ABSTRACT

Title of dissertation: Techniques for Image Retrieval:
Deformation Insensitivity and
Automatic Thumbnail Cropping

Haibin Ling, Doctor of Philosophy, 2006

Dissertation directed by: Professor David W. Jacobs
Department of Computer Science

We study several problems in image retrieval systems. These problems and proposed techniques are divided into three parts.

Part I: This part focuses on robust object representation, which is of fundamental importance in computer vision. We target this problem without using specific object models. This allows us to develop methods that can be applied to many different problems. Three approaches are proposed that are insensitive to different kind of object or image changes. First, we propose using the inner-distance, defined as the length of shortest paths within shape boundary, to build articulation insensitive shape descriptors. Second, a deformation insensitive framework for image matching is presented, along with an insensitive descriptor based on geodesic distances on image surfaces. Third, we use a gradient orientation pyramid as a robust face image representation and apply it to the task of face verification across ages.

Part II[1]: This part concentrates on comparing histogram-based descriptors that are widely used in image retrieval. We first present an improved algorithm of the Earth Mover's Distance (EMD), which is a popular dissimilarity measure between histograms. The new algorithm is one order faster than original EMD algorithms. Then, motivated by the new algorithm, a diffusion-based distance is designed that is more straightforward and efficient. The efficiency and effectiveness of the proposed approaches are validated

[1] This work is supervised by Dr. Kazunori Okada at Siemens Corporate Research during my internship.

in experiments on both shape recognition and interest point matching tasks, using both synthetic and real data.

Part III[2]: This part studies the thumbnail generation problem that has wide application in visualization tasks. Traditionally, thumbnails are generated by shrinking the original images. These thumbnails are often illegible due to size limitation. We study the ability of computer vision systems to detect key components of images so that intelligent cropping, prior to shrinking, can render objects more recognizable. With this idea, we propose an automatic thumbnail cropping technique based on the distribution of pixel saliency in an image. The proposed approach is tested in a carefully designed user study, which shows that the cropped thumbnails are substantially more recognizable and easier to find in the context of visual search.

[2] Part of this work is jointly done with Bongwon Suh under supervision of Professor Ben B. Bederson.

Techniques in Image Retrieval:
Deformation Insensitivity and
Automatic Thumbnail Cropping

by

Haibin Ling

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:

Professor David W. Jacobs, Advisor
Professor Yiannis Aloimonos
Professor Ben B. Bederson
Professor Rama Chellappa
Professor Larry S. Davis
Professor Min Wu

ACKNOWLEDGMENTS

First and foremost I would like to thank my advisor, Professor David W. Jacobs for giving me an invaluable opportunity to work on challenging and interesting problems. Step by step, he guided me into the area of computer vision and taught me the way to do research, with his stimulating comments and infinite patience. It has been a pleasure to work with and learn from such a great advisor and I will cherish this forever.

I am grateful to my committee members, Professors Yiannis Aloimonos, Ben B. Bederson, Rama Chellappa, Larry S. Davis, and Min Wu, for their insightful comments on my research. I would also greatly thank Dr. Kazunori Okada for the supervision during my internship at Siemens Corporate Research (SCR) in 2005. I am also indebted to Dr. Kevin S. Zhou for helpful discussions and encouragements. In addition, I would like to thank Professor Amitabh Varshney for his help during my Ph.D. study.

I owe my thanks to helpful discussions and collaborations from my colleagues: Gaurav Agarwal, Feng Guo, Narayanan Ramanathan, Sameer Shirdhonkar, Bongwon Suh, and Zhanfeng Yue. I enjoyed a pleasant stay at the Center for Automation Research (CfAR) and thank all my friends who make this true: Jian Li, Xue Mei, Hui Ji, Arunkumar Mohananchettiar, Yang Ran, Jie Shao, Hao Wu, etc.

I owe my deepest thanks to my family - I take this opportunity thank my parents and parents-in-law for their support and to wish them the best. Finally, I thank my wife, Mei, for her patience, encouragement, and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Part I

Robust Descriptors for Object Recognition

In this dissertation we describe our study on several problems in computer vision and image retrieval. The whole dissertation contains three parts. The first part describes three robust object representations, focusing on shape articulation, image deformation and an application to face verification across age respectively. The second part proposes two robust and efficient approaches for comparing histogram-based descriptors that are widely used in computer vision tasks. The third part introduces a novel automatic thumbnail cropping technique based on visual attention models.

Chapter 1

Introduction to Part I

1.1 Deformation in Computer Vision

According to the Merriam-Webster Online Dictionary, *deformation* has three meanings[1]:

1. Alteration of form or shape; also : the product of such alteration.
2. The action of deforming : the state of being deformed.
3. Change for the worse.

Similar definitions can be found in WordNet (http://wordnet.princeton.edu/). The first two explanations relate to shape change that appears pervasively in computer vision. The third explanation, seemingly irrelevant to computer vision, happens to imply the unpleasant consequences brought to computer vision tasks by deformation.

[1] http://www.m-w.com/dictionary/deformation

In computer vision, since the inputs are usually images, the definition of deformation needs to be broadened because image deformations may not necessary reflect true shape change of the contents of the image. In the following, we discuss several common examples of deformation in computer vision (see Fig. 1.1), including articulation that is a special case of deformation, the "deformation" in illumination, and the deformation caused by viewpoint change.

- Shapes can deform by themselves. A person walks, a flag flies, a leaf grows, a man gains weight while a woman becomes thinner, etc. Fig. 1.1 (a) and (b) show two examples.

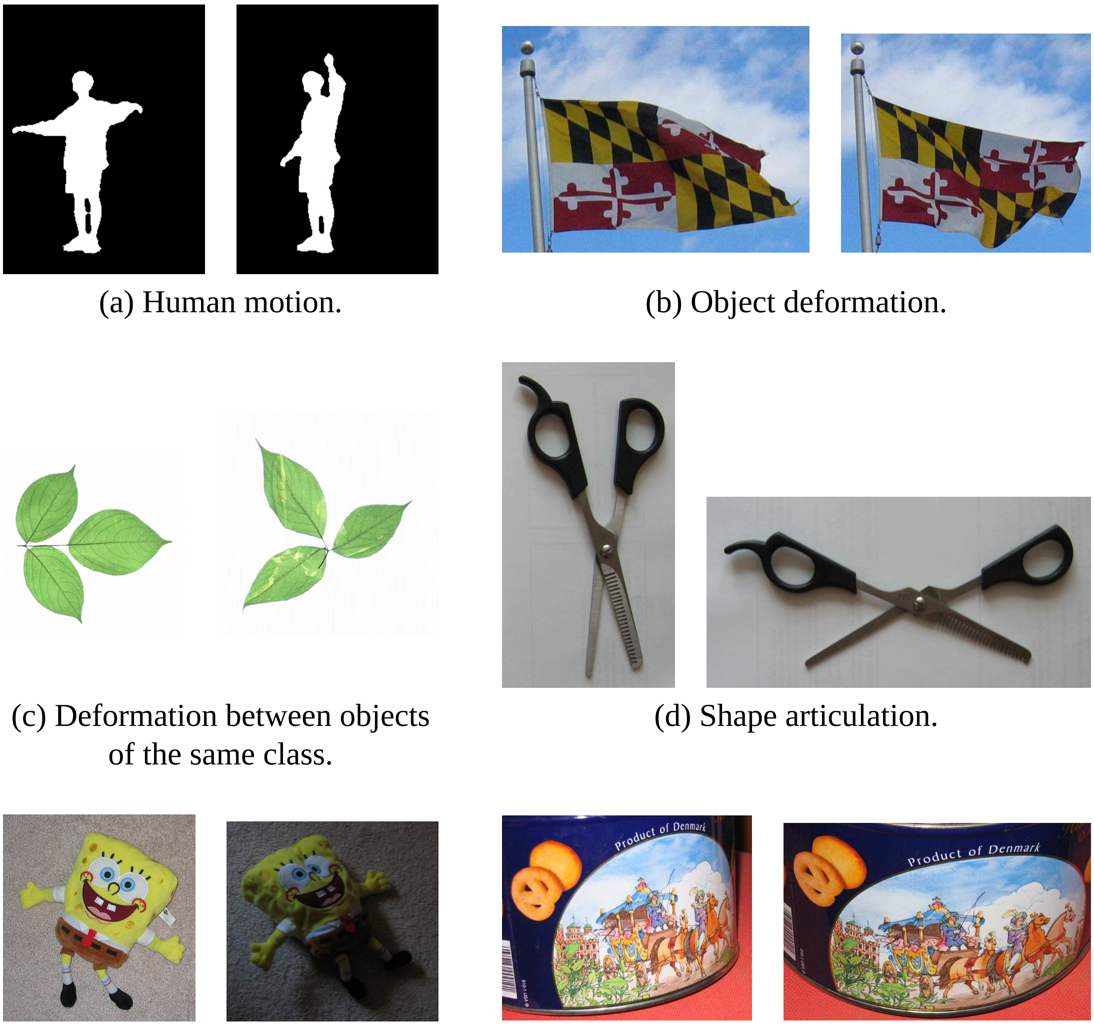
- Objects from the same category often have similar shapes related by deformation, usually non-linear, e.g. the two compound leaves in Fig. 1.1 (c)).

- Articulation is a special case of deformation, in that some *parts* of objects are rigid while the geometric relations between parts, through *junctions*, may change. Fig. 1.1 (d) gives an example of articulation. Articulation often happens together with other shape changes, e.g. Fig. 1.1 (c).

- As mentioned above, in computer vision, deformation sometimes happens without real shape change. The most important example is the deformation caused by viewpoint change. Particularly, for non-planar structures, viewpoint change usually generates non-linear deformation, such as in Fig. 1.1 (f).

- In addition to geometric change in images, illumination variation can also be treated as a special case of deformation, in that the deformation is in the intensity (or color) space. An example is shown in Fig. 1.1 (d).

- To further generalize the definition of deformation, it can even include other photometric changes such as texture change. For example, the face of the same person over years can have different albedos (see Fig. 1.2), and even different geometric properties (due to wrinkles, etc).



(a) Human motion.

(b) Object deformation.

(c) Deformation between objects of the same class.

(d) Shape articulation.

(e) “Deformation” caused by illumination change.

(f) Deformation caused by viewpoint change of non-planar structure.

Figure 1.1: Examples of deformation in computer vision.

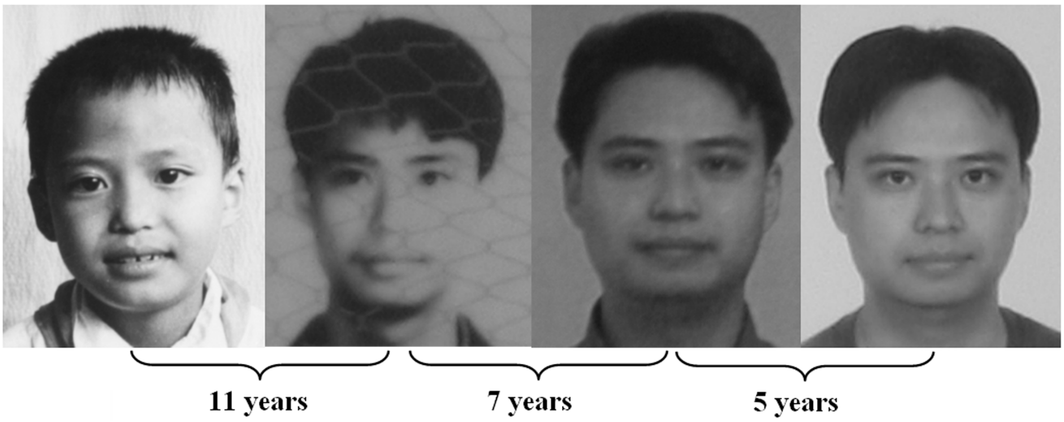


Figure 1.2: Examples of four images from a person taken at different ages. The true gaps between neighboring images are indicated under each pair.

The above examples, though far from comprehensive, demonstrate the pervasiveness of deformation in computer vision. Not surprisingly, it attracts a lot of research efforts in lots of computer vision areas, such as in object recognition [51, 15, 18, 36, 38, 165], face recognition [39, 30], shape analysis [19, 37, 12, 138, 146], medical image analysis [28, 152, 153], invariant description [88, 96, 103], stereo matching [100, 148], image retrieval [130, 132], motion analysis [121, 163], etc. Previous work dealing with deformation can be roughly categorized into two classes. The first class uses specified models to handle deformation, such as a model of an object with several parts and a statistical model of the relation between parts. These approaches are efficient for specific tasks such as face recognition, human tracking, etc. The second class represents deformation in a general way. Examples include the work of invariant descriptors, the shock graph as a general model, etc. Detailed discussions and examples of these works can be found in the following chapters.

In this dissertation, we will study the second class of deformations. Specifically, we consider the problem from two points of view. On the one hand, we are interested in building descriptions that are invariant to deformation. On the other hand, we search for methods to compare existing descriptors in a way insensitive to deformation. In the next section, we will describe motivations for three proposed approaches in these two directions.

1.2 Motivations

In the following we will describe the motivation for three approaches for building robust object descriptors. The first one is specialized to handle shape articulation. The second one focuses on deformation invariance in images. The third one is a robust face representation for face verification across age.

1.2.1 Articulation Insensitivity and the Inner-Distance

Intuitively, the articulation of shapes can be defined as a *part-wise* rigid transformation, with non-rigidity restricted to *junctions*. Given two points on an object, the length of the shortest path between them, when restricted within the object boundary, is invariant to articulation in the ideal case. This is because 1) the rigidity of parts will not change the shortest path length and 2) the sizes of junctions are zero. We call this length the *inner-distance*.

In practice, junction sizes are usually not zero. However, they are very small compared to the rigid parts - otherwise they become parts perceptually. This implies that the length of the above mentioned shortest path may change a little, but it is still safe to say that it is *insensitive* to articulation.

How should we use the inner-distance? A natural way is to use the inner-distance to replace the Euclidean distance, which is widely used in building shape descriptors. In Chapter 2, we will show how to extend the shape context [15] using the inner-distance and describe experiments with the new descriptors.

1.2.2 Deformation Invariant Image Descriptor

What is an invariant for intensity images? During a general deformation, i.e., a one-to-one continuous transformation, pixel positions can change wildly, but pixel intensities remain constant. A simple count of existing intensities is deformation invariant. Unfortunately, this loses too much spatial information to be truly useful. This hints to us to combine the intensity and spatial coordinates together. A natural solution is to treat an image as a two dimensional surface in three dimensional space, where intensity serves as the third dimension.

Based on the above idea, we propose a novel framework to build descriptors of local intensity that are invariant to general deformations. In this framework, an image is embedded as a 2D surface in 3D space (this technique can be traced back to Koenderink [74] or earlier), with intensity weighted relative to distance in x-y. We show that as this weight increases, geodesic distances on the embedded surface are less affected by image deformations. In the limit, distances are deformation invariant. We use geodesic sampling to get neighborhood samples for interest points, then use a geodesic-intensity histogram (GIH) as a deformation invariant local descriptor. In addition to its invariance, the new descriptor automatically finds its support region. This means it can safely gather information from a large neighborhood to improve discriminability. In an experiment of interest point matching on image pairs with both synthetic and real deformation, our method shows promising matching results compared to several other approaches.

1.2.3 Using A Gradient Orientation Pyramid for Robust Photo Verification

Face recognition and detection has been widely studied for several decades [164]. In comparison, face verification across ages is far less studied [123]. This is a challenging task because human faces can vary a lot over time in many aspects, including facial texture (e.g. from winkles), shape (e.g. from weight gain), facial hair, presence of glasses, etc. In addition, the image acquisition conditions and environment for taking face photos often undergoes a large change, which can cause non-uniform illumination and scale changes. Because of this reason, a robust representation is desired for the task.

Inspired by the work on lighting insensitivity with a Lambertian assumption and its application to face recognition [25], we propose using a gradient orientation pyramid (GOP) as a reliable face descriptor. First, by discarding magnitude information, the gradient orientation is insensitive to lighting change [25]. Second, we use the pyramid to bring in hierarchical information. Then, given a face image pair, we use the cosines between gradient orientations at all scales to build the “difference” between the pair. Finally, the “difference” is combined with the support vector machine (SVM) [149] for face verification tasks. We applied the proposed approach to passport verification tasks and tested it on two passport image datasets with large age differences. Promising results are observed in comparison with several other approaches, including the Bayesian+PointFive face [123], the SVM+difference space [120], two commercial face recognition products, etc.

1.3 Outline of Part I

In the rest of the dissertation, we will describe these three approaches in detail.

Chapter 2 presents the inner-distance, including its properties and how to use it to improve existing descriptors. In particular, we will discuss the inner-distance shape context and its application to foliage image retrieval.

Chapter 3 focuses on the deformation invariant framework for matching intensity images. The theoretic discussion and the Geodesic-intensity histogram is discussed in detail, along with experiments.

Chapter 4 describes the propose robust image representation for a face verification task. We introduce the gradient orientation pyramid and show how it is combined with support vector machines for face verification tasks. The experimental results on two passport datasets containing large age variations are demonstrated and analyzed.

Chapter 2

Shape Descriptors Using the Inner-Distance

2.1 Introduction

Part structure plays a very important role in classifying complex shapes in both human vision and computer vision [58, 18, 73] etc. However, capturing part structure is not a trivial task, especially considering articulations, which are nonlinear transformations between shapes. To make things worse, sometimes shapes can have ambiguous parts (e.g. [12]). Unlike many previous methods that deal with part structure explicitly, we propose an implicit approach to this task.

In this chapter we introduce the *inner-distance*, defined as the length of the shortest path within the shape boundary, to build shape descriptors. It is easy to see that the inner-distance is insensitive to shape articulations. For example, in Fig. 2.1, although the points on shape (a) and (c) have similar spatial distributions, they are quite different in their part structures. On the other hand, shapes (b) and (c) appear to be from the same category with different articulations. The inner-distance between the two marked points is quite different in (a) and (b), while almost the same in (b) and (c). Intuitively, this example shows that the inner-distance is insensitive to articulation and sensitive to part structures, a desirable property for complex shape comparison. Note that the Euclidean distance does not have these properties in this example. This is because, defined as the length of the line segment between landmark points, the Euclidean distance does not consider whether the line segment crosses shape boundaries. In this example, it is clear that the inner-distance reflects part structure and articulation without explicitly decomposing shapes into parts. We will study this problem in detail and give more examples in the following sections.

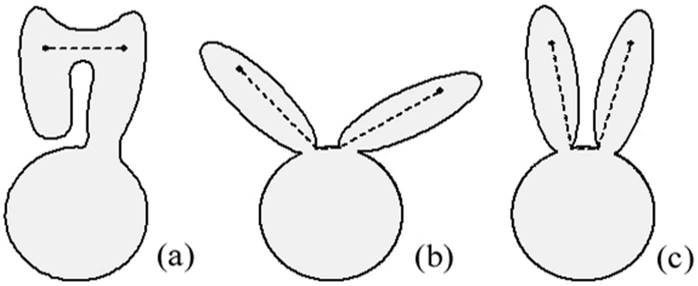


Figure 2.1: Three objects. The dashed lines denote shortest paths within the shape boundary that connect landmark points.

It is natural to use the inner-distance as a replacement for other distance measures to build new shape descriptors that are invariant/insensitive to articulation. In this chapter we propose and experiment with two approaches. In the first approach, by replacing the geodesic distance with the inner-distance, we extend the bending invariant signature for 3D surfaces [36] to the articulation invariant signature for 2D articulated shapes. In the second method, the inner-distance replaces the Euclidean distance to extend the shape context [15]. We design a dynamic programming method for silhouette matching that is fast and accurate since it utilizes the ordering information between contour points. Both approaches are tested on a variety of shape databases, including an articulated shape database[1], MPEG7 CE-Shape-1 shapes, Kimia's silhouette [136, 134], ETH-80 [82], a Swedish leaf database [141] and a Smithsonian leaf database [7]. The excellent performance demonstrates the inner-distance's ability to capture part structures (not just articu-

[1] This is a dataset we collected and available at http://www.cs.umd.edu/∼hbling/Research/data/articu.zip

lations).

In practice, it is often desirable to combine shape and texture information for object recognition. For example, leaves from different species often share similar shapes but have different vein structures (see Fig. 2.13 for examples). Using the gradient information along the shortest path, we propose a new shape descriptor that naturally takes into account the texture information inside a given shape. The new descriptor is applied to a foliage image task and excellent performance is observed.

The rest of this chapter is organized as follows. Sec. 2.2 discusses related works. Sec. 2.3 first gives the definition of the inner-distance and its computation. Then the articulation insensitivity of the inner-distance is proved. After that we address the inner-distance's ability to capture part structures. Sec. 2.4 describes using the inner-distance and MDS to build articulation insensitive signatures for 2D articulated shapes. Sec. 2.5 describes the extension of the shape context using the inner-distance, and gives a framework for using dynamic programming for silhouette matching and comparison. Sec. 2.6 introduces the new shape descriptor that captures texture information. Sec. 2.7 presents and analyzes all experiments. Sec. 2.8 concludes the paper.

Part of this work appears in [91, 89, 7].

2.2 Related Work

2.2.1 Representation and Comparison of Shapes with Parts and Articulation

For general shape matching, a recent review is given in [151]. Roughly speaking, works handling parts can be classified into three categories. The first category (e.g. [8, 51, 39, 131, 40, 155] etc.) builds part models from a set of sample images, and usually with some prior knowledge such as the number of parts. After that, the models are used for retrieval tasks such as object recognition and detection. These works usually use statistical methods to describe the articulation between parts and often require a learning process to find the model parameters. For example, Grimson [51] proposed some early work performing matching with precise models of articulation. Agarwal et al. [8] proposed a framework for object detection via learning sparse, part-based representations. The method is targeted to objects that consist of distinguishable parts with relatively fixed spatial configuration. Felzenszwalb and Huttenlocher [39] described a general method to statistically model objects with parts for recognition and detection. The method models appearance and articulation separately through parameter estimation. After that, the matching algorithm is treated as an energy minimization problem that can be solved efficiently by assuming that the pictorial representation has a tree structure. Schneiderman and Kanade [131] used a general definition of parts that corresponds to a transform from a subset of wavelet coefficients to a discrete set of values, then builds classifiers based on their statistics. Fergus et al. [40] treated objects as flexible constellations of parts and probabilistically represented objects using their shape and appearance information. These methods have been successfully used in areas such as face and human motion analysis etc. However, for tasks where the learning process is prohibited, either due to the lack of training samples or due to the complexity of the shapes, they are hard to apply.

In contrast, the other two categories (e.g. [73, 12, 134, 138, 47, 95] etc.) capture part structures from only one image. The second category (e.g. [12, 95]) measures the similarity between shapes via a part-to-part (or segment-to-segment) matching and junction parameter distribution. These methods usually use only the boundary information such as the convex portions of silhouettes and curvatures of boundary points.

The third category, which our method belongs to, captures the part structure by considering the interior of shape boundaries. The most popular examples are the skeleton based approaches, particularly the *shock graph*-based techniques ([73, 138, 134] etc.). Given a shape and its boundary, shocks are defined as the singularities of a curve evolution process that usually extracts the skeleton simultaneously. The shocks are then organized into a shock graph, which is a directed, acyclic tree. The shock graph forms a hierarchical representation of the shape and naturally captures its part structure. The shape matching problem is then reduced to a tree matching problem. Shock graphs are closely related to shape skeletons or the medial axis [19, 73]. Therefore, they benefit from the skeleton's ability to describe shape, including robustness to articulation and occlusion. However, they also suffer from the same difficulties as the skeleton, especially in dealing with boundary noise. Another related unsupervised approach is proposed by Gorelick et al. [47]. They used the average length of random walks of points inside a shape silhouette to build shape descriptors. The average length is computed as a solution to the Poisson equation. The solution can be used for shape analysis tasks such as skeleton and part extraction, local orientation detection, shape classification, etc.

The inner-distance is closely related to the skeleton based approaches in that it also considers the interior of the shape. Given two landmark points, the inner-distance can be "approximated" by first finding their closest points on the shape skeleton, then measuring the distance along the skeleton. In fact, the inner-distance can also be computed via the evolution equations starting from boundary points. The main difference between the inner-distance and the skeleton based approaches is that the inner-distance discards the structure of the path once their lengths are computed. By doing this, the inner-distance is more robust to disturbances along boundaries and becomes very flexible for building shape descriptors. For example, it can be easily used to extend existing descriptors by replacing Euclidean distances. In addition, the inner-distance based descriptors can be used for landmark point matching. This is very important for some applications such as motion analysis. The disadvantage is the loss of the ability to perform part analysis. It is an interesting future work to see how to combine the inner-distance and skeleton based techniques.

2.2.2 Geodesic Distances for 3D Surfaces

The inner-distance is very similar to the geodesic distance on surfaces. The geodesic distances between any pair of points on a surface is defined as the length of the shortest path on the surface between them. One of our motivations comes from Elad and Kimmel's work [36] using geodesic distances for 3D surface comparison through multidimensional scaling (MDS). Given a surface and sample points on it, the surface is distorted using MDS, so that the Euclidean distances between the stretched sample points are as similar as possible to their corresponding geodesic distances on the original surface. Since the geodesic distance is invariant to bending, the stretched surface forms a bending invariant signature of the original surface.

Bending invariance is quite similar to the 2D articulation invariance in which we are interested. However, the direct counterpart of the geodesic distance in 2D does not work for our purpose. Strictly speaking, the geodesic distance between two points on the "surface" of a 2D shape is the distance between them along the contour. If a simple (i.e. non self-intersecting), closed contour has length M, then for any point, p, and any $d < M/2$, there will be exactly two points q_1, q_2 that are a distance d away from p, along the contour (see Fig. 2.2 for examples). Hence, a histogram of the geodesic distance to all points on the contour degenerates into something trivial, which does not capture shape. Unlike the geodesic distance, the inner-distance measures the length of the shortest path within the shape boundary instead of along the shape contour (surface). We will show that the inner distance is very informative and insensitive to articulation.

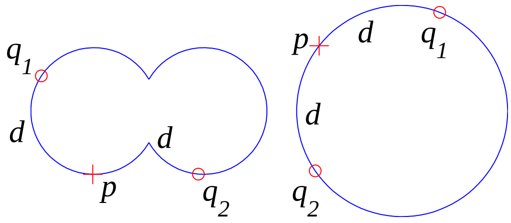


Figure 2.2: Geodesic distances on 2D shapes. Using the geodesic distances along the contours, the two shapes are indistinguishable.

There are other works using geodesic distances in shape descriptions. For example, Hamza and Krim [54] applied geodesic distance using *shape distributions* ([114]) for 3D shape classification. Zhao and Davis [163] used the color information along the shortest path within a human silhouette. The articulation invariance of shortest paths is also utilized by them, but in the context of background subtraction. Ling and Jacobs [90] proposed using the geodesic distance to achieve deformation invariance in intensity images.

2.2.3 Shape Contexts for 2D Shapes

The *shape context* was introduced by Belongie et al. [15]. It describes the relative spatial distribution (distance and orientation) of landmark points around feature points. Given n sample points $x_1, x_2, ..., x_n$ on a shape, the shape context at point x_i is defined as a histogram h_i of the relative coordinates of the remaining $n-1$ points

$$h_i(k) = \#\{x_j : j \neq i, x_j - x_i \in bin(k)\} \tag{2.1}$$

where the bins uniformly divide the log-polar space. The distance between two shape context histograms is defined using the χ^2 statistic.

For shape comparison, Belongie et al. used a framework combining shape context and thin-plate-splines [20] (SC+TPS). Given the points on two shapes A and B, first the point correspondences are found through a weighted bipartite matching. Then, TPS is used iteratively to estimate the transformation between them. After that, the similarity D between A and B is measured as a weighted combination of three parts

$$D = aD_{ac} + D_{sc} + bD_{be} \tag{2.2}$$

where D_{ac} measures the appearance difference. D_{be} measures the bending energy. The D_{sc} term, named the *shape context distance*, measures the average distance between a point on A and its most similar counterpart on B (in the sense of (2.10)). a, b are weights ($a = 1.6$, $b = 0.3$ in [15]).

The shape context uses the Euclidean distance to measure the spatial relation between landmark points. This causes less discriminability for complex shapes with articulations (e.g., Fig. 2.8 and 2.9). The inner-distance is a natural way to solve this problem since it captures the shape structure better than the Euclidean distance. We use the inner-distance to extend the shape context for shape matching. The advantages of the new descriptor are strongly supported by experiments.

Belongie et al. showed that the SC+TPS is very effective for shape matching tasks. Due to its simplicity and discriminability, the shape context has become quite popular recently. Some examples can be found in [110, 145, 147, 162, 111, 82]. Among these works, [145] is most related to our approach. Thayananthan et al. [145] suggested including a figural continuity constraint for shape context matching via an efficient dynamic programming scheme. In our approach, we also include a similar constraint by assuming that contour points are ordered and use dynamic programming for matching the shape context at contour sample points. Notice that usually dynamic programming encounters problems with shapes with multiple boundaries (e.g., scissors with holes). The inner-distance has no such problem since it only requires landmark points on the outermost silhouette, and the shortest path can be computed taking account of holes. This will be discussed in the following sections.

2.3 The Inner-Distance

In this section, we will first give the definition of the inner-distance and discuss how to compute it. Then, the inner-distance's insensitivity to part articulations is proven. After that, we will discuss its ability to capture part structures.

2.3.1 The Inner-Distance and Its Computation

First, we define a shape O as a connected and closed subset of $\mathbb{R}^2$. Given a shape O and two points $x, y \in O$, the inner-distance between x, y, denoted as $d(x, y; O)$, is defined as the length of the shortest path connecting x and y within O. One example is shown in Fig. 2.3.

Note: 1) There may exist multiple shortest paths between given points. However, for most cases, the path is unique. In rare cases where there are multiple shortest paths, we arbitrarily choose one. 2) We are interested in shapes defined by their boundaries, hence only boundary points are used as landmark points. In addition, we approximate a shape with a polygon formed by their landmark points.

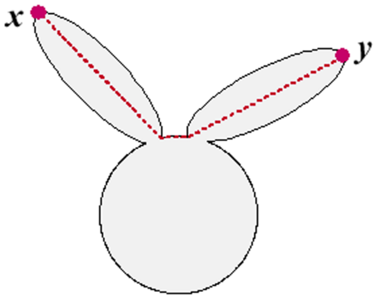


Figure 2.3: Definition of the inner-distance. The dashed polyline shows the shortest path between point x and y.

A natural way to compute the inner-distance is using shortest path algorithms. It consists of two steps:

1. Build a graph with the sample points. First, each sample point is treated as a node in the graph. Then, for each pair of sample points p_1 and p_2, if the line segment connecting p_1 and p_2 falls entirely within the object, an edge between p_1 and p_2 is

added to the graph with its weight equal to the Euclidean distance $\|p_1 - p_2\|$. An example is shown in Fig. 2.4. Note 1) Neighboring boundary points are always connected; 2) The inner-distance reflects the existence of holes without using sample points from hole boundaries[2], which allows dynamic programming algorithms to be applied to shapes with holes.

2. Apply a shortest path algorithm to the graph. Many standard algorithms [31] can be applied here, among them Johnson or Floyd-Warshall's algorithms have $O(n^3)$ complexity (n is the number of sample points).

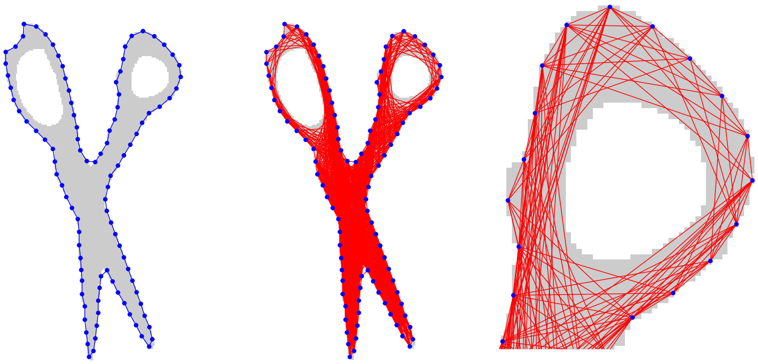


Figure 2.4: Computation of the inner-distance. Left, the shape with the sampled silhouette landmark points. Middle, the graph built using the landmark points. Right, a detail of the right top of the graph. Note how the inner-distance captures the holes.

In this chapter we are interested in the inner-distance between all pairs of points. Now we will show that this can be computed with $O(n^3)$ time complexity for n sample points. First, it takes time $O(n)$ to check whether a line segment between two points

[2] The points along hole boundaries may still be needed for computing the inner-distance, but not for building descriptors.

is inside the given shape (by checking the intersections between line p_1p_2 and all other boundary line segments, with several extra tests). As a result, the complexity of graph construction is of $O(n^3)$. After the graph is ready, the all-pair shortest path algorithm has complexity of $O(n^3)$. Therefore, the whole computation takes $O(n^3)$.

Note that when O is convex, the inner-distance reduces to the Euclidean distance. However, this is not always true for non-convex shapes (e.g., Fig. 2.1). This suggests that the inner-distance is influenced by part structure to which the concavity of contours is closely related [58, 37]. In the following subsections, we discuss this in detail.

2.3.2 Articulation Insensitivity of the Inner-Distance

As shown in Fig. 2.1, the inner-distance is insensitive to articulation. Intuitively, this is true because an articulated shape can be decomposed into rigid parts connected by junctions. Accordingly, the shortest path between landmark points can be divided into segments within each parts. We will first give a very general model for part articulation and then formally prove articulation insensitivity of the inner-distance.

A Model of Articulated Objects

Before discussing the articulation insensitivity of the inner-distance, we need to provide a model of articulated objects. Note that our method does not involve any part models, the model here is only for the analysis of the properties of the inner-distance. Intuitively, when a shape O is said to have articulated parts, it means

- O can be decomposed into several *parts*, say, $O_1, O_2, ..., O_n$, where n is the number of parts. These parts are connected by *junctions*.

- The junctions between parts are very small compared to the parts they connect.

- The articulation of O as a transformation is rigid when limited to any part O_i, but can be non-rigid on the junctions.

- The new shape O' achieved from articulation of O is again an articulated object and can articulate *back* to O.

Based on these intuition, we define an articulated object $O \subset \mathbb{R}^2$ of n parts together with an articulation f as:

$$O = \{\bigcup_{i=1}^{n} O_i\} \bigcup \{\bigcup_{i \neq j} J_{ij}\}$$

where

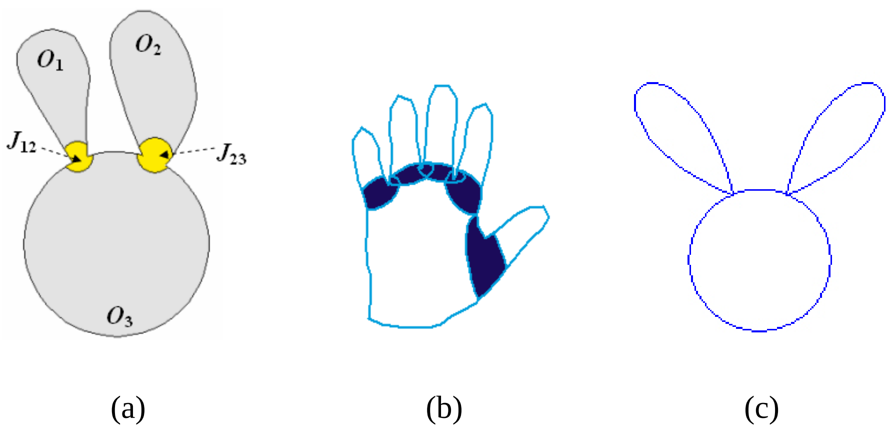
- $\forall i, 1 \leq i \leq n$, part $O_i \subset \mathbb{R}^2$ is connected and closed, and $O_i \bigcap O_j = \emptyset, \forall i \neq j, 1 \leq i, j \leq n$.

- $\forall i \neq j, 1 \leq i, j \leq n$, $J_{ij} \subset \mathbb{R}^2$, connected and closed, is the junction between O_i and O_j. If there is no junction between O_i and O_j, then $J_{ij} = \emptyset$. Otherwise, $J_{ij} \bigcap O_i \neq \emptyset$, $J_{ij} \bigcap O_j \neq \emptyset$.

- $diam(J_{ij}) \leq \epsilon$, where $diam(P) \doteq max_{x,y \in P}\{d(x, y; P)\}$ is the *diameter* of a point set $P \subset \mathbb{R}^2$ in the sense of the inner-distance. $\epsilon \geq 0$ is constant and very small compared to the size of the articulated parts. A special case is $\epsilon = 0$, which means that all junctions degenerate to single points and O is called an *ideal articulated object*.

Fig. 2.5 (a) shows an example articulated shape with three parts and two junctions.

The articulation from an articulated object O to another articulated object O' is a one-to-one continuous mapping f, such that:



(a) (b) (c)

Figure 2.5: Examples of articulated objects. (a) An articulated shape with three parts, $O_1 \bigcup O_2 \bigcup O_3 \bigcup J_{12} \bigcup J_{23}$. (b) Overlapping junctions (the five dark areas). (c) Ideal articulation.

- O' has the decomposition $O' = \{\bigcup_{i=1}^{n} O'_i\} \bigcup \{\bigcup_{i \neq j} J'_{ij}\}$. Furthermore, $O'_i = f(O_i)$, $\forall i, 1 \leq i \leq n$ are parts of O' and $J'_{ij} = f(J_{ij})$, $\forall i \neq j, 1 \leq i, j \leq n$ are junctions in O'. This preserves the topology between the articulated parts. In particular, the deformed junctions still have a diameter less than or equal to ϵ.

- f is rigid (rotation and translation only) when restricted to O_i, $\forall i, 1 \leq i \leq n$. This means inner-distances within each part will not change.

Notes: 1) In the above and following, we use the notation $f(P) \doteq \{f(x) : x \in P\}$ for short. 2) It is obvious from the above definitions that f^{-1} is an articulation that maps O' to O.

The above model of articulation is very general and flexible. For example, there is no restriction on the shape of the junctions. Junctions are even allowed to overlap each other. Furthermore, the articulation f on the junctions are not required to be smooth. Fig. 2.5 (b) and (c) gives two more examples of articulated shapes.

Articulation Insensitivity

We are interested in how the inner-distance varies under articulation. From previous paragraphs we know that changes of the inner-distance are due to junction deformations. Intuitively, this means the change is very small compared to the size of parts. Since most pairs of points have inner-distances comparable to the sizes of parts, the relative change of the inner-distances during articulation are small. This roughly explains why the inner-distances are articulation insensitive.

