Proc. IEEE CVPR First Workshop Component Analysis, 2007.

[80] Y. Fu, S. Yan, and T.S. Huang, “Correlation Metric for Generalized Feature Extraction,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 30, no. 12, pp. 2229-2235, Dec. 2008.

[81]F. Devernay, “A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy,” Technical Report RR-2724, Institut National de Recherche en Informatique et en Automatique, 1995.

[82] C. Liu, “The Bayes Decision Rule Induced Similarity Measures,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1086-1090, June 2007.

[83] R. Duda, P. Hart, and D. Stark, Pattern Classification, second ed.John Wiley and Sons, Inc., 2000.

[84] J. F. Canny. A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell., 8(6):679–698, Nov 1986.

[85] B.J. Super. Fast correspondence-based system for shape-retrieval. *Patt. Recog. Lett.*, 25:217–225, 2004. pp. 642–651, 1996.

[86] J. Zhang, X. Zhang, H. Krim, and G.G. Walter. Object representation and recognition in shape spaces. *Patt. Recog.*, 36:1143–1154, 2003.

[87] S. Wang, T. Kubota, and T. Richardson. Shape correspondence through landmark sliding. In *IEEE Conf. on Comp. Vis. and Patt. Recog.*, volume 1, pages

143–150, 2004.

[88] K. Mikolajczyk and C. Schmid, Scale & Affine Invariant Interest Point Detectors, *International Journal of Computer Vision*, no. 60, pp. 63-86, 2004

[89]G.Z. Yang, P. Burger, D.N. Firmin, S.R Underwood, Structure adaptive anistropic image filtering, *Image and Vision Computing*, vol.14, pp. 135-145, 1996

[90]F.Chabat, G.Z Yang, D.M. Hansell, A corner orientation detector, *Image and Vision Computing*, vol.17, pp. 135-145, 1999

[91]A. Kutkus, Object Analysis Using the Covariance Matrix of Local Intensity Derivative Information, MSc thesis, EEE, Imperial College, Sep, 2004

[92]T. Lindeberg, Feature Detection with Automatic Scale Selection, *International Journal of Computer Vision*, vol.30, no.2, 1998.

[93] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In Eighth IEEE International Conference on Computer Vision, volume 1, pages 454-461, Vancouver, Canada, July, 2001.

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

[95] H. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute, 1980.

[96] A. Noble, Descriptions of Image Surfaces, PhD thesis, Department of Engineering Science, Oxford University 1989, p45.

[97] Harold W. Kuhn, Variants of the Hungarian method for assignment problems, *Naval Research Logistics Quarterly*, **3**: 253–258, 1956.

[98] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.

[99] S. Chiu. Fuzz model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3), 1994

[100] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 1150-1157, 1999.

[101] Witkin, A. P. 1983. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, pp. 1019-1022

[102] T Lindeberg. Scale-space theory: A basis tool for analysis structures at different scales. *Journal of Applied Statistics*, 21(2):224-270

[103] M. Brown and D.G. Lowe, Invariant features from interest point groups. In *British Machine Vision Conference*, Cardiff, Wales, pp. 656-665

[104] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1992, pp. 502–503.