Object Matching Using Deformable Templates

By

Yu Zhong

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Department of Computer Science

June 24, 1997

Abstract

Object Matching Using Deformable Templates

By

Yu Zhong

We have proposed and implemented a general object localization and retrieval scheme based on object shape using deformable templates. Prior knowledge of an object shape is described by a hand-drawn prototype template which consists of the representative contour/edges. The shape variations in an object class are achieved using a set of probabilistic deformation transformations on the template. The deformed shape template then interacts with the input image via a directional edge potential field calculated from the salient edge features. A Bayesian scheme, which is based on the prior knowledge and the edge information in the input image, is employed to find a match between the deformed template and objects in the image. The scheme is invariant to location, rotation, and moderate scale changes of the objects.

We have investigated three different deformation transform basis functions, namely, the trigonometric basis in the 2D domain, the spline representation, and the wavelet basis. We address the advantages and disadvantages of these deformation basis functions.

A coarse-to-fine algorithm is implemented for efficient and automatic object localization. We have successfully applied the deformable template matching algorithm to digital library retrieval tasks, including a hierarchical shape-based retrieval system for a trademark image database and a two-stage retrieval system using object color, texture, and shape. We have also applied the deformable template matching algorithm to track objects in image sequences, where shape and gradient information, combined with the consistency between corresponding object regions throughout the sequence, and the inter-frame motion are used to track the boundary of moving objects.

Future work on this topic could be directed on the following topics: (i) learning the template and its inherent variations from representative training samples; and (ii) annotating and indexing image databases using deformable templates.

To My Parents **Xinmin** and **Qiyu**

ACKNOWLEDGMENTS

I would like to acknowledge all the people who have assisted me during the years of my graduate study at Michigan State University. I am the most grateful to my advisor, Dr. Anil K. Jain, for both his professional and personal advice and guidance. He has provided me with numerous valuable ideas, insights, and comments. He has also been very understanding and supportive. I am very fortunate to have him as my advisor. I would also like to thank the other members of my dissertation committee. I would like to thank Dr. Sridhar Lakshmanan from the University of Michigan-Dearborn for his many insightful discussions. He has also exposed me to a rich literature of deformable template-related work. Dr. John J. Weng has always been available to me to discuss ideas and concepts in computer vision and learning. Dr. V. Mandrekar gave me many suggestions on stochastic processes, and Markov Fields. Dr. Eric Torng has helped me to appreciate the beauty underlying the complexity of computing theory. Besides the committee members, I would like to thank our lab director, Dr. George C. Stockman, who has been very supportive over the years.

My sincere thanks go to all the members of my family. I am very grateful to my parents, Xinmin and Qiyu, for their never-fading love, care, understanding, and

encouragement. I could have accomplished nothing without their love and support. I own my life and every achievement to them. I would like to dedicate this dissertation to them. I would also like to thank my husband Yuntao and my baby boy Richard, who make my life even more joyful. I have learned to appreciate and been fascinated by the young life through Richard's growth.

My fellow students and colleagues at the MSU PRIP lab have provided help and moral support throughout my stay there. I would like to thank them for their interests and concerns, especially Jinlong Chen, Shaoyun Chen, Yao Chen, Yuntao Cui, Chitra Dorai, Hansye Dulimarta, Ling Hong, Sally Howden, Qian Huang, Wey Shiuan Hwang, Kelle Karu, Gongjun Li, Yonghong Li, Jianchang Mao, Aniati Murni, Sharathcha Pankanti, Nalini Ratha, Jarle Strand, Dan Swets, Oivind Due Trier, Aditya Vailaya, Marilyn Wulfekuhler, Bin Yu, and especially the lab managers Lisa Lees and Karissa Miller. I would also like to thank Dr. Alok Gupta and Dr. Marie-Pierre Jolly of Siemens Corporate Research Lab, Princeton for their support of my research.