We will use following notations: 1) $\Gamma(x_1, x_2; P)$ denotes a shortest path from $x_1{\in}P$ to $x_2{\in}P$ for a closed and connected point set $P{\subset}\mathbb{R}^2$ (so $d(x_1, x_2; P)$ is the length of $\Gamma(x_1, x_2; P)$). 2) $'$ indicates the image of a point or a point set under f, e.g., $P'{\doteq}f(P)$ for point set P, $p'{\doteq}f(p)$ for a point p. 3) “[” and “]” denote the concatenation of paths.

Let us first point out two facts about the inner-distance within a part or crossing a junction. Both facts are direct results from the definitions in sec. 7.3.1.

$$d(x, y; O_i) = d(x', y'; O_i') \qquad \forall x, y{\in}O_i, 1{\le}i{\le}n \tag{2.3}$$

$$|d(x, y; O) - d(x', y'; O')| \le \epsilon, \forall x, y \in J_{ij} \qquad \forall i{\ne}j, 1{\le}i, j{\le}n, J_{ij} \ne \emptyset \tag{2.4}$$

Note that (2.4) does not require the shortest path between x, y to lie within the junction J_{ij}. These two facts describe the change of the inner-distances of restricted point pairs. For the general case of $x, y{\in}O$, we have the following theorem:

Theorem: Let O be an articulated object and f be an articulation of O as defined above. $\forall x, y{\in}O$, suppose the shortest path $\Gamma(x, y; O)$ goes through m different junctions in O and $\Gamma(x', y'; O')$ goes through m' different junctions in O', then

$$|d(x, y; O) - d(x', y'; O')| \le max\{m, m'\}\epsilon \tag{2.5}$$

Proof: The proof uses the intuition mentioned above. First we decompose $\Gamma(x, y; O)$ into segments. Each segment is either within a part or across a junction. Then, applying (2.3) and (2.4) to each segment leads to the theorem.

First, $\Gamma(x, y; O)$ is decomposed into l segments:

$$\Gamma(x, y; O) = [\Gamma(p_0, p_1; R_1), \Gamma(p_1, p_2; R_2), ..., \Gamma(p_{l-1}, p_l; R_l)]$$

using point sequence $p_0, p_1, ..., p_l$ and regions $R_1, ..., R_l$ via the steps using Algorithm 1.

An example of this decomposition is shown in Fig. 2.6 (a). With this decomposition, $d(x, y; O)$ can be written as:

$$d(x, y; O) = \sum\nolimits_{1 \leq i \leq l} d(p_{i-1}, p_i; R_i)$$

Suppose m_1 of the segments cross junctions (i.e., segments not contained in any single part), then obviously $m_1 {\leq} m$.

In O', we construct a path from x' to y' corresponding to $\Gamma(x, y; O)$ as follows (e.g. Fig. 2.6 (b)):

$$\widetilde{C}(x', y'; O') = [\Gamma(p'_0, p'_1; R'_1), \Gamma(p'_1, p'_2; R'_2), ..., \Gamma(p'_{l-1}, p'_l; R'_l)]$$

Note that $\widetilde{C}(x', y'; O')$ is not necessarily the shortest path in O'. Denote $\widetilde{d}(x', y'; O')$ as the length of $\widetilde{C}(x', y'; O')$, it has the following property due to (2.3), (2.4):

$$|d(x, y; O) - \widetilde{d}(x', y'; O')| \leq m_1\epsilon \leq m\epsilon \tag{2.6}$$

On the other hand, since O can be articulated from O' through f^{-1}, we can construct $\widetilde{C}(x, y; O)$ from $\Gamma(x', y'; O')$ in the same way we constructed $\widetilde{C}(x', y'; O')$ from $\Gamma(x, y; O)$. Then, similar to (2.6), there is

$$|d(x', y'; O') - \widetilde{d}(x, y; O)| \leq m'\epsilon \tag{2.7}$$

Algorithm 1 Decompose $\Gamma(x, y; O)$

```
$p_0 \leftarrow x$, $i \leftarrow 0$
**while** $p_i \neq y$ **do** $\{$/*find $p_{i+1}$*/$\}$
  $i \leftarrow i + 1$
  $R_i \leftarrow$ the region (a part or a junction) $\Gamma(x, y; O)$ enters after $p_{i-1}$
  **if** $R_i = O_k$ for some $k$ ($R_i$ is a part) **then** $\{$/*enter a part*/$\}$
    Set $p_i$ as a point in $O_k$ such that
    1) $\Gamma(p_{i-1}, p_i; O_k) \subseteq \Gamma(x, y; O)$
    2) $\Gamma(x, y; O)$ enters a new region (a part or a junction) after $p_i$ or $p_i = y$
  **else** $\{$/*$R_i = J_{rs}$ for some $r, s$ ($R_i$ is a junction), enter a junction*/$\}$
    Set $p_i$ as the point in $J_{rs} \bigcap \Gamma(x, y; O)$ such that $\Gamma(x, y; O)$ never re-enters $J_{rs}$ after $p_i$.
    $R_i \leftarrow$ the union of all the parts and junctions $\Gamma(p_{i-1}, p_i; O)$ passes through (note $J_{rs} \subseteq R_i$).
  **end if**
**end while**
$l \leftarrow i$
```

Combining (2.6) and (2.7),

$$d(x,y;O) - m'\epsilon \leq \widetilde{d}(x,y;O) - m'\epsilon \leq d(x',y';O') \leq \widetilde{d}(x',y';O') \leq d(x,y;O) + m\epsilon$$

This implies (2.5). ■

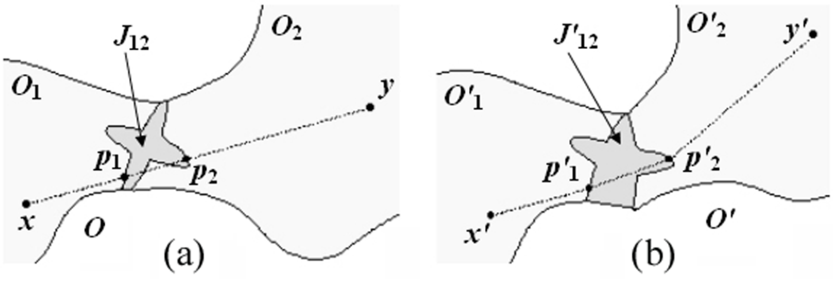


Figure 2.6: (a) Decomposition of $\Gamma(x,y;O)$ (the dashed line) with $x = p_0, p_1, p_2, p_3 = y$. Note that a segment can go through a junction more than once (e.g. p_1p_2). (b) Construction of $\widetilde{C}(x',y';O')$ in O' (the dashed line). Note that $\widetilde{C}(x',y';O')$ is not the shortest path.

From (2.5) we can make two remarks concerning changes of inner-distances under articulation:

1. The inner-distance is strictly invariant for ideal articulated objects. This is obvious since $\epsilon = 0$ for ideal articulations.

2. Since ϵ is very small, for most pairs of x, y, the relative change of inner-distance is very small. This means the inner-distance is insensitive to articulations.

We further clarify several issues. First, the proof depends on the size limitation of junctions. The intuition is that a junction should have a relatively smaller size compared to parts, otherwise it is more like a part itself. A more precise part-junction definition may provide a tighter upper bound but sacrifice some generality. The definition also captures our intuition about what distinguishes articulation from deformation. Second, the part-junction model is not actually used at all when applying the inner-distance. In fact, one advantage of using the inner-distance is that it *implicitly* captures part structure, whose definition is still not clear in general.

2.3.3 Inner-Distances and Part Structures

In addition to articulation insensitivity, we believe that the inner-distance captures part structures better than the Euclidean distance. This is hard to prove because the definition of part structure remains unclear. For example, Basri et al. [12] gave a shape of shoe (Fig. 2.7) which has no clear part decomposition, although it feels like it has more than one part.

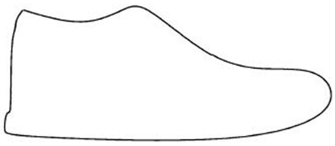


Figure 2.7: A shape from [12], which has no clear part decomposition.

Instead of giving a rigorous proof, we show how the inner-distance captures part structure with examples and experiments. Figures 2.1, 2.8 and 2.12 show examples where the inner-distance distinguishes shapes with parts while the Euclidean distance runs into trouble because the sample points on the shape have the same spatial distributions. For example, the original shape context [15] may fail on these shapes. One may argue that the Euclidean distance will also work on these examples with an increased number of landmark points. This argument has several practical problems. First, the computational cost will be increased, usually in a quadratic order or higher. Second, no matter how many points are used, there can always be finer structures. Third, as shown in Fig. 2.9, for some shapes this strategy will not work.

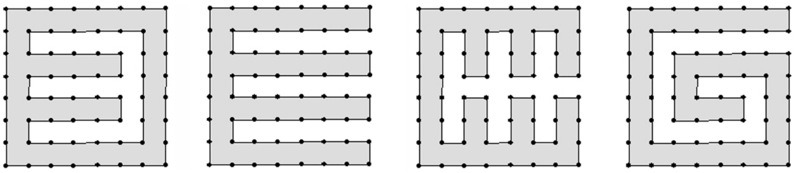


Figure 2.8: With the same sample points, the distributions of Euclidean distances between all pair of points are indistinguishable for the four shapes, while the distributions of the inner-distances are quite different.

During retrieval experiments using several shape databases, the inner-distance based descriptors all achieve excellent performance. Through observation we have found that some databases (e.g., MPEG7) are difficult for retrieval mainly due to the complex part structures in their shapes, though they have little articulation. These experiments show that the inner-distance is effective at capturing part structures (see Sec. 2.7.2 and Figures 2.12 and 2.18 for details).

Aside from part structures, examples in Fig. 2.9 show cases where the inner-distance can better capture some shapes without parts. We expect further studies on the relationship between inner-distances and shape in the future.

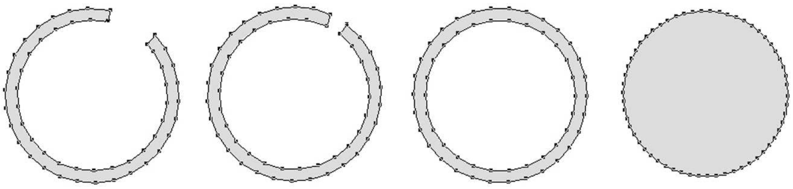


Figure 2.9: With about the same number of sample points, the four shapes are virtually indistinguishable using distribution of Euclidean distances, as in Fig. 2.8. However, their distributions of the inner-distances are quite different except for the first two shapes. Note: 1) None of the shapes has (explicit) parts. 2) More sample points will not affect the above statement.

2.4 Articulation Invariant Signatures

To build shape descriptors with the inner-distance is straightforward. Theoretically it can be used to replace other distance measures (e.g. the Euclidean distance) in any existing shape descriptors. In this section, the inner-distance is used to build articulation invariant signatures for 2D shapes using multidimensional scaling (MDS) similar to [36]. In the next section, we will show how to use the inner-distance to extend the shape context for shape matching.

Given sample points $P \doteq \{p_i\}_{i=1}^n$ on a shape O and the inner-distances $\{d_{ij}\}_{i,j=1}^n$ between them, MDS finds the transformed points $Q \doteq \{q_i\}_{i=1}^n$ such that the Euclidean distances $\{e_{ij}(Q) = \|q_i - q_j\|\}_{i,j=1}^n$ minimize the *stress* $S(Q)$ defined as:

$$S(Q) = \frac{\sum_{i<j} w_{ij}(d_{ij} - e_{ij}(Q))^2}{\sum_{i<j} d_{ij}^2} \tag{2.8}$$

where w_{ij} are weights. In our experiment, we use the least squares MDS with $w_{ij} = 1$. The stress can be minimized using the SAMCOF (Scaling by Maximizing a Convex

Function) algorithm [21]. SAMCOF is an iterative algorithm that keeps decreasing the objective function, i.e., the stress (2.8). The details can be find in Elad and Kimmel's paper [36].

Fig. 2.10 shows two examples of the articulation invariant signatures computed by the above approach. It can be seen that although the global shape of the two original objects are quite different due to the articulation, their signatures are very similar to each other. More examples of the articulation invariant signatures can be seen in Fig. 6.8.

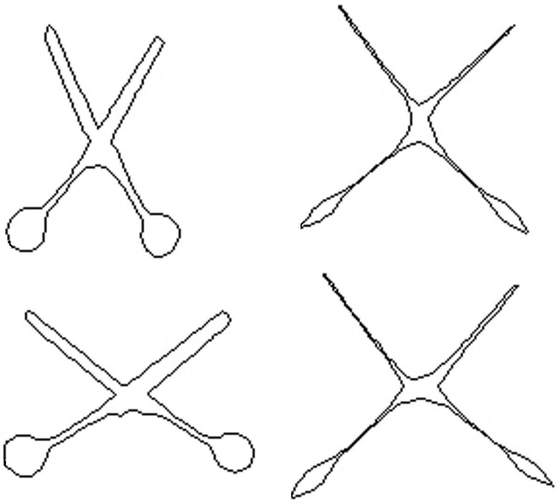


Figure 2.10: Articulation invariant signatures. Left: two shapes related by articulation. Right: their signatures.

It is attractive to use the articulation invariant signature for classifying articulated shapes. In our experiments we combine it with the shape context. The method contains three steps: 1) use the inner-distance and MDS to get the articulation invariant signatures; 2) build the shape context on the signatures; 3) use dynamic programming for shape context matching. The third step is described in detail in the next section. We call this approach MDS+SC+DP. The experimental results show significant improvement compared to the shape context on the original shapes.

2.5 Inner-Distance Shape Context: Matching and Retrieval

2.5.1 Inner-Distance Shape Context (IDSC)

To extend the shape context defined in (2.1), we redefine the bins with the inner-distance. The Euclidean distance is directly replaced by the inner-distance. The relative orientation between two points can be defined as the tangential direction at the starting point of the shortest path connecting them. However, this tangential direction *is* sensitive to articulation. Fortunately, for a boundary point p and its shortest path $\Gamma(p, q; O)$ to another point q, the angle between the contour tangent at p and the direction of $\Gamma(p, q; O)$ at p is insensitive to articulation (invariant to ideal articulation). We call this angle the ***inner-angle*** (e.g., see Fig. 2.11) and denote it as $\theta(p, q; O)$. The inner-angle is used for the orientation bins. This is similar to using the local coordinate system suggested in [15] to get rotation invariance. In practice, the shape boundary may be distorted by noise that reduces the stability of the inner-angle. To deal with this problem, we smooth the contour using a small neighborhood before computing the inner-angle.

Fig. 2.12 shows examples of the shape context computed by the two different methods. It is clear that SC is similar for all three shapes, while IDSC is only similar for the beetles. From this figure we can see that the inner-distance is better at capturing parts than SC.

The inner-angle is just a byproduct of the shortest path algorithms and does not affect the complexity. Once the inner-distances and orientations between all pair of points

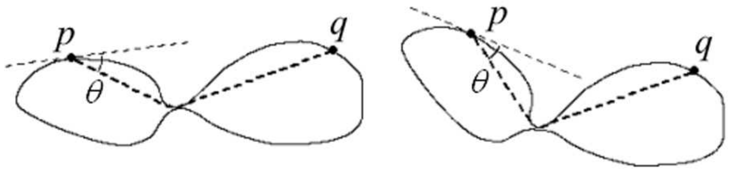


Figure 2.11: The inner-angle $\theta(p, q; O)$ between two boundary points.

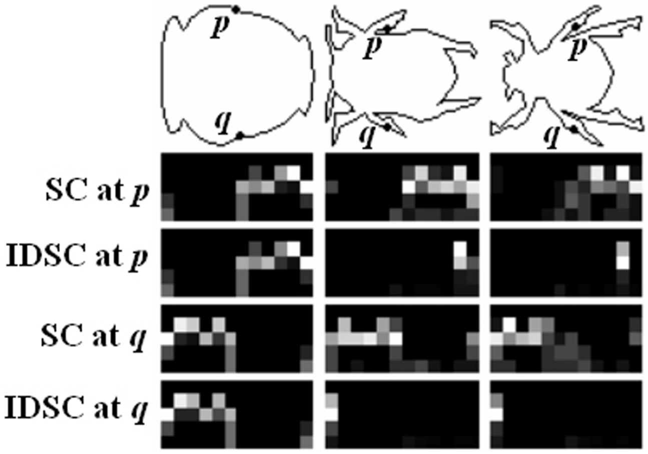


Figure 2.12: Shape context (SC) and inner-distance shape context (IDSC). The top row shows three objects from the MPEG7 shape database (Sec. 2.7.2), with two marked points p, q on each shape. The next rows show (from top to bottom), the SC at p, the IDSC at p, the SC at q, the IDSC at q. Both the SC and the IDSC use local relative frames (i.e. aligned to the tangent). In the histograms, the x axis denotes the orientation bins and the y axis denotes log distance bins.

are ready, it takes $O(n^2)$ time to compute the histogram (2.1).

2.5.2 Shape Matching Through Dynamic Programming

The contour matching problem is formulated as follows: Given two shapes A and B, describe them by point sequences on their contour, say, $p_1 p_2 ... p_n$ for A with n points, and

$q_1q_2...q_m$ for B with m points. Without loss of generality, assume $n \geq m$. The matching π from A to B is a mapping from $1, 2, ..., n$ to $0, 1, 2, ..., m$, where p_i is matched to $q_{\pi(i)}$ if $\pi(i) \neq 0$ and otherwise left unmatched. π should minimize the match cost $H(\pi)$ defined as

$$C(\pi) = \sum\nolimits_{1\leq i\leq n} c(i, \pi(i)) \tag{2.9}$$

where $c(i, 0) = \tau$ is the penalty for leaving p_i unmatched, and for $1 \leq j \leq m$, $c(i, j)$ is the cost of matching p_i to q_j. This is measured using the χ^2 statistic as in [15]

$$c(i,j) \equiv \frac{1}{2} \sum\nolimits_{1\leq k\leq K} \frac{[h_{A,i}(k) - h_{B,j}(k)]^2}{h_{A,i}(k) + h_{B,j}(k)} \tag{2.10}$$

Here $h_{A,i}$ and $h_{B,j}$ are the shape context histograms of p_i and q_j respectively, and K is the number of histogram bins.

Since the contours provide orderings for the point sequences $p_1p_2...p_n$ and $q_1q_2...q_m$, it is natural to restrict the matching π with this order. To this end, we use dynamic programming (DP) to solve the matching problem. DP is widely used for contour matching. Detailed examples can be found in [145, 12, 118]. We use the standard DP method [31] with the cost functions defined as (2.9) and (2.10).

By default, the above method assumes the two contours are already aligned at their start and end points. Without this assumption, one simple solution is to try different alignments at all points on the first contour and choose the best one. The problem with this solution is that it raises the matching complexity from $O(n^2)$ to $O(n^3)$. Fortunately, for the comparison problem, it is often sufficient to try aligning a fixed number of points, say, k points. Usually k is much smaller than m and n, this is because shapes can be first rotated according to their moments. According to our experience, for $n, m = 100$,

$k = 4$ or 8 is good enough and larger k does not demonstrate significant improvement. Therefore, the complexity is still $O(kn^2) = O(n^2)$.

Bipartite graph matching is used in [15] to find the point correspondence π. Bipartite matching is more general since it minimizes the matching cost (2.9) without additional constraints. For example, it works when there is no ordering constraint on the sample points (while DP is not applicable). For sequenced points along silhouettes, however, DP is more efficient and accurate since it uses the ordering information provided by shape contours.

2.5.3 Shape Distances

Once the matching is found, we use the matching cost $C(\pi)$ as in (2.9) to measure the similarity between shapes. One thing to mention is that dynamic programming is also suitable for shape context. In the following, we use IDSC+DP to denote the method of using dynamic programming matching with the IDSC, and use SC+DP for the similar method with the SC.

In addition to the excellent performance demonstrated in the experiments, the IDSC+DP framework is simpler than the SC+TPS framework (2.2) [15]. First, besides the size of shape context bins, IDSC+DP has only two parameters to tune: 1) The penalty τ for a point with no matching, usually set to 0.3, and 2) The number of start points k for different alignments during the DP matching, usually set to 4 or 8. Second, IDSC+DP is easy to implement, since it does not require the appearance and transformation model as well as the iteration and outlier control. Furthermore, the DP matching is faster than bipartite matching, which is important for retrieval in large shape databases.

The time complexity of the IDSC+DP consists of three parts. First, the computation of inner-distances can be achieved in $O(n^3)$ with Johnson or Floyd-Warshall's shortest path algorithms, where n is the number of sample points. Second, the construction of the IDSC histogram takes $O(n^2)$. Third, the DP matching costs $O(n^2)$, and only this part is required for all pairs of shapes, which is very important for retrieval tasks with large image databases. In our experiment using partly optimized Matlab code on a regular Pentium IV 2.8G PC, a single comparison of two shapes with $n = 100$ takes about 0.31 second.

2.6 Shortest Path Texture Context

In real applications, the shape information is often not enough for object recognition tasks. On the one hand, shapes from different classes sometimes are more similar than those from the same class (e.g., Fig. 2.13). On the other hand, shapes are often damaged due to occlusion and self-overlapping (some examples can be found in Fig. 2.24). Naturally, the combination of texture and shape information is desirable for this problem. In [15] the appearance information is included into the SC+TPS framework by considering appearance around landmark points. In this section, we will introduce a new descriptor that considers the texture information inside the whole shape.

In previous sections, the inner-distance is shown to be articulation insensitive due to the fact that the shortest paths within shape boundaries are robust to articulation. Therefore, the texture information along these paths provides a natural articulation insensitive texture description. Note that this is true only when the paths are robust. In this sec-

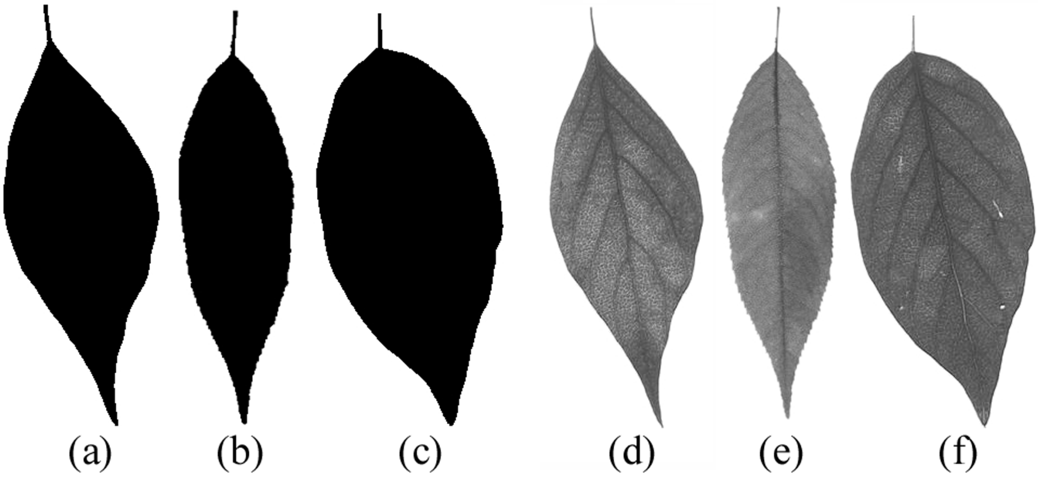


Figure 2.13: Shapes of three leaves ((a), (b) and (c)) are not enough to distinguish them. Their texture ((d), (e) and (f) respectively) apparently helps.

tion, we use local intensity gradient orientations to capture texture information because of their robustness and efficiency. To gain articulation invariance, the angles between intensity gradient directions and shortest path directions are used. In the following we call these angles *relative orientations*. Given shape O and two points p, v on it, we use $\alpha(p, v; O)$ to denote the relative orientation with respect to the shortest path $\Gamma(p, v; O)$. An example is shown in Fig. 2.14.

Based on the above idea, we propose the *shortest path texture context* (SPTC) as a combined shape and texture descriptor. SPTC is an extension of the IDSC in that it measures the distributions of (weighted) relative orientations along shortest paths instead of the joint distributions of inner-distance and inner-angle distributions of landmark points. In our application, the relative orientations are weighted by gradient magnitudes when building into SPTC. For texture undergoing large non-uniform illumination change, it might be better to use non-weighted relative orientations.

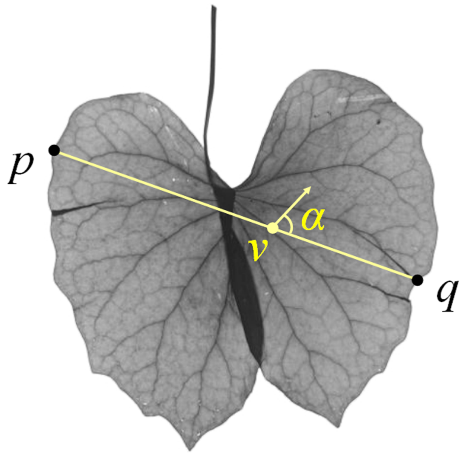


Figure 2.14: Relative orientation $\alpha(p, v; O)$ at point v. The arrow points to local intensity gradient direction.

Given n landmark points $x_1, x_2, ..., x_n$ sampled from the boundary of shape O, the SPTC for each x_i is a three-dimensional histogram h_i (we abuse notation to use h_i again for the histograms). Similarly to IDSC, SPTC uses the inner-distance and the inner-angle as the first two dimensions. The third dimension of SPTC is the (weighted) relative orientation that takes into account the texture information along shortest paths. To build h_i, for each x_j, $j \neq i$, a normalized histogram of relative orientation along the shortest path $\Gamma(x_i, x_j; O)$ is added into the relative orientation bin located at the inner-distance and inner-angle bin determined by x_j. The algorithm is described in Algorithm 2. Note that when the number of relative orientation bins $n_r = 1$, SPTC reduces to IDSC.

A similar idea of using "relative orientation" is used by Lazebnik et al. [80] for rotation invariant texture description. Shape context had also been extended for texture description by including intensity gradient orientation (e.g. [104]). SPTC is different from these methods in three ways. First, SPTC combines texture information and global

Algorithm 2 Shortest path texture context h_i at landmark point x_i

```
$h_i \leftarrow$ 3-D matrix with zero entries everywhere
**for** $j = 1$ to $n$, $j \neq i$ **do**
    $\Gamma(x_i, x_j; O) \leftarrow$ shortest path from $x_i$ to $x_j$
    $\hat{h} \leftarrow$ 1-D weighted histogram of the relative orientations along $\Gamma(x_i, x_j; O)$
    $\hat{h} \leftarrow \hat{h} / \parallel \hat{h} \parallel_1$    $\{$/* Normalize $\hat{h}$, where $\parallel . \parallel_1$ is the $L_1$ norm */$\}$
    $d_{id} \leftarrow$ the inner-distance bin index computed from $d(x_i, x_j; O)$
    $\theta_{id} \leftarrow$ the inner-angle bin index computed from $\theta(x_i, x_j; O)$
    **for** $\alpha_{id} = 1$ to $n_r$ **do** { /* $n_r$ is the number of relative orientation bins */$\}$
        $h_i(d_{id}, \theta_{id}, \alpha_{id}) \leftarrow h_i(d_{id}, \theta_{id}, \alpha_{id}) + \hat{h}(\alpha_{id})$
    **end for**
**end for**
$h_i \leftarrow h_i / |h_i|$    $\{$/* Normalize $h_i$ */$\}$
```

shape information while the above methods work for local image patches. Second, the above methods sample the orientations at a large number of pixels inside a patch, which is too expensive for our task without utilizing shortest paths. Third, none of the previous methods is articulation invariant. Another related work by Zhao and Davis [163] used the color information along the shortest path for background subtraction. Instead of color information, we use gradient orientation, which is more robust to lighting change [25], which is very important for classification tasks. In the next section, SPTC is tested in two leaf image databases and excellent performance is observed.

2.7 Experiments

This section describes the experiments testing proposed approaches. First, we test the inner-distance's articulation insensitivity with an articulated shape dataset. After that, the inner-distance is tested in comparison with other state-of-the-art approaches on several widely tested shape data sets, including the MPEG7 CE-Shape-1 shapes, Kimia's silhouette [136, 134], ETH-80 [82]. Then, the proposed approach is tested on two foliage image datasets, a Swedish leaf dataset [141] and a Smithsonian leaf dataset. These experiments show how the inner-distance works in real applications and how the SPTC performs on shapes with texture. Finally, we will show the potential use of the IDSC on human motion analysis.

Now we describe the parameters used in the experiments. We use n to denote the number of landmark points (on the outer contour of shapes). Landmark points are sampled uniformly (same as in [15]) to avoid bias. n is chosen according to tasks. In general, larger n will produce greater accuracy with less efficiency. For the size of histograms, n_d, n_θ, and n_r are used for the number of inner-distance bins, the number of inner-angle bins, and the number of relative orientation bins respectively. A typical setting for the bin number is $n_d = 5, n_\theta = 12$ and $n_r = 8$. In our experiments, we sometimes use $n_d = 8$ to get a better results. For dynamic programming, k denotes the number of different starting points for alignment (uniformly chosen from landmark points). The choice of k was discussed in Sec. 2.5.3. In general, a larger k increases the accuracy. However in practice we found that $k = 4 - 8$ usually gives satisfiable results. For example, $k = 8$ is used for the MPEG7 dataset. However, we did notice that larger k can improve the performance

Table 2.1: Retrieval result on the articulate dataset.

Distance Type	Top 1	Top 2	Top 3	Top 4
L_2 (baseline)	25/40	15/40	12/40	10/40
SC+DP	20/40	10/40	11/40	5/40
MDS+SC+DP	**36/40**	**26/40**	**17/40**	**15/40**
IDSC+DP	**40/40**	**34/40**	**35/40**	**27/40**

further, e.g., $k = 16$ is used for the ETH-80 dataset that involves wildly varied rotations. We did not rotate shapes according to their moments, which might be helpful for tasks involving a large variation in orientations. The penalty τ for one occlusion is always set to be 0.3 (our experiments show that different τ in the range of $[0.25, 0.5]$ do not affect the results too much). In all the experiments, the parameters for MDS+SC+DP are the same as in IDSC+DP. Furthermore, for datasets that have no previously reported shape context matching results, we run the SC+DP for comparison with the same parameters as IDSC+DP.

2.7.1 Articulated Database

To show the articulation insensitivity of the inner-distance, we apply the proposed articulation invariant signature and the IDSC+DP approach to an articulated shape data set we collected. The dataset contains 40 images from 8 different objects. Each object has 5 images articulated to different degrees (see Fig. 6.8). The dataset is very challenging because of the similarity between different objects (especially the scissors). The holes of the scissors make the problem even more difficult.

The parameters in the experiment are: $n = 200$, $n_d = 5$, $n_\theta = 12$. Since all the objects are at the same orientation, we align the contours by forcing them to start from the bottom-left points and then set $k = 1$ for DP matching. The articulation invariant

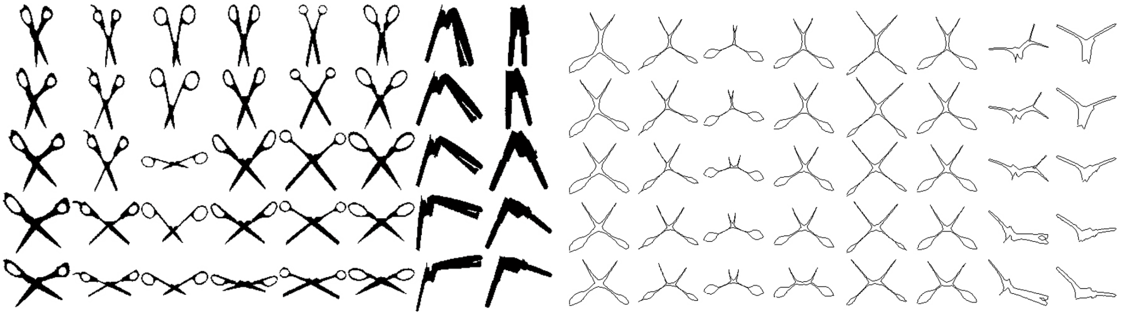


Figure 2.15: Left: Articulated shape database. This dataset contains 40 images from 8 objects with articulation. Each column contains five images from the same object. Right: MDS of the articulated shape database using the inner-distances.

signatures of the shapes are computed and shown in Fig. 6.8.

To evaluate the recognition result, for each image, the four most similar matches are chosen from other images in the dataset. The retrieval result is summarized as the number of 1st, 2nd, 3rd and 4th most similar matches that come from the correct object. Table 2.1 shows the retrieval results. It demonstrates that both the articulation invariant signature and the IDSC help to improve recognition a lot. This verifies our claim that the inner-distance is very effective for objects with articulated parts. Fig. 2.16 shows some detailed retrieval results for some of the images. The experiment also shows that IDSC works better than MDS for the articulated shapes. One reason is that the MDS may cause loss of information since it uses the Euclidean distance to *approximate* the inner-distance. To give an intuition of the difficulty of the database, a baseline algorithm using L_2 distance was also tested.

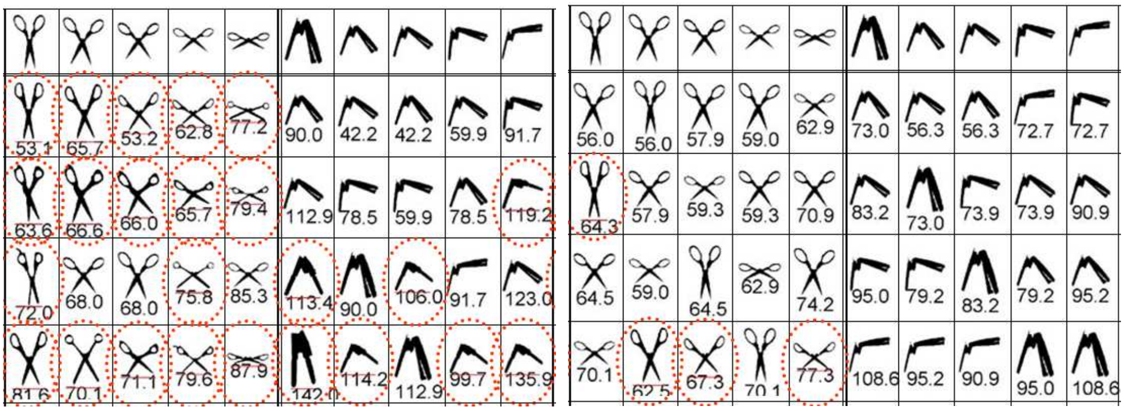


Figure 2.16: Left: SC+DP on the articulated shape database. The top 4 retrieval results of 20 images are shown here. The top row shows the querying images. Row two to row five show the top one to top four retrieval results respectively. The numbers below the results are the matching scores. Incorrect hits are circled in dotted lines. Right: IDSC+DP on the articulated shape database, same notations as for SC+DP.

2.7.2 MPEG7 Shape Database

The widely tested MPEG7 CE-Shape-1 [79] database consists of 1400 silhouette images from 70 classes. Each class has 20 different shapes (see Fig. 2.17 for some typical images). The recognition rate is measured by the so-called Bullseye test: For every image in the database, it is matched with all other images and the top 40 most similar candidates are counted. At most 20 of the 40 candidates are correct hits. The score of the test is the ratio of the number of correct hits of all images to the highest possible number of hits (which is 20x1400).

The parameters in our experiment are: $n = 100$ (300 were used in [15]), $n_d = 8$, $n_\theta = 12$ and $k = 8$. To handle mirrored shapes, we compare two point sequences (corresponding to shapes) with the original order and reversed order. Table 2.2 lists reported

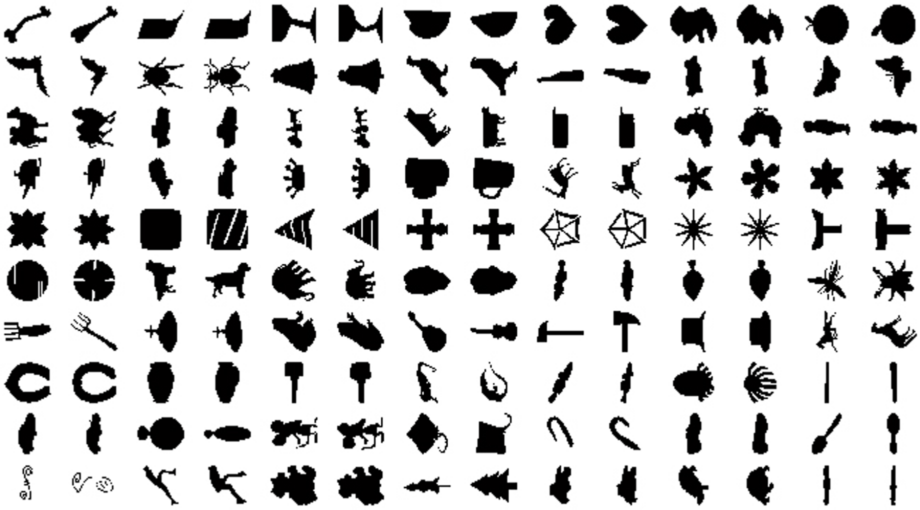


Figure 2.17: Typical shape images from the MPEG7 CE-Shape-1, two images from each class.

results from different algorithms. It shows that our algorithms outperform all the alternatives. The speed of our algorithm is in the same range as those of shape contexts [15], curve edit distance [133] and generative model [147]. Again, we observed that IDSC performs a little better than the articulation invariant signatures.

Note that unlike the original SC+TPS framework used in [15], the appearance and bending information are not included in our experiment. The reason is twofold: 1) we want to focus more on the inner-distance itself; 2) this also makes our framework easy to use. In addition, the dynamic programming scheme is used to take advantage of the ordering information of the landmark points and the local coordinate framework (along the tangential of landmark points) are used to achieve rotation invariance.

To help understand this performance, we did two other experiments in the same setting where the only difference is the descriptors used: one uses SC, another IDSC. The parameters in both experiments are: 64 sample points on each silhouette, 8 distance

Table 2.2: Retrieval rate (bullseye) of different methods for the MPEG7 CE-Shape-1.

Alg.	CSS [109]	Visual Parts [79]	SC+TPS [15]	Curve Edit [133]	Dis. Set [50]
Score	75.44%	76.45%	76.51%	78.17%	78.38%
Alg.	MCSS [64]	Gen. Model [147]	**MDS+SC+DP**	**IDSC+DP**	
Score	78.8%	80.03%	**84.35%**	**85.40%***	

* A higher score of 86.56 is achieved using EMD-L_1 instead of χ^2 distance [92].

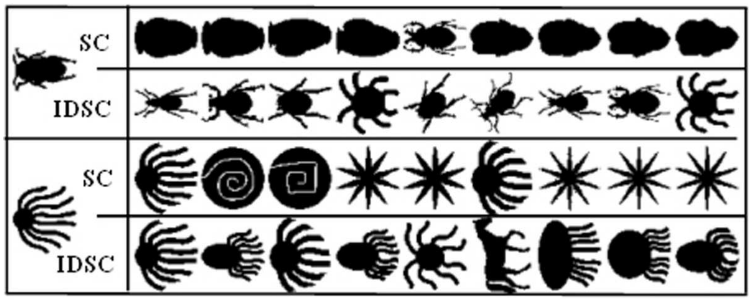


Figure 2.18: Two retrieval examples for comparing SC and IDSC on the MPEG7 data set. The left column show two shapes to be retrieved: a beetle and an octopus. The four right rows show the top 1 to 9 matches, from top to bottom: SC and IDSC for the beetle, SC and IDSC for the octopus.

bins and 8 orientation bins. To avoid the matching effect, shapes are compared using the simple shape context distance measure D_{sc} instead of DP (see Sec. 2.2.3 or [15]). The Bullseye score with SC is 64.59%, while IDSC gets a higher score of 68.83%. Fig. 2.18 shows some retrieval results, where we see that the IDSC is good for objects with parts while the SC favors global similarities. Examination of the MPEG7 data set shows that the complexity of shapes are mainly due to the part structures but not articulations, so the good performance of IDSC shows that the inner-distance is more effective at capturing part structures.

2.7.3 Kimia's database

IDSC+DP and MDS+SC+DP are tested on two shape databases provided by Kimia's group [136, 134]. The first database [136] contains 25 images from 6 categories (Fig. 2.19 (a)). It has been tested by [15, 136, 45]. We use parameters $n = 100$, $n_d = 5$, $n_\theta = 12$ and $k = 4$. The retrieval result is summarized as the number of 1st, 2nd and 3rd closest matches that fall into the correct category. The results are listed in Table 2.3. It shows that IDSC slightly outperforms the other three reported methods and the MDS-based approach.

The second database [134] contains 99 images from 9 categories (Fig. 2.19 (b)) and has been tested by [134, 147]. We use parameters $n = 300$, $n_d = 8$, $n_\theta = 12$ and $k = 4$. Similar to results described above, the retrieval result is summarized as the number of top 1 to top 10 closest matches (the best possible result for each of them is 99). Table 2.4 lists the numbers of correct matches of several methods, which shows that our approaches performs comparably to the best approaches. One interesting observation is that the IDSC performs very similarly to the shock edit. This suggests a close relation between them as mentioned in the related work section.

Table 2.3: Retrieval result on Kimia dataset 1 [136] (Fig. 2.19 (a)).

Method	Top 1	Top 2	Top 3
Sharvit et. al [136]	23/25	21/25	20/25
Gdalyahu and Weinshall [45]	25/25	21/25	19/25
Belongie et. al [15]	25/25	24/25	22/25
MDS+SC+DP	**23/25**	**20/25**	**19/25**
IDSC+DP	**25/25**	**24/25**	**25/25**

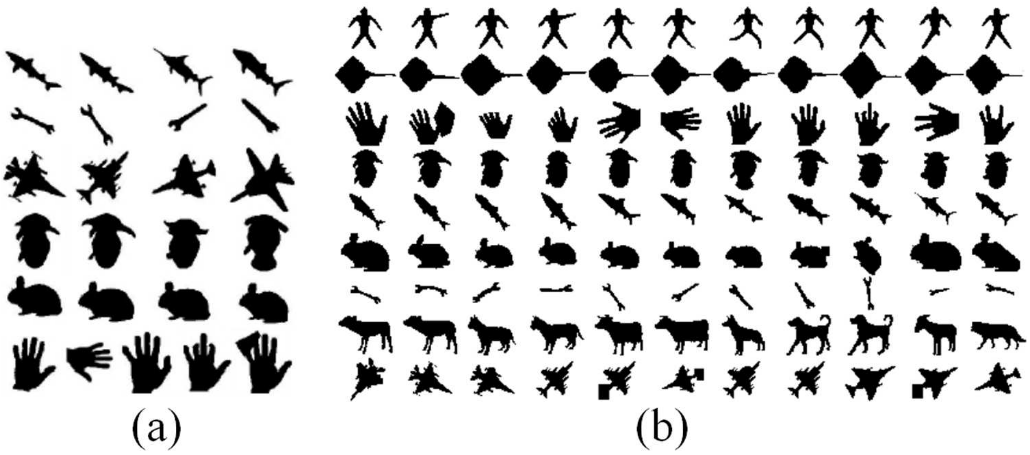


Figure 2.19: Kimia shape datasets. (a) Kimia dataset 1 [136], 25 instances from 6 categories. (b) Kimia set 2 [134], 99 instances from 9 categories.

Table 2.4: Retrieval result on Kimia dataset 2 [134] (Fig. 2.19 (b)).

Algorithm	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
SC [134]	97	91	88	85	84	77	75	66	56	37
Gen. Model [147]	99	97	99	98	96	96	94	83	75	48
Shock Edit [134]	99	99	99	98	98	97	96	95	93	82
MDS+SC+DP	**99**	**98**	**98**	**98**	**97**	**99**	**97**	**96**	**97**	**85**
IDSC+DP	**99**	**99**	**99**	**98**	**98**	**97**	**97**	**98**	**94**	**79**

2.7.4 The ETH-80 Image Set

The ETH-80 database [82] contains 80 objects from 8 categories. For each object, there are 41 images from different viewpoints. So the database contains 3280 images in total. To analyze appearance and contour based methods for object categorization, [82] first applied seven different approaches (including SC+DP), each with a single cue (either appearance or shape). Decision trees were then used to combine those approaches to get better performance. The test mode is leave-one-object-out cross-validation. Specifically, for each image in the database, it is compared to all the images from the other 79 objects. The recognition rate is averaged over all the objects.

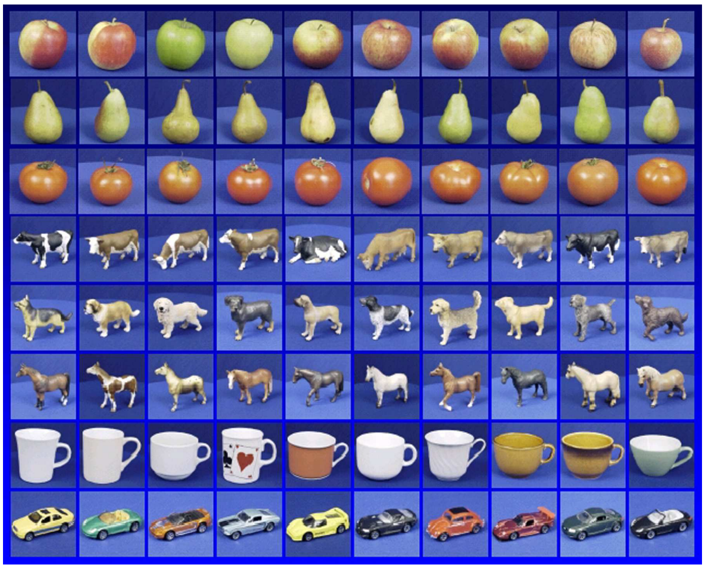


Figure 2.20: ETH-80 image set [82]. This data set contains 80 objects from 8 classes, with 41 images of each object obtained from different viewpoints.

We tested the MDS+SC+DP and the IDSC+DP on this data set with parameters: $n = 128$, $n_d = 8$, $n_\theta = 12$ and $k = 16$. Since only shape information is used, we compared the result with the seven single cue approaches in [82]. The recognition results are listed in Table 2.5. It shows that the IDSC works the best among all the single cue approaches.

Table 2.5: Recognition rates of single cue approaches on ETH-80 database [82]. All experiments results are from [82] except for MDS+SC+DP and IDSC+DP.

Alg.	Color Hist.	$D_x D_y$	Mag-Lap	PCA Masks	PCA Gray
Rec. Rate	64.85%	79.79%	82.23%	83.41%	82.99%

Alg.	SC Greedy	SC+DP	Decision Tree*	**MDS+SC+DP**	**IDSC+DP**
Rec. Rate	86.40%	86.40%	93.02%	**86.80%**	**88.11%**

* The decision tree is a multi-cue method which combines all the previous seven single-cue methods.

2.7.5 Foliage Image Retrieval

In this subsection we will demonstrate the application of the inner-distance on a real and challenging application, foliage image retrieval. Leaf images are very challenging for retrieval tasks due to their high between class similarity and large inner class deformations. Furthermore, occlusion and self-folding often damage leaf shape. In addition, some species have very similar shape but different texture, which therefore makes the combination of shape and texture desirable.

Swedish Leaf Database

The Swedish leaf dataset comes from a leaf classification project at Linköping University and the Swedish Museum of Natural History [141]. The dataset contains isolated leaves from 15 different Swedish tree species, with 75 leaves per species. Fig. 2.21 shows some representative silhouette examples. Some preliminary classification work has been done in [141] by combining simple features like moments, area and curvature etc. We tested with Fourier descriptors, SC+DP, MDS+SC+DP, IDSC+DP and SPTC+DP with parameters $n = 128$, $n_d = 8$, $n_\theta = 12$, $n_r = 8$ and $k = 1$. Each species contains 25 training samples and 50 testing samples per species. The recognition results with 1-nearest-neighbor are summarized in Tab. 2.6. Notice that unlike other experiments, the articulation invariant signature works a little better than IDSC on the leaf images. One possible explanation is that, as a real image dataset, the inner-angle for leaves are less robust due to boundary noise. Also notice that SPTC improves IDSC as we had expected.

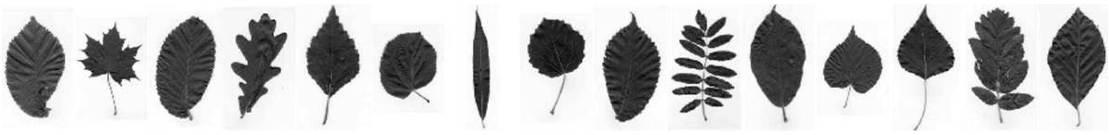


Figure 2.21: Typical images from Swedish leaf data base, one image per species. Note that some species are quite similar, e.g. the 1st, 3rd and 9th species.

Table 2.6: Recognition rates on the Swedish leaf dataset. Note that MDS+SC+DP and SPTC got same rates.

Alg.	Soderkvist [141]	Fourier	SC+DP	MDS+SC+DP	IDSC+DP	SPTC+DP
Rec. Rate	82%	89.6%	88.12%	95.33%	94.13%	95.33%

Smithsonian Isolated Leaf Database

This data set comes from the Smithsonian project [4] which is aimed to “build a digital collection of the Smithsonian’s collection of specimens and provide means to access it with text and photos of plants”. We designed an Electronic Field Guide image retrieval system that allows online visual searching. For example, during a filed test, a botanist can input a picture of an unknown leaf to the system and get the most visually similar leaves in a database. A detailed description of the system can be found in [7]. The task is very challenging because it requires querying from a database containing more than one hundred species and real time performance requires an efficient algorithm. In addition, the pictures taken in the filed are vulnerable to lighting changes and the leaves may not be flattened well.

We evaluated the proposed approaches on a representative subset of the leaf image database in the system[3]. The subset contains 343 leaves from 93 species (the number of leaves from different species varies). In the experiment, 187 of them are used as the

[3] http://www.cs.umd.edu/∼hbling/Research/data/SI-93.zip

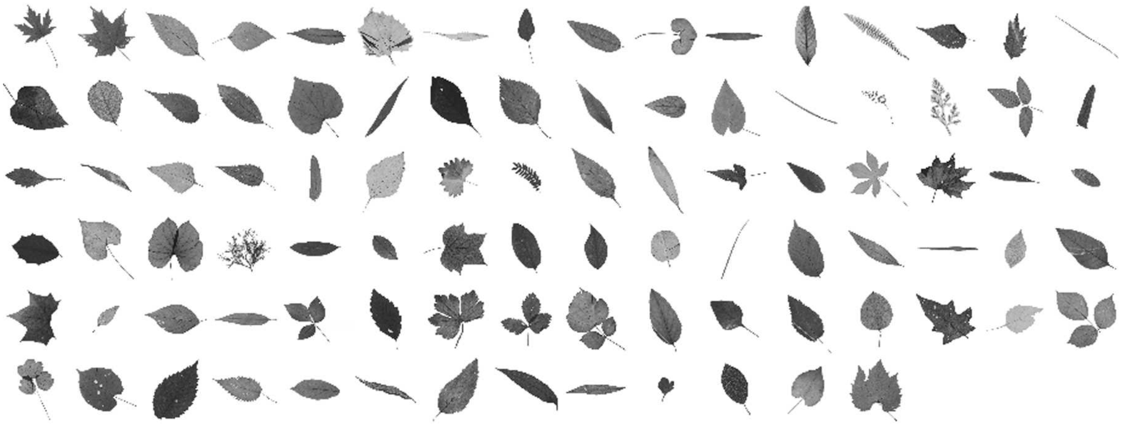


Figure 2.22: Smithsonian data set, containing 343 leaf images from 93 species. One typical image from each species is shown.

training set and 156 as the testing set. Note that there are only two instances per class in the training set on average. The retrieval performance is evaluated using performance curves which show the recognition rate among the top N leaves, where N varies from 1 to 16.

For the efficiency reasons mentioned above, only 64 contour points are used (i.e. $n = 64$). The similarity between leaves is measured by the shape context distance D_{sc} (see Sec. 2.2.3 or [15][4]) because it is faster than DP. Other parameters used in the experiment are $n_d = 5$, $n_\theta = 12$, and $n_r = 8$. Note that k is not needed because DP is not used here. The performance is plotted in Fig. 2.23. It shows that SPTC works significantly better than other methods. Fig. 2.24 gives some detailed query results of SPTC and IDSC, from which we can see how SPTC improves retrieval result by also considering texture information.

[4] It is based on a greedy matching and should not be confused with the bipartite matching based approach.

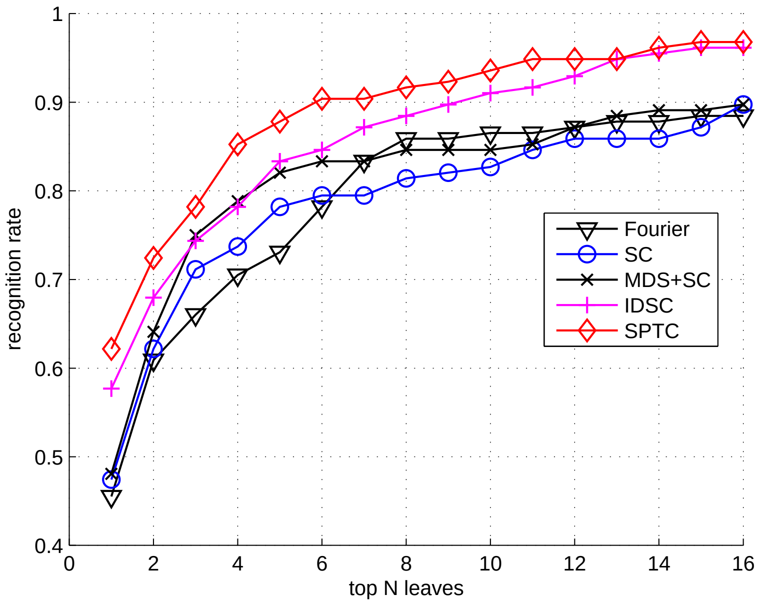


Figure 2.23: Recognition result on the Smithsonian leaf dataset. The ROC curves shows the recognition rate among the top N matched leaves.

2.7.6 Human body matching

In this experiment, we demonstrate the potential for using the proposed method on human body matching, which is important in human motion analysis. The dataset is a human motion sequence from a stationary camera, collected at the Keck lab at the University of Maryland. Silhouettes are extracted with background subtraction. Our task is to match the silhouettes from different frames. For adjacent frames, IDSC+DP performs very well, as demonstrated in the left of Fig. 2.25. For two silhouettes separated by 20 frames, the articulation turns out to be large and the matching becomes challenging. The IDSC+DP also gives promising results (see the right part in Fig. 2.25, for example). An application of the inner-distance to human motion analysis can be found in [85].

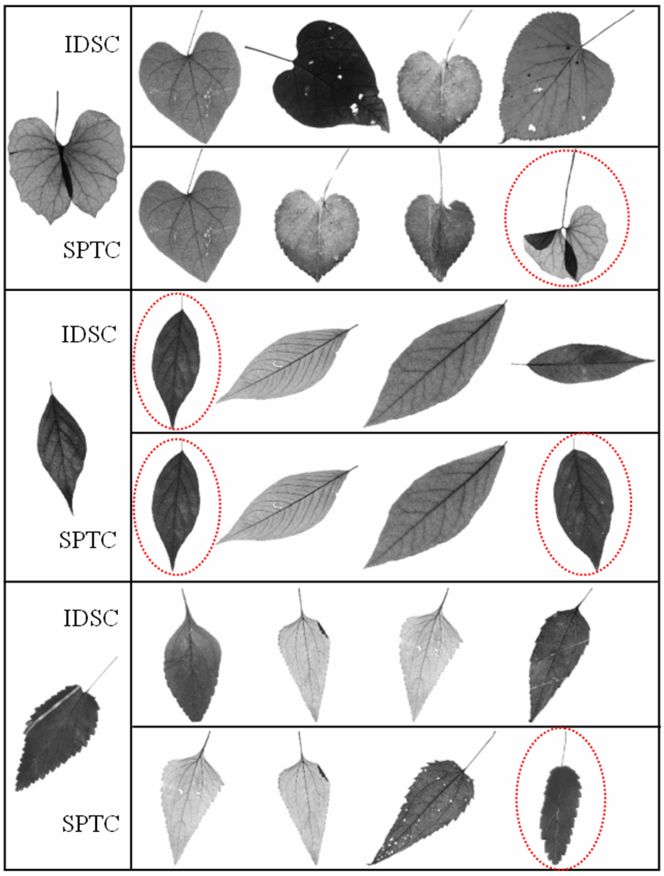


Figure 2.24: Three retrieval examples for IDSC and SPTC. The left column shows the query images. For each query image, the top four retrieving results are shown to its right, using IDSC and SPTC respectively. The circled images come from the same species as the query image.

2.8 Conclusion and Discussion

In this chapter we proposed using the inner-distance to build shape descriptors. We show that the inner-distance is articulation insensitive and is good for complicated shapes with part structures. Then the inner-distance is used to build better shape representations. We first build articulation invariant signatures for 2D shapes by combining the inner-distance and MDS. After that, we extended the shape context with the inner-distance to form a

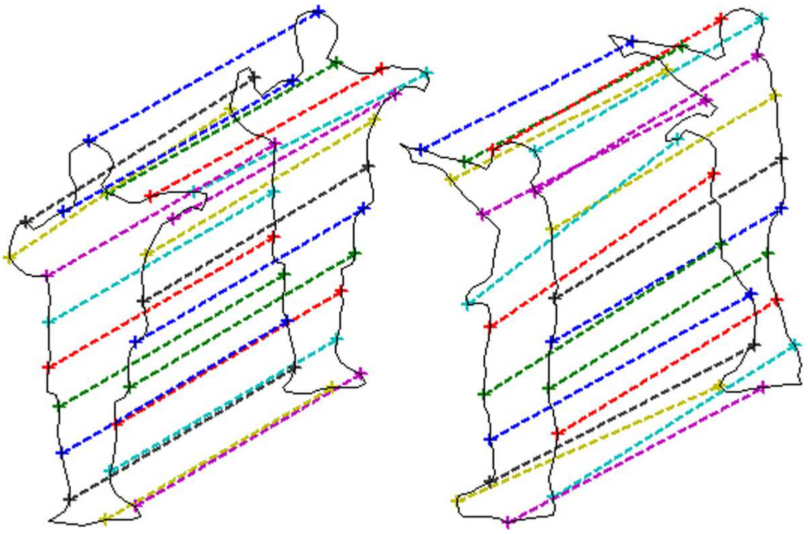


Figure 2.25: Human silhouettes matching. Left: between adjacent frames. Right: silhouettes separated by 20 frames, note that the hands are correctly matched. Only half of the matched pairs are shown for illustration.

new descriptor, and designed a dynamic programming based method for shape matching and comparison. Then, the descriptor is extended to capture texture information in a natural and efficient way. In retrieval experiments on several data sets, our approach demonstrated excellent retrieval results in comparison with several other algorithms. In addition, the approach is tested on sequential human silhouettes. Good matching results show the potential for using inner-distances in tracking problems. From these experiments, we are confident that the inner-distance works for shapes with complex part structure, particularly with large articulation. In addition, it is worth noting that the technique had been applied for a real electronic field guide system [7].

There are several interesting issues about the inner-distance we want to address here. First, to compute the inner-distance the shape boundary is assumed to be known.

This limits the approach to applications where the segmentation is available. Second, the inner-distance is sensitive to shape topology which sometimes causes problems. For example, the occlusion may cause the topology change of shapes. In addition, the inner-distance may not be proper for shapes involving little part structure and large deformation (no articulation).

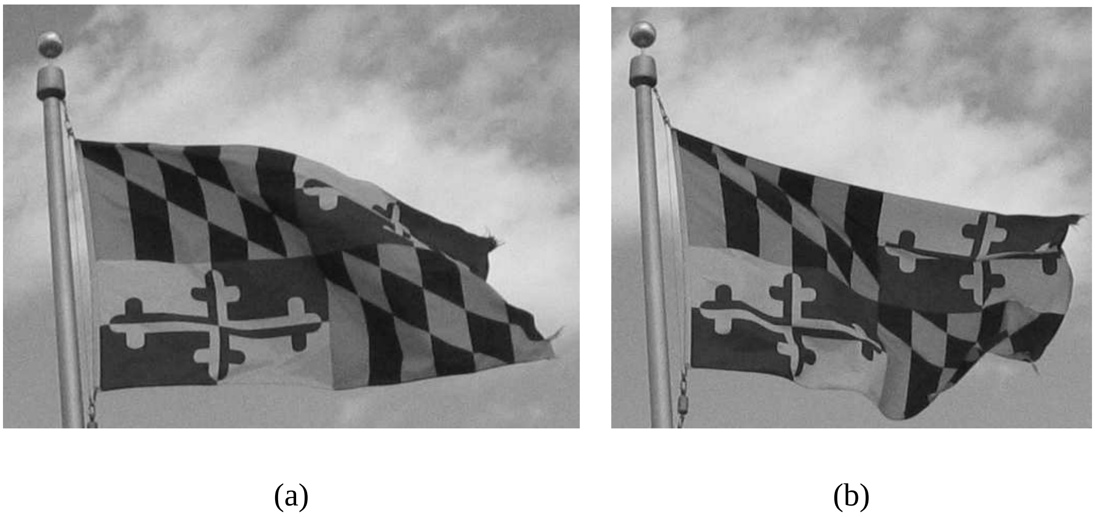
Chapter 3

Deformation Invariant Image Descriptors

3.1 Introduction

We propose a novel framework for building image descriptions that are invariant to deformations. An intensity image is treated as a surface embedded in 3D space, with the third coordinate proportional to the intensity values with an *aspect weight* α and the first two coordinates proportional to x-y with weight $1 - \alpha$. As α increases, the geodesic distance on the embedded surface becomes less sensitive to image deformations. In the limit when $\alpha \to 1$, the geodesic distance is exactly deformation invariant. Based on this idea, we use *geodesic sampling* to get sample points on the embedded surface, then build the *geodesic-intensity histogram* (GIH) as a local descriptor. GIH captures the spatial distribution of intensities on the embedded manifold. With $\alpha = 1$, it is exactly invariant to deformation. In matching experiments on data sets with both synthetic and real deformations, GIH demonstrates promising results in comparison to other approaches.

Our work builds on much recent work on invariant local descriptors. This work has found wide application in areas such as object recognition [40, 96], wide baseline matching [129, 148], image retrieval [130]. However, this previous work focuses on invariance to specific transformation groups, such as affine transformations. Affine invariant matching is useful when viewpoint changes relative to a rigid object that has locally planar regions large enough to contain distinctive intensity variations. It is less appropriate for objects without planar parts. For example, with a white Lambertian object, all intensity variations are due to 3D shape variations that lead to non-affine deformations as viewpoint changes. We are also interested in matching images of non-rigid objects, such as a flag waving or an animal moving its body (see Figure 3.1).



(a) (b)

Figure 3.1: Two images to be matched. Note that the right bottom corner of the flag in (b) is folded.

We make two main contributions. 1) To our knowledge, GIH is the first local image descriptor that is invariant to general deformations. Note that although intensity is invariant to deformation, color histogram is not in general. This is because the deformation may change the count of pixels with given intensity values. 2) Embedding images as surfaces in 3D and varying their aspect weight provides a novel framework for dealing with image deformation. This also suggests methods for finding deformation invariant feature points, and for building descriptors in which we trade off discriminability and deformation insensitivity with our choice of α. In addition, GIH does not require a specific interest region detector, since geodesic sampling automatically determines regions for gathering information.

The rest of the chapter is organized as follows. Sec. 2 discusses related works. Sec. 3 discusses deformation invariant features. Sec. 3.1 provides intuitions about deformation invariance through a 1D example. Sec. 3.2 shows how the aspect weight α relates to deformations by studying its relation to curve lengths on embedded image surfaces. Sec. 3.3 talks about geodesic distances and their computation and Sec. 3.4 explains geodesic sampling. Sec. 3.5 introduces the proposed descriptor, the geodesic-intensity histogram. Sec. 3.6 discusses several practical issues in using GIH, including illumination change, the interest points, and the choice of α. Sec. 4 describes all the experiments and analyzes the results. Sec. 5 concludes.

Part of this work appears in [90].

3.2 Related Work

Deformation appears in a large range of computer vision areas. A non-exclusive list includes object recognition [51, 36, 38, 91], face detection [39, 30], shape analysis [12, 138], stereo matching [100, 148], motion analysis [121, 163], etc.

Lots of research effort contributes to the study of local descriptors which are invariant to geometric or more general invariant descriptors. Among them the affine transformation is the mostly studied, because its relative simplicity and the fact that many complex image transformation can be locally approximated by it. For example, Lindeberg [88] proposed extracting scale invariant regions via the extremum of the scale space. Lowe [96] proposed a fast and efficient way to compute the scale invariant feature transform (SIFT), which measures the gradient distribution in detected scale invariant regions. Mikolajczyk and Schmid [103] proposed an affine invariant interest point detector through combining a scale invariant detector and the second moment of Harris corners [55]. Other work on affine invariant features can be found in [68, 119]. Mikolajczyk and Schmid [102] gave a review and performance evaluation of several local descriptors including steerable filters [43], moment invariants [46], complex filters [129, 13], scale invariant feature transform (SIFT) [96] and cross-correlation.

Though achieving success in a wide range of vision tasks, the affine invariants have some limitations. First, object shape change can not always be accurately approximated by affine models, e.g. the flying flags in Fig. 3.1. Second, viewpoint change of non-planar surfaces is non-linear in general. Invariants to more general deformation are therefore desired, which is the main interest in this chapter.

Matas et al. [100] proposed using *maximally stable extremal region* (MSER) for robust wide baseline matching.

Vedaldi and Soatto [150] gave a theoretic study of viewpoint invariants for non-planar scenes. With the Lambertian and non-occlusion assumption, they proved the existence of non-trivial viewpoint invariants and discusses trade-offs in shape-discriminability.

Lepetit et al. [83] treated wide baseline points matching as a classification problem and used randomized trees [10] for a solution. Their method is robust to viewpoint change by synthesizing features from different viewpoints during the training stage. Furthermore, this also makes the approach very time-efficient for real applications. Benefitting from this work, Pilet et al. [121] presented a real-time approach for detecting non-rigid surfaces. They model a deformable surface with a mesh structure, and the optimum deformation is obtained through minimizing a criterion that balances distortion and smoothing.

Our method can be categorized with so-called distribution based descriptors, which use histograms to capture local image information. These include the spin image [80, 65], shape context [15], and PCA-SIFT [71]. Our method differs from all these in two ways. First, our method is invariant to all deformations. Second, our descriptor automatically detects its support region, i.e., it does not need a special deformation invariant detector.

Treating images as 2D surfaces embedded in 3D space is not new. Our work is particularly motivated by the Beltrami framework proposed in [140]. They treat images as 2D manifolds and operators on the manifold are applied directly on the surface for low level vision tasks such image enhancement. Our work focuses more on feature extraction. Also we are more interested in using large aspect weights to produce deformation invariance.

Geodesic distance has also been used in object recognition. For example, in [36] it is used to build bending invariant signatures for real surfaces. Our work is different in that we are using embedded surfaces that vary according to aspect weights. This achieves deformation invariance for 2D images, as opposed to bending invariance for 3D data.

This work can also be viewed as a general version of our previous work of using the inner-distance for shape matching [91]. If we treat a given shape as a binary image, the inner-distance between two boundary points is the same as the geodesic distance used in this chapter for α near 1.

3.3 Deformation Invariant Features

In this section we first discuss deformation invariance within the framework of embedded image surfaces, then introduce geodesic sampling and the geodesic-intensity histogram, which is invariant to deformation. After that, some practical issues in using this descriptor are discussed.

3.3.1 Intuitions about Deformation Invariance

We consider deformation as homeomorphisms (continuous, one-to-one transformations) between two images. Intensities change position, but not their value, with deformations. To obtain distinctive descriptors we collect intensities from a neighborhood. Our problem is to do this in a way that is invariant to deformation.

To gain intuition, we consider a one dimensional image. Figure 3.2(a) shows two 1D images I_1, I_2, with height denoting image intensity (dashed lines are the geodesics between marked points). They look quite different, but they are related by a deformation (composed of stretching and compressing in different places). Consider the images as 1D surfaces embedded in a 2D space, where intensity is scaled by α, the *aspect weight* and x is scaled by $1 - \alpha$. Using different αs produces the images in Figure 3.2 (b,c,d). We see that when α increases, the graphs of the two embedded images look more and more similar. It is natural to expect that they become exactly the same as $\alpha \rightarrow 1$. One way to explain this is that α controls the weight on the intensity compared to the weight on the image coordinate. A larger α means that we place more importance on the intensity, which is deformation invariant. So $\alpha = 1$ leads to a deformation invariant view of the image.

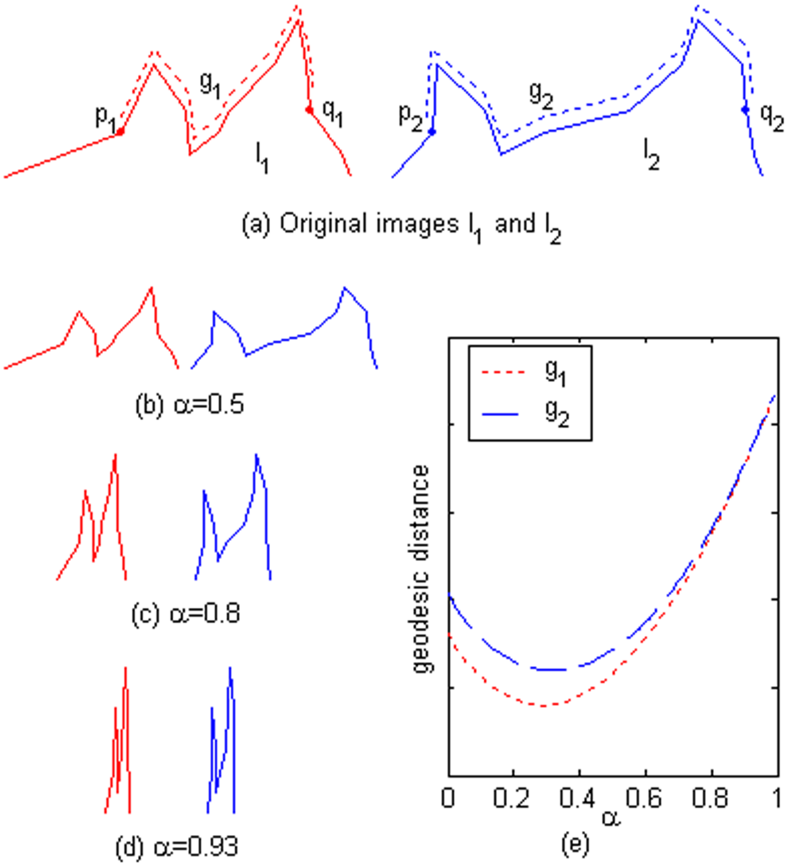


Figure 3.2: Deformation invariance for one dimensional images. Details in Section 3.3.1

How does α work? Let p_1, q_1 be two points on I_1, with their deformed counterparts p_2, q_2 on I_2. Consider the geodesic distance g_1 between p_1 and q_1 on αI_1 and g_2 between p_2 and q_2 on αI_2. Figure 3.2 (e) shows how g_1 and g_2 vary for different α, from which we see that g_1 and g_2 tend to become similar when α increases. This implies that the geodesic distance on the embedded surface tends to be deformation invariant when $\alpha \rightarrow 1$.

This provide a solution for the problem of finding deformation invariant neighborhoods. For a given interest point, we can use the geodesic distance to find and sample within a neighborhood, and then build a descriptor based on the intensities of the sampled points. If the procedure is done with a very large α (in the limit approaching 1), then the descriptor is deformation invariant, which is exactly what we want.

3.3.2 Curve Lengths on Embedded Surfaces

In this section we show that as α increases, the lengths of curves change less and less when an object deforms. Let $I_1(x, y)$ be an image defined as $I : R^2 \rightarrow [0, 1]$. Let $I_2(u, v)$ be a deformation of I_1. Because deformation is a homeomorphism, and so it is invertible, we can write $u = u(x, y), v = v(x, y)$, $x = x(u, v), y = y(u, v)$, and $I_2(u, v) = I_1(x(u, v), y(u, v))$.

Denote the embedding of an image $I(x, y)$ with aspect weight α as $\sigma(I; \alpha) = (x' = (1 - \alpha)x, y' = (1 - \alpha)y, z' = \alpha I(x, y))$. Denote σ_1, σ_2 as the embeddings of I_1, I_2 respectively

$$\sigma_1 = (x' = (1 - \alpha)x, y' = (1 - \alpha)y, z' = \alpha I_1(x, y))$$

$$\sigma_2 = (u' = (1 - \alpha)u, v' = (1 - \alpha)v, w' = \alpha I_2(u, v))$$

Let γ_1 be a regular curve on σ_1, $t \in [a, b]$, and γ_2 the deformed version of this curve on σ_2

$$\begin{aligned}
\gamma_1(t) &= (x'(t), y'(t), z'(t)) \\
&= ((1 - \alpha)x(t), (1 - \alpha)y(t), \alpha I(x(t), y(t))) \\
\gamma_2(t) &= (u'(t), v'(t), w'(t)) \\
&= ((1 - \alpha)u(t), (1 - \alpha)v(t), \alpha I(u(t), v(t)))
\end{aligned}$$

Where

$$w'(t) = \alpha I_2(u(t), v(t)) = \alpha I(t) = \alpha I_1(x(t), y(t)) = z(t)$$

because the intensity is invariant to deformation,

Now we can study the length of γ_1, γ_2, denoted as l_1, l_2 respectively. We have

$$\begin{aligned} l_1 &= \int_a^b \sqrt{x_t'^2 + y_t'^2 + z_t'^2} dt \\ &= \int_a^b \sqrt{(1-\alpha)^2 x_t^2 + (1-\alpha)^2 y_t^2 + \alpha^2 I_t^2} dt \qquad (3.1) \\ l_2 &= \int_a^b \sqrt{u_t'^2 + v_t'^2 + w_t'^2} dt \\ &= \int_a^b \sqrt{(1-\alpha)^2 u_t^2 + (1-\alpha)^2 v_t^2 + \alpha^2 I_t^2} dt \qquad (3.2) \end{aligned}$$

Where the subscripts denote partial derivatives, e.g., $x_t \doteq dx/dt$, $u_t \doteq \partial u/\partial t$, etc....

From (3.1) and (3.2) it is clear that for a large α, the curve length is dominated by the intensity changes along the curve. In the limit when $\alpha \to 1$, l_1, l_2 converge to the same value. Also, the length of curves with constant intensities tend to be trivial compared to lengths of curves with non-constant intensities.

In the rest of the chapter, when talking about deformation invariance, we implicitly assume that $\alpha \to 1$.

3.3.3 Geodesic Distance and Level Sets

It follows from the last subsection that the *geodesic distance*, which is the distance of the shortest path between two points on the embedded surfaces, is deformation invariant. Given an interest point $p_0 = (x_0, y_0)$, the geodesic distances from it to all other points on the embedded surface $\sigma(I;\alpha)$ can be computed using the level set framework [135]. Points with identical geodesic distances from p_0 are treated as level curves. For images defined on discrete grids, the fast marching algorithm ([135]) provides an efficient method of computing these curves.

Figure 3.3 shows two example results of the geodesic distances computed for real images. It shows that when α is small (in (c),(d)), the geodesic distances are almost like Euclidean distances in the image plane. With a large α (in (e),(f)), the geodesic distance captures the geometry of image intensities and automatically adapts to deformation.

One interesting issue is that since real images are defined on discrete grids, the fast marching method we use implicitly assumes that the surface is piecewise constant (constant within the region of each pixel). The image can also be interpolated as a smooth surface, in which case the arguments above still hold.

In the real implementation, instead of using $\alpha \in [0,1]$, we fix the weight for $x - y$ as 1 and use β as the weight for intensity $\beta \in (0,\infty)$. This is practically useful because it avoids the possible singularities. An obvious relation between α and β is $\beta = \alpha/(1-\alpha)$.

Fast marching

$$\phi_t + F|\nabla\phi| = 0 \tag{3.3}$$

Where we use

$$F = \frac{1}{\sqrt{1+\sigma_x^2+\sigma_y^2}}$$

as the front marching speed.