My special thanks go to Scott Connell for proofreading the draft of this dissertation. I would also like to thank Cathy Davison, Lora Mae Higbee, and Linda Moore for their administrative assistance.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The ultimate goal of computer vision is to simulate the human perception and interpretation of the world around us. Given an image of a scene, in terms of a pixel array of different greyscales or colors, it is extremely difficult to locate and recognize different objects present in it. In spite of numerous efforts in the computer vision field, very little progress has been made in recent years [77, 72]. A general solution to computer vision problems is not envisioned, at least, in the foreseeable future. Successful computer vision (machine vision) applications are limited to specific domains and for specific applications [101].

One major difficulty in image processing and object recognition tasks is how to integrate and interpret the diverse local image cues (gradient, texture, intensity, etc.) [18, 49]. The bottom-up methods often fail due to poor-contrast, occlusion, adverse viewing conditions, and noise. A model or structure-free interpretation is doomed by the underconstrained nature of the problem. Imperfect image data can be augmented with extrinsic information such as geometrical models of the objects likely to be

present in the scene. The object shape is one of the most important characteristics which distinguishes an object from the background in an image [122]. The object shape model can be used to complete the information provided by local image features, such as gray level, texture, or color. Furthermore, the global shape information is more robust than local features in the presence of image noise and poor imaging conditions. However, it should be pointed out that it is more difficult to define a shape mathematically than other visual cues such as color and texture. The interpretation of an object shape is often subjective, which depends not only on the geometrical pixel patterns, but also on the context and the previous experience and knowledge of the observer.

The geometrical shape constraints can vary from local and generic to global and specific, taking various forms. For example, they can incorporate the smoothness or stretchness constraints, or, they can be specified using a hand-crafted parametric or tabular form. Such model information is determined based on a specific application of interest, and should be incorporated explicitly in an integrated and robust computer vision system.

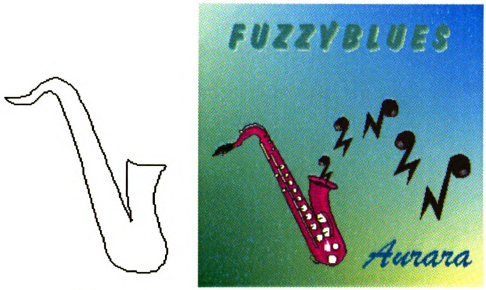
We take full advantage of the global shape of an object, and address the problem of locating and retrieving an object from a complex image using its 2D shape/boundary information [76].

1.1 Problem Statement

The problem under investigation is stated as follows: given a hand-drawn template, which gives an inexact description of the salient or characteristic edge/boundary information of a $2D$ object of interest, how to use this template to locate and identify all the objects in the image which resemble this template? By object location and identification we mean that a description of the object boundary is given in terms of a deformed template. This match is also quantified by a numerical value which indicates the goodness of the match between the template and located object. Figure 1.1 illustrates the problem of interest, where Fig. 1.1(a) shows the hand drawn sketch of a saxophone and Fig. 1.1(b) shows an image which contains a saxophone which resembles the given template in shape. Note that while the sketch resembles the saxophone in the CD cover image, there is also an observable discrepancy in the two shapes.

1.2 Motivations and Challenges

The deformable object matching problem has wide applications in image processing and computer vision, including image/video database retrieval, object recognition and identification, image segmentation, and object tracking. In many of these applications, an *a priori* shape information is available in the form of an inexact model of the object which needs to be matched to the objects present in the input image. We give a few examples in the following:



(a) (b)

Figure 1.1: Deformable template matching. **(a)** a prototype template (bitmap of contour) of a saxophone, **(b)** an image (of a CD cover) containing a saxophone to be matched with the template in (a).

Image segmentation When an approximate shape of the object to be segmented is available, we can apply the deformable template matching model to segment the object from the background. We place an approximate template in the vicinity of the object in the image. This coarse template is then attracted to the salient edges in the image in a similar manner as a "snake" [81] until the deformed template agrees with the object boundary. For example, in medical image segmentation, the general shape of an organ or a tissue of interest is often available, and we need to trace or segment the organ or tissue from a given image for diagnostic purposes.