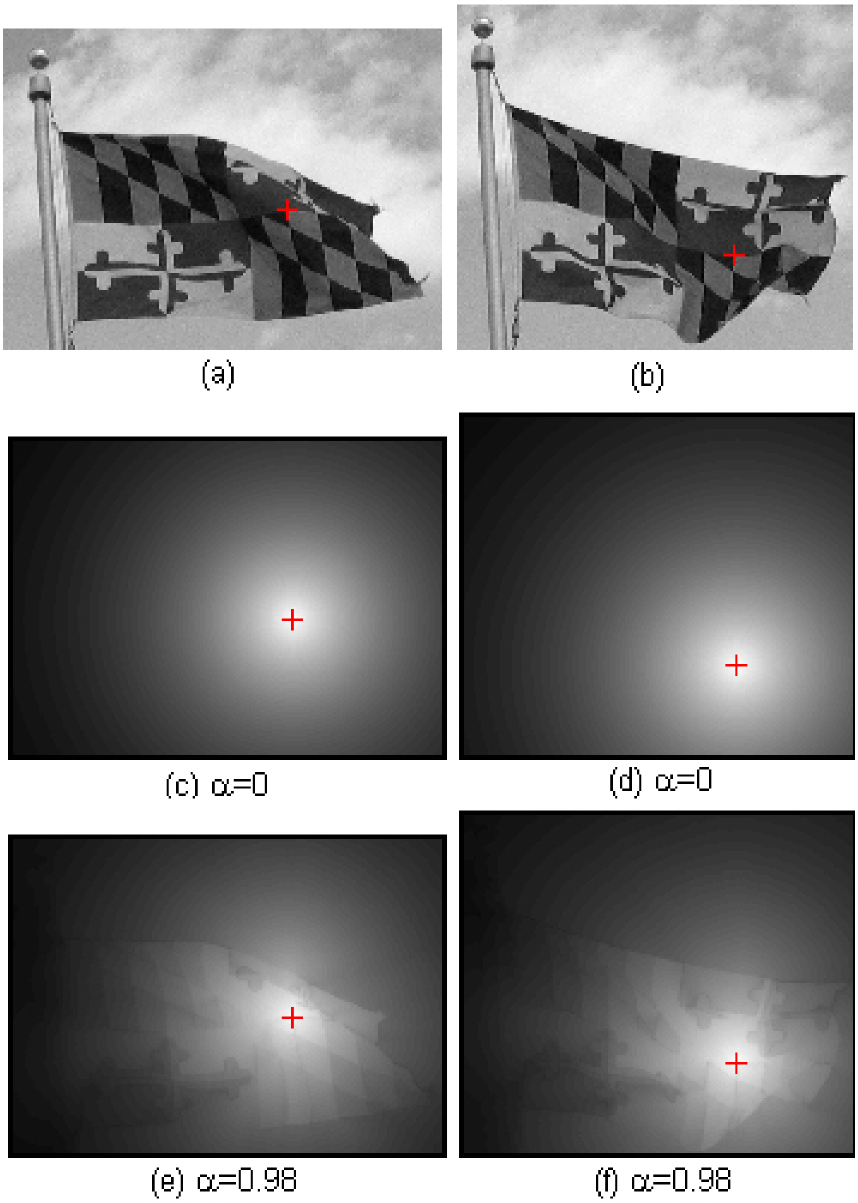


Figure 3.3: Geodesic distances computed via fast marching. The marked point in (a) corresponds to the marked point in (b) after deformation. (c),(e) shows the geodesic distances of all pixels in (a) from the marked point, with different α's. Darker intensities mean large distances. (d),(f) shows the same thing for the marked point in (b). Note that image structures of (a) and (c) are captured in the distance map in (e) and (f).

3.3.4 Deformation Invariant Sampling

Geodesic level curves provide us a way to find deformation invariant regions surrounding interest points. These regions can be used as support regions for extracting deformation invariant descriptors. To derive invariant descriptors, we must also sample these regions using geodesic distances, to find deformation invariant sample points. In the following Δ is used to denote the sampling interval.

Geodesic sampling for 2D images is done in two steps.

- The level curves are extracted at intervals of Δ.

- Points are sampled from each level curve at intervals of Δ.

Figure 3.3.4 gives examples of 2D geodesic sampling. Note that the sampling along uniform intensity regions is sparser than along regions with large intensity variation. Intuitively, this implies that deformations (such as stretching) will not change the number of sample points, although it may change their locations. We sample densely, so that changes in the location of sample points do not have much effect on the resulting histogram.

3.3.5 The Geodesic-Intensity Histogram

Now we introduce the *geodesic-intensity histogram* (GIH), which is a deformation invariant descriptor extracted from geodesic sampling. It captures the joint distribution of the geodesic distance and the intensity of the sample points. Since both the geodesic distance and the intensity are deformation invariant, so is the GIH. It is based on spin images, which produce a related descriptor using Euclidean distance ([80, 65]).

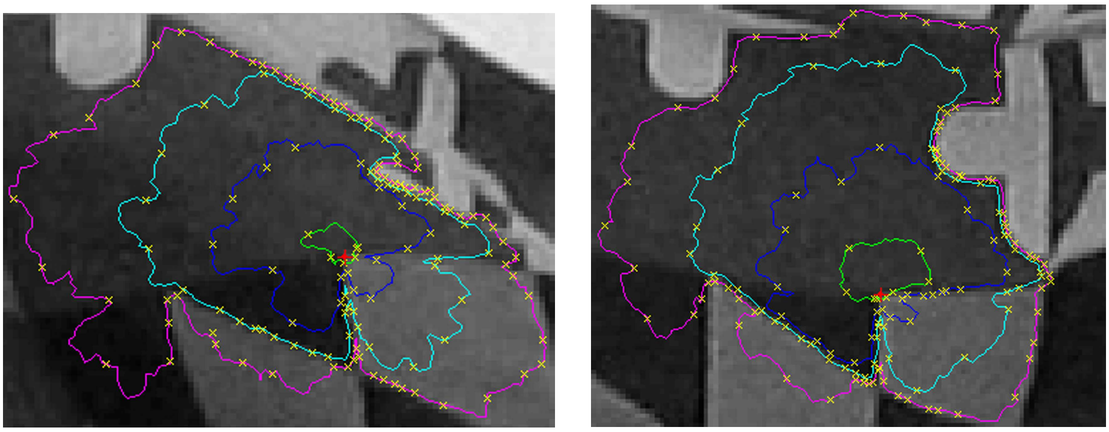

(a) (b)

Figure 3.4: 2D geodesic sampling. $\alpha = 0.98$. A large interval is used for better illustration. The interest points (the red '+' points in the center) are the same as in Figure 3.3 (a),(b). Level curves of the same colors correspond to the same geodesic distances. The yellow cross points are the sampled points.

Given an interest point p, together with a sample point set P_p obtained via geodesic sampling, the GIH H_p at p is a normalized two dimensional histogram obtained through the following steps:

1. Divide the 2D intensity-geodesic distance space into $K \times M$ bins. Here K is the number of intensity intervals, and M the number of geodesic distance intervals. The geodesic intervals can be segmented either linearly or at log scale.

2. Insert all points in P_p into H_p: $\forall 1 \leq k \leq K, \forall 1 \leq m \leq M$, $H_p(k,m) = \#\{q \in H_p : (I(q), g(q)) \in B(k,m)\}$. Here $I(q)$ is the intensity at q, $g(q)$ is the geodesic distance at q (from p), and $B(k,m)$ is the bin corresponding to the kth intensity interval and the mth geodesic interval.

3. Normalize each column of H_p (representing the same geodesic distance). Then

normalize the whole H_p.

Figure 3.5 displays examples of the geodesic-intensity histograms of two points with deformation. The two histograms are quite similar, although the deformation between the two images is quite large.

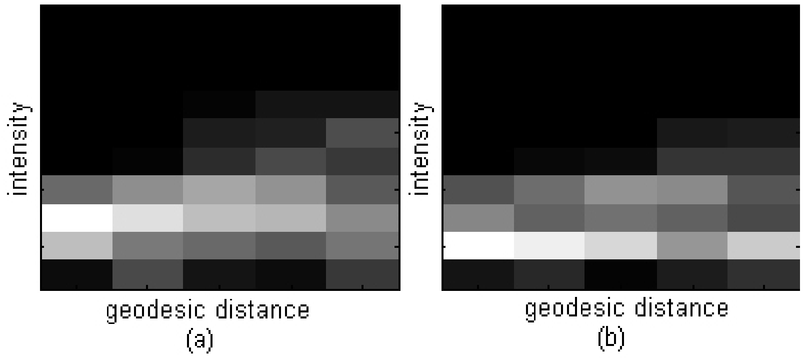


Figure 3.5: Geodesic-intensity histograms, $\alpha = 0.98$, $K = 10$, $M = 5$. (a), (b) for points in Figure 3.3 (a),(b) respectively.

Given two geodesic-intensity histogram H_p, H_q, the similarity between them is measured using the χ^2 distance:

$$\chi^2(p, q) \equiv \frac{1}{2} \sum_{k=1}^{K} \sum_{m=1}^{M} \frac{[H_p(k, m) - H_q(k, m)]^2}{H_p(k, m) + H_q(k, m)} \tag{3.4}$$

3.3.6 Practical Issues

Dealing with illumination change. We use an affine model for lighting change ([102]), i.e., $aI(x, y) + b$ for the illumination change of the pixel at (x, y). There are two steps to make GIH insensitive to lighting change. 1) GIH is made invariant to lighting in the same way as [102]. That is, when building the histogram, the intensity is normalized by subtracting the mean and then divided by the standard deviation, where the mean and deviation is estimated on the sampled point set H_p. 2) Compensate for the effect of lighting change on the geodesic sampling. For large α, the intensity change dominates the geodesic distance (3.1,3.2). So the change of geodesic distance is approximately linear with rate a under the lighting model, which is equivalent to changing α to $a\alpha$. So when we compare two interest points, we compare several GIH's that use different α's, and pick the match with minimal χ^2 distance (3.4).

Interest points. GIH does not require a special interest point detector since it automatically locates the support region. However, there are problems with using some feature points. First, deformation invariance makes points within a constant region indistinguishable. Second, for real images the intensity on edges or corners may vary due to sampling. We have found that *extreme points*, where images have local intensity extremum, are less affected by the above factors (they are locally unique in the continuous cases). The extreme point can be viewed as a deformation invariant version of the DoG point proposed by Lowe [96], which is scale invariant. In Sec. 3.4.3 we tested the performance of GIH using several different interest point operators.

Choosing α. In the following experiments we will use a very large α (0.98) because we want to deal with large deformations. However, in domains involving only small deformations, a relatively smaller α might be a better choice. Smaller α's can lead to descriptors that are somewhat insensitive to deformations, but that provide more information since they do not treat images related by large deformations as identical. It is obvious that GIH with $\alpha = 0$ becomes equivalent to spin images [80].

3.4 Experiments

In this section we will describe our experiments using the GIH for interest point matching. Experiments were conducted on two groups of image pairs. One contains synthetic deformation as well as illumination change, the other contains real non-affine deformations. We have two experiments. The first one compares the GIH's matching ability to several other approaches. The second experiment studies the performance of GIH using several different kinds of interest points including the proposed extreme points.

3.4.1 Experimental Setup

Data set We evaluate the proposed method using two groups of images. The first group contains eight image pairs with synthetic deformation and illumination change (see Figure 3.6, the original images are from the Berkeley segmentation dataset [1]). The deformation is created by mapping the original images to non-flat surfaces and viewing them from different viewpoints. The lighting change is generated through an affine model (intensities limited to $[0..1]$). The second group contains three pairs of images with real deformations (see Figure 3.7).

Interest point We use Harris-affine points [103] for the matching experiments. The interest point is detected using the online code provided by Mikolajczyk [101]. One reason for this choice is its affine invariance. This makes the other descriptors invariant to affine transformation, although it is not necessary for our descriptor. The other reason is that [101] also provides executable codes for several state-of-art descriptors that we can use

[1] http://www.cs.berkeley.edu/projects/vision/grouping/segbench/

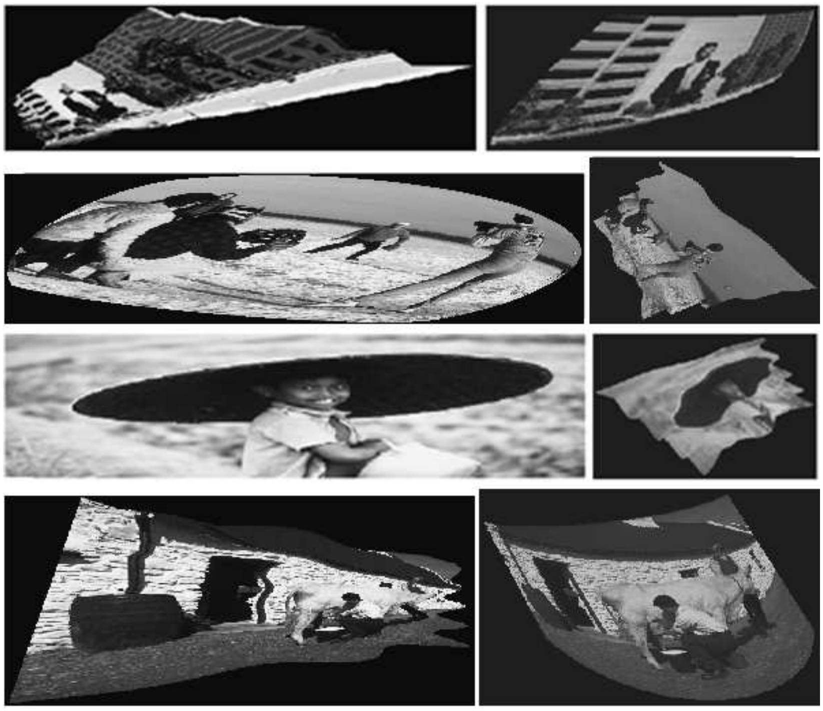


Figure 3.6: Four of the eight pairs of images with synthetic deformation and illumination change.

for comparison. For each image, we pick the 200 points extracted by the detector with the largest cornerness.

Evaluation criterion For each pair of images together with their interest points, we first obtained the ground truth matching (automatically for synthetic images, manually for real images). Then, for efficiency we removed those points in image 1 with no correct matches. After that, every interest point in image 1 is compared with all interest points in image 2 using the descriptors to be compared. An interest point p_1 in image 1 is treated as a correct match of another point p_2 in image 2 if the deformation of p_1 is within a three pixel distance of p_2. The detection rate among the top N matches is used to study the

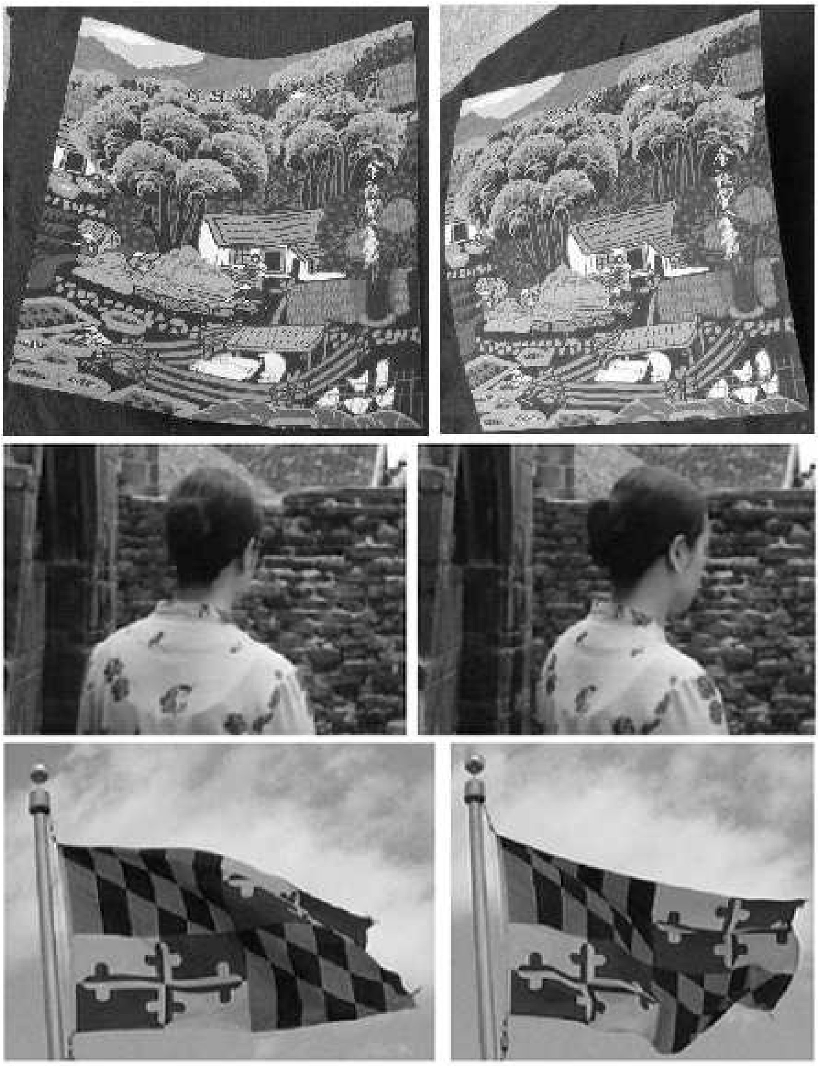


Figure 3.7: Images with real non-affine deformation.

performance. The detection rate is defined in a way similar to [102]:

$$r = \frac{\#correct\ matches}{\#possible\ matches} = \frac{\#correct\ matches}{\#points\ in\ image\ 1} \tag{3.5}$$

3.4.2 Matching Experiment

In this experiment we will study the performance of GIH in comparison with several other methods. The experiments are conducted on both the synthetic and real deformation data sets. All of them use the Harris-Affine interest point.

Mikolajczyk and Schmid [102, 101] provided convenient online code for several state-of-the-art local descriptors. The descriptors are normalized to enable direct comparison using sum of square differences (SSD). Benefitting from their code, we compare the geodesic-intensity histogram with steerable filters [43], SIFT [96], moments invariants [46], complex filters [129] and spin images [80].

The main difference between the evaluation here and that in [102] lies in that [102] focused more on the evaluation of region-like descriptors. For example, some of their experiments use interest regions instead of interest points. Furthermore, their matching criterion between two features is also related to their support regions. Also note that the Harris-Affine point is chosen because it provides affine invariant support regions to the descriptors we will compare to, although it is not necessary for GIH (see Sec. 3.4.3).

We tested two versions of the geodesic-intensity histogram. Version one uses $\alpha = 0.98, K = 13, M = 8$. This tests the ability of the GIH. The other version is a degenerate version where $\alpha = 0, K = 10, M = 5$. This demonstrates that GIH becomes like spin images for $\alpha = 0$.

A Receiver Operating Characteristics (ROC) based criterion is used which is similar to the one in [102]. Instead of using the false positive rate, we study the detection rates among the top N matches, as N varies.

Figure 3.8 displays the ROC curves for the experiment on the synthetic deformation data set and Figure 3.9 for the real deformation data set. From the ROC curves we can see that GIH performs better than other methods in both data sets regardless of illumination changes. Note that for $\alpha = 0$, the performance drops a lot, with the performance similar to spin images (with no affine invariant support region).

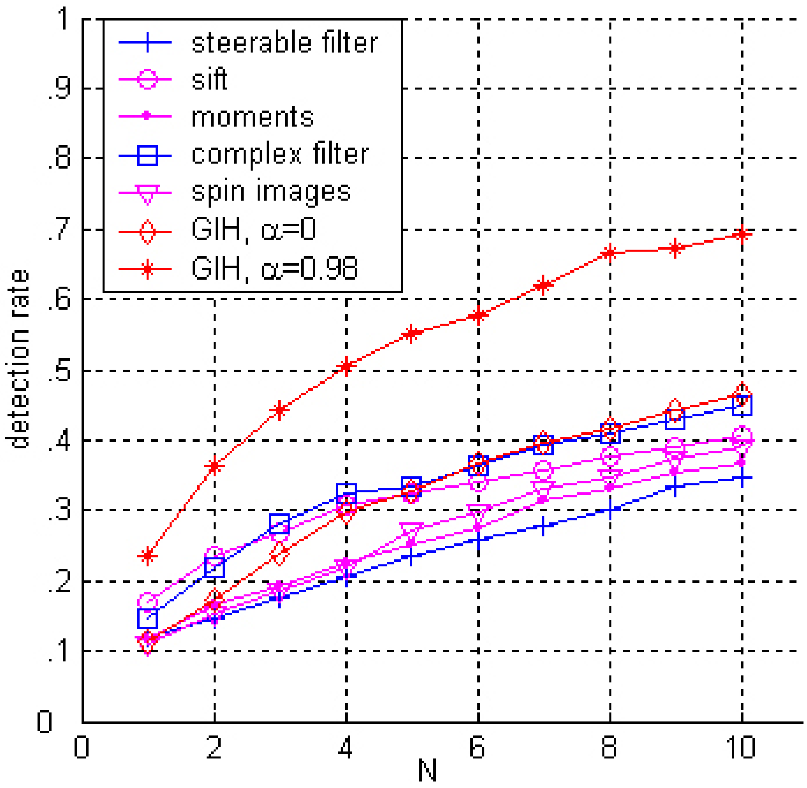


Figure 3.8: Experiment results on the synthetic deformation data set (see Figure 3.6).

3.4.3 Interest Points

This experiment is to test the performance of GIH using several kinds of interest points. In addition to extreme points and DoG [96] points, we also tested on Harris corners [55] and Harris-Affine points [103]. We use the code provided at [101] except for the extreme points. The experiment is conducted on the synthetic deformation data set. For each image, 200 points are picked with the largest detector responses (cornerness, for example). For extreme points, the response is computed through Laplace-of-Gaussian filtering. The same parameters for GIH are used for all kinds of interest points, $\alpha = 0.98$, $K = 13$, $M = 8$.

Since different interest point detectors may generate different number of correct

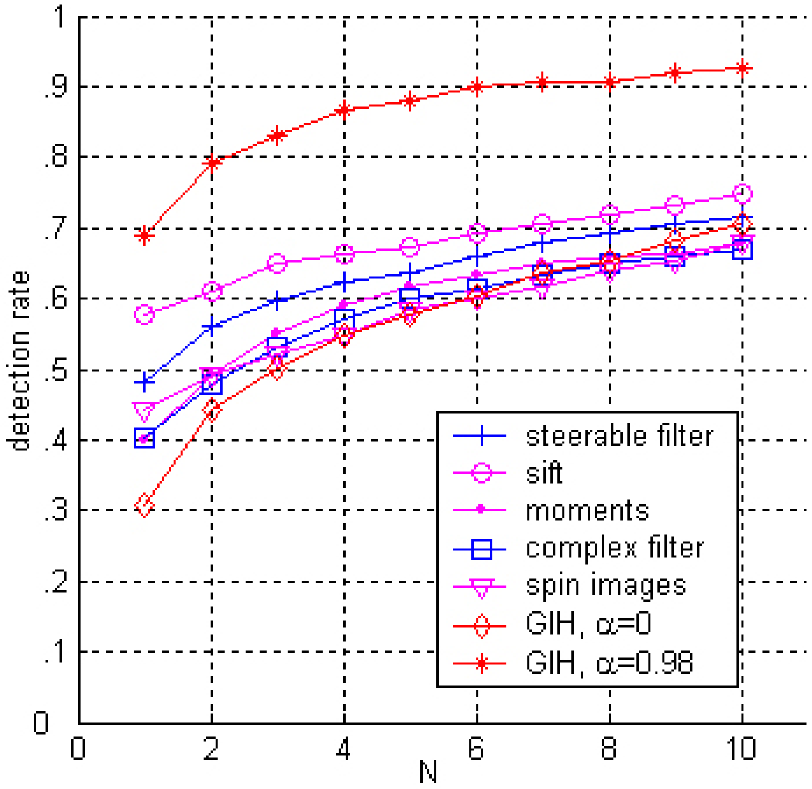


Figure 3.9: Experiment results on the real deformation data set (see Figure 3.7).

correspondences, the ROC curves is plotted as the detection rate versus the false positive rate instead of N as in the previous experiment. The false positive rate is defined as (similar to [102])

$$r_{false} = \frac{\#false\ matches}{(\#points\ in\ image\ 1)(\#points\ in\ image\ 2)}$$

Figure 3.10 shows the ROC curves. From the figure we can see that GIH works better than the others for small false positive rate less than 0.03 (this roughly corresponds to the top 6 matches). For large false positive rates, DoG performs the best. The Harris corner works the worst with GIH, which is consistent with our previous discussion.

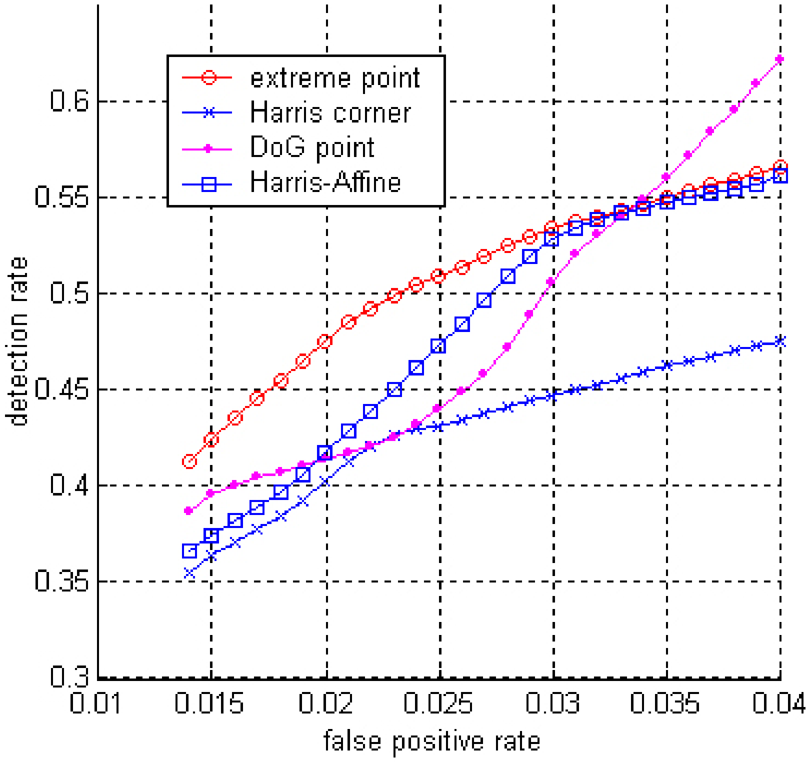


Figure 3.10: GIH using different kinds of interest points. The false positive rate 0.04 roughly corresponds to $N = 8$ as in Figure 3.9 and 3.8.

3.5 Conclusions

In this chapter we proposed a novel deformation invariant feature, the geodesic-intensity histogram, for intensity images. Images are treated as 2D surfaces embedded in 3D spaces. We then showed that the geodesic distance along the surface is invariant to deformation when the embedding aspect weight $\alpha \to 1$. The geodesic-intensity histogram is a 2D histogram measuring the geodesic distances and the intensities surrounding an interest point. With the geodesic sampled neighborhood points and an $\alpha \to 1$, the proposed histogram becomes deformation invariant. After that, we discussed practical issues including how to deal with illumination change and the option of choosing α to balance deformation invariance and discriminativity. The proposed descriptor is tested on data sets with both synthetic and real deformations. In all the experiments the new descriptor performs excellently in comparison with several other methods.

Chapter 4

Using A Gradient Orientation Pyramid for Robust Passport Photo Verification

4.1 Introduction

Face verification across ages is an important problem and has many applications, such as passport photo verification, image retrieval, face animation, surveillance, etc. This is a challenging task because human faces can vary a lot over time in many aspects, including facial texture (e.g. from winkles), shape (e.g. from weight gain), facial hair, presence of glasses, etc. In addition, the image acquisition conditions and environment for taking face photos often undergoes a large change, which can cause non-uniform illumination and scale changes.

In this chapter we are interested in passport photo verification. Since real photos are used instead of digital ones, scanning is needed in image acquisition. This process often causes additional challenges. For example, the original photo can be smudged. Scanning sometimes also causes saturation and/or additional noise. Some typical passport images with different age gaps are shown in Fig. 4.1.

It is natural to model face verification as a two-class classification problem, i.e., *intra-personal* and *extra-personal* classification [108]. Previous works did this using the intensity difference [120] or (normalized) intensity [67] as an input feature for support vector machines (SVM) [149]. These approaches largely relied on SVM's ability to automatically extract the relevant information from the input features. However, for face images involving large age differences, it is difficult for SVM to handle the previously mentioned problems, such as non-uniform lighting variation.

We are interested in finding robust face descriptions especially to lighting. Inspired by the work on lighting insensitivity with a Lambertian assumption and its application to face recognition [25], we propose using a gradient orientation pyramid (GOP) as a reliable face descriptor. First, by discarding magnitude information, the gradient orientation is insensitive to lighting change [25]. Second, we use the pyramid to bring in hierarchical information. Then, given a face image pair, we use the cosines between gradient orientations at all scales to build the "difference" between the pair. Finally, the "difference" is combined with SVM for face verification tasks. We applied the proposed approach for passport verification tasks and tested it on two passport image datasets with large age differences. Promising results are observed in comparison with several other approaches, including the Bayesian+PointFive face [123], the SVM+difference space [120], two commercial face recognition products, etc.

The main contribution of this chapter is the proposed orientation direction + SVM approach for face verification. Though simple, the proposed approach achieves very promising results on the challenging passport photo verification task. A side message from our experiments is that a better representation (e.g. gradient direction) can sometimes improve the performance of SVM dramatically (e.g. compared to the intensity difference) - this is interesting because SVM is known to be good at automatic feature selection/extraction.

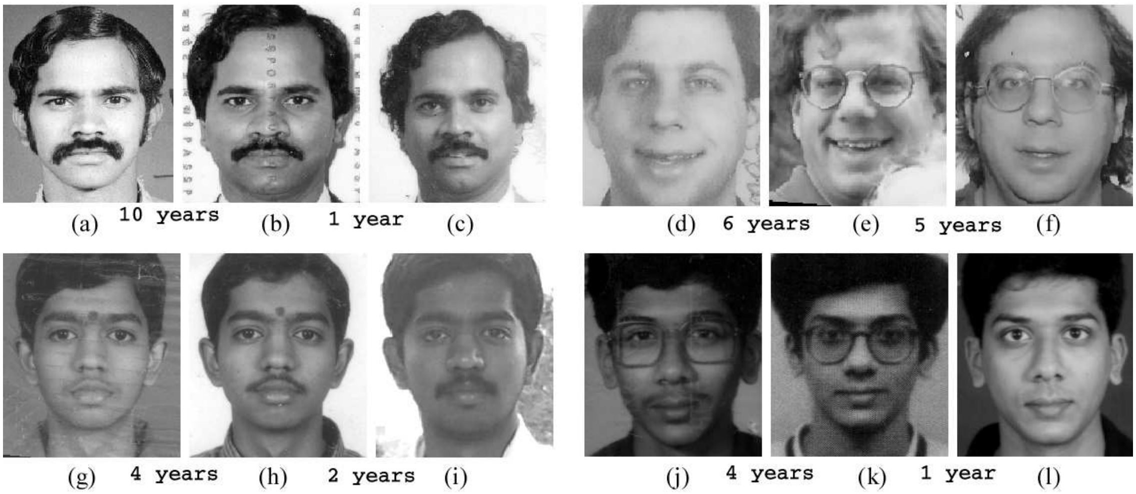


Figure 4.1: Typical passport images with age differences [123]. Note the smudging on image (b) and the saturation caused by scanning in (e).

The rest of the chapter is organized as following. Sec. 4.2 discusses the related work. After that, the framework for face verification using the support vector machine is described in Sec. 4.3. Then, we introduce the gradient orientation pyramid in Sec. 4.4. Sec. 4.5 describes our experiments on two passport image datasets involving large age separations. Finally, Sec. 4.6 concludes.

Part of this work appears in [52].

4.2 Related Work

Face recognition and detection has been widely studied for several decades. A thorough survey can be found in [164]. A lot of work has been done to handle the problem under different conditions, including lighting, pose, expression, etc. The aging process and its effect on face analysis, which we are interested in, has recently attracted research effort. Most work has focused on modelling the aging process [124], age estimation

[75, 77, 123, 166], and simulation [78]. In comparison, face verification across ages is far less studied [123].

Modelling face verification as a two-class classification problem is not new. Moghaddam et al. [108] used a Bayesian framework for the intra-personal and extra-personal face classification. Phillips [120] used SVM for face recognition problems and observed good results on the FERET dataset compared to component based approaches. Jonsson et al. [67] used SVM for face authentication problems. Our work is different in that we use the gradient orientation pyramid instead of intensity differences [108, 120] or the intensity itself [67] as a face description. Furthermore, we are more interested in passport photos with age differences.

Ramanathan and Chellappa [123] adapted the probabilistic eigenspace framework [107] for face identification across age progression, which is the most related to our work. Instead of using a whole face, only a half face (called a PointFive face) is used to alleviate the non-uniform illumination problem. Then, the eigenspace technique and a Bayesian model is combined to model the inter-personal and extra-personal image differences. Model parameters are learnt by an EM scheme and the inference is done in a standard MAP fashion. The approach demonstrated promising results on a passport photo verification task. Our work also focuses on face verification across ages but differs from their work in two ways. First, different descriptors are used. Second, we use SVM instead of the Bayesian eigenspace framework.

Image gradients are widely used for feature building. Lowe [96] proposed the scale invariant feature transform (SIFT) that uses the weighted histogram of gradient orientation as local descriptors. SIFT has been used for many applications including face detection

[139]. Dalal and Triggs [32] proposed using histograms of oriented gradients (HoG) for human detection. The work is further extended by Zhu et al. [168] using a cascade scheme to gain more efficiency. Unlike these work, we exclude magnitudes of image gradients and use only the orientations.

The direction of image gradient has been proposed for lighting insensitive recognition (e.g. [17]) and was shown to be insensitive to changes in lighting direction under a Lambertian assumption [25]. To the best of our knowledge, this is the first time the gradient direction is combined with SVM for face verification problems. In addition, we also propose using hierarchical structure to further improve descriminability.

4.3 Problem Formulation

4.3.1 Task Description

Passport photo verification is important in the process of passport renewal and related face authentication applications. For example, when a person submits a new photo for renewal, the ideal system can automatically tell whether he is an imposter by comparing the new photo to previous photos that are usually taken years before.

Due to the challenges of this task, it is not realistic to expect one hundred percent accuracy. However, we can still use computer vision techniques to save human effort on this task. A common way to evaluate verification uses two criterion: the rate of correct rejection on imposter images (correct reject rate) and the rate of correct acceptance (correct accept rate) on true images. These two rates conflict although we want both rates to be as high as possible. In practice for the passport verification task, the correct reject rate is most important because rejected images will be examined by a human. For example, we can fix such a rate at a very high level (e.g., 99%) while making the correct acceptance rate as high as possible.

4.3.2 Classification Framework

We model face verification as a two-class classification problem [108, 120, 67]. Given an input image pair I_1 and I_2, the task is to assign it as either *intra-personal* or *extra-personal*. In this section we briefly describe the framework of using a support vector machine for this task. Details about support vector machines can be found in [149, 22].

Given any image pair (I_1, I_2), it is first mapped it into the feature space. Formally, we have,

$$\mathbf{x} = \mathcal{F}(I_1, I_2) \in \mathbb{R}^d$$

where $\mathbf{x}$ is the feature vector extracted from the image pair (I_1, I_2) through the feature extraction function (functional) $\mathcal{F}(.,.)$, and $\mathbb{R}^d$ is the d-dimensional feature space.

Then the support vector machine is used to divide the feature space into two classes, one for intra-personal pairs and the other for extra-personal pairs. Using same terminology as in [120], we denote the separating boundary with the following equation

$$\sum_{i=1}^{N_s} \alpha_i y_i K(\mathbf{s_i}, \mathbf{x}) + \mathbf{b} = \mathbf{\Delta} \tag{4.1}$$

where N_s is the number of support vectors and s_i is the i-th support vector. Δ is used to trade off the correct reject rate and correct accept rate as described in Sec. 4.5. $K(.,.)$ is the kernel function that provides SVM with non-linear abilities. The RBF kernel is chosen in our experiment due to its effectiveness and efficiency. The RBF kernel is defined as

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma * |\mathbf{x}_1 - \mathbf{x}_2|^2) \tag{4.2}$$

where γ is a parameter determining the size of RBF kernels. In our experiments, we use the OSU SVM toolbox [97] that provides a convenient Matlab interface to the LibSVM library [24].

4.4 Gradient Orientation Pyramid

There is one question left open in the previous section: what is $\mathcal{F}(.,.)$? A natural choice is to use the intensity difference between I_1 and I_2, which is called *difference space* in [108] and also been used in [123, 120]. With an appropriate normalization scheme, the intensity difference can be made robust to affine lighting changes. However, the affine lighting model is not always sufficient for face images, especially for images taken at times separated by years. Instead, to alleviate this problem, we propose using the cosine of gradient orientations as features of image pairs, because gradient orientation is known to be insensitive to lighting change [25]. In addition, we organize the gradient orientation in a hierarchical fashion, which further improves its discriminability.

4.4.1 Gradient Orientation Pyramid

Our proposed features are partly motivated by recent work on using gradient information for object representations ([96, 32]). In these works, the gradient directions were weighted by gradient magnitudes. To make the feature more reliable, we discard the gradient magnitudes inspired by [25]. Furthermore, the gradient directions at different scales are combined to make a hierarchical representation.

Given an image $I(\mathbf{p})$, where $\mathbf{p} = (x, y)$ indicates pixel locations. We first define the pyramid of I as $\mathcal{P}(I) = \{I(\mathbf{p}; \sigma)\}_{\sigma=0}^{s}$ as

$$\begin{aligned} I(\mathbf{p}; 0) &= I(\mathbf{p}) \\ I(\mathbf{p}; \sigma) &= [I(\mathbf{p}; \sigma - 1) * \Phi(\mathbf{p})] \downarrow_2 \quad \sigma = 1, ..., s \end{aligned} \tag{4.3}$$

where $\Phi(\mathbf{p})$ is the Gaussian kernel (0.5 is used as the standard deviation in our experiments), $\downarrow_2$ denotes half size downsampling, and s is the number of pyramid layers. Note that in (4.3) the notation I is used both for the original image and the images at different scales for convenience.

Then, the gradient orientation at each scale σ is defined by its normalized gradient vectors at each pixel.

$$g(I(\mathbf{p}; \sigma)) = \frac{\nabla(I(\mathbf{p}, \sigma))}{|\nabla(I(\mathbf{p}, \sigma))|} \tag{4.4}$$

Naturally, the *gradient orientation pyramid* (GOP) of I, is defined as $\mathcal{G}(I) = \{g(I(\mathbf{p}, \sigma))\}_{\sigma=0}^{s}$.

Fig. 4.2 illustrates the computation of a GOP from an input image.