Content-based retrieval from image databases In an image database retrieval system, the user may provide a set of curves and ask to retrieve all database images which contain such a set of curves. An automatic content-based image retrieval

system [41, 55, 63, 64, 65, 127] should be able to search the database for the images which contain objects with similar characteristics as specified by the user. Shape features have already been incorporated in content-based database retrieval systems [41, 74, 138].

Object tracking In object tracking, the shape of the object varies from frame to frame, but the inter-frame differences are small. The deformable template approach can be used to track the object due to its capability to incorporate both global structure information and variations in the shape class.

Digital video encoding In very low bit rate digital video encoding, it is desirable to track an object in the sequence and encode it using its shape, texture, and motion. Such a coding can give enormous savings in storage and transmission. It also provides a convenient representation for searching, indexing and retrieval. The deformable template model provides a good solution to the representation and tracking problems.

Object recognition and classification Such an approach can also be used to recognize and identify objects. A prototype template can be created for each object class. The identification and recognition can be performed by applying each prototype to the input image and identifying the object class whose prototype interprets the given object the best.

The deformable template matching problem is challenging because of the following factors:

- There are intrinsic variations in an object's shape. As has been said, "there

are no two leaves of the same shape", so an object shape will have intrinsic variations. Object deformation is expected in most imaging applications because of the varying imaging conditions, sensor noise, occlusion and imperfect segmentations. A solution to this problem should handle the shape variations in a meaningful way.

- The approach should be general enough to handle shapes with different appearances. Furthermore, the given template which describes the characteristic edge/boundary of the object of interest can be either open or closed, singly connected or multiply connected. For example, we want the same scheme to be able to locate hands using a hand template and a face sketch to locate a face.

- It should sensibly combine both the prior information about the shape and the input image information (likelihood) to draw an inference.

- The presence of the object of interest in the image is not known. If the object is indeed present, we do not know the number of its occurrences, its position, scale, and orientation.

- The objects of interest are not presegmented from the background. Comparing two segmented shapes is an easier problem: One can first align the two shapes in scale and orientation, and extract some shape features [56] such as moments [67], area, circularity, major axis orientation [111] eccentricity, etc., to see if the two are topologically close in the feature space. In our problem, object localization involves both a segmentation and a matching problem. As both

the segmentation and matching problems are well known to be difficult, this makes the template matching problem even more challenging. This fact along with the dimensionality of the pose parameter vector (position, orientation, scale) makes it a computationally difficult problem.

1.3 Our Approach

We approach the problem of general object localization and identification as a process of matching a deformable template to the object boundary in an input image. The prior shape information of the object of interest is specified as a sketch or binary template. This prototype template is not parameterized, but it contains the edge/boundary information in the form of a bitmap. Deformed templates are obtained by applying parametric transforms to the prototype, and the variability in the template shape is achieved by imposing a probability distribution on the admissible mappings. Among all such admissible transformations, the one that minimizes a Bayesian objective function is selected.

The deformation transformation determines the set of deformed templates which can be obtained using the prototype, i.e., the variety of shapes that the deformable template can take. We would preferably like to use a small set of parameters to represent a large class of deformations. It is desirable to use that deformation transform which approximates the shape classes well. We have investigated three different kinds of deformation transforms, namely,

- a set of two-dimensional trigonometric basis functions with different frequency

components,

- a set of spline basis functions with local compact support, and
- a set of wavelet basis functions with local compact support.

These transformations can model a large set of shape deformations, although each of the deformation basis functions has its own advantages and deficiencies.

The objective function we try to minimize consists of two terms. The first term plays the role of a Bayesian data likelihood. This likelihood term is a potential energy linking the edge positions and gradient directions in the input image to the object boundary specified by the deformed template. The second term corresponds to a Bayesian prior. This prior term penalizes the various deformations of the template — large deviations from the prototype result in a large penalty.

A match between the deformable template and the object in the input image requires minimization of the objective function with respect to the set of deformation and pose parameters. We minimize the objective function by iteratively updating the transformation parameters to alter the shape of the template so that the best match between the deformed template and the edges in the image is obtained. We use the gradient descent method because of its simplicity and efficiency compared to the stochastic global energy optimizers. A problem with the gradient descent method is that it needs a good initialization to avoid locking to local extremas. In most applications, the objective function is non-convex. Therefore, we have used a number of techniques to find good initializations.

- We have used a multiresolution coarse-to-fine search algorithm. The coarse level

is targeted to find good candidates for the finer level match at a relatively low computational cost. At the coarse level, the inputs are subsampled for savings in computations and we use coarse and smooth energy fields to attract the deformable template to regions of interest and to avoid local extremas. At finer levels, the matching is performed only at the outputs from the previous coarser level. Finer energy fields are used for more accurate localization.

- We have also used region information (inside the template) to find locations in an input image where the desired objects are likely to occur. The deformable templates are only initialized at locations with similar texture and/or color. Furthermore, we can compute the color and texture features directly from Discrete Cosine Compressed images. So our approach can be directly applied to JPEG images or the I-frames in MPEG videos.

The low-cost localization processes help to reduce the computational cost of the automatic localization/searching process using deformable template models.

1.4 Relationship to Existing Approaches

We note that certain elements of this approach bear some resemblance to existing studies. These similarities are described below.

Deformation model The idea of representing the deformation as probabilistic transformations on the prototype template is akin to the work of Grenander and his colleagues [4, 28, 97, 98], where such transformations are used to derive a set

of object images from the "ideal" one. However, the use of a bitmap to represent the characteristic/salient curves of the object makes the modeling very general and flexible.

Potential energy The use of potential functions to influence template deformations towards salient image features is akin to the work in [81]. However, our work is different in that we relate the potential to the *nearest* input edge pixels, and this potential is further modified based on the angle between the template pixels and the closest point on the image edge. We also use additional edge direction information to suppress the adversary effects of noisy/spurious edges.

Multi-resolution coarse-to-fine localization We have used a multi-resolution coarse-to-fine algorithm to automatically locate objects of interest in a given image. At a coarse level, we use smoother potential fields, coarse step sizes, and subsampled templates/images which help to escape from local extremas and be attracted to the desired features. At finer levels we use finer potential fields and step sizes for accurate matching. This approach is akin to the classical multi-resolution algorithms and the more recent scale-space approaches.

Combining region cues Texture and color are the most commonly used region information for matching, segmentation, and retrieval tasks. Numerous texture and color features are proposed in the literature. We have extracted texture/color features from Discrete Cosine Transform compressed images (JPEG and MPEG).

Object tracking Deformable contours such as snakes have been used to track features in image sequences. We have also applied our deformable shape model to image tracking problems. Because our deformable shape model is different from a snake in that it incorporates prior shape information, this model shows some advantages in tracking when weak gradient or partial occlusion is present. Instead of being attracted to spurious features, it tries to maintain its prior shape when the gradient is not strong enough or when object features are missing due to occlusion.

1.5 Contributions

As an effective way to integrate both the prior, structural shape knowledge and the local, pixel information, this dissertation exhibits potential applications of deformable template matching in image segmentation, localization, matching, retrieval from image/video databases, and object tracking. We have used the paradigm to successfully solve the following problems.

- Given a rough contour of the object of interest, segment the object embedded in a complex background.
- Automatically localize objects in images using a coarse-to-fine multiresolution matching scheme;
- Retrieve images from image databases using hierarchical architectures which achieve both efficiency and accuracy;
- Track objects in image sequences using the boundary information.

Our experimental results show that under this new paradigm, we can

- match objects that are curved or polygonal, closed or open, simply-connected or multiply-connected;
- retrieve objects based on boundary information alone, even in complex images;
- localize objects independent of their location, orientation, size, and number in the image; and
- Select an appropriate initialization of the template for computational efficiency.