4.4.2 Differences Between GOPs

Given an image pair (I_1, I_2) and corresponding GOPs $(\mathcal{G}(I_1), \mathcal{G}(I_2))$, the feature vector $\mathbf{x} = \mathcal{F}(I_1, I_2)$ is computed as the concatenation of the cosines of the difference between gradient orientations at each pixels and all scales. The computation can be efficiently achieved through the inner product of the corresponding entries of GOP, i.e., for pixel $\mathbf{p}$ at scale σ, it is computed as

$$f(I_1(\mathbf{p}; \sigma), I_2(\mathbf{p}; \sigma)) = g(I_1(\mathbf{p}; \sigma)) \cdot g(I_2(\mathbf{p}; \sigma)) \tag{4.5}$$

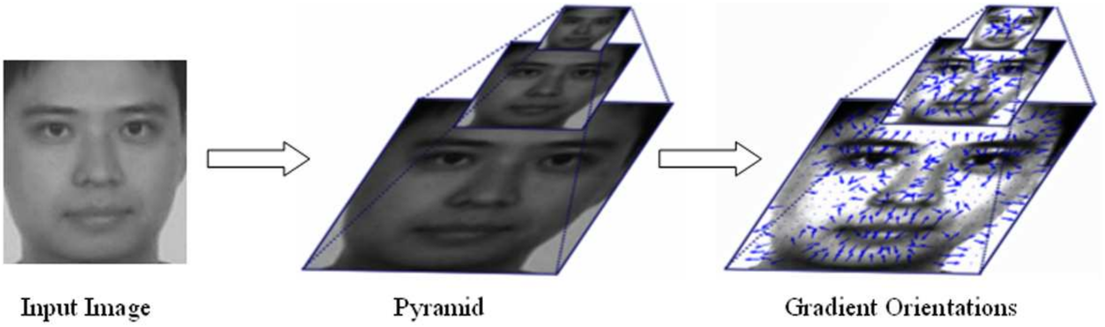


Figure 4.2: Compute a GOP from an input image. Note: 1) In the right figure, the gradient orientations at "flat" regions are excluded. 2) The images in the right figure are made brighter for better illustration.

The cosine values are organized into a feature vector $\mathbf{x}$ as

$$\mathbf{x} = \mathcal{F}(I_1, I_2) = (\dots\ ,\ f(I_1(\mathbf{p};\sigma), I_2(\mathbf{p};\sigma))\ ,\ \dots)^\top \tag{4.6}$$

where $\mathbf{p}$ is organized in lexicographic order and σ in increasing order.

We summarize the advantages of using GOPs for face verification tasks as following.

- GOP is insensitive to illumination change [25]. As a result, no normalization is needed on the input images.
- The pyramid technique provides a natural way to perform face comparison at different scales.
- The inner product between normalized gradient lies in a finite range ($[-1, 1]$) that automatically limits the effect of outliers.

In the following, we use SVM+GOP to indicate the proposed approach.

4.5 Experiments

In this section, we describe experiments applying SVM+GOP on passport image verification tasks, in comparison with several other approaches.

4.5.1 Datasets

We tested the proposed approach on two real passport image datasets, which we will refer to as Passport I and Passport II respectively. Passport I is the dataset used in [123], and contains 452 intra-personal image pairs (several duplicated pairs are removed). Passport II contains 1824 intra-personal image pairs. Images in both datasets are scanned passport images. They are in general frontal images with small pose variations. The lighting condition varies, and can be non-uniform and saturated. The age differences between image pairs are summarized in Table 4.1. It shows that both datasets have significant age gaps for intra-personal images. Fig. 4.3 further shows the distribution of age differences in the datasets.

Intuitively, Passport II is more challenging than Passport I for verification tasks because of the relatively larger age differences. Furthermore, we observed that the image resolution change in Passport II is also larger that in Passport I.

In order to created a verification task, 2251 extra-personal image pairs are randomly generated for Passport I and 9492 for Passport II.

In our experiments (SVM-based approaches), the images are preprocessed using the same scheme as in [123]. This includes manual eye location labelling, alignment by eyes and cropping with an elliptic region. For memory efficiency, image sizes are reduced

Table 4.1: Passport datasets for identification tasks. "Std." is short for standard deviation.

Dataset	# intra-personal pairs	# extra-personal pairs	mean age	std. age	mean age diff.	std. age diff.
Passport I	452	2251	39	10	4.27	2.9
Passport II	1824	9492	48	14.7	7.45	3.2

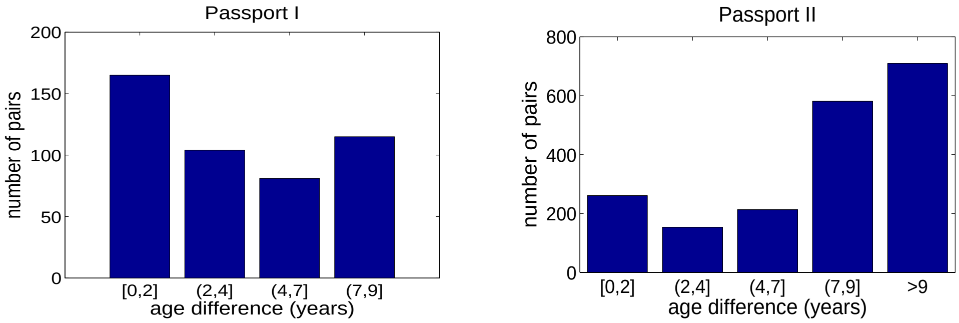


Figure 4.3: Distribution of age differences in the passport image databases. Left: Passport I. Right: Passport II.

to 96×84 for Passport I and 72×63 for Passport II. In comparison, 207×180 is used in the Bayesian based approach [123].

To alleviate the alignment problem, when comparing two GOPs, we tried different alignments with small shiftings (2 pixels). In our experiments it helped to improve the performance by around 0.5% (equal error rate). A similar technique is used by [99].

4.5.2 Methods

We compared the following approaches:

- SVM+GOP. The approach proposed in this chapter.

- SVM+GO. This is similar to SVM+GOP, except that only the gradient orientation (GO) at the finest scale is used without a hierarchical representation.

- SVM+G. This one is similar to SVM+GO, except that the gradient (G) itself is used instead of gradient orientation. It can also be viewed as weighting gradient orientations with gradient magnitudes.

- SVM+diff [120]. As in [120], we use the differences of normalized images as input features combined with SVM.
- Bayesian+PFF [123]. This is the approach combining Bayesian framework [107] and PointFive Face (PFF) [123].
- Vendor A. This is a commercial face recognition product.
- Vendor B. This is a commercial face recognition product.

Note that Vendor A and Vendor B are two commercial softwares[1] and the original passport images were used as input.

4.5.3 Experimental Evaluation

For verification tasks, the correct reject rate (CRR) and the correct acceptance rate (CAR) are two critical criterion, which are defined as

$$\begin{aligned} CRR &= \frac{\text{\# correct rejected extra-personal pairs}}{\text{\# total extra-personal pairs}} \\ CAR &= \frac{\text{\# correct accepted intra-personal pairs}}{\text{\# total intra-personal pairs}} \end{aligned} \tag{4.7}$$

The performances of algorithms are evaluated using the CRR-CAR curves that are usually created by varying some classifier parameters. We want both rates to be as high as possible, though they usually conflict. In many face verification tasks such as passport review tasks, CRR plays more an important role in practice.

[1] Anonymous due to agreements with the companies.

For SVM-based approaches, we used three-fold cross validation. For each dataset, we first (randomly) divide the image pairs into three parts (for both intra-personal and extra-personal pairs). Then we conducted three experiments, each time picking a different part as the testing set and the rest as the training set. For each experiment, the CRR-CAR curve is created by adjusting parameter Δ in (4.1). The total performance is evaluated as the average of the output CRR-CAR curves of the three experiments. For Bayesian+PFF, we use the results reported according to the experiments in [123].

Fig. 4.4 shows the CRR-CAR curves for the experiments. In addition, Table 4.2 lists the equal error rates (i.e. when CRR=CAR). There are several observations from the experimental results.

- The proposed SVM+GOP approach demonstrated excellent performance compared to other approaches. This is particularly true for passport verification tasks because a high correct reject rate is very important for these tasks. In contrast, the Bayesian-based approach [123] is more suitable on tasks where high correct accept rate is desired.

- Among the SVM-based approaches, GOP works the best. The gradient direction obviously plays a main role in GOP's excellent performance. In comparison, the gain of the hierarchical structure is less significantly, especially for a task that requires high correct reject rates. A preliminary experiment is described in the following paragraphs.

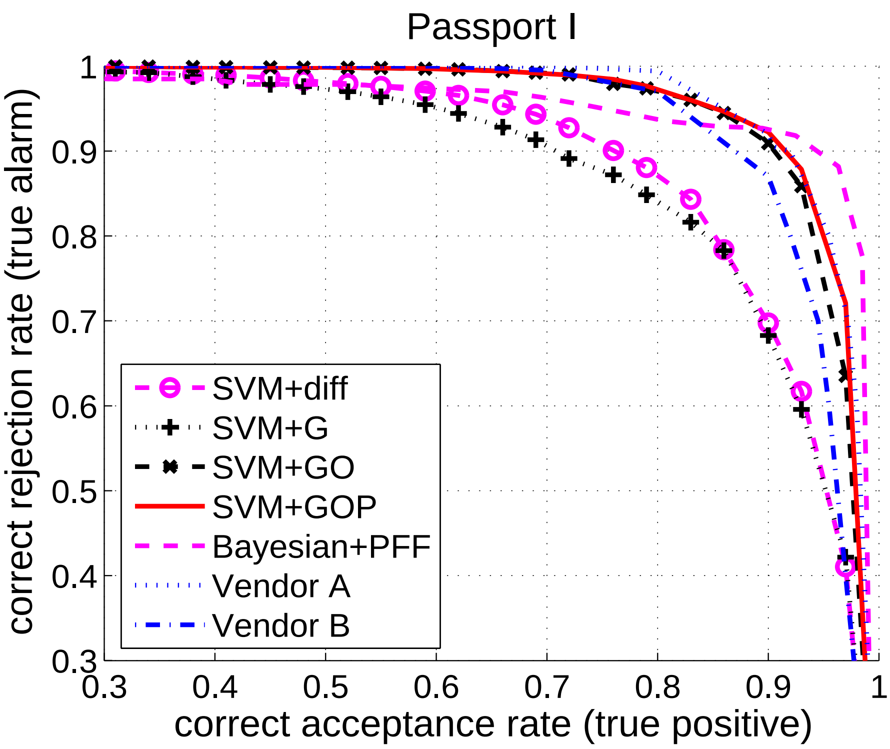
- The performance of SVM+diff drops a lot on Passport II compared to Passport I. The main reason is that the image resolutions in Passport II vary more widely than in Passport I. In contrast, this shows that the gradient based descriptors are more robust to resolution change.

Table 4.2: Equal error rates.

Approach	SVM+diff [120]	SVM+G	SVM+GO	SVM+GOP	Bayesian+ PFF [123]	Vendor A	Vendor B
Passport I	16.5%	17.8%	9.5%	8.9%	8.5%	9.5%	11.5%
Passport II	32.2%	17.4%	12.0%	11.2%	12.5%	13.5%	8.0%

To get a sense on how aging affects different descriptors, we analyzed the falsely rejected intra-personal pairs versus their ages. Specifically, we first collect the false rejected image pairs for experiments on both datasets in the SVM-based approaches. We use the threshold that leads to the equal error rate. Then we divide the errors into different groups corresponding to different age gaps. The distributions of the errors versus the age differences are shown in Fig. 4.5. From these distributions we see clearly that SVM+GOP works consistently better than other descriptors for different age gaps.

To study the effect of the pyramid with larger image size, we conducted experiments comparing SVM+GOP and SVM+GO on large resolution images. We used randomly chosen subsets of Passport I and Passport II due to memory limitations. Specifically, for Passport I, one third of the image pairs are used with the size 207×180. For Passport II, one fourth of the image pairs are used with the size 160×140. The CRR-CAR curves are shown in Fig. 4.6. Compared to the results in Fig. 4.4, the improvement from using pyramids is more obvious.



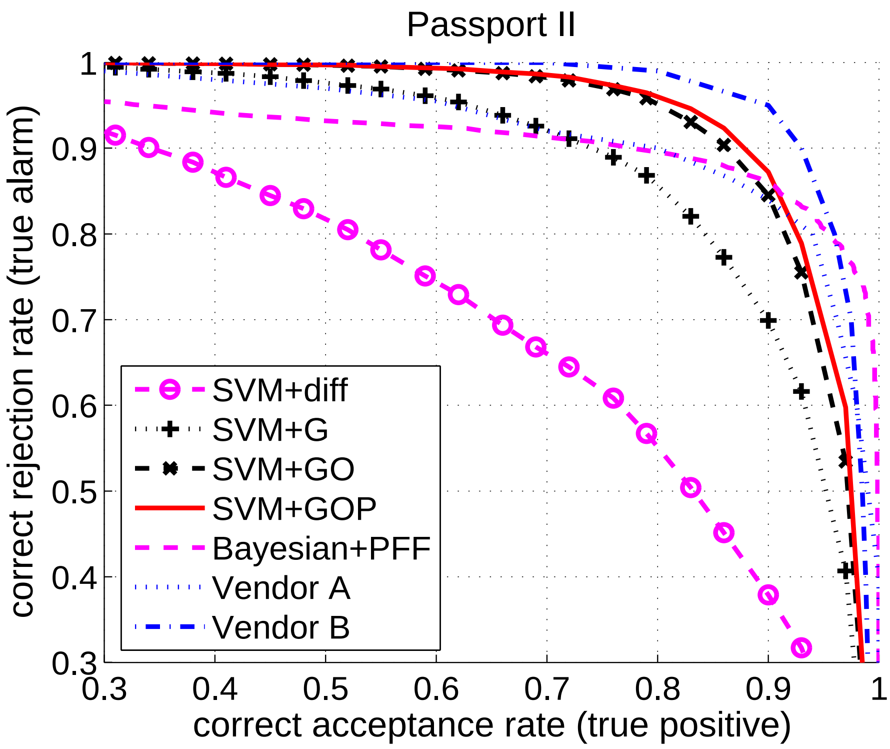


Figure 4.4: CRR-CAR curves. Top: on Passport I. Bottom: on Passport II.

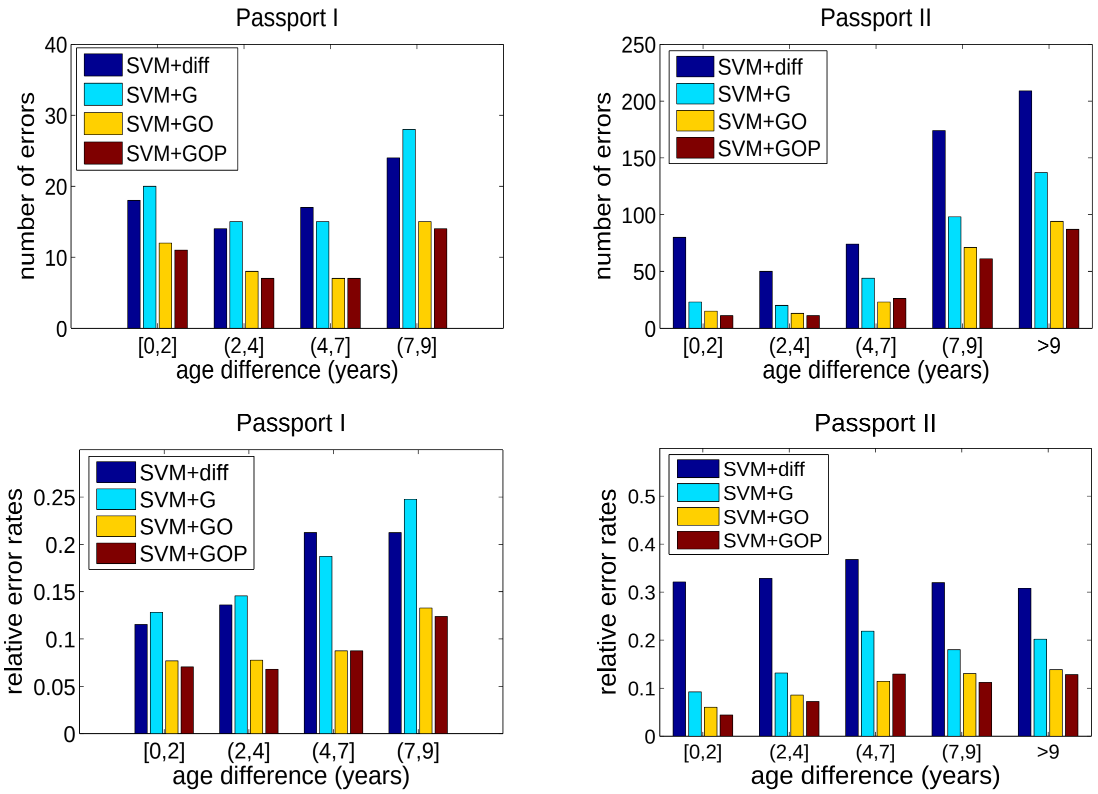


Figure 4.5: Falsely rejected intra-personal pairs versus age differences. Top row: the number of errors. Bottom row: the rate of errors. Left column: Passport I. Right column: Passport II.

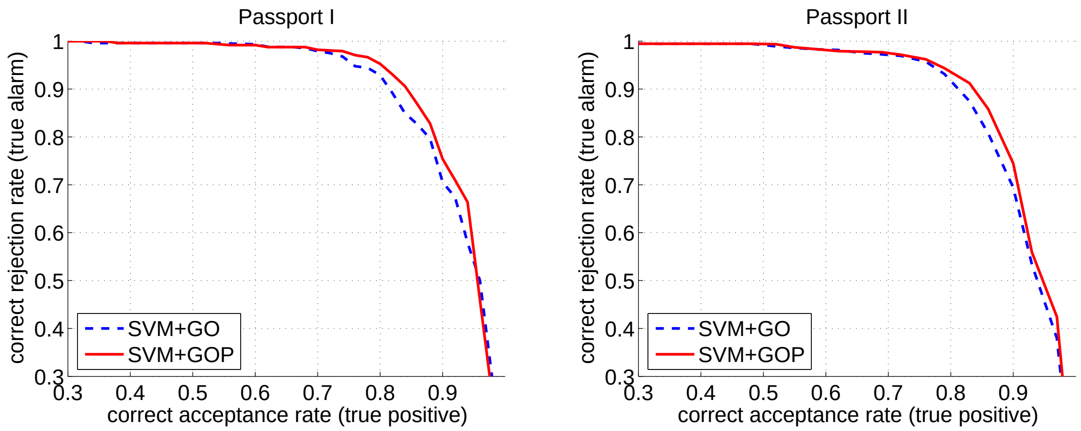


Figure 4.6: CRR-CAR curves for testing the effectiveness of hierarchical structures. Left: on Passport I (one third subset). Right: on Passport II (one fourth subset).

4.6 Conclusion

In this chapter we proposed a robust face descriptor, the gradient orientation pyramid, for face verification tasks across ages. Compared to previously used descriptors such as image intensity, the new descriptor is more robust and performs well on face images with large age differences. In addition, the pyramid technique enables the descriptor to capture hierarchical facial information. In our experiments with comparison to several techniques, the new approach demonstrated very promising results on two challenging passport databases.

Part II

Robust Comparison of Histogram-Based Local Descriptors

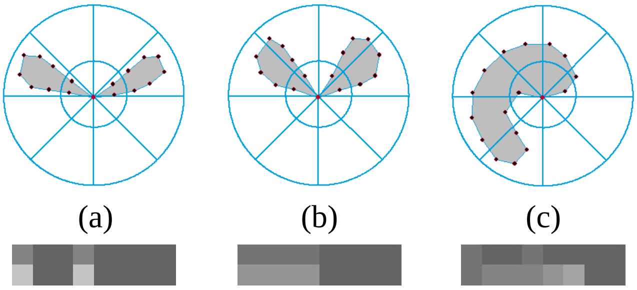
Chapter 5

Introduction to Part II

[1]

Histogram-based local descriptors (HBLDs) are ubiquitous tools in numerous computer vision tasks, such as shape matching [15, 110, 145, 147, 91], image retrieval [96, 71, 111, 104, 90], texture analysis [80], color analysis [128, 143], 3D object recognition [65, 114], stereo matching [129, 148], to name a few. For comparing these descriptors, it is common to apply *bin-to-bin* distance functions, including L_p distances, χ^2 statistics, KL divergence, and Jensen-Shannon (JS) divergence [87]. In applying these functions, we often assume that the domain of the histograms are previously aligned and call them *bin-to-bin* distances accordingly. However, in practice, such an assumption can be violated due to various factors, such as shape deformation, non-linear lighting change, and heavy noise. The *Earth Mover's Distance* (EMD) [128] is a *cross-bin* distance function that addresses this alignment problem. By modelling one histogram as piles of earth and another histogram as a set of pits, EMD defines the dissimilarity between two histograms as the minimum work required to move all the earth to all the pits. In other words, EMD is the optimization solution of the *transportation problem* that is a special case of the linear programming (LP). Beyond the color signature application originally sought by Rubner et al. [128], we demonstrate in this chapter that EMD is useful for more general classes

[1] The main part of this work is done during my internship at Siemens Corporate Research with Dr. Kazunori Okada.



(a) (b) (c)

(d) (e) (f)

Distance	$d(a,b)$	$d(b,c)$
L_1	1.0	0.875
L_2	0.3953	0.3644
χ^2	0.6667	0.6625
EMD	0.5	1.5625

(g)

Figure 5.1: An example where bin-to-bin distances meet problems. (a),(b) and (c) show three shapes and log-polar bins on them. (d),(e) and (f) show the corresponding 2D histograms (shape context) of (a),(b) and (c) using the same 2D bins, respectively. The distances between (d) and (e) and the distance between (e) and (f) are summarized in table (g). All EMDs here use the L_1 ground distance.

of histogram descriptors such as SIFT [96] and shape context [15].

Fig. 5.1 illustrates an example with the shape context, demonstrating the advantage of the cross-bin EMD over common bin-to-bin functions. The small articulation of two blobs between (a) and (b) causes a large change in their corresponding shape contexts as 2D histograms. EMD correctly describes the perceptual similarity of (a) and (b), while the three bin-to-bin distance functions, L_1, L_2 and χ^2, falsely state that (b) is more similar to (c) than to (a). Despite this favorable robustness property, EMD has seldom been applied to general histogram-based descriptors (especially local descriptors) to the best of our knowledge. The main reason lies in its expensive computational cost, which is larger than $O(N^3)$ (super-cubic[2]) for a histogram with N bins.

Targeting this problem, our contribution is twofold. First, we propose two new fast

[2] By super-cubic, we mean a complexity in $\Omega(N^3) \cap O(N^4)$.

cross-bin algorithms, *EMD-L_1* and *diffusion distance*.

- **EMD-L_1**. The formulation of EMD-L_1 is much simpler than the original EMD formulation. It has only $O(N)$ unknown variables, which is significantly less than the $O(N^2)$ variables required in the original EMD. Furthermore, EMD-L_1 has only half the number of constraints and a more concise objective function. We prove that EMD-L_1 is formally equivalent to the original EMD with L_1 ground distance. As an optimization solver for EMD-L_1 computation, we designed an efficient tree-based algorithm. The new algorithm greatly improves the efficiency of the original transportation simplex algorithm as used in [128]. An empirical study demonstrates that the time complexity of EMD-L_1 is around $O(N^2)$, which is much faster than the previous super-cubic algorithm.

- **Diffusion Distance**. The diffusion distance models the difference between two histograms as a temperature field and considers the diffusion process on the field. Then, the integration of a norm on the diffusion field over time is used as a dissimilarity measure between the histograms. For computational efficiency, a Gaussian pyramid is used to discretize the continuous diffusion process. The diffusion distance is then defined as the sum of norms over all pyramid layers. The new distance allows cross-bin comparison. This makes it robust to distortions such as deformation, lighting change and noise that often causes problems for HBLDs. Due to the exponentially decreasing layer sizes in the Gaussian pyramid, the new approach has a linear time complexity, which is much faster than previously used cross-bin distances with quadratic complexity or higher.

Second, for the first time, the cross-bin distances are successfully applied to compare HBLDs and evaluated in comparison to other bin-to-bin distances. The speedup gained by the two proposed approaches enables them to be applied directly to multi-dimensional histograms without reducing the discriminability by introducing approximation. We tested the proposed approach in two tasks (shape matching and interest point matching) with three different state-of-the-art HBLDs, shape context [15], SIFT [96], and spin images [65, 80]. The experiments are conducted on both synthetic and real image pairs, under significant geometrical deformation, lighting change, and intensity noise. In all experiments, the proposed approaches performs excellently, as do other cross-bin distances, while running much faster.

The rest of this part is organized as follows. Sec. 5.1 discusses related works. After that, Chapter 6 proposes two approaches for histogram comparison. In Section 6.1, we first review the original Earth Mover's Distance and present its formulation for histograms. Then we briefly introduce the proposed EMD-L_1. After that, Section 6.2 presents the proposed diffusion distance and discusses its relationship to EMD and previously proposed pyramid-based approaches. Section 6.3 describes experiments comparing the diffusion distance to other methods (including EMD-L_1) on shape matching and interest point matching tasks. Section 6.4 concludes the chapter.

Part of this work appears in [94, 93, 92].

5.1 Related Work

Dis/similarity measures between histograms can be categorized into bin-to-bin and cross-bin distances. Our approach falls into the latter category. Early works using cross-bin matching costs for histogram comparison can be found in [137], [156] and [116]. Particularly, in Peleg et al. [116], images are modeled as sets of pebbles after normalization. The similarity between two images is the minimum cost required to move one set of pebbles to match the other. Another cross-bin distance is the *quadratic-form distance* [112, 53]. Quadratic-form distance is another cross-bin distance. It allows comparison of histograms across different bin locations whose connectivity is heuristically determined by a quadratic-form. An evaluation of different histogram dissimilarity measures for texture retrieval (color based) can be found in [126]. Recently, Domke and Aloimonos [35] developed an approach to create deformation and viewpoint invariant color histograms. The idea is to use unequal pixel weights that are extracted from gradients of different color channels to cancel the changes induced by deformation. In the following, we discuss the cross-bin distances that are most related to our study.

Adapted from previous work, the Earth Mover's Distance (EMD) is proposed by Rubner et al. [128] and Rubner and Tomasi [127] to compare distributions for image retrieval tasks. By modelling distribution comparison as a transportation problem [57] (a.k.a. the Monge-Kantorovich problem [122]), a specialized efficient linear programming algorithm, the *transportation simplex* (TS) algorithm [56] is proposed to solve the EMD. It is shown in [128] that TS has a super-cubic empirical time complexity. In [128], EMD is applied to signatures of distributions instead of directly to histograms. Signatures are abstracted representations of distributions and are usually clustered versions of histograms. This approach is very efficient and effective for distributions with sparse structures, e.g., the color histograms in the CIE-Lab space [128]. However, for histogram-based local descriptors that are not sparse in general, e.g. SIFT [96], EMD should be applied to histograms directly. In a typical setting to solve real vision problems, the number of required comparisons between these descriptors is very large, which forbids the use of the original TS algorithm. For example, to compare two images with 300 local features each, 90,000 comparisons are needed! Note that, although there is a fast exact EMD algorithm for 1D histograms [116], such a solution does not scale to higher dimensions - while most of the histogram-based local descriptors have two or three dimensions.

Since it was initially proposed by Rubner et al. [128], EMD has attracted a large amount of research interest. Here we briefly summarize some examples. Cohen and Guibas [29] studied the problem of computing a transformation between distributions with minimum EMD. Levina and Bickel [84] proved that EMD is equivalent to the Mallows distance [98] when applied to probability distributions. Tan and Ngo [144] applied EMD to common pattern discovery using EMD's partial matching ability. In addition, Indyk and Thaper [61] proposed a fast approximation EMD algorithm and it is used for image retrieval [61] and shape matching [48]. The L_1 formulation had been introduced by Wesolowsky [157] and then Cohen and Guibas [29]. In this paper we extend it to general multi-dimensional histograms. In addition, Holmes et al. [60, 59] touched on several areas explored in the paper, including EMD approximations in a Euclidean space for classes of derivative histograms and partial matching.

The fast algorithm proposed by Indyk and Thaper [61] is through embedding the

EMD metric into a Euclidean space. The approach first embeds EMD between point sets into a L_1 space. This is done via some hierarchical distribution analysis. Then fast nearest neighbor retrieval is achieved via Locality-Sensitive Hashing (LSH). The EMD can then be approximated by the L_1 distance in the Euclidean space. Grauman and Darrell [48] extended this approach for fast contour matching. For this purpose, a shape is treated as a set of features on the contours, where each feature is treated as a point in the feature space. The time complexity of these algorithms are $O(Nd \log \Delta)$, where N is the number of points, d is the dimension of the feature space, and Δ is the diameter of the union of the two feature sets to be compared. These approaches are very efficient for retrieval tasks and global shape comparison [61, 48]. However, the approximation due to the embedding may sacrifice precision, reducing the discriminability of descriptors. As indicated in [61], the distortion upper bound is $O(\log \Delta)$ and empirical distortion is about 10%. In addition, these approaches focused on point set matching rather than the histogram comparison in which we are interested. Recently, Grauman and Darrell [49] proposed *pyramid matching kernel* (PMK) for feature set matching. PMK can be viewed as a further extension of the fast EMD embedding in that it also compares the two distributions in a hierarchical fashion. PMK also handles partial matching through histogram intersections [143]. PMK is very similar to the proposed diffusion-based distance. The main differences lie in the motivation and the application, as will be clarified in Chapter II.

Unlike the above previous work, we focus on designing a distance metric for the histogram-based local descriptors, which have attracted a lot of research interest recently [15, 110, 145, 147, 91, 96, 71, 111, 104]. Three representative examples are chosen in our experiments. First, the shape context introduced by Belongie et al. [15] captures the distribution of landmark points. It is demonstrated to be very discriminative for shape matching. Some extensions of the shape context can be found in [110, 145, 147, 91]. The second is the scale invariant feature transform (SIFT) proposed by Lowe [96], which is a three-dimensional histogram measuring local gradient distributions. SIFT and its extensions are widely used for image matching and retrieval, e.g. [96, 71, 111, 104]. The third one is the spin image that basically computes the joint distribution of the intensity and distance of pixels around given interest points. It was first proposed by Johnson and Hebert [65] for 3D object recognition and later extended to a 2D texture descriptor by Lazebnik et al. [80]. A review of other descriptors and their performance evaluation can be found in [104]. Previously, these histogram-based local descriptors are compared by bin-to-bin metrics, especially the χ^2 distance and the L_p norms (e.g., Euclidean distance, Manhattan distance). In this paper, we will show that the proposed EMD comparison achieves better performance, especially for tasks involving large distortions including geometric deformation, illumination change and heavy intensity noise.

Our work differs from the above works in several ways. First, we model the similarity between histograms with a diffusion process. Second, we focus on comparing histogram-based local descriptors such as shape context [15] and SIFT [96], while the above works focus on feature distributions in the image domain. The difference between the proposed approach and the pyramid matching kernel in [49] is studied in Sec. 6.2.

Previously, we proposed a fast EMD algorithm, EMD-L_1 [94], for histogram comparison. EMD-L_1 utilizes the special structure of the L_1 ground distance on histograms for a fast implementation of EMD. Therefore it still solves the transportation problem, which is fundamentally different from the motivation of this chapter. The diffusion distance is much faster than EMD-L_1 and performs similarly in the case of large deformations. However, in a preliminary experiment with only small quantization errors, EMD-L_1 performed better than the diffusion distance. More comprehensive comparisons between them remains as an interesting future work.

The diffusion process has widely been used for the purpose of data smoothing and scale-space analysis in the computer vision community. Some earlier work introducing this idea can be found in [158, 74]. These works axiomatically demonstrated that a PDE model of the linear heat dissipation or diffusion process has Gaussian convolution as a unique solution. More recent well-known diffusion-based methods include anisotropic diffusion for edge-preseving data smoothing [117] and automatic scale selection with γ-normalized Laplacian [88]. It also provides a theoretical foundation to other vision techniques such as Gaussian pyramids and the SIFT feature detector [96], anisotropic bandwidth selection [113]. Despite its ubiquitousness, to the best of our knowledge, this is the first attempt to exploit the diffusion process to compute a histogram distance. In addition, Berg and Malik [16] used diffusion to alleviate shape deformation in template matching.

Chapter 6

Robust Histogram Comparison

6.1 EMD-L_1 for Histogram Comparison

6.1.1 The Original EMD between Signatures

The Earth Mover's Distance (EMD) is proposed by Rubner et al. [128] to measure the dissimilarity between signatures that are compact representations of distributions. A signature of size N is defined as a set $S = \{s_j = (w_j, m_j)\}_{j=1}^{N}$, where m_j is the position of the j-th element and w_j is its weight.

Given two signatures $P = \{(p_i, u_i)\}_{i=1}^{m}$ and $Q = \{(q_j, v_j)\}_{j=1}^{n}$ with size m, n respectively, the EMD between them is modeled as a solution to a transportation problem. Treat elements in P as "supplies" located at u_i and elements in Q as "demands" at v_j. Then p_i and q_j indicates the amount of supply and demand respectively. The EMD is defined as the minimum (normalized) work required for resolving the supply-demand transports, i.e.

$$EMD(P, Q) = \min_{F=\{f_{ij}\}} \frac{\sum_{i,j} f_{ij} d_{ij}}{\sum_{i,j} f_{ij}}$$

with the following constraints:

$$\sum_j f_{ij} \le p_i \ , \quad \sum_i f_{ij} \le q_j \ , \quad \sum_{i,j} f_{ij} = \min\{\sum_i p_i, \sum_j q_j\} \ , \quad f_{ij} \ge 0 \ ,$$

where $F = \{f_{ij}\}$ denotes a set of *flows*. Each flow f_{ij} represents the amount transported from the i-th supply to the j-th demand. We call d_{ij} the *ground distance* between the position u_i and v_j. Fig. 6.1 gives an example, where P has four elements and Q has three.

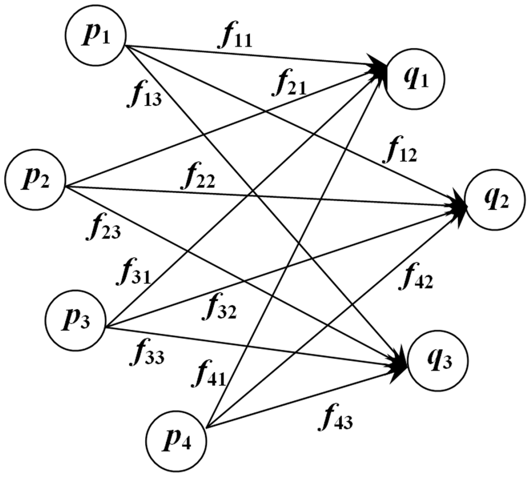


Figure 6.1: EMD between two signatures ($m = 4$, $n = 3$) as a transportation problem.

The transportation problem is a special case of linear programming (LP) problems. The constraint matrix in this case has a very sparse structure that enables an efficient algorithmic solution. One such efficient algorithm is the transportation simplex (TS) [128, 56]. Modified from the standard simplex algorithm, TS greatly reduces the number of operations to maintain the constraint matrix by taking advantage of its special structure. The empirical study in [128] shows that the time complexity is super-cubic for signatures with size N. Other possible solutions mentioned in [128] include interior-point algorithms [70] and incapacitated minimum network flow [9] that have similar time complexities.

6.1.2 The EMD between Histograms

Histograms can be viewed as a special type of signatures in that each histogram bin corresponds to an element in a signature. In this view, the histogram values are treated as the weights w_j in a signature S, and the grid locations (indices of bins) are treated as positions m_j in S.

In the following we assume two dimensional histograms for illustrative simplicity. They are widely used for shape and image descriptors and derivations for higher dimensional cases are straightforward. Without loss of generality, we use the following assumptions and notations.

- Histograms have m rows and n columns and $N = m \times n$ bins.
- The index set for bins is defined as $\mathcal{I} = \{(i,j) : 1{\le}i{\le}m, 1{\le}j{\le}n\}$. We use (i,j) to denote a bin or a node corresponding to it.
- The index set for flows is defined as $\mathcal{J} = \{(i,j,k,l) : (i,j) \in \mathcal{I}, (k,l) \in \mathcal{I}\}$.
- $P = \{p_{ij} : (i,j) \in \mathcal{I}\}$ and $Q = \{q_{ij} : (i,j) \in \mathcal{I}\}$ are the two histograms to be compared.
- Histograms are normalized to a unit mass, i.e., $\sum_{i,j} p_{ij} = 1$, $\sum_{i,j} q_{ij} = 1$. As will be clear later, the normalization is not essential for the algorithm we will propose.
- The bin sizes in both dimensions are equal. Without loss of generality, each bin is assumed to be a unit square.

With these notations and assumptions, we obtain the following new definition of

EMD between two histograms P and Q

$$EMD(P,Q) = \min_{F=\{f_{i,j;k,l}:(i,j,k,l)\in\mathcal{J}\}} \sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} \tag{6.1}$$

$$\text{s.t.} \begin{cases} \sum_{(k,l)\in\mathcal{I}} f_{i,j;k,l} & = & p_{ij} & \forall (i,j)\in\mathcal{I} \\ \sum_{(i,j)\in\mathcal{I}} f_{i,j;k,l} & = & q_{kl} & \forall (k,l)\in\mathcal{I} \\ f_{i,j;k,l} & \geq & 0 & \forall (i,j,k,l)\in\mathcal{J} \end{cases} \tag{6.2}$$

where F is a flow from P to Q and $f_{i,j;k,l}$ denotes a flow from bin (i,j) to (k,l). Note that we use the term “flow” to indicate both the set of flows in a graph and a single flow between two nodes, when there is no confusion. A flow F satisfying (6.2) is called *feasible*.

The ground distance $d_{i,j;k,l}$ is commonly defined by L_p distance

$$d_{i,j;k,l} = \|(i,j)^\top - (k,l)^\top\|_p = (|i-k|^p + |j-l|^p)^{1/p} \tag{6.3}$$

For example, the original EMD proposed by Rubner et al. [128] employed the L_1 (for texture) and L_2 (for color) ground distances.

6.1.3 EMD-L_1

Despite its good performance, EMD is known to suffer from efficiency problems. The transportation simplex algorithm has a super-cubic complexity. Recently, we proposed a fast EMD algorithm, EMD-L_1 [94], for histogram comparison. EMD-L_1 utilizes the special structure of the L_1 ground distance on histograms for a fast implementation of EMD. In this section we briefly introduce EMD-L_1, details of related algorithms can be found in [92].

Formulation of EMD-L_1

The robustness and efficiency of the L_1 norm often makes it preferable to the L_2 norm in computer vision and related areas, such as low-level vision learning [44], stereo analysis [66, 33], 1-norm support vector machine [167], etc. In addition, the L_1 and L_2 norms often perform similarly for image retrieval tasks [11]. Inspired by this evidence, we choose L_1 as EMD's ground distance. In the rest of the paper, unless indicated otherwise, the L_1 ground distance is implicitly assumed when talking about EMD. With the L_1 ground distance, formula (6.3) becomes

$$d_{i,j;k,l} = |i-k| + |j-l|$$

Note that the ground distance takes only integer values now. For illustrative purpose, the flow index set $\mathcal{J}$ is divided into three disjoint subsets $\mathcal{J} = \mathcal{J}_0 \bigcup \mathcal{J}_1 \bigcup \mathcal{J}_2$, each of which corresponds to one of the following types of flows.