The primary contribution of this research is that it sensibly combines existing ideas along with new ones to provide a systematic paradigm for *general* object matching. This scheme can be applied to objects with different appearances. We do not need a new algorithm to apply to a new shape class; the shape template can be very flexible, open or closed, have a single component or multiple components. The other contributions of the dissertation include:

- A probabilistic transform-based deformation model is used in a novel way; several deformation transforms are utilized. A new likelihood function based on the edge map is proposed which utilizes both the position and direction information for robust matching results; these two components are integrated in a Bayesian formulation. This model is very flexible and general in that we can easily design different likelihood or deformation transforms to serve different application requirements.

- We have addressed the problem of selecting an appropriate initialization of the template in two ways. A coarse-to-fine multiresolution algorithm is proposed to obtain a small set of plausible candidate poses at low resolutions which are then screened at a higher accuracy. When object region information (e.g., color and texture) is available, we use it to find locations with similar region characteristics and then perform the shape matching only at those locations. One attractive feature of this method is that the color and texture features can be computed directly from Discrete Cosine Transform (DCT) compressed images (JPEG or MPEG). A proper initialization of the deformable template avoids many unnecessary computations.

1.6 Dissertation Overview

The rest of the dissertation is organized as follows.

Chapter 2 gives a review and analysis of the previous work on template matching, and, in particular, the deformable template matching approach. The advantages and disadvantages of each of the approaches are addressed.

Chapter 3 formulates the existing energy-based deformation methods in a Bayesian framework. The prior and likelihood for each of the methods (free-form active contour model, deformation model using parametric forms, and deformation model via parametric mappings) are discussed. It is shown that the corresponding MAP estimate is consistent with the solution to the energy minimization problem.

Chapter 4 proposes the shape-based *general* object matching method based on deformable template matching.

Chapter 5 addresses the applications to image segmentation and object tracking.

Chapter 6 discusses the implementation issues for efficient localization. A multiresolution algorithm is proposed to automatically localize the desired objects in an image.

Chapter 7 presents the applications of the deformable template matching method to digital library applications. Two image database retrieval systems are described.

Chapter 8 summarizes the proposed approach, experimental results and limitations. It also gives an outline of the future work.

Chapter 2

Literature Review

Model-based shape matching is a well-known problem in the computer vision and image processing domain. Its applications include image segmentation, object matching and tracking, object recognition and interpretation, and image database retrieval. Early research in this area concentrated mainly on rigid shape matching, where the matched shapes were obtained by applying simple transformations such as translation, rotation, scaling, and the affine transformation to the model template [26, 62]. The transformations are characterized by a set of global parameters, which cannot effectively incorporate complex variations such as local deformation. Examples include correlation-based matching and the Hough transform [11, 39, 66, 104, 123]. Because of the rigidness of the above approaches, their utility is limited. In most applications, an exact model of the object is not available because of the variability in the imaging process and inherent within-class variabilities. Deformable template matching, which is receiving increasing attention, is more versatile and flexible in dealing with the deficiencies of rigid shape matching. It is a more powerful technique because of its

capability to deal with a variety of shape deformations and variations.

In the following, we give a survey of the existing work in this field. We first briefly review the classical template matching methods, which are rigid in the sense that the appearance of an object of interest in the image is known accurately, that is, it is described exactly by a template. Two well known template matching methods, namely, the correlation-based method [2, 9] and the generalized Hough transform, are discussed. Then, we survey the more powerful *deformable template matching* methods. We partition these methods into two classes: **(i)** *free-form*, and **(ii)** *parametric*, based on whether the deformed templates are parameterized or not. The parameterized deformable template matching methods are further classified as either *analytical form-based* or *prototype-based*, where the first class is identified by specification of an analytic form for the templates, and in the second class, a template instance is obtained by transforming (parametrically) a prototype template. An overview of the various template matching techniques is given in Fig. 2.1.

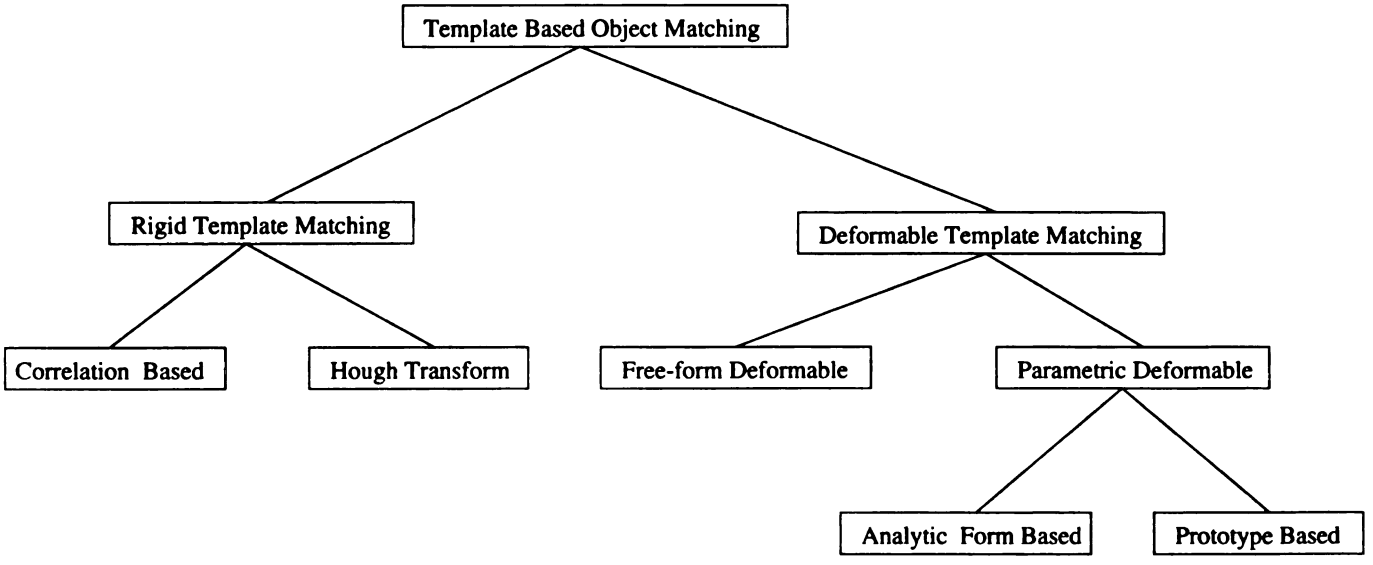


Figure 2.1: An overview of the template matching techniques.

2.1 Rigid Template Matching

When a template is compared to an object in a rigid way, a match is attempted either directly or indirectly, accounting for possible pose and scale changes. By rigid template matching we mean that the object shape can be obtained from the model template via a sequence of operations of translation, rotation, and scaling. An instance of such an example is given in Fig. 2.2 where, for the template fish on the left, successful matches are obtained between this template and the three fish on the right as indicated. When the pose and scale are known, the crosscorrelation between the aligned template and object gives a high matching score. Otherwise, when the pose parameters are unknown, the Hough transform provides a clever way to detect the presence of an object as well as an estimate of the pose. Both methods are examples of rigid template matching, which are typically used to detect image features (e.g., straight lines, corners, or objects) [46, 57, 114, 113].

2.1.1 Correlation-based Matching

Correlation-based matching [2, 9] is a simple filtering method [134] to detect a particular shape or object in an image. An object can be detected if its appearance is known accurately in terms of the template.

Given a template $t(\mathbf{x})$ and an image $i(\mathbf{x})$, the crosscorrelation between $i(x)$ and $t(x)$ at position $\mathbf{y}$ is defined as:

$$R_{ft}(y) = \sum_{\mathbf{x}} i(\mathbf{x})t(\mathbf{x}-\mathbf{y}). \tag{2.1}$$