- $\mathcal{J}_0 = \{(i,j,i,j) : (i,j) \in \mathcal{I}\}$ is for flows between bins at the same location. We call this kind of flows *self-flows* or *s-flows* for short.

- $\mathcal{J}_1 = \{(i,j,k,l) : (i,j,k,l) \in \mathcal{J}, d_{i,j;k,l} = 1\}$ is for flows between neighbor bins. We call this kind of flows *n-flows*.

- $\mathcal{J}_2 = \{(i,j,k,l) : (i,j,k,l) \in \mathcal{J}, d_{i,j;k,l} > 1\}$ is for other flows which are called *f-flows* because of their *far* distances.

An important property of the L_1 ground distance is that every positive f-flow can be replaced by a sequence of n-flows. This is because L_1 distance forms a shortest path system on the integer lattice. For example, given an f-flow $f_{i,j;k,l}$, $i{\le}k, j{\le}l$, the L_1 ground distance has the following decomposition

$$d_{i,j;k,l} = d_{i,j;i,l} + d_{i,l;k,l} = \sum_{j\le x<l} d_{i,x;i,x+1} + \sum_{i\le y<k} d_{y,l;y+1,l} \tag{6.4}$$

In another words, any L_1 shortest path from (i,j) to (k,l) can be decomposed into a sum of edges with ground distance one. It follows that, without changing the total weighted flow $\sum_{f\in F} fd$, the f-flow $f_{i,j;k,l}$ can be removed by increasing all n-flows along the path $[(i,j),(i,j+1),\dots,(i,l),(i+1,l),\dots,(k,l)]$ with $f_{i,j;k,l}$. This is illustrated in Fig. 6.2

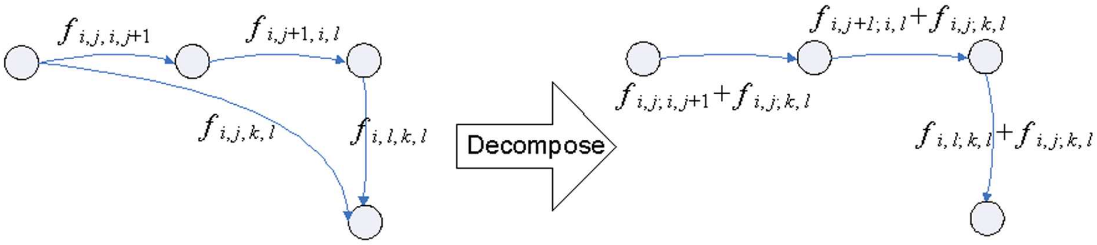

$f_{i,j,i,j+1}$
$f_{i,j+1,i,l}$
$f_{i,j,k,l}$
$f_{i,l,k,l}$
Decompose
$f_{i,j+l;i,l}+f_{i,j;k,l}$
$f_{i,j;i,j+1}+f_{i,j;k,l}$
$f_{i,l;k,l}+f_{i,j;k,l}$

Figure 6.2: Decompose an f-flow $f_{i,j;k,l}$, $k = i+1, l = j+2$. Only flows involved in decomposition are shown.

In addition to f-flows, s-flows can also be removed due to the zero ground distances associated with them while maintaining the total weighted flow. With these intuitions, we propose *EMD-*L_1: a new simplified formulation of EMD that only uses n-flows

$$EMD\text{-}L_1(P,Q) = \min_{G=\{g_{i,j;k,l}:(i,j,k,l)\in\mathcal{J}_1\}} \sum_{\mathcal{J}_1} g_{i,j;k,l} \tag{6.5}$$

$$\text{s.t.} \left\{ \begin{array}{rcll} \sum_{k,l:(i,j,k,l)\in\mathcal{J}_1} (g_{i,j;k,l} - g_{k,l;i,j}) & = & b_{ij} & \forall (i,j)\in\mathcal{I} \\ g_{i,j;k,l} & \ge & 0 & \forall (i,j,k,l)\in\mathcal{J}_1 \end{array} \right. \tag{6.6}$$

where $b_{ij} = p_{ij} - q_{ij}$ is the difference between the two histograms at a bin (i,j). We call a flow G satisfying (6.6) a *feasible* flow, analogous to that in the original EMD.

EMD-L_1 is largely simplified compared to the original EMD (6.1) and (6.2). The specific simplifications include

1. There are only $4N$ variables in (6.5), one order of magnitude less than that in (6.1). This is critical for speedup since the number of variables is a dominant factor in the time complexity of all LP algorithms. In addition, the memory efficiency gained by this is very favorable for large histograms.

2. The number of equality constraints is reduced by half. This is another important factor for deriving an efficient LP algorithm.

3. All the ground distances involved in the EMD-L_1 become ones. This is practically useful, because it removes all the distance computation and thus each flow g is equivalent to the corresponding weighted flow gd. It also allows the use of integer operations to handle the coefficients.

One important property of EMD-L_1 is that it is equivalent to the original EMD with a L_1 ground distance. The equivalence here is in the sense of the weighted total flows. For example, a flow G for EMD-L_1 and a flow F in the original EMD is said to be equivalent if $\sum_{\mathcal{J}_1} g_{i,j;k,l} = \sum_{\mathcal{J}} d_{i,j;k,l} f_{i,j;k,l}$, i.e., they have same total weighted flow. The following proposition states the equivalence in which we are interested.

Proposition Given two histograms P and Q as defined above

$$EMD(P, Q) = EMD\text{-}L_1(P, Q)\ . \tag{6.7}$$

Intuitively, the discussion in previous paragraphs suggests that, for any flow F for the original EMD, an equivalent flow G for EMD-L_1 can be created by eliminating all f-flows in F by using the decomposition and removing s-flows. This implies $EMD(P,Q) \geq EMD\text{-}L_1(P,Q)$. Now we need to verify the other direction. Given a flow G for EMD-L_1, find an equivalent F for the original EMD. The key issue is how to satisfy the constraints (6.2) in the original EMD. To do this, we introduce a "merge" procedure. The idea is to merge input and output flows at each bin so that either input or output flows disappear as a result. Notice that, for this proof, we only need an F to have a total weight not greater than that of G. This makes the proof with the merge procedure much simpler, allowing us to merge any pair of input and output flows. We left the details of the proof in [92].

Network Flow Formulation of EMD-L_1 and A Fast Solution

EMD-L_1 can be interpreted as a network flow model illustrated in Fig. 6.3. In the model, each bin (i,j) is treated as a node with weight b_{ij}, and eight flow edges (as shown in Fig. 6.3) between the node and its four neighbors. The total weight of the nodes is 0 (i.e. $\sum_{\mathcal{I}} b_{ij} = 0)$. The task is to redistribute the weights via the flows to make all weights vanish. In this interpretation, EMD-L_1 is given by a solution with the minimum total flow.

To compute EMD-L_1 between histograms is equivalent to solving the linear programming (LP) problem in (6.5) and (6.6). We designed a tree-based algorithm, Tree-EMD, as an efficient discrete optimization solver, which extends the original simplex algorithm. The tree-based algorithm is significantly faster than the original simplex, and has a more intuitive interpretation as a network flow problem. As a reference, we will first briefly describe the standard simplex applied to EMD-L_1. After that, an extended trans-

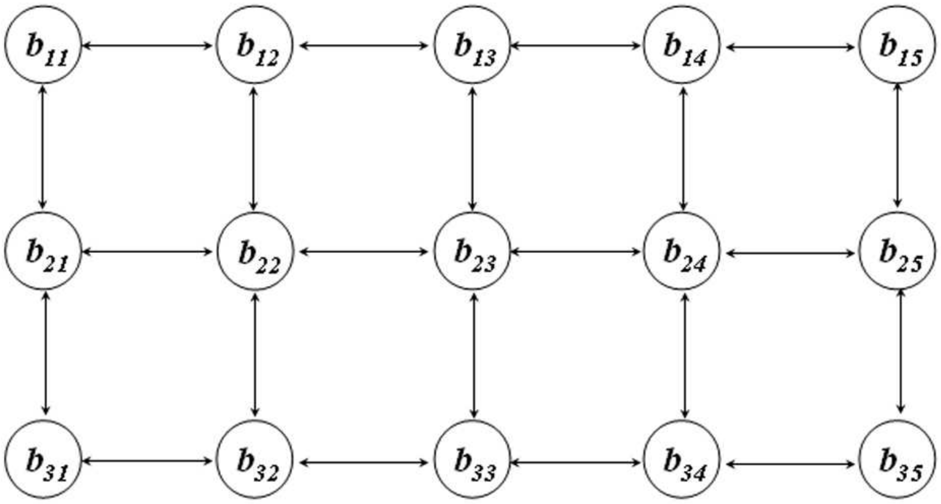


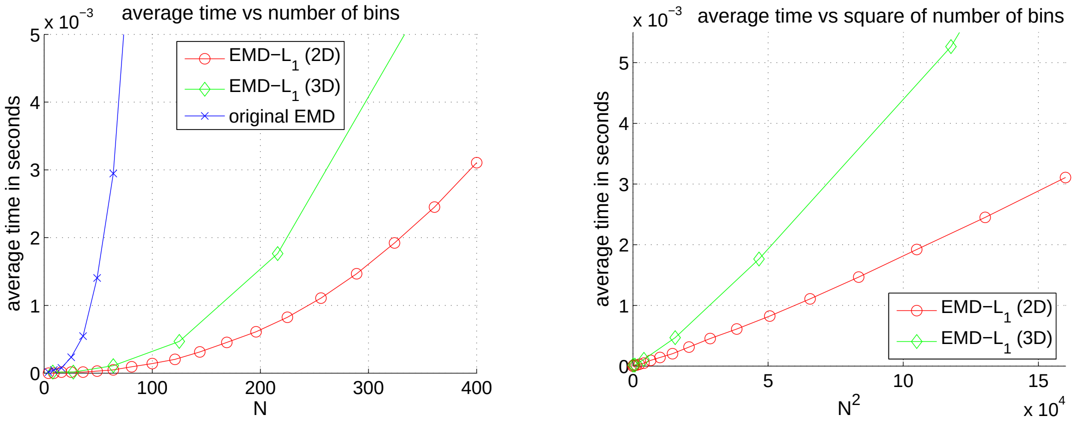
Figure 6.3: The EMD-L_1 as a network flow problem for 3×5 histograms.

portation simplex algorithm for EMD-L_1 is designed based on the original transportation simplex [56] used in [128]. Finally, the tree-based algorithm is derived by further extending the fast simplex.

In the following we will briefly describe the basic idea of Tree-EMD. First, EMD-L_1 is modeled as a minimum flow problem as illustrated above. The trick lies in that there exists at least a *spanning tree* (may not be unique) that achieves the minimum. Therefore, given an initial (non-optimal) solution, it can be iteratively improved till the optimum is reached. Derived from the transportation simplex algorithm, Tree-EMD is much faster due to the efficiency provided by using a tree structure. The details of the algorithm, including its relation to simplex, can be found in [92].

6.1.4 Empirical Study for Time Complexity

The simplex algorithm is known to have good empirical time complexity but poor worst case time complexity. Therefore, to evaluate the time complexity of the proposed algorithm, we conduct an empirical study similar to that in [128]. First, two sets of 2D random histograms are generated for sizes: $n \times n$, $2 \leq n \leq 20$. For each n, 1000 random histograms are generated for each set (i.e. 2000 for all). Then, the two sets are paired and the average time to compute EMD for each size n is recorded. We compare EMD-L_1 (with Tree-EMD) and the original EMD (with TS algorithm[1]). In addition, EMD-L_1 is tested for 3D histograms with similar settings, except using $2 \leq n \leq 8$. In summary, three algorithms are compared: EMD-L1 for 2D, EMD-L1 for 3D, and the original EMD. The results are shown in Fig. 6.4. From (a) it is clear that EMD-L_1 is much faster than the original one. Fig. 6.4 (b) shows that EMD-L_1 has a complexity of $O(N^2)$, where N is the number of bins (n^2 for 2D and n^3 for 3D). Furthermore, in our image feature matching experiments (Sec. 6.3.2), EMD-L_1 shows similar running time as the quadratic form distance (see Table 6.3), which has a quadratic time complexity.



(a) (b)

Figure 6.4: Empirical time complexity study of EMD-L_1 (Tree-EMD). (a) In comparison to the original EMD (TS Algorithm). (b) Average running time vs. square of histogram sizes.

[1] With Rubner's code, http://ai.stanford.edu/∼rubner/emd/default.htm

In addition to the above experiment, we also compared Tree-EMD and ETS in a pilot experiments for 2D histograms with 80 bins. We observed that Tree-EMD is roughly six times faster than the ETS algorithm.

By far EMD-L_1 has been shown to be more efficient than the original EMD. However, for sparse histograms, especially in high-dimensional spaces, the original EMD might have an advantage as it uses signatures that can compactly represent the sparse spaces with a relatively low number of features (bins).

6.2 The Diffusion Distance Between Histograms

6.2.1 Motivation

In last section EMD-L_1 is proposed as a fast algorithm to compute the EMD with a L_1 ground distance. Note that the network flow formulation is actually not limited to the L_1 ground distance if we allow flows between all pairs of nodes. This implies, in general, the EMD between histograms can be viewed as the minimum flow that makes the weights vanish at all nodes. Naturally, a question is raised: is the minimum flow the best way to measure histogram dissimilarity? Or, in other words, are there other effective or efficient ways for weight exchanging?

Motivated by the questions above, we propose use the diffusion process to model the weight exchanging and designed a simple while efficient solution. Intuitively, instead of using the minimum flow, we can exchange weights greedily to make all weights vanish. It can be iteratively done until convergence. This procedure follows a diffusion process. Given two histograms $h_1,\ h_2$ and their difference $d = h_1 - h_2$, d can be treated as initial temperatures of a temperature field T. Then, without any external heat exchanging, T diffuses according to time. The diffusion process naturally redistribute the weights so that every weight vanishes ($T \to 0$) in the limit.

The rest of this section is organized as follows. Section 6.2.2 presents the diffusion model for histogram differences. Then, Section 6.2.3 discusses the relationship between the proposed distance and the EMD. After that, Section 6.2.4 proposed a pyramid-based approximation of the proposed distance measure. Then, Section 6.2.5 describes the difference between our approach to the pyramid matching kernel [49].

6.2.2 Modelling Histogram Difference with a Diffusion Process

Let us first consider 1D distributions $h_1(x)$ and $h_2(x)$. It is natural to compare them by their difference, denoted as $d(x) = h_1(x) - h_2(x)$. Instead of putting a metric on d directly, we treat it as an isolated temperature field $T(x,t)$ at time $t = 0$, i.e. $T(x,0) = d(x)$. It is well known that the temperature in an isolated field obeys the heat diffusion equation

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} \tag{6.8}$$

It has a unique solution

$$T(x,t) = T_0(x) * \phi(x,t) \tag{6.9}$$

given initial condition $T_0(x)$

$$T(x,0) = T_0(x) \doteq d(x) \tag{6.10}$$

where $\phi(x,t)$ is the Gaussian filter

$$\phi(x,t) = \frac{1}{(2\pi t)^{1/2}} \exp\{-\frac{x^2}{2t}\} \tag{6.11}$$

Note that the mean of the difference field is zero, therefore $T(x,t)$ becomes zero everywhere when t increases. In this sense, $T(x,t)$ can be viewed as a process of histogram value exchange which makes h_1 and h_2 equivalent. Intuitively, the process *diffuses* the difference between two histograms, therefore a dissimilarity can be extracted by measuring the process. A distance between h_1 and h_2 is defined as

$$\widehat{K}(h_1,h_2) = \int_0^{\bar{t}} k(|T(x,t)|)dt \tag{6.12}$$

where $\bar{t}$ is a positive constant upper bound of the integration, which can be ∞ as long as the integration converges. $k(.)$ is a norm that measures how $T(x,t)$ differs from 0. In this chapter, we use the L_1 norm because of its computational simplicity and good performance in our pilot studies.

Next we will show how $\widehat{K}$ handles deformation with a simple 1D example.

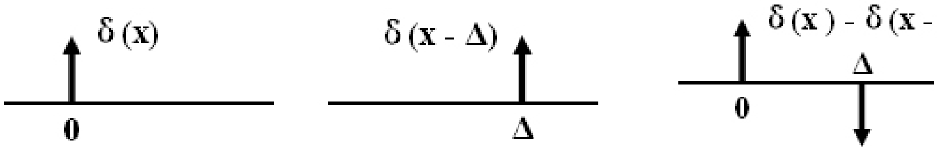
Assume a simple case where $h_1(x) = \delta(x)$ and $h_2(x) = \delta(x - \Delta)$, as shown in Fig. 6.5 (a) and (b). This means the histogram is shifted by $\Delta \geq 0$. The initial value of $T(x,t)$ is therefore $T_0 = \delta(x) - \delta(x - \Delta)$, as shown in Fig. 6.5 (c). The diffusion process becomes

$$\begin{aligned} T(x,t) &= (\delta(x) - \delta(x-\Delta)) * \phi(x,t) \\ &= \phi(x,t) - \phi(x-\Delta,t) \end{aligned} \tag{6.13}$$

Use the L_1 norm for $k(.)$,

$$\begin{aligned} k(|T(x,t)|) &= \int_{-\infty}^{\infty} |\phi(x,t) - \phi(x-\Delta,t)| dx \\ &= 2\int_{-\infty}^{\Delta/2} (|\phi(x,t) - \phi(x-\Delta,t)|) dx \\ &= 2\left(\int_{-\infty}^{\Delta/2} \phi(x,t)dx - \int_{-\infty}^{-\Delta/2} \phi(x,t)dx\right) \\ &= 2\left(2\int_{-\infty}^{\Delta/2} \phi(x,t)dx - 1\right) \qquad (6.14) \end{aligned}$$

From (6.12) and (6.14), it is clear that $k(.)$ and $\widehat{K}$ are monotonically increasing with Δ. This suggests that $\widehat{K}$ indeed measures the degree of deformation between two histograms.



(a) (b) (c)

Figure 6.5: Two histograms with shift Δ between them and their difference. (a) h_1. (b) h_2. (c) $d = h_1 - h_2$.

6.2.3 Relation to the Earth Mover's Distance

From the above discussion, it is clear that $\widehat{K}$ is a cross-bin distance, which allows comparison between bins at different locations. In this subsection we will discuss its relation with EMD [128], which is another effective cross-bin histogram distance.

Given two histograms h_1 and h_2, EMD models h_1 as a set of supplies and h_2 as a set of demands. The minimum *work* to transport all supplies to demands is used as the distance between h_1 and h_2. In other word, EMD measures the dissimilarity between histograms with a transportation problem [128].

Note that bins of h_1 and h_2 share same lattice locations, which means that it takes zero work to transport supplies from a bin in h_1 to the same bin in h_2. This leads to an intuitive interpretation of EMD with the difference $d = h_1 - h_2$: EMD is the minimum work of exchanging values in d to make d vanish everywhere.

This provides an intuition about the difference between EMD and $\widehat{K}$. EMD seeks the exchanging scheme which has the minimum work, while $\widehat{K}$ measures a more "natural" exchanging scheme, i.e. diffusion process. While EMD has been successfully applied to several vision tasks (e.g. [128, 48]), the diffusion-based distances have not been evaluated with any vision tasks. Our conjecture is that they may fit to different tasks. In our experiments (see Sec. 6.3) on the HBLDs suffering large deformation, both approaches perform quite similarly. Below we demonstrate an example, in which $\widehat{K}$ performs better than EMD.

Consider three one-dimensional histograms h_1, h_2 and h_3 as illustrated in the left of Fig. 6.6. h_2 is shifted from h_1 by Δ, while h_3 can not be linearly transformed from h_1. We want to compare h_1 to h_2 and h_3. Subtracting h_2 and h_3 from h_1, we get the differences d_{12}, d_{13} as shown in the right of Fig. 6.6. It is clear that the EMD between h_1 and h_2 are the same as the EMD between h_1 and h_3. Perceptually, however, h_1 seems to be more similar to h_2 than to h_3.

Fig. 6.7 shows the diffusion process $T(x,t)$ at $t = 0, 6, 12$. From the figure we see that $k(|T(x,t)|)$ for h_1 and h_2 is always smaller than that for h_1 and h_3. Therefore, $\widehat{K}(h_1, h_2) < \widehat{K}(h_1, h_3)$. This is more consistent with our perception.

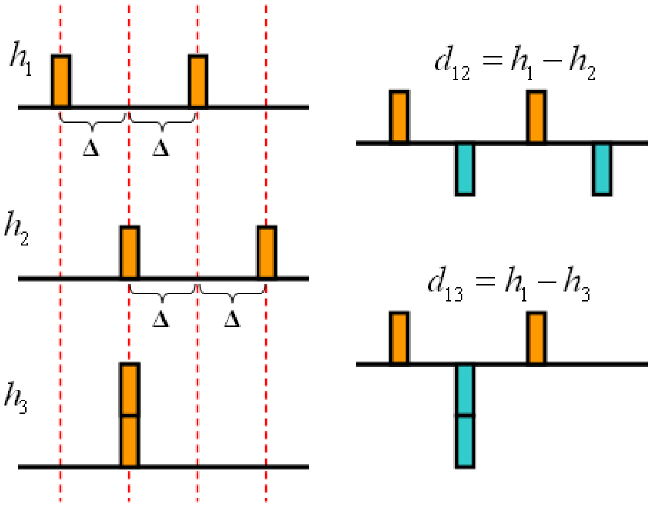


Figure 6.6: Left: Three 1D histograms. Right: The differences between them.

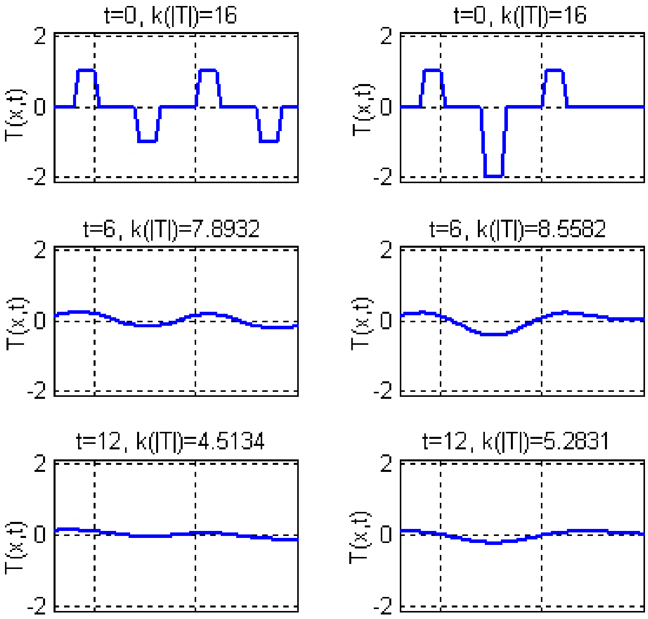


Figure 6.7: The diffusion process of the difference d_{12} (left column) and d_{13} (right column). Each row shows the diffusion result at a different time t. $k(|T|)$ is measured using the L_1 norm; the values show that d_{12} decays faster than d_{13}.

6.2.4 Diffusion Distance

It is straightforward to extend previous discussions to higher dimensions. Consider two m-dimensional histograms $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^m$ is a vector. The definition of

$\widehat{K}(h_1, h_2)$ is the same as in Sec. 6.2.2, except that equations (6.8) and (6.11) are replaced by (6.15) and (6.16), respectively.

$$\frac{\partial T}{\partial t} = \nabla^2 T \tag{6.15}$$

$$\phi(\mathbf{x}, t) = \frac{1}{(2\pi t)^{m/2}} \exp\{-\frac{\mathbf{x}^\top \mathbf{x}}{2t}\} \tag{6.16}$$

Now the problem is how to compute $\widehat{K}$. Direct computation of equation (6.14) is expensive. Instead, we use an alternative distance function based on the Gaussian pyramid. The Gaussian pyramid is a natural and efficient discritization of the continuous diffusion process $T(\mathbf{x}, t)$. It is justified because smoothing allows subsampling without aliasing. With this idea, we propose the *diffusion distance* $K(h_1, h_2)$ as

$$K(h_1, h_2) = \sum_{l=0}^{L} k(|d_l(\mathbf{x})|) \tag{6.17}$$

where

$$d_0(\mathbf{x}) = h_1(\mathbf{x}) - h_2(\mathbf{x}) \tag{6.18}$$

$$d_l(\mathbf{x}) = [d_{l-1}(\mathbf{x}) * \phi(\mathbf{x}, \sigma)] \downarrow_2 \quad l = 1, ..., L \tag{6.19}$$

are different layers of the pyramid. The notation "$\downarrow_2$" denotes half size downsampling. L is the number of pyramid layers and σ is the constant standard deviation for the Gaussian filter ϕ.

Note that as long as $k(.)$ is a metric, $K(h_1, h_2)$ forms a metric on histograms. In particular, in this chapter we choose $k(.)$ as the L_1 norm, which makes the diffusion distance a true metric. Equation (6.17) is then simplified as

$$K(h_1, h_2) = \sum_{l=0}^{L} |d_l(x)| \tag{6.20}$$

The computational complexity of $K(h_1, h_2)$ is O(N), where N is the number of hitogram bins. This can be easily derived by two facts. First, the size of d_l exponentially reduces. Second, only a small Gaussian filter ϕ is required which makes the convolution take time linear in the size of d_l for each scale l.

6.2.5 Relation to the Pyramid Matching Kernel

The diffusion distance (6.20) is similar to the pyramid matching kernel (PMK) recently proposed by Grauman and Darrell [49] in that both methods compare histograms by summing the distances over all pyramid layers.

As mentioned in the related work section, our approach focuses on histogram-based local descriptors, while PMK focuses on feature set matching. The two methods have the following differences.

First, when comparing each pyramid layer, PMK counts the number of newly matched feature pairs via the difference of histogram intersection [143]. This is particularly effective for handling occlusions for feature set matching. However, this is not an effective strategy for HBLDs because they are usually normalized. In contrast, we employ the L_1 norm to compare each pyramid layer.

Second, PMK uses varying weights for different scales by emphasizing finer scales more. This is reasonable for feature set matching as mentioned in [49]. However in the diffusion distance, uniform weights are used - this seems more natural and performs better than non-uniform weights in our preliminary experiments.

Third, the diffusion distance uses Gaussian smoothing before downsampling according to the underlying diffusion process.

Fourth, PMK requires random shifting when extracting histograms from feature sets to alleviate quantization effects. The proposed method avoids such a strategy by using the intuitive cross-bin referencing imposed by the diffusion.

6.3 Experiments

In this section the diffusion distance is tested for two kinds of vision tasks using HBLDs. The first experiment is for shape features, where the diffusion distance is used to compare shape context [15] in a data set with articulated objects. The second experiment is for interest point matching on a data set with synthetic deformation, illumination change and heavy noise. Both experiments demonstrate that the proposed method is robust for quantization problems.

6.3.1 Shape Matching with Shape Context

This subsection compares the diffusion distance for shape matching with shape context (SC) [15] and the inner-distance shape context (IDSC) [91]. Shape context is a shape descriptor that captures the spatial distribution of landmark points around every interest key point [15]. IDSC is an extension of SC using the shortest path distance instead of Euclidean distance. In [91], SC and IDSC are used for contour comparison with a dynamic programming (DP) scheme. We use the same framework, except for replacing the χ^2 distance with the diffusion distance and EMD (with Rubner's code[2]) for measuring dissimilarity between (inner-distance) shape contexts.

The experiment is conducted on an articulated shape database tested in [91]. The

[2] http://ai.stanford.edu/∼rubner/emd/default.htm

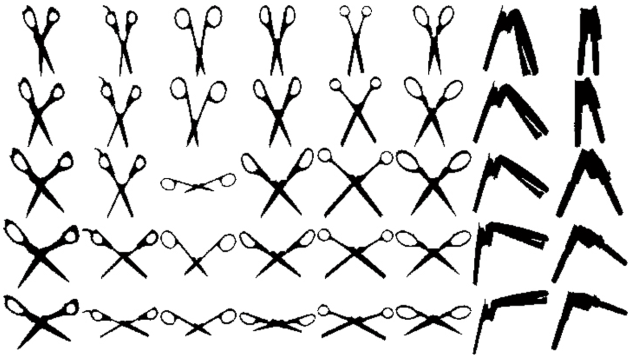


Figure 6.8: Articulated shape database. This dataset contains 40 images from 8 objects. Each column contains five images from the same object with different articulation.

database contains 40 images from 8 different objects. Each object has 5 images articulated to different degrees (see Figure 6.8). This data set is designed for testing articulation, which is a special and important case of deformation. [91] shows that the original shape context with χ^2 distance does not work well for these shapes. The reason is that the articulation causes a large deformation in the histogram.

We use exactly the same experimental setup as used in [91]: 200 points are sampled along the outer contours of every shape; 5 log-distance bins and 12 orientation bins are used for shape context histograms. The same dynamic programming matchings are used to compute distances between pairs of shapes. The recognition result is evaluated as following: For each image, the 4 most similar matches are chosen from other images in the dataset. The retrieval result is summarized as the number of 1st, 2nd, 3rd and 4th most similar matches that come from the correct object. Table 6.1 shows the retrieval results using the shape context. It demonstrates that the diffusion distance works much better than the χ^2 distance.

Table 6.2 shows the results for inner-distance shape context. In this case, though the

Table 6.1: Retrieval result on the articulated dataset with shape context [15]. The running time (in seconds) of using χ^2 was not reported in [91].

Distance	Top 1	Top 2	Top 3	Top 4	Time
χ^2 [91]	20/40	10/40	11/40	5/40	N/A
EMD [128]	37/40	33/40	24/40	16/40	1355s
Diffu. Dist.	34/40	27/40	19/40	14/40	67s

Table 6.2: Retrieval result on the articulated dataset with the inner-distance shape context [91]. The running time (in seconds) of using χ^2 was not reported in [91].

Distance	Top 1	Top 2	Top 3	Top 4	Time
χ^2 [91]	40/40	34/40	35/40	27/40	N/A
EMD [128]	39/40	38/40	26/40	28/40	1143s
Diffu. Dist.	40/40	36/40	37/40	23/40	68s

inner-distance is already insensitive to articulation, the diffusion distance still improves the result. From the tables we also see that the diffusion distance works similarly to EMD, while being more efficient.

6.3.2 Image Feature Matching

This subsection describes the experiment for interest point matching with several state-of-the-art image descriptors. The experiment was conducted on two image data sets. The first data set contains ten image pairs with synthetic deformation, noise and illumination change, see Fig. 6.9 for some examples. The second one contains six image pairs with real deformation and lighting changes, some of them are shown in Fig. 6.10[3]. The experimental configuration and results are described below.

Dissimilarity measures. We tested the diffusion distance along with several popular bin-to-bin distances, as well as cross-bin distances. The bin-to-bin distances include

[3] Two pairs of images with large lighting change are not shown here due to copyright issues. They are available at http://www.cs.umd.edu/˜hbling/Research/Publication/data/RD-cvpr06.zip.

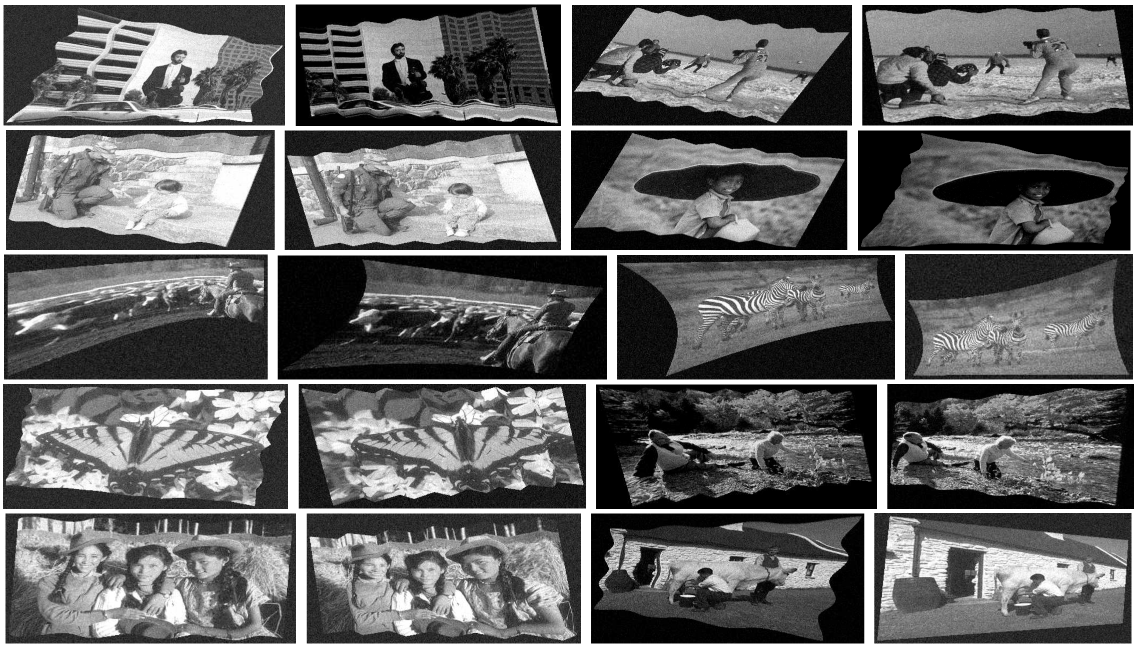


Figure 6.9: Synthetic image pairs with synthetic deformation, illumination change and noise.

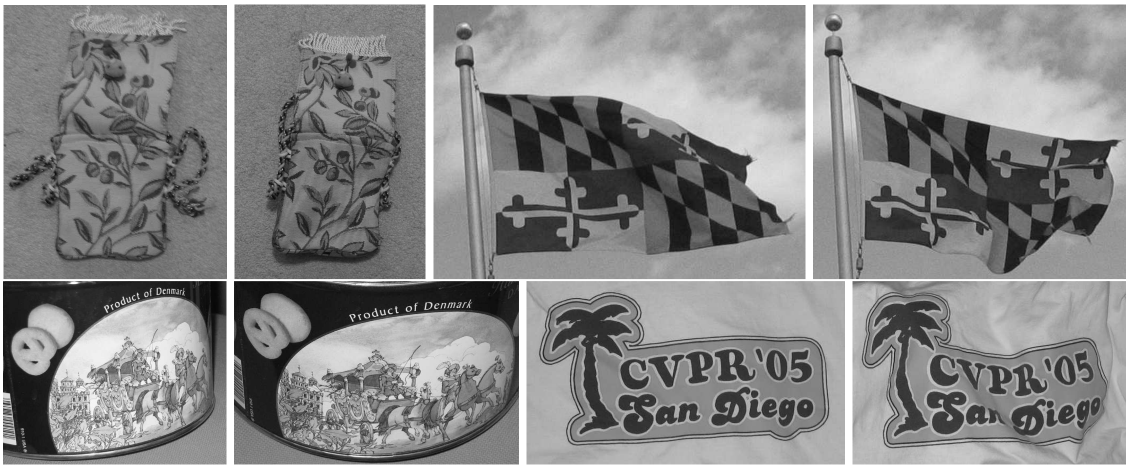


Figure 6.10: Four of the six image pairs containing real deformation and lighting change.

the χ^2 statistics, the symmetric Kullback-Leibler divergence (KL), symmetric Jensen-Shannon(JS) divergence [87], L_2 distance and Bhattacharyya distance (BT). Cross-bin distances include EMD, EMD-L_1 and quadratic-form(QF). For EMD, we use Rubner's online code with L_2 ground distance. The quadratic-form distance is implemented according to [128]. For the diffusion distance, we set the Gaussian standard deviation $\sigma = 0.5$ and use a window of size 3×3 ($3 \times 3 \times 3$ for 3D histograms). We did not compare with PMK [49] because it requires random shifting when building a initial histogram (zero-th layer) and it uses the intersection focusing on un-normalized histograms extracted from feature sets.

Interest point. We use Harris corners [55] for the matching experiments. The reason for this choice is that, due to the large deformation, noise and lighting change, it is hard to apply other interest point detectors. On the other hand, we focus more on comparing descriptors than the interest points. For the synthetic data set, we pick 200 points per image pair with the largest cornerness responses. To compute the descriptors, a circular support region around each interest point is used. The region diameter is 41 pixels, which is similar to the setting used in [104]).

Descriptors. We tested all the distances on three different histogram-based descriptors. The first one is SIFT proposed by [96]. It is a weighted three-dimensional histogram, 4 bins for each spatial dimensions and 8 bins for gradient orientation. The second one is the shape context [15]. The shape context for images is extracted as a two-dimensional histogram counting the local edge distribution in a similar way to [104]. In our experiment, we use 8 bins for distance and 16 bins for orientation. The third one is the spin image [80, 65] which measures the joint spatial and intensity distribution of pixels around interest points. We use 8 distance bins and 16 intensity bins.

Evaluation criterion. For each pair of images with their interest points, we first find the ground-truth correspondence. This is done automatically for the synthetic data set and manually for the real image pairs. Then, for efficiency we removed those points in Image 1 with no correct matches (this also makes the maximum detection rate to 1). After that, every interest point in Image 1 is compared with all interest points in Image 2 by comparing the SIFT extracted on them. The detection rate among the top N matches is used to study the performance. The detection rate r is defined similarly to [104] as

$$r = \frac{\text{\# correct matches}}{\text{\# possible matches}} \tag{6.21}$$

Experiment results. For evaluation, a performance curve for each distance measure is plotted showing the detection rates versus N, which is the number of the most similar matches allowed. The curves on the synthetic and real image pairs are shown in Fig. 6.11. In addition, the running time of each method is recorded. The average running time over real image pairs is summarized in Table 6.3. From these results, we see that the cross-bin distances work better than bin-to-bin distances. EMD, EMD-L_1 and the diffusion distance perform consistently better than the quadratic-form distance. For efficiency, it is clear that the diffusion distance is much faster than all three other cross-bin distances - this is due to its linear computational complexity.