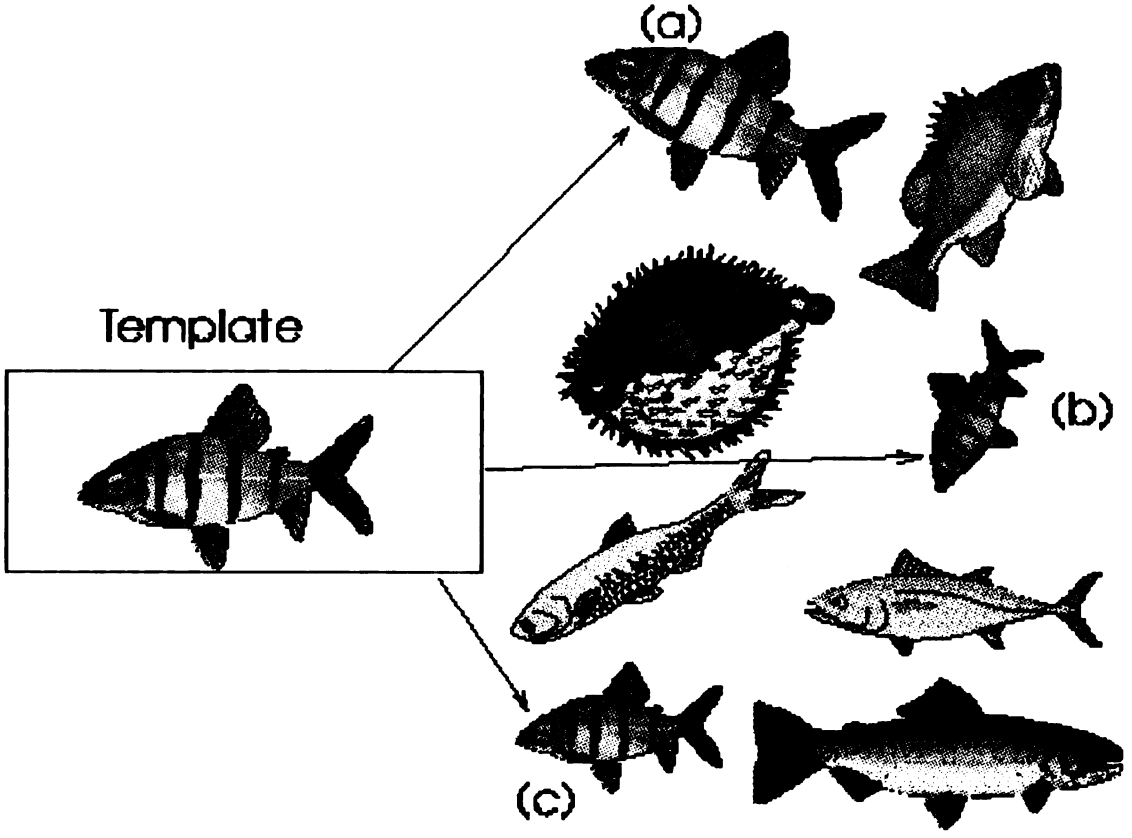


Figure 2.2: An example of rigid template matching. Figs. (a), (b) and (c) can be successfully matched to the rigid template using translation, scaling, and rotation.

The crosscorrelation is maximized when the portion of the image i overlapped with the template is identical to t, i.e., when there is a perfect match (Fig. 2.3). The idea of correlation-based matching is straightforward. The template is shifted in the image field, and the crosscorrelation at each position is calculated. A match is reported at positions where the correlation exceeds a threshold value. In the example in Fig. 2.3, the given template is applied to an image with noise in the lower-right corner. The correlation at each position is computed. Ideally a peak in the correlation array indicates a position of good match, as the one in the upper-left corner. However, a "false" match is caused by the bright noisy pixel with a value 7. The "X"s in the correlation array indicate "not available".

Template	**Image**	**Correlation**
1 1 1	1 1 0 0 0	7 4 2 x x
1 1 1	1 1 1 0 0	5 3 2 x x
1 1 1	1 0 1 0 0	2 1 8 x x
	0 0 0 0 0	x x x x x
	0 0 0 0 7	x x x x x

Figure 2.3: An example of correlation-based matching.