6.4 Conclusion and Future Work

We model the difference between two histograms as an isolated temperature field. Therefore the difference can be studied with a diffusion process. Combining this idea and the connection between a diffusion process and the Gaussian pyramid, we proposed a new distance between histograms, diffusion distance. We show that the diffusion distance is

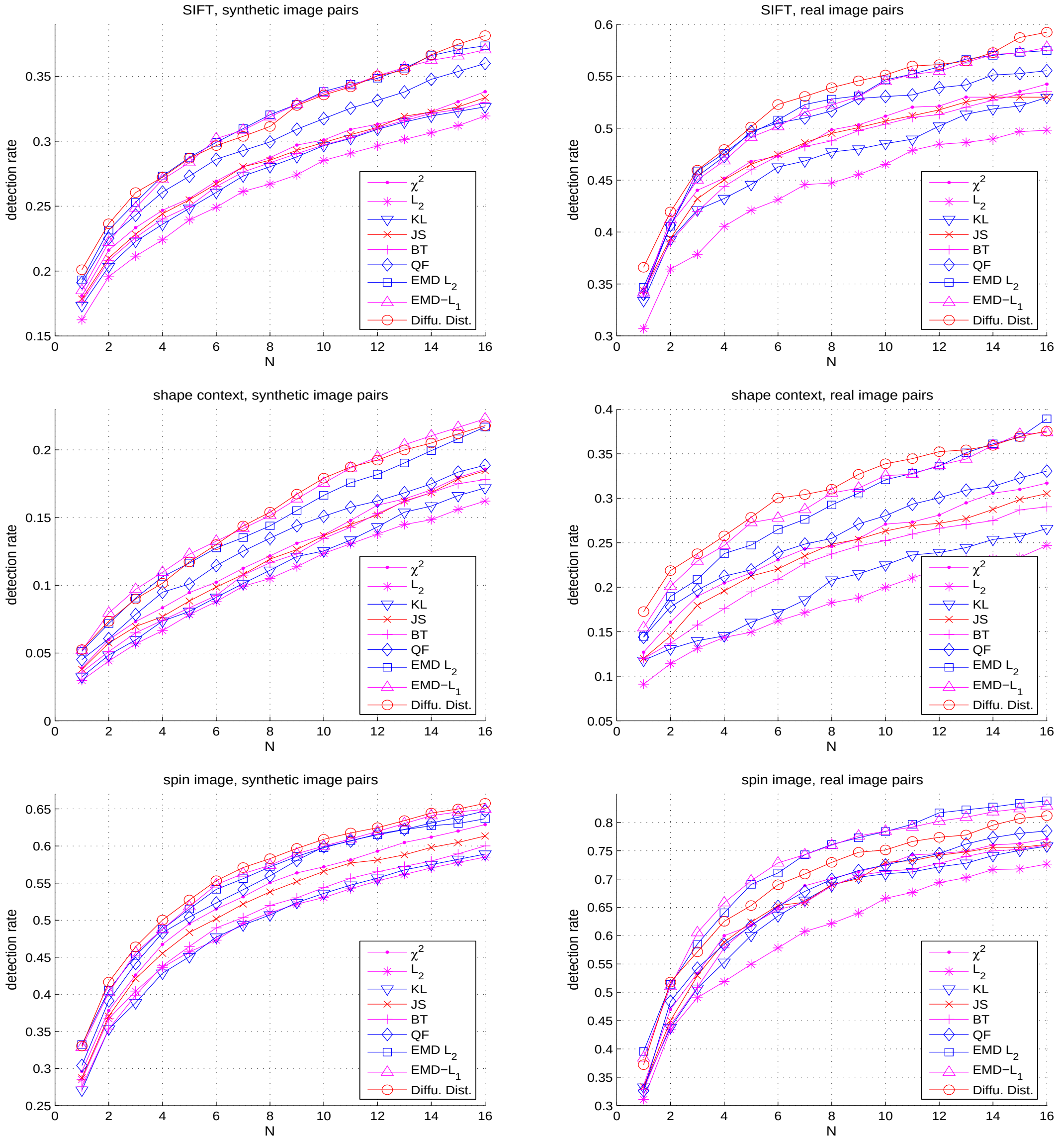


Figure 6.11: ROC curves for interest point matching experiments. Left column is for synthetic image pairs and right for real image pairs. First row is for experiments with SIFT [96], second row for shape context [15], and third row for spin image [80, 65]

Table 6.3: Average time (in seconds) for interest point matching between a real image pair. SC is short for shape context and SI for spin image.

Approach	SIFT [96]	SC [15]	SI [80, 65]
χ^2	0.055	0.047	0.042
L_2	0.007	0.009	0.01
KL	0.161	0.229	0.2
JS	0.317	0.284	0.299
BT	0.044	0.034	0.047
QF	3.622	3.625	3.675
EMD(L_2)	603.955	418.419	468.955
EMD-L_1	6.041	3.693	3.74
Diffu. Dist.	0.909	0.117	0.112

robust for comparing histogram-based local descriptors since it alleviates deformation problems as well as quantization effects that often occur in real vision problems. In the experiments on both shape features and image features, the proposed approach demonstrates very promising performance in both accuracy and efficiency in comparison with other state-of-the-art histogram distances.

We are interested in deepening our understanding of how the diffusion process models the histogram difference, including further theoretical analysis of the deformation problem and the relationship between the diffusion process and other cross-bin distances, especially the Earth Mover's Distance. We are also interested in applying the proposed approach to other histogram comparison problems aside from local descriptors.

Part III

Automatic Thumbnail Cropping and Its Effectiveness

Chapter 7

Automatic Thumbnail Cropping and Its Effectiveness

[1]

7.1 Introduction

Thumbnail images are now a widely used technique for visualizing large numbers of images given limited screen real estate. The *QBIC* system developed by Flickner et al. [41] is a notable image database example. A zoomable image browser, *PhotoMesa* [14], lays out thumbnails in a zoomable space and lets users move through the space of images with a simple set of navigation functions. *PhotoFinder* applied thumbnails as a visualization method for personal photo collections [69]. Popular commercial products such as Adobe Photoshop Album [2] and ACDSee [1] also use thumbnails to represent images files in their interfaces. In addition to image thumbnails, *Summary thumbnail* [76]is proposed for web pages to improve web browsing on small screens.

Current systems generate thumbnails by shrinking the original image. This method is simple. However, thumbnails generated this way can be difficult to recognize, especially when the thumbnails are very small. This phenomenon is not unexpected, since shrinking an image causes detailed information to be lost. An intuitive solution is to keep the more informative part of the image and cut less informative regions before shrinking.

[1] Part of this work is jointly done with Bongwon Suh supervised by Professor Ben B. Bederson.

Some commercial products allow users to manually crop and shrink images [6]. Burton et al. [23] proposed and compared several image simplification methods to enhance the full-size images before subsampling. They chose edge-detecting smoothing, lossy image compression, and self-organizing feature map as three different techniques in their work.

In quite a different context, DeCarlo and Santella [34] tracked a user's eye movements to determine interesting portions of images, and generated non-photorealistic, painterly images that enhanced the most salient parts of the image. Chen et al. [26] use a visual attention model as a cue to conduct image adaptation for small displays.

In this part, we study the effectiveness of saliency based cropping methods for preserving the recognizability of important objects in thumbnails. Our first method is a general cropping method based on the saliency map of Itti and Koch based on a model of human visual attention [63, 62]. A saliency map of a given image describes the importance of each position in the image. In our method, we use the saliency map directly as an indication of how much information each position in images contains. The merit of this method is that the saliency map is built up from low-level features only, so it can be applied to general images. We then select the portion of the image of maximal informativeness.

Although this saliency based method is useful, it does not consider semantic information in images. We show that semantic information can be used to further improve thumbnail cropping, using automatic face detection. We choose this domain because a great many pictures of interest show human faces, and also because face detection methods have begun to achieve high accuracy and efficiency [161, 164].

In this part we describe saliency based cropping and face detection based cropping after first discussing related work from the field of visual attention. We then explain the design of a user study that evaluates the thumbnail methods. This part concludes with a discussion of our findings and future work.

Part of this work appears in [142].

7.2 Related Work

Visual attention is the ability of biological visual systems to detect interesting parts of the visual input [63, 62, 106, 105]. The saliency map of an image describes the degree of saliency of each position in the image. The saliency map is a matrix corresponding to the input image that describes the degree of saliency of each position in the input image. Itti and Koch [63, 62] provided an approach to compute a saliency map for images. Their method first uses pyramid technology to compute three feature maps for three low level features: color, intensity, and orientation. For each feature, saliency is detected when a portion of an image differs in that feature from neighboring regions. Then these feature maps are combined together to form a single saliency map. After this, in a series of iterations, salient pixels suppress the saliency of their neighbors, to concentrate saliency in a few key points.

Chen et al. [26] proposed using semantic models together with the saliency model of Itti and Koch to identify important portions of an image, prior to cropping. Their method is based on the attention model, which uses attention objects as the basic elements. The attention value of each attention object is calculated by combining attention value from different models. For semantic attention models they use a face detection technique

[86] and a text detection technique [27] to compute two different models. The method provides a way to combine semantic information with low-level features. However, when combining the different values, their method uses a heuristic weight that is different for five different predefined image types. Images need to be manually categorized into the five categories prior to applying their method. Furthermore, it heavily relies on semantic extraction techniques. When the corresponding semantic technique is not available or when the technique failed to provide good result (e.g. no face found in the image), it is hard to expect a good result from the method.

There are some recent work on combining the saliency with visualization/browsing tasks. Xie et al. [160] proposed improving the browsing interface by learning user interest from their previous activity. Lee et al. [81] applied the saliency to 3D mesh visualization. Wang et al. [154] presented a perceptual scale space technique for adaptive image displaying. Rother et al. [125] presented a saliency-based technique to smartly merge multiple images (by their parts) into a single one. Khella and Bederson [72] designed the *Pocket PhotoMesa* to facilitate image browsing on pocket screens.

7.3 Thumbnail Cropping

7.3.1 Problem Definition

We define the thumbnail cropping problem as follows: Given an image I, the goal of thumbnail cropping is to find a rectangle R_C, containing a subset of the image I_C so that the main objects in the image are visible in the subimage. We then shrink I_C to a thumbnail. In the rest of this chapter, we use the word “cropping” to indicate thumbnail cropping.

In the next subsection, we propose a general cropping method, which is based on the saliency map and can be applied to general images. Next, a face detection based cropping method is introduced for images with faces.

7.3.2 A General Cropping Method Based on the Saliency Map

In this method, we use the saliency value to evaluate the degree of informativeness of different positions in the image I. The cropping rectangle R_C should satisfy two conditions: having a small size and containing most of the salient parts of the image. These two conditions generally conflict with each other. Our goal is to find the optimal rectangle to balance these two conditions.

Compute Saliency Map

We use Itti and Koch's saliency algorithm [62] because their method is based on low-level features and hence independent of semantic information in images. We choose Itti and Koch's model also because it is one of the most practical algorithms on real images.

The direct implementation of Itti and Koch's approach is time consuming. First, the algorithm to compute the saliency map involves several series of iterations. Some of the iterations (especially for surround inhibition) involve convolutions using very large filter templates (on the order of the size of the saliency map). These convolutions make the computation very time consuming. Instead, we use an approximated algorithm to simplify the computation with 1) using fewer iterations and smaller filter templates during the saliency map calculation; 2) squaring the saliency to enhance it. Experiments show that the approximated algorithm makes no distinguishable difference for thumbnail cropping, while the speed is much faster. Figure 7.1 shows the framework of the algorithm adapted from [62].

An example saliency map is given in Figure 7.2.

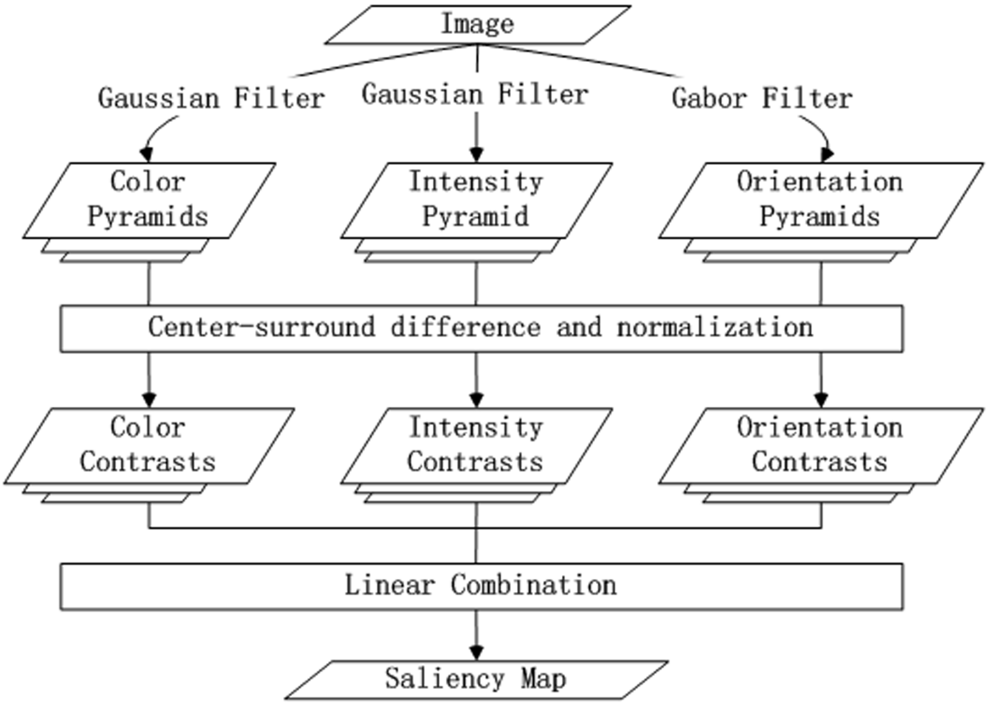


Figure 7.1: Simplified saliency computation. Adapted from [62].

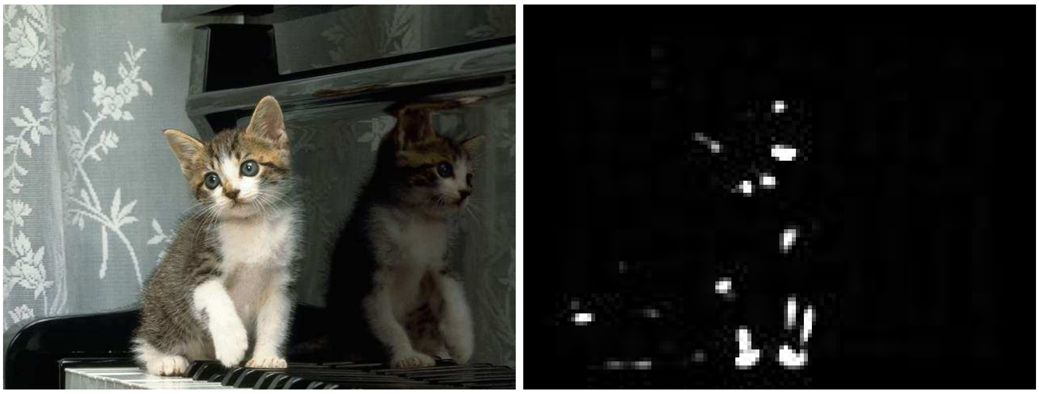


Figure 7.2: Left: original image, right: saliency map of the image shown left.

Find Cropping Rectangle with Fixed Threshold using Brute Force Algorithm

Once the saliency map S_I is ready, our goal is to find the crop rectangle R_C that is expected to contain the most informative part of the image. Since the saliency map is used as the criteria of importance, the sum of saliency within R_C should contain most of the saliency value in S_I. Based on this idea, we can find R_C as the smallest rectangle containing a fixed fraction of saliency. To illustrate this formally, we define candidates set $\mathcal{R}(\lambda)$ for R_C and the fraction threshold λ as

$$\mathcal{R} = \left\{ r : \frac{\sum_{(x,y)\in r} S_I(x,y)}{\sum_{(x,y)} S_I(x,y)} > \lambda \right\} \tag{7.1}$$

Then R_C is given by

$$R_C = \text{argmin}_{r\in\mathcal{R}(\lambda)}\{\text{area}(r)\} \tag{7.2}$$

R_C denotes the minimum rectangle that satisfies the threshold defined above. A brute force algorithm was developed to compute R_C. A brute force algorithm was developed to compute R_C.

Find Cropping Rectangle with Fixed Threshold using Greedy Algorithm

The brute force method works, however, it is not time efficient since the brute force algorithm basically searches all sub-rectangles exhaustively. To solve this problem, we propose using a greedy search instead of brute force method by only considering rectangles that include the peaks of the saliency.

Algorithm 3 shows the algorithm GREEDY-CROPPING to find the cropping rectangle with fixed saliency threshold λ. The greedy algorithm calculates R_C by incrementally including the next most salient peak point P. Also when including a salient point P in R_C, we union R_C with a small rectangle centered at P. This is because if P is within

Algorithm 3 GREEDY-CROPPING (S, λ). Algorithm to find cropping rectangle with fixed saliency threshold. S is the input saliency map and λ is the threshold.

```
thresholdSum ← $\lambda \times$ Total saliency value in $S$
$R_C$ ← the center of $S$
currentSaliencySum ← saliency value of $R_C$
while currentSaliencySum $<$ thresholdSum do
    $P$ ← Maximum saliency point outside $R_C$
    $R'$ ← Small rectangle centered at P
    $R_C$ ← UNION$(R_C, R')$
    Update currentSaliencySum with new region $R_C$
end while
return $R_C$
```

the foreground object, it is expected that a small region surrounding P would also contain the object.

This algorithm can be modified to satisfy further requirements. For example, the UNION function in Algorithm 3 can be altered when the cropped rectangle should have the same aspect ratio as the original image. Rather than just merging two rectangles, UNION needs to calculate the minimum surrounding bounds that have the same aspect ratio as the original image. As another example, the initial value of R_C can be set to either the center of image, S, or the most salient point or any other point. Since the initial point always falls in the result thumbnail, it can be regarded as a point with extremely large saliency. When the most salient point is selected as an initial point, the result can be optimized to have the minimum size. But, we found that to begin the algorithm with the center of images gives more robust and faster results even though it might increase the size of the result thumbnail especially when all salient points are skewed to one side of an image.

Find Cropping Rectangle with Dynamic Threshold

Experience shows that the most effective threshold varies from image to image. We therefore have developed a method for adaptively determining the threshold λ.

Intuitively, we want to choose a threshold at a point of diminishing returns, where adding small amounts of additional saliency requires a large increase in the rectangle. We use an area-threshold graph to visualize this. The X axis indicates the threshold (fraction of saliency) while the Y axis shows the normalized area of the cropping rectangle as the result of the greedy algorithm mentioned above. Here the normalized area has a value between 0 and 1. The solid curve in Figure 7.3 gives an example of an area-threshold graph.

A natural solution is to use the threshold with maximum gradient in the area-threshold graph. We approximate this using a binary search method to find the threshold in three steps: First, we calculate the area-threshold graph for the given image. Second, we use a binary search method to find the threshold where the graph goes up quickly. Third, the threshold is tuned back to the position where a local maximum gradient exists. The dotted lines in Figure 7.3 demonstrate the process of finding the threshold for the image given in Figure 7.2.

Examples of Saliency Map Based Cropping

After getting R_C, we can directly crop the input image I. Thumbnails of the image given in Figure 7.2 are shown in Figure 7.4. It is clear from Figure 7.4 that the cropped

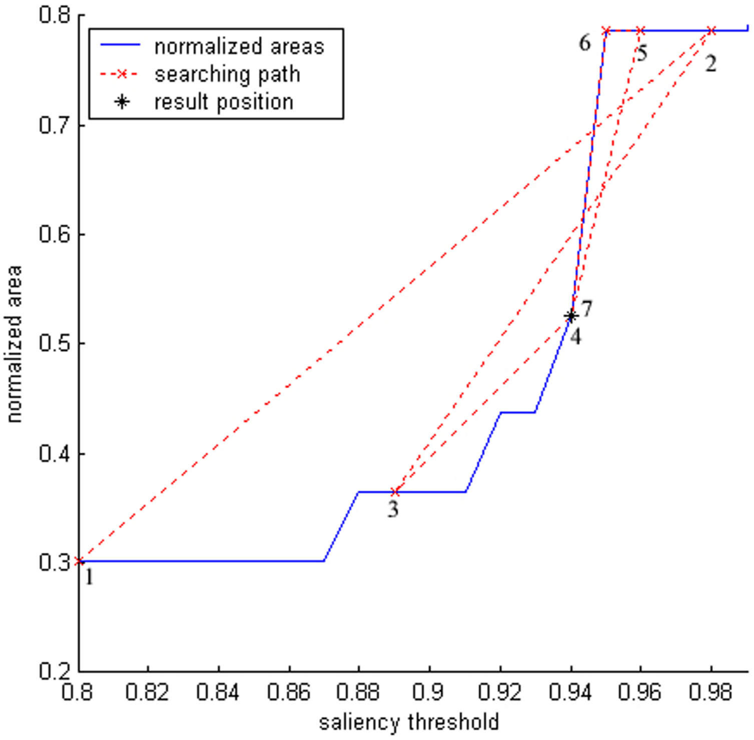


Figure 7.3: The solid line represents the area-threshold graph. The dotted lines show the process of searching for the best threshold. The numbers indicate the sequence of searching.

thumbnail can be more easily recognized than the thumbnail without cropping.

Figure 5 shows the result of an image whose salient parts are more scattered. Photos focusing primarily on the subject and without much background information often have this property. A merit of our algorithm is that it is not sensitive to this.

7.3.3 Face Detection Based Cropping

In the above section, we proposed a general method for thumbnail cropping. The method relies only on low-level features. However, if our goal is to make the objects of interest in an image more recognizable, we can clearly do this more effectively when we are able

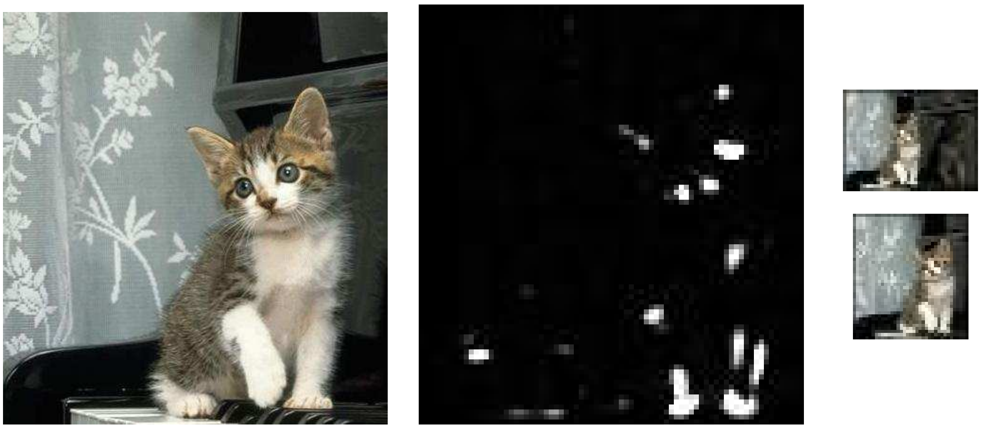


Figure 7.4: (left): the image cropped based on the saliency map; (middle): the cropping rectangle which contains most of the saliency parts; (right top): a thumbnail subsampled from the original image; (right bottom): a thumbnail subsampled from the cropped image (left part of this figure).

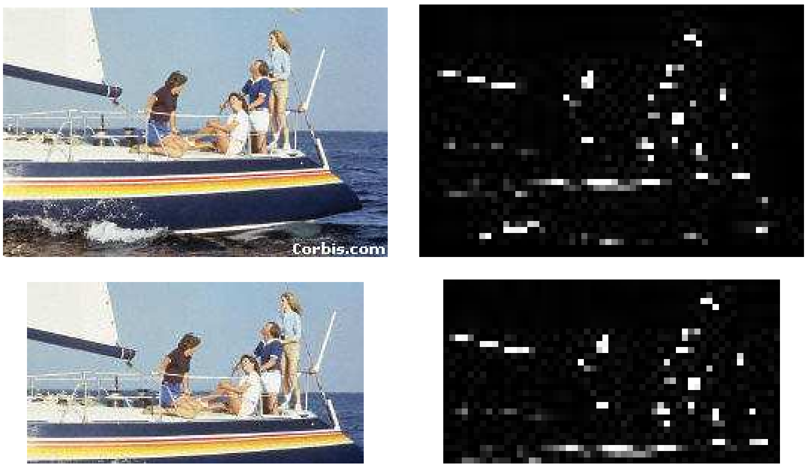


Figure 7.5: (left top): the original image (courtesy of Corbis [3]); (right top): the saliency map; (left bottom): the cropped image; (right bottom): the cropped saliency map which contains most of the salienct parts.

to automatically detect the position of these objects.

Images of people are essential in a lot of research and application areas. At the same time, face processing is a rapidly expanding area and has attracted a lot of research effort in recent years. Face detection is one of the most important problems in the area. Surveys the numerous methods proposed for face detection can be found in [161] and [164].

For human image thumbnails, we claim that recognizability will increase if we crop the image to contain only the face region. Based on this claim, we designed a thumbnail cropping approach based on face detection. First, we identify faces by applying CMU's on-line face detection [5, 131] to the given images. Then, the cropping rectangle R_C is computed as containing all the detected faces. After that, the thumbnail is generated from the image cropped from the original image by R_C.

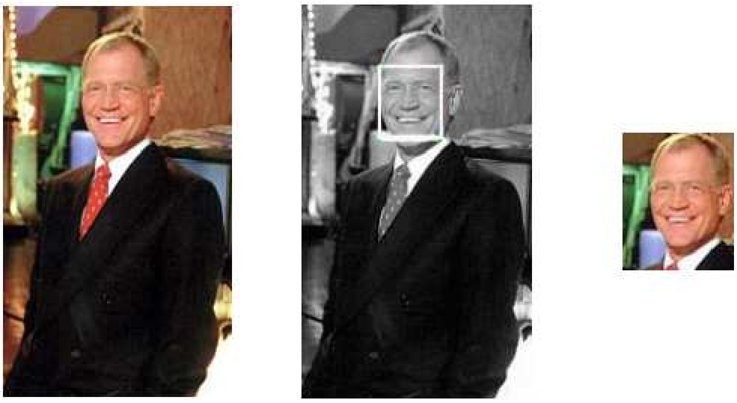


Figure 7.6: (left): the original image; (middle): the face detection result from CMU's online face detection [9]; (right): the cropped image based on the face detection result.

Figure 7.6 shows an example image, its face detection result and the cropped image. Figure **??** shows the three thumbnails generated via three different methods. In this example, we can see that face detection based cropping method is a very effective way to create thumbnails, while saliency based cropping produces little improvement because the original image has few non-salient regions to cut.

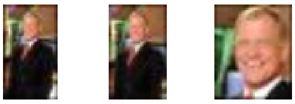


Figure 7.7: Thumbnails generated by the three different methods. (left): without cropping; (middle): saliency based cropping; (right): face detection based cropping.

7.4 User Study

We ran a controlled empirical study to examine the effect of different thumbnail generation methods on the ability of users to recognize objects in images. The experiment is divided into two parts. First, we measured how recognition rates change depending on thumbnail size and thumbnail generation techniques. Participants were asked to recognize objects in small thumbnails (Recognition Task). Second, we measured how the thumbnail generation technique affects search performance (Visual Search Task). Participants were asked to find images that match given descriptions.

7.4.1 Design of Study

The recognition tasks were designed to measure the successful recognition rate of thumbnail images as three conditions varied: image set, thumbnail technique, and thumbnail size. We measured the correctness as a dependent variable.

The visual search task conditions were designed to measure the effectiveness of image search with thumbnails generated with different techniques. The experiment employed a 3x3 within-subjects factorial design, with image set and thumbnail technique as independent variables. We measured search time as a dependant variable. But, since the face-detection clipping is not applicable to the Animal Set and the Corbis Set, we omitted the visual search tasks with those conditions as in Table 7.1. The total duration of the experiment for each participant was about 45 minutes.

Table 7.1: Visual search task design. Checkmarks (✓) show which image sets were tested with which image cropping techniques.

Thumbnail Technique	Animal Set	Corbis Set	Face Set
Plain shrunken thumbnail	✓	✓	✓
Saliency based cropping	✓	✓	✓
Face detection based cropping	×	×	✓

7.4.2 Participants

There were 20 participants in this study. Participants were college or graduate students at the University of Maryland at College Park recruited on the campus. All participants were familiar with computers. Before the tasks began, all participants were asked to pick ten familiar persons out of fifteen candidates. Two participants had difficulty with choosing them. Since the participants must recognize the people whose images are used for identification, the results from those two participants were excluded from the analysis.

7.4.3 Image Sets

We used three image sets for the experiment. We also used filler images as distracters to minimize the duplicate exposure of images in the visual search tasks. There were 500 filler images and images were randomly chosen from this set as needed. These images were carefully chosen so that none of them were similar to images in the three test image sets.

Animal Set (AS) The ”Animal Set” includes images of ten different animals and there are five images per animal. All images were gathered from various sources on the Web. The reason we chose animals as target images was to test recognition and visual search performance with familiar objects. The basic criteria of choosing animals were 1) that the animals should be very familiar so that participants can recognize them without prior learning; and 2) they should be easily distinguishable from each other. As an example, donkeys and horses are too similar to each other. To prevent confusion, we only used horses.

Corbis Set (CS) Corbis is a well known source for digital images and provides various types of tailored digital photos [3]. Its images are professionally taken and manually cropped. The goal of this set is to represent images already in the best possible shape. We randomly selected 100 images out of 10,000 images. We used only 10 images as search targets for visual search tasks to reduce the experimental errors. But during the experiment, we found that one task was problematic because there were very similar images in the fillers and sometimes participants picked unintended images as an answer. Therefore we discarded the result from the task. A total of five observations were discarded due to this condition.

Face Set (FS) This set includes images of fifteen well known people who are either politicians or entertainers. Five images per person were used for this experiment. All images were gathered from the Web. We used this set to test the effectiveness of face detection based cropping technique and to see how the participants' recognition rate varies with different types of images.

Some images in this set contained more than one face. In this case, we cropped the image so that the resulting image contains all the faces in the original image. Out of 75 images, multiple faces were detected in 25 images. We found that 13 of them contained erratic detections. All erroneously detected faces were included in the cropped thumbnail sets since we intended to test our cropping method with available face detection techniques, which are not perfect.

7.4.4 Thumbnail Techniques

Plain shrinking without cropping The images were scaled down to smaller dimensions. We prepared ten levels of thumbnails from 32 to 68 pixels in the larger dimension. The thumbnail size was increased by four pixels per level. But, for the Face Set images, we increased the number of levels to twelve because we found that some faces are not identifiable even in a 68 pixel thumbnail.

Table 7.2: Ratio of cropped to original image size.

Cropping Technique and Image Set	Ratio	Variance
Saliency based cropping (CS)	61.3%	0.110
Saliency based cropping (AS)	53.9%	0.127
Saliency based cropping (FS)	54.3%	0.128
Saliency based cropping (CS)	57.6%	0.124
Face detection based cropping (FS)	16.1%	0.120

Saliency based cropping By using the saliency based cropping algorithms described above, we cropped out background of the images. Then we shrunk cropped images to ten sizes of thumbnails. Table 7.2 shows how much area was cropped for each technique.

Face detection based cropping Faces were detected by CMU's algorithm as described above. If there were multiple faces detected, we chose the bounding region that contains all detected faces. Then twelve levels of thumbnails from 36 to 80 pixels were prepared for the experiment.

7.4.5 Recognition Task

We used the “Animal Set” and the “Face Set” images to measure how accurately participants could recognize objects in small thumbnails. First, users were asked to identify animals in thumbnails. The thumbnails in this task were chosen randomly from all levels of the Animal Set images. This task was repeated 50 times. When the user clicked the “Next” button, a thumbnail was shown as in Figure 7.8 for two seconds. Since we intended to measure pure recognizability of thumbnails, we limited the time thumbnails were shown. According to our pilot user study, users tended to guess answers even though they could not clearly identify objects in thumbnails when they saw them for a long time. To discourage participants’ from guessing, the interface was designed to make thumbnails disappear after a short period of time, two seconds. For the same reason, we introduced more animals in the answer list. Although we used only ten animals in this experiment, we listed 30 animals as possible answers as seen in Figure 7.8, to limit the subject’s ability to guess identity based on crude cues. In this way, participants were prevented from choosing similarly shaped animals by guess. For example, when participants think that they saw a bird-ish animal, they would select swan if it is the only avian animal. By having multiple birds in the candidate list, we could prevent those undesired behaviors.

After the Animal Set recognition task, users were asked to identify a person in the same way. This Face Set recognition task was repeated 75 times. In this session, the candidates were shown as portraits in addition to names as seen in Figure 7.8.

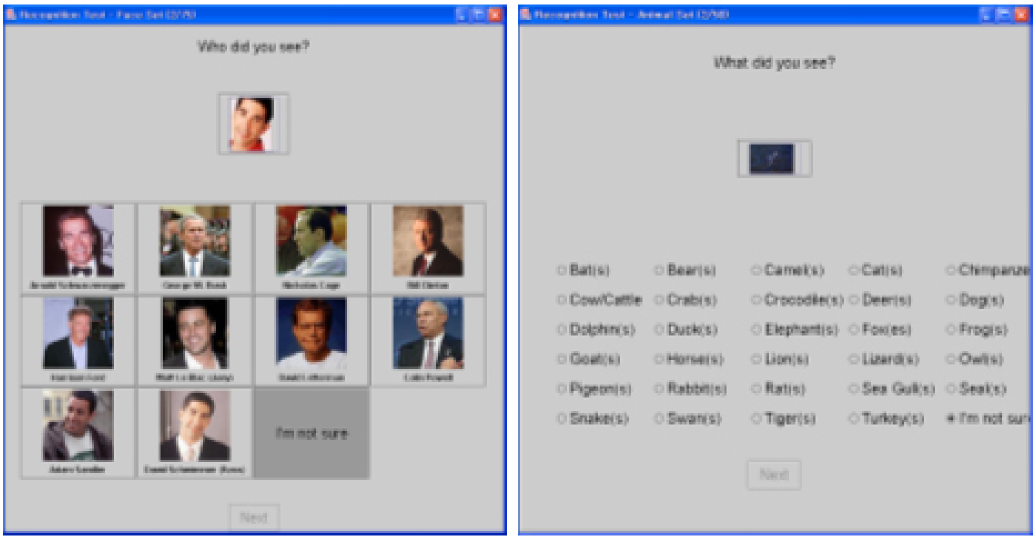

Recognition Test - Face Set (2/75)
Who did you see?
Arnold Schwarzenegger
George W. Bush
Nicholas Cage
Bill Clinton
Harrison Ford
Matt LeBlanc (Joey)
David Letterman
Colin Powell
Adam Sandler
David Schwimmer (Ross)
I'm not sure
Next
Recognition Test - Animal Set (2/50)
What did you see?
○ Bat(s) ○ Bear(s) ○ Camel(s) ○ Cat(s) ○ Chimpanze
○ Cow/Cattle ○ Crab(s) ○ Crocodile(s) ○ Deer(s) ○ Dog(s)
○ Dolphin(s) ○ Duck(s) ○ Elephant(s) ○ Fox(es) ○ Frog(s)
○ Goat(s) ○ Horse(s) ○ Lion(s) ○ Lizard(s) ○ Owl(s)
○ Pigeon(s) ○ Rabbit(s) ○ Rat(s) ○ Sea Gull(s) ○ Seal(s)
○ Snake(s) ○ Swan(s) ○ Tiger(s) ○ Turkey(s) ● I'm not sur
Next

Figure 7.8: Recognition task interfaces. Participants were asked to click what they saw or "I'm not sure" button. Left: Face Set recognition interface, Right: Animal Set recognition interface.

7.4.6 Visual Search Task

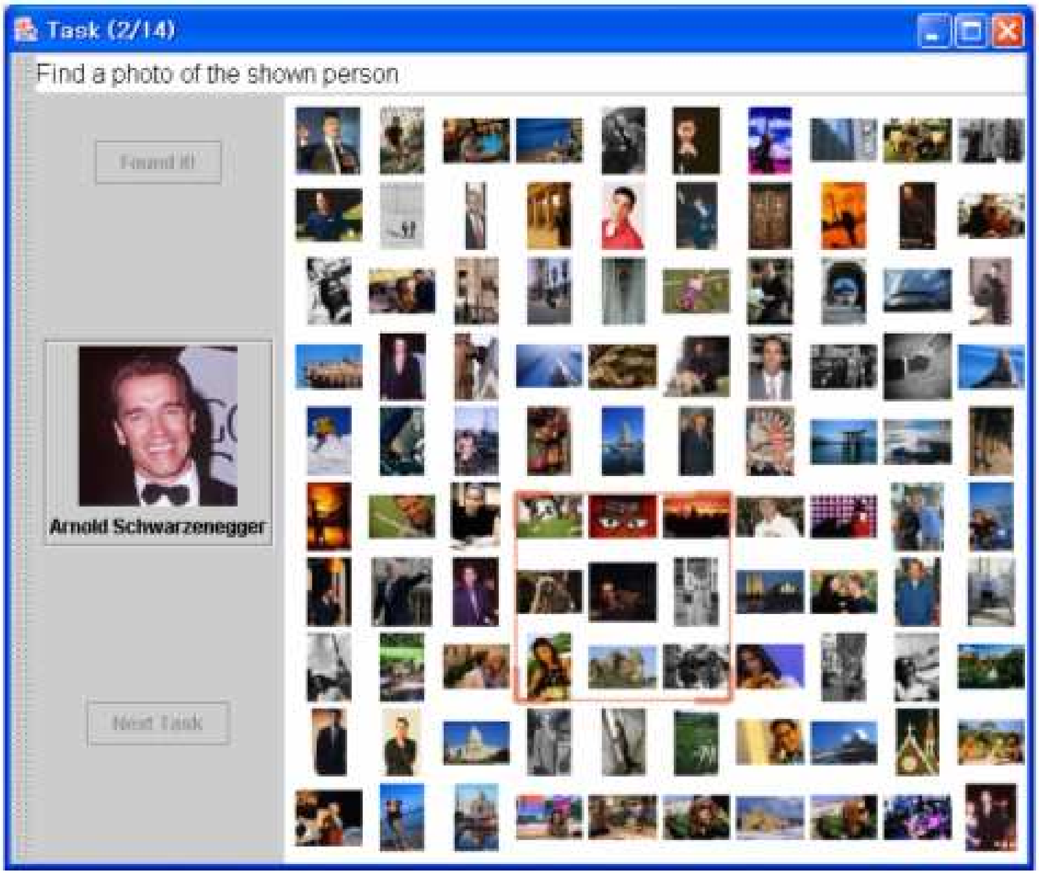
For each testing condition in Table 7.1, participants were given two tasks. Thus, for each visual search session, fourteen search tasks were assigned per participant. The order of tasks was randomized to reduce learning effects.

As shown in Figure 7.9, participants were asked to find one image among 100 images. For the visual search task, it was important to provide equal search conditions for each task and participant. To ensure fairness, we designed the search condition carefully. We suppressed the duplicate occurrences of images and manipulated the locations of the target images.

For the Animal Set search tasks, we randomly chose one target image out of 50 Animal Set images. Then we carefully selected 25 non-similar looking animal images.

After that we mixed them with 49 more images randomly chosen from the filler set as distracters. For the Face Set and Corbis Set tasks, we prepared the task image sets in the same way.

The tasks were given as verbal descriptions for the Animal Set and Corbis set tasks. For the Face Set tasks, a portrait of a target person was given as well as the person's name. The given portraits were separately chosen from an independent collection so that they were not duplicated with images used for the tasks.



Task (2/14)

Find a photo of the shown person

Figure 7.9: Visual search task interface. Participant were asked to find an image that matches a given task description. Users can zoom in, zoom out, and pan freely until they find the right image.

We used a custom-made image browser based on PhotoMesa [14] as our visual search interface. PhotoMesa provides a zooming environment for image navigation with a simple set of control functions. Users click the left mouse button to zoom into a group of images (as indicated by a red rectangle) to see the images in detail and click the right mouse button to zoom out to see more images to overview. Panning is supported either by mouse dragging or arrow keys. The animation between zooming helps user to remember where things fit together based on spatial relationships. PhotoMesa can display a large number of thumbnails in groups on the screen at the same time. Since this user study was intended to test pure visual search, all images were presented in a single cluster as in Figure 7.9.

Participants were allowed to zoom in, zoom out and pan freely for navigation. When users identify the target image, they were asked to zoom into the full scale of the image and click the "Found it" button located on the upper left corner of the interface to finish the task. Before the visual search session, they were given as much time as they wanted until they found it comfortable to use the zoomable interface. Most participants found it very easy to navigate and reported no problem with the navigation during the session.

7.4.7 Recognition Task Results

Figure 7.10 shows the results from the recognition tasks. The horizontal axis represents the size of thumbnails and the vertical axis denotes the recognition accuracy. Each data point in the graph denotes the successful recognition rate of the thumbnails at that level. As shown, the bigger the thumbnails are, the more accurately participants recognize objects in the thumbnails. And this fits well with our intuition. But the interesting point here is that the automatic cropping techniques perform significantly better than the original thumbnails.

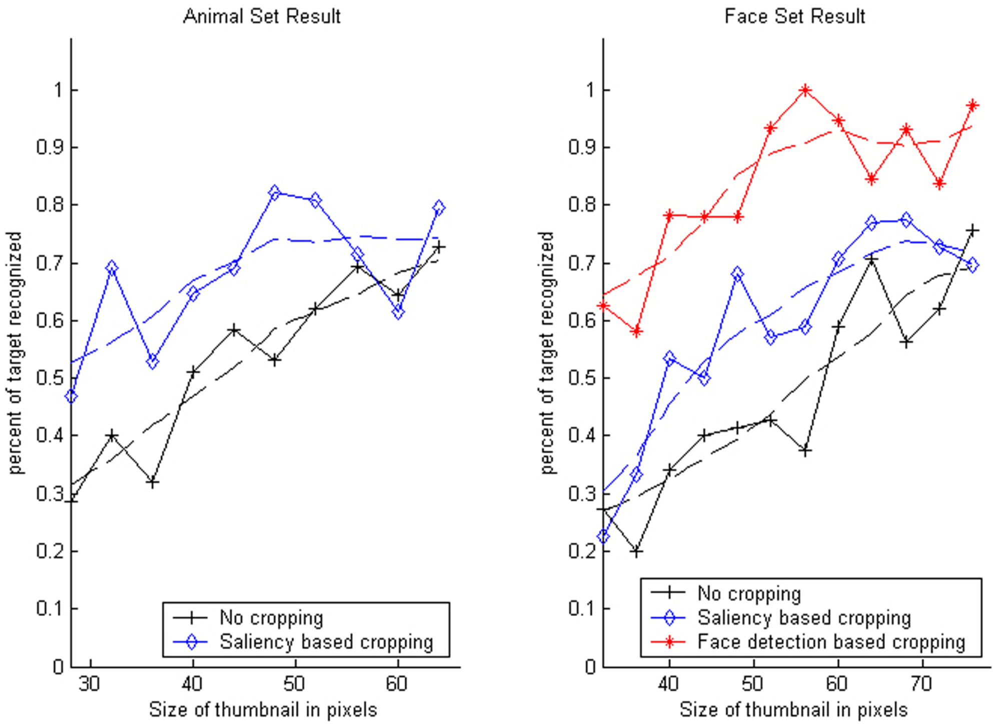


Figure 7.10: Recognition Task Results. Dashed lines are interpolated from jagged data points.

There were clear correlations in the results. Participants recognized objects in bigger thumbnails more accurately regardless of the thumbnail techniques. Therefore, we used Paired T-test (two tailed) to analyze the results. The results are shown in Table 7.3.

The first graph shows the results from the “Animal Set” with two different thumbnail techniques, no cropping and saliency based cropping. As clearly shown, users were able to recognize objects more accurately with saliency based cropped thumbnails than with plain thumbnails with no cropping. One of the major reasons for the difference can be attributed to the fact that the effective portion of images is drawn relatively larger in saliency based cropped images. But, if the main object region is cropped out, this would not be true. In this case, the users would see more non-core parts of images and the recognition rate of the cropped thumbnails would be less than that of plain thumbnails. The goal of this test is to measure if saliency based cropping cut out the right part of images. The recognition test result shows that participants recognize objects better with saliency based thumbnails than plain thumbnails. Therefore, we can say that saliency based cropping cut out the right part of images.

Table 7.3: Analysis results of Recognition Task (Paired T-Test). Every curve in Figure 12 is significantly different from each other.

Condition	$t-$value	P value
No cropping vs. Saliency based cropping on AS	4.33	0.002
No cropping vs. Saliency based cropping on FS	4.16	0.002
No cropping vs. Face Detection based cropping on FS	9.56	< 0.001
Saliency based cropping vs. Face detection based cropping on FS	7.34	< 0.001
AS vs. FS with no cropping	5.00	0.001
AS vs. FS with saliency based cropping	3.08	0.005

During the experiment, participants mentioned that the background sometimes helped with recognition. For example, when they saw blue background, they immediately suspected that the images would be about sea animals. Similarly, the camel was well identified in every thumbnail technique even in very small scale thumbnails because the images have unique desert backgrounds (4 out of 5 images).

Since saliency based cropping cuts out large portion of background (42.4%), we suspected that this might harm recognition. But the result shows that it is not true. Users performed better with cropped images. Even when background was cut out, users still could see some of the background and they got sufficient help from this information. It implies that the saliency based cropping is well balanced. The cropped image shows the main objects bigger while giving enough background information.

The second graph shows results similar to the first. The second graph represents the results from the “Face Set” with three different types of thumbnail techniques, no cropping, saliency based cropping, and face detection based cropping. As seen in the graph, participants perform much better with face detection based thumbnails. It is not surprising that users can identify a person more easily with images with bigger faces.

Compared to the Animal Set result, the Face Set images are less accurately identified. This is because humans have similar visual characteristics while animals have more distinguishing features. In other words, animals can be identified with overall shapes and colors but humans cannot be distinguished easily with those features. The main feature that distinguishes humans is the face. The experimental results clearly show that participants recognized persons better with face detection based thumbnails.

The results also show that saliency cropped thumbnails are useful for recognizing humans as well as animals. We found that saliency based cropped images include persons in the photos so that persons in the images can be presented larger in cropped images. The test results show that the saliency based cropping does increase the recognition rate.

In this study, we used two types of image sets and three different thumbnail techniques. To achieve a higher recognition rate, it is important to show major distinguishing features. If well cropped, a small sized thumbnail would be sufficient to represent the whole image. Face detection based cropping shows benefits when this type of feature extraction is possible. But, in a real image browsing task, it is not always possible to know users’ searching intention. For the same image, users’ focus might be different for browsing purposes. For example, users might want to find a person at some point, but the next time, they would like to focus on costumes only. We believe that the saliency based cropping technique can be applied in most cases when semantic object detection is not available or users' search behavior is not known.

In addition, the recognition rate is not the same for different types of images. This implies that the minimum recognizable size should be different depending on image types.

7.4.8 Visual Search Task Results

Figure 7.11 shows the result of the visual search tasks. Most participants were able to finish the tasks within the 120 second timeout (15 timeouts out of 231 tasks) and also chose the desired answer (5 wrong answers out of 231 tasks). Wrong answers and timed out tasks were excluded from the analysis.

A two way analysis of variance (ANOVA) was conducted on the search time for two conditions, thumbnail technique and image sets. As shown, participants found the answer images faster with cropped thumbnails. Overall, there was a strong difference for visual search performance depending to thumbnail techniques, F(2, 219) = 5.58, p = 0.004.

Since we did not look at face detection cropping for the Animal Set and the Corbis Set, we did another analysis with the two thumbnail techniques (plain thumbnail, saliency based cropped thumbnail) to see if the saliency based algorithm is better. The result shows a significant improvement on visual search with saliency based cropping, F(1, 190) = 3.823, p = 0.05. We therefore believe that the proposed saliency based cropping algorithm make a significant contribution to visual search.

When the results from the Face Set alone were analyzed by one way ANOVA with three thumbnail technique conditions, there also was a significant effect, F(2, 87)=4.56,

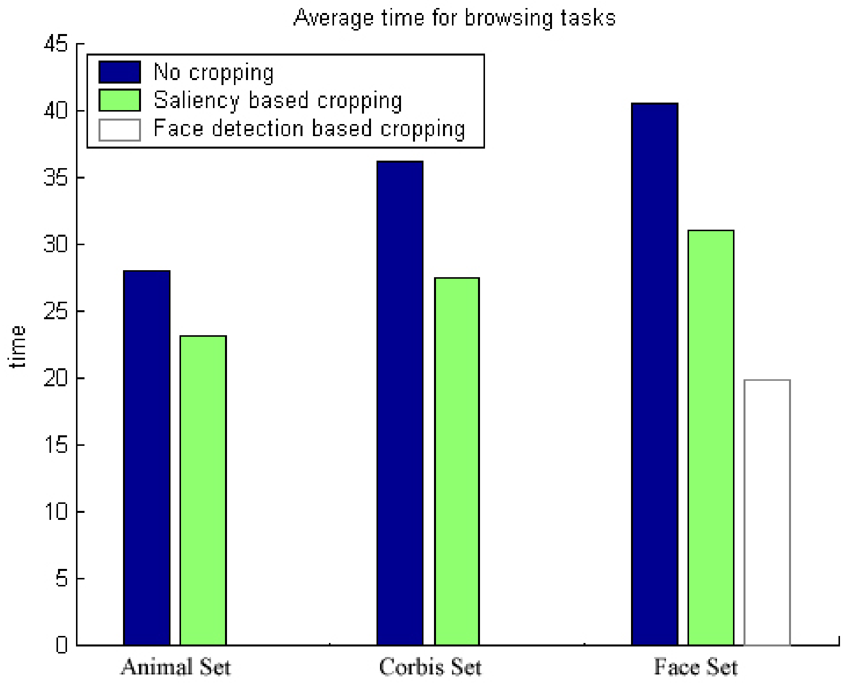


Figure 7.11: Visual search task results.

Table 7.4: List of ANOVA results from the visual search task.

Condition	F value	P value
Thumbnail techniques on three sets	5.58	0.004
Thumbnail techniques on FS	4.56	0.013
No cropping vs. Saliency based thumbnail on three image sets	3.82	0.052
Three image sets regardless of thumbnail techniques	2.44	0.089

p = 0.013. But for the Animal Set and the Corbis Set, there was only a borderline significant effect over different techniques. We think that this is due to the small number of observations. We believe those results would also be significant if there were more participants because there was a clear trend showing an improvement of 18% on the Animal Set and 24% on the Corbis Set. Lack of significance can also be attributed to the fact that the search task itself has large variances by its nature. We found that the location of answer images affects the visual search performance. Users begin to look for images from anywhere in the image space (Figure 7.9). Participants scanned the image space from the upper-left corner, from the lower-right corner, or sometimes randomly. If the answer image is located in the initial position of users' attention, it would be found much earlier. Since we could not control users' behavior, we randomized the location of the answer images. But as a result, there was large variance.

Before the experiment, we were afraid that the cropped thumbnails of the Corbis Set images would affect the search result negatively since the images in the Corbis Set are already in good shape and we were concerned that cutting off their background would harm participants' visual search. But according to our result, saliency based cropped thumbnails does not harm users' visual search. Rather, it showed a tendency to increase participants' search performance. We think that this is because the saliency based cropping algorithm cut the right amount of information without removing core information in the images. At least, we can conclude that it did not make visual search worse to use the cropped thumbnails.

Another interesting thing we found is that the visual search task with the Animal Set tends to take less time than with the Corbis Set and the Face Set, F(2, 219) = 2.44, p = 0.089. This might be because the given Corbis Set and Face Set tasks were harder than the Animal Set. But we think there is another interesting factor. During the experiment, when he found the answer image after a while, one participant said that "Oh . . . This is not what I expected. I expected blue background when I'm supposed to find an airplane." Since one of the authors was observing the experiment session, it was observed that the participant passed over the correct answer image during the search even though he saw the image at reasonably big scale. Since all of the visual search tasks except finding faces were given as verbal descriptions, users did not have any information about what the answer images would be like. We think that this verbal description was one of the factors in performance differences between image sets. We found that animals are easier to find by guessing background than other image sets.

7.5 Discussion and Conclusion

We developed and evaluated two automatic cropping methods. A general thumbnail cropping method based on a saliency model finds the informative portion of images and cuts out the non-core part of images. Thumbnail images generated from the cropped part of images increases users' recognition and helps users in visual search. This technique is general and can be used without any prior assumption about images since it uses only low level features. Furthermore, it also can be used for images already in good shape. Since it dynamically decides how much to cut away, it can prevent cutting out too much.

The face detection based cropping technique shows how semantic information can be used to enhance thumbnail cropping. With a face detection technique, we created more effective thumbnails, which significantly increased users' recognizing and finding performance.

Our study shows strong empirical evidence that the more salient a portion of image, the more informative it is. We also showed that using more recognizable thumbnails increases visual search performance.

Another finding of interest is that users tend to have mental models about search targets. Users tend to develop a model about what a target will look like by guessing its color and shape. We observed that they spent a long time searching or even skipped the correct answer when their guesses were wrong or they were unable to guess. It is known that humans have an “attentional control setting” - a mental setting about what they are (and are not) looking for while performing a given task. Interestingly, it is also known that humans have difficulty in switching their attentional control setting instantaneously [42]. This theory explains our observation. We think that this phenomenon should be regarded in designing image browsing interfaces especially in situations where users need to skim a large number of images.

There are several interesting directions for future research. One direction involves determining how to apply these techniques to other browsing environments. In our study, we used a zoomable interface for visual search. We believe that the image cropping techniques presented in this chapter can benefit other types of interfaces that deal with a large number of images as well. While our research confirms that well cropped thumbnails can increase users’ visual search performance, we did not try to build a model about recognition, attention and its relationship on image browsing. Further research about human’s attention and perception model [115, 159] would help designing a better image browsing system.

Another interesting direction would be to combine image adaptation techniques (i.e. saliency based smoothing) with the image cropping techniques. This would allow faster thumbnail processing and delivery for thumbnail-based retrieval systems.

Chapter 8

Future Work

In this dissertation, several problems in image retrieval have been studied and some solutions were proposed. In the future, we expect extensions of these work, both theoretically and practically. Some of the future works are discussed in the following paragraphs.

Robust object representation. First, we are very interested in finding out the relationship between the part structure and the inner-distance. Part structure is a fundamental problem in computer vision that is still open. Considering the fact that the inner-distance captures part structure and the strong connection between articulation and parts, we expect more understanding of part structure from the study of the inner-distance. Second, we hope to deepen our understanding of deformation invariant framework for intensity images. In the same time, we are also interested in designing efficient algorithm to compute the geodesic-intensity histograms. Third, we will extend the gradient orientation to other applications such as object class classification [40].

Robust feature comparison. For this topic, we would like to move forward mainly in two directions. On the one hand, we want to find out how diffusion processes are theoretically related to cross-bin distances such as the Earth Mover's Distance. This may lead some distance measures with closed form (thus efficient) solutions. On the other hand, we are interested in applying machine learning techniques to estimate dissimilarity metrics.

Automatic thumbnail cropping. There are several interesting directions for future research on thumbnail cropping. One direction is to adaptively determine the thumbnail size. This involves the human attention and perception model [115, 159]. The other direction is to use learning techniques to study human's ability on image summarization, and then apply it to computer based thumbnail cropping.

BIBLIOGRAPHY

[1] "Acdsee, acd systems." [Online]. Available: http://www.adsystems.com

[2] "Adobe photoshop album, adobe systems inc." [Online]. Available: http://www.adobe.com/products/photoshopalbum/

[3] "Corbis." [Online]. Available: http://www.corbis.com

[4] "An electronic field guide: Plant exploration and discovery in the 21st century." [Online]. Available: http://www1.cs.columbia.edu/cvgc/efg/index.php

[5] "Face detection demonstration. robotics institute, carnegie mellon university." [Online]. Available: http://www.vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi

[6] "Vimas technologies." [Online]. Available: http://www.vimas.com

[7] G. Agarwal, H. Ling, D. Jacobs, S. Shirdhonkar, W. J. Kress, R. Russell, N. A. Bourg, P. Belhumeur, N. Dixit, S. Feiner, D. Mahajan, K. Sunkavalli, R. Ramamoorthi, and S. White, "First steps toward an electronic field guide for plants," *Taxon, to appear*.

[8] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1475–1490, 2004.

[9] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[10] Y. Amit and D. Geman, “Shape quantization and recognition with randomized trees,” *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, 1997.

[11] D. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos, “A novel vector-based approach to color image retrieval using a vector angular-based distance measure,” *Comput. Vis. Image Underst.*, vol. 75, no. 1-2, pp. 46–58, 1999.

[12] R. Basri, L. Costa, D. Geiger, and D. Jacobs, “Determining the similarity of deformable shapes,” *Vision Research*, vol. 38, no. 5-16, pp. 2365–2385, 1998.

[13] A. Baumberg, “Reliable feature matching across widely separated views,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2000, pp. 774–781.

[14] B. B. Bederson, “Photomesa: a zoomable image browser using quantum treemaps and bubblemaps,” in *Proceedings of the 14th annual ACM symposium on User interface software and technology (UIST)*, New York, NY, USA, 2001, pp. 71–80.

[15] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[16] A. C. Berg and J. Malik, “Geometric blur for template matching,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. 607–614.

[17] M. Bichsel, “Strategies of robust object recognition for the automatic identification of human faces,” Ph.D. dissertation, ETH Zurich, 1991.

[18] I. Biederman, “Recognition–by–components: A theory of human image understanding,” *Psychological Review*, vol. 94, no. 2, pp. 115–147, 1987.

[19] H. Blum, “Biological shape and visual science,” *J. Theor. Biol.*, vol. 38, pp. 205–287, 1973.

[20] F. L. Bookstein, “Principal warps: Thin-plate splines and the decomposition of deformations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, 1989.

[21] I. Borg and P. Groenen, *Modern Multidimensional Scaling : Theory and Applications*. Springer, 1997.

[22] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition.” *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[23] C. Burton, L. Johnston, and E. Sonenberg, “Case study: An empirical investigation of thumbnail image recognition,” in *Proceedings on Information Visualization*, 1999, pp. 115–121.

[24] C.-C. Chang and C.-J. Lin, “Libsvm : a library for support vector machines,” 2001. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm

[25] H. Chen, P. Belhumeur, and D. Jacobs, “In search of illumination invariants,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2000, pp. 254–261.

[26] L.-Q. Chen, X. Xie, X. Fan, W.-Y. Ma, H.-J. Zhang, and H.-Q. Zhou, “A visual attention model for adapting images on small displays,” *ACM Multimedia Systems Journal*, vol. 9, no. 4, pp. 353–364, 2003.

[27] X. Chen and H. Zhang, “Text area detection from video frames,” in *IEEE Pacific-Rim Conference on Multimedia*, 2001, pp. 222–228.

[28] H. Chui and A. Rangarajan, “A new point matching algorithm for non-rigid registration,” *Comput. Vis. Image Underst.*, vol. 89, no. 2-3, pp. 114–141, 2003.

[29] S. Cohen and L. Guibas, “The earth mover’s distance under transformation sets,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 1999, pp. 1076–1083.

[30] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models - their training and application,” *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2001.

[32] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[33] A. Darabiha, J. Rose, and W. J. MacLean, “Video-rate stereo depth measurement on programmable hardware,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2003, pp. 203–210.

[34] D. DeCarlo and A. Santella, “Stylization and abstraction of photographs,” in *SIGGRAPH ’02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 769–776.

[35] J. Domke and Y. Aloimonos, “Deformation and viewpoint invariant color histograms,” in *BMVC: British Machine Vision Conference*, 2006.

[36] A. Elad and R. Kimmel, “On bending invariant signatures for surfaces.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1285–1295, 2003.

[37] J. Feldman and M. Singh, “Information along contours and object boundaries,” *Psychological Review*, vol. 112, no. 1, pp. 243–252, 2005.

[38] P. F. Felzenszwalb, “Representation and detection of deformable shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 208–220, 2005.

[39] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition.” *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[40] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2003, pp. 264–271.

[41] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, “Query by image and video content: The qbic system,” *Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[42] C. L. Folk, R. W. Remington, and J. C. Johnston, “Involuntary covert orienting is contingent on attentional control settings,” *Journal of Experimental Psychology: HP&P*, vol. 18, pp. 1030–44, 1992.

[43] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, 1991.

[44] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, “Learning low-level vision.” *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

[45] Y. Gdalyahu and D. Weinshall, “Flexible syntactic matching of curves and its application to automatic hierarchal classification of silhouettes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 12, pp. 1313–1328, 1999.

[46] L. V. Gool, T. Moons, and D. Ungureanu, “Affine / photometric invariants for planar intensity patterns,” in *European Conference on Computer Vision (ECCV)*, 1996, pp. 642–651.

[47] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, “Shape representation and classification using the poisson equation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2004, pp. 61–67.

[48] K. Grauman and T. Darrell, “Fast contour matching using approximate earth mover’s distance,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2004, pp. 220–227.

[49] ——, “The pyramid match kernel: Discriminative classification with sets of image features,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2005, pp. 458–1465.

[50] C. Grigorescu and N. Petkov, “Distance sets for shape filters and shape recognition.” *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1274–1286, 2003.

[51] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*. Cambridge, MA, USA: MIT Press, 1990.

[52] N. R. H.L̃ing and D. W. Jacobs, “Using a gradient orientation pyramid for robust passport photo verification,” in *under review*.

[53] J. L. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, “Efficient color histogram indexing for quadratic form distance functions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 7, pp. 729–736, 1995.

[54] A. B. Hamza and H. Krim, “Geodesic object representation and recognition,” in *Discrete Geometry for Computer Imagery, 11th International Conference (DGCI)*, 2003, pp. 378–387.

[55] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, 1988, pp. 147–151.

[56] F. S. Hillier and G. J. Lieberman, *Introduction to Mathematical Programming*. New York, NY: McGraw-Hill, 1990.

[57] F. L. Hitchcock, “The distribution of a product from several sources to numerous localities,” *Jour. Math. Phys.*, vol. 20, pp. 224–230, 1941.

[58] D. D. Hoffman and W. A. Richards, “Parts of recognition,” *Cognition*, vol. 18, pp. 65–96, 1985.

[59] A. S. Holmes, C. J. Rose, and C. J. Taylor, “Measuring similarity between pixel signatures.” *Image Vision Comput.*, vol. 20, no. 5-6, pp. 331–340, 2002.

[60] ——, “Transforming pixel signatures into an improved metric space.” *Image Vision Comput.*, vol. 20, no. 9-10, pp. 701–707, 2002.

[61] P. Indyk and N. Thaper, “Fast image retrieval via embeddings,” in *In 3rd Workshop on Statistical and computational Theories of Vision*, 2003.

[62] L. Itti and C. Koch, “A comparison of feature combination strategies for saliency-based visual attention systems,” in *SPIE human vision and electronic imaging IV(HVEI’99)*, 1999, pp. 473–482.

[63] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

[64] A. Jalba, M. H. F. Wilkinson, and J. B. T. M. Roerdink, “Shape representation and recognition through morphological curvature scale spaces,” *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 331– 341, 2006.

[65] A. E. Johnson and M. Hebert, “Using spin images for efficient object recognition in cluttered 3d scenes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, 1999.

[66] D. G. Jones and J. Malik, “A computational framework for determining stereo correspondence from a set of linear spatial filters,” in *European Conference on Computer Vision*, 1992, pp. 395–410.

[67] K. Jonsson, J. Kittler, Y. Li, and J. Matas, “Support vector machines for face authentication,” in *BMVC: British Machine Vision Conference*, 1999, pp. 543– 553.

[68] T. Kadir, A. Zisserman, and M. Brady, “An affine invariant salient region detector,” in *European Conference on Computer Vision*, vol. 1, 2004, pp. 228–241.

[69] H. Kang and B. Shneiderman, “Visualization methods for personal photo collections: Browsing and searching in the photofinder,” in *IEEE International Conference on Multimedia and Expo*, 2000, pp. 1539–1542.

[70] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” in *ACM Symposium on Theory of Computing (STOC)*, 1984, pp. 302–311.

[71] Y. Ke and R. Sukthankar, “Pca-sift: a more distinctive representation for local image descriptors,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2004, pp. 506–513.

[72] A. Khella and B. B. Bederson, “Pocket photomesa: a zoomable image browser for pdas,” in *MUM ’04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, 2004, pp. 19–24.

[73] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, “Shapes, shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space,” *Int. J. Comput. Vision*, vol. 15, no. 3, pp. 189–224, 1995.

[74] J. J. Koenderink, “The structure of images,” *Biological Cybernetics*, vol. 50, pp. 363–370, 1984.

[75] Y. H. Kwon and N. da Vitoria Lobo, “Age classification from facial images,” *Comput. Vis. Image Underst.*, vol. 74, pp. 1–21, 1999.

[76] H. Lam and P. Baudisch, “ummary thumbnails: Readable overviews for small screen web browsers,” in *In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*, 2005, pp. 681–690.

[77] A. Lanitis, C. Draganova, and C. Christodoulou, “Comparing different classifiers for automatic age estimation,” *IEEE Trans. Syst Man Cybern B Cybern*, vol. 34, no. 1, pp. 621–628, February 2004.

[78] A. Lanitis, C. J. Taylor, and T. F. Cootes, “Toward automatic simulation of aging effects on face images.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 442–455, 2002.

[79] L. J. Latecki, R. Lakamper, and U. Eckhardt, “Shape descriptors for non-rigid shapes with a single closed contour,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 424–429.

[80] S. Lazebnik, C. Schmid, and J. Ponce, “A sparse texture representation using affine-invariant regions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, 2005.

[81] C. H. Lee, A. Varshney, and D. W. Jacobs, “Mesh saliency,” *ACM Trans. Graph. (SIGGRAPH)*, vol. 24, no. 3, pp. 659–666, 2005.

[82] B. Leibe and B. Schiele, “Analyzing appearance and contour based methods for object categorization,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2003, pp. 409–415.

[83] V. Lepetit, P. Lagger, and P. Fua, “Randomized trees for real-time keypoint recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 775–781.

[84] E. Levina and P. Bickel, “The earth mover’s distance is the mallows distance: Some insights from statistics,” in *IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 251–256.

[85] J. Li, S. K. Zhou, and R. Chellappa, “Appearance modeling under geometric context,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2005, pp. 1252–1259.

[86] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, “Statistical learning of multi-view face detection,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 4, 2002, pp. 67–81.

[87] J. Lin, “Divergence measures based on the shannon entropy,” *IEEE Trans. on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

[88] T. Lindeberg, “Feature detection with automatic scale selection,” *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[89] H. Ling and D. W. Jacobs, “Shape classification using the inner-distance,” *IEEE Trans. Pattern Anal. Mach. Intell., accepted*.

[90] ——, “Deformation invariant image matching.” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE Computer Society, 2005, pp. 1466–1473.

[91] ——, “Using the inner-distance for classification of articulated shapes,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE Computer Society, 2005, pp. 719–726.

[92] H. Ling and K. Okada, “An efficient earth mover’s distance algorithm for robust histogram comparison,” *IEEE Trans. Pattern Anal. Mach. Intell., accepted*.

[93] ——, “Diffusion distance for histogram comparison,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 246–253.

[94] ——, “Emd-l_1: An efficient and robust algorithm for comparing histogram-based descriptors,” in *European Conference on Computer Vision (ECCV)*, ser. LNCS, H. B. Aleš Leonardis and A. Pinz, Eds., vol. 3953. Springer, 2006, pp. 330–343.

[95] T.-L. Liu and D. Geiger, “Visual deconstruction: Recognizing articulated objects,” in *EMMCVPR ’97: Proceedings of the First International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer-Verlag, 1997, pp. 295–309.

[96] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[97] J. Ma, Y. Zhao, and S. Anhalt, “Osu svm classifier matlab toolbox.”

[98] C. L. Mallows, “A note on asymptotic joint normality,” *nnals of Mathematical Statistics*, vol. 43, no. 2, pp. 508–515, 1972.

[99] A. M. Martínez, “Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 748–763, 2002.

[100] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions.” in *BMVC: British Machine Vision Conference*, 2002, pp. 384–393.

[101] K. Mikolajczyk, “Affine covariant features.” [Online]. Available: http://www.robots.ox.ac.uk/ vgg/research/affine/

[102] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2003, pp. 257–264.

[103] ——, “Scale & affine invariant interest point detectors,” *Int. J. Comput. Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[104] ——, “A performance evaluation of local descriptors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[105] R. Milanese, “Detecting salient regions in an image: from biological evidence to computer implementation,” *Ph.D. thesis, Univ. of Geneva*, 1993.

[106] R. Milanese, H. Wechsler, S. Gil, J. Bost, and T. Pun, “ntegration of bottom-up and top-down cues for visual attention using non-linear relaxation,” in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 781–785.

[107] B. Moghaddam and A. Pentland, “Probabilistic visual learning for object representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 696–710, 1997.

[108] B. Moghaddam, W. Wahid, and A. Pentland, “Beyond eigenfaces: Probabilistic matching for face recognition.” in *International Conference on Face & Gesture Recognition (FG)*, 1998, pp. 30–35.

[109] F. Mokhtarian, S. Abbasi, and J. Kittler, “Efficient and robust retrieval by shape content through curvature scale space,” in *International Workshop on Image DataBases and MultiMedia Search*, 1996, pp. 35–42.

[110] G. Mori and J. Malik, “Recognizing objects in adversarial clutter: Breaking a visual captcha.” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2003, pp. 134–144.

[111] E. N. Mortensen, H. Deng, and L. G. Shapiro, “A sift descriptor with global context,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 184–190.

[112] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, “The qbic project: Querying images by content, using color, texture, and shape,” in *Storage and Retrieval for Image and Video Databases (SPIE)*, 1993, pp. 173–187.

[113] K. Okada, D. Comaniciu, and A. Krishanan, “cale selection for anisotropic scale-space: Application for volumetric tumor characterization,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2004, pp. 594–601.

[114] R. Osada, T. A. Funkhouser, B. Chazelle, and D. P. Dobkin, “Shape distributions,” *ACM Trans. Graph.*, vol. 21, no. 4, pp. 807–832, 2002.

[115] J. Palmer, C. T. Ames, and D. T. Lindsey, “Measuring the effect of attention on simple visual search,” *Journal of Experimental Psychology: Human Perception & Performance*, vol. 19, p. 19, 1993.

[116] S. Peleg, M. Werman, and H. Rom, “A unified approach to the change of resolution: Space and gray-level,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 739–742, 1989.

[117] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.

[118] E. G. M. Petrakis, A. Diplaros, and E. Milios, “Matching and retrieval of distorted and occluded shapes using dynamic programming,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 11, pp. 1501–1516, 2002.

[119] M. Petrou and A. Kadyrov, “Affine invariant features from the trace transform,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 30–44, 2004.

[120] P. J. Phillips, “Support vector machines applied to face recognition,” in *Advances in Neural Information Processing Systems 16 (NIPS)*, vol. 2, 1999, pp. 803–809.

[121] J. Pilet, V. Lepetit, and P. Fua, “Real-time non-rigid surface detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 822–828.

[122] S. T. Rachev, “The monge-kantorovich mass transference problem and its stochastic applications,” *Theory of Probability and its Applications*, vol. XXIX, no. 4, pp. 647–676, 1984.

[123] N. Ramanathan and R. Chellappa, “Face verification across age progression,” *IEEE Transactions on Image Processing*, vol. to appear.

[124] ——, “Modeling age progression in young faces,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[125] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake, “Digital tapestry,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 589–596.

[126] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann, “Empirical evaluation of dissimilarity measures for color and texture,” *omputer Vision and Image Understanding*, vol. 84, pp. 25–43, 2001.

[127] Y. Rubner and C. Tomasi, *Perceptual Metrics for Image Database Navigation*. Boston, MA: Kluwer Academic Publishers, 2001.

[128] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *Int. J. Comput. Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[129] F. Schaffalitzky and A. Zisserman, “Multi-view matching for unordered image sets,” in *International Conference on Pattern Recognition (ICPR)*, 2002, pp. 414–431.

[130] C. Schmid and R. Mohr, “Local grayvalue invariants for image retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 530–535, 1997.

[131] H. Schneiderman and T. Kanade, “Object detection using the statistics of parts,” *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[132] T. B. Sebastian, P. N. Klein, and B. B. Kimia, “Shock-based indexing into large shape databases,” in *European Conference on Computer Vision (ECCV)*, vol. 3, 2002, pp. 731–746.

[133] ——, “On aligning curves,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 116–125, 2003.

[134] ——, “Recognition of shapes by editing their shock graphs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 550–571, 2004.

[135] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” *Proc. Nat. Acad. Sci.*, vol. 93, no. 4, pp. 1591–1595, 1996.

[136] D. Sharvit, J. C. H. Tek, and B. Kimia, “A fast marching level set method for monotonically advancing fronts,” *J. Visual Communication and Image Representation*, vol. 9, no. 4, pp. 366–380, 1998.

[137] H. C. Shen and A. K. C. Wong, “Generalized texture representation and metric,” *omputer Vision, Graphics, and Image Processing*, vol. 23, pp. 187–206, 1983.

[138] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, “Shock graphs and shape matching,” *Int. J. Comput. Vision*, vol. 35, no. 1, pp. 13–32, 1999.

[139] J. Sivic and A. Zisserman, “Video data mining using configurations of viewpoint invariant regions,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2004, pp. 488–495.

[140] N. Sochen, R. Kimmel, and R. Malladi, “A general framework for low level vision,” *IEEE Trans. in Image Processing, Special Issue on Geometry Driven Diffusion*, vol. 7, no. 3, pp. 310–318, 1998.

[141] O. Söderkvist, “Computer vision classification of leaves from swedish trees,” *MS thesis, Linköping University*, 2001.

[142] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs, “Automatic thumbnail cropping and its effectiveness,” in *Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST)*, 2003, pp. 95–104.

[143] M. J. Swain and D. H. Ballard., “Color indexing,” *Int. J. Comput. Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[144] H. Tan and C. Ngo, “Common pattern discovery using earth movers distance and local flow maximization,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2005, pp. 1222–1229.

[145] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, “Shape context and chamfer matching in cluttered scenes.” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2003, pp. 127–133.

[146] D. W. Thompson, *On Growth and Form*. Dover Publication, 1992.

[147] Z. Tu and A. L. Yuille, “Shape matching and recognition - using generative models and informative features,” in *European Conference on Computer Vision (ECCV)*, vol. 3, 2004, pp. 195–209.

[148] T. Tuytelaars and L. J. V. Gool, “Matching widely separated views based on affine invariant regions,” *International Journal of Computer Vision*, vol. 59, no. 1, pp. 61–85, 2004.

[149] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer, 1995.

[150] A. Vedaldi and S. Soatto, “Features for recognition: Viewpoint invariance for non-planar scenes,” in *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2005, pp. 1474–1481.

[151] R. C. Veltkamp and M. Hagedoorn, “State of the art in shape matching,” *Principles of visual information retrieval*, pp. 87–119, 2001.

[152] S. Wang, J. X. Ji, and Z. Liang, “Landmark-based shape deformation with topology-preserving constraints,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2003, pp. 923–930.

[153] S. Wang, W. Zhu, and Z. Liang, “Shape deformation: Svm regression and application to medical image segmentation,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2001, pp. 209–216.

[154] Y. Wang, S. Bahrami, and S. C. Zhu, “Perceptual scale space and its applications,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2005, pp. 58–65.

[155] I. Weiss and M. Ray, “Recognizing articulated objects using a region-based invariant transform,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1660–1665, 2005.

[156] M. Werman, S. Peleg, and A. Rosenfeld, “A distance metric for multidimensional histograms,” *Computer Vision, Graphics, and Image Processing*, vol. 32, pp. 328–336, 1985.

[157] G. Wesolowsky, “The weber problem: History and perspectives,” *Location Science*, vol. 1, no. 1, pp. 5–23, 1993.

[158] A. P. Witkin, “Scale-space filtering.” in *IJCAI*, 1983, pp. 1019–1022.

[159] J. M. Wolfe, “Guided search 2.0: A revised model of visual search,” *Psychonomic Buttletin and Review*, vol. 1, no. 2, pp. 202–238, 1994.

[160] X. Xie, H. Liu, S. Goumaz, and W.-Y. Ma, “Learning user interest for image browsing on small-form-factor devices,” in *ACM Conference on Human Factors in Computing Systems (CHI 2005)*, 2005, pp. 671–680.

[161] M.-H. Yang, D. J. Kriegman, and N. Ahuja, “Detecting faces in images: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, 2002.

[162] H. Zhang and J. Malik, “Learning a discriminative classifier using shape context distances,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2003, pp. 242–247.

[163] L. Zhao and L. S. Davis, “Segmentation and appearance model building from an image sequence,” in *IEEE International Conference on Image Processing (ICIP)*, vol. 1, 2005, pp. 321–324.

[164] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, 2003.

[165] Y. Zheng and D. S. Doermann, “Robust point matching for two-dimensional non-rigid shapes,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2005, pp. 1561–1566.

[166] S. K. Zhou, B. Georgescu, X. S. Zhou, and D. Comaniciu, “Image based regression using boosting method,” in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2005, pp. 541–548.

[167] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, “1-norm support vector machines.” in *Advances in Neural Information Processing Systems 16 (NIPS)*, 2003.

[168] Q. Zhu, S. Avidan, M.-C. Ye, and K.-T. Cheng, “Fast human detection using a cascade of histograms of oriented gradients,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.