The crosscorrelation array can be efficiently calculated using the correlation theorem for Fourier transforms:

$$\mathbf{F}(\mathbf{i} \star \mathbf{t})_j = (\mathbf{I})_j(\mathbf{T}^*)_j, \quad \mathbf{I} = \mathbf{F}(\mathbf{i}), \ \mathbf{T} = \mathbf{F}(\mathbf{t}), \tag{2.2}$$

where $\mathbf{F}$ denotes the Fourier transform operation, and $\mathbf{T}^*$ is the complex conjugate of $\mathbf{T}$. This theorem says that point-by-point multiplication in the Fourier transform domain is equivalent to convolution in the spatial domain [90]. Because efficient algorithms for Fourier transforms are available [31], the correlation theorem enables costly correlations in the spatial domain to be computed efficiently in the transformed domain. The correlation array can be obtained by

1. taking the Fourier transforms of both the feature image and the template image,
2. performing multiplication in the frequency domain,
3. taking the inverse Fourier transform.

A shortcoming of the correlation-based matching is that it requires that the template be very accurate, and it is sensitive to shape deformations including scale, orientation change, and partially offset features, etc. Nevertheless, it is still a very powerful tool for image matching because of its simplicity and computational efficiency, and has been used directly or indirectly in many applications. One successful instance is offered in the QBIC [102] system for content-based image database retrieval, where correlation is performed on images of reduced resolutions to quickly locate shapes similar to the query sketch. A moderate amount of robustness to shape deformations is achieved by calculating the correlations at a low resolution.

It is noted that the correlation-based matching is a special example of filtering operations in image processing. Filtering is a very general notion of transforming the image intensities in some way so as to enhance or suppress certain image features [5, 54, 111]. In a general sense, many difficult low level image matching and segmentation

tasks can be cast in a similar scenario as the correlation-based feature or object detection, where we need to find:

- a set of filters which best captures the desired image features, and
- appropriate orthogonal basis and transforms which guarantee the computational efficiency.

However, the filtering technique is application-dependent and in most cases, it is very difficult to define good filters and to find the appropriate transform space.

2.1.2 Hough Transform

An elegant and versatile technique to detect parameterized shapes (of object boundaries) was first proposed by Hough [66]. It was later generalized by Ballard [11] to detect any arbitrary shape which can be represented in a tabular form [39, 104, 123]. Basically, the Hough method transforms points in the image (spatial) space into a parameter space. Note that each point in the parameter space determines a curve in the image plane, and each spatial feature corresponds to a curve in the parameter space by reversing the roles of the parameters and the spatial coordinates. So, given a parametric form for a family of curves or shapes which we expect to detect, and a collection of "interesting" image points, we increment, for each such point, the corresponding entries in the quantized parameter space. The count of each quantized bin in the parameter space is the number of image points that lie on the specific curve defined by the corresponding parameter values. The specified shape is detected by finding the peak(s) in the parameter space, or in other words, the set of parameter

values that best describe the given image points. In this way, a global evidence accumulation process of shape detection is transformed into a search for local peaks. Such an example is illustrated in Fig. 2.4, where a straight line $y = m_0x + c_0$ in the $(x - y)$ space is determined by parameters m_0 and c_0. Given a point (x_1, y_1) on this line in the $x - y$ space, all the straight lines which pass through this point are described by $y_1 = mx_1 + c$, where x_1, y_1 are fixed and m, c are variables. They correspond to a straight line $c = y_1 - x_1m$, in the transformed parameter space $(m - c)$ in (b), where y_1, x_1 are the parameters, and m, c are free variables. In particular, this line should pass through the point (m_0, c_0). With the same reasoning, another point (x_2, y_2) on the line $y = mx + c$ is also associated with a line $c = y_2 - x_2m$ in the $(m - c)$ space, which also passes the point (m_0, c_0). In a similar manner, all points on the line $y = m_0x + c_0$ in the $x - y$ space have an associated line in the parameter space $(m - c)$. All these lines intersect at (m_0, c_0) in the $(m - c)$ plane. So the line $y = m_0x + c_0$ can be detected by finding the peak at (m_0, c_0) in the transformed space $(m - c)$.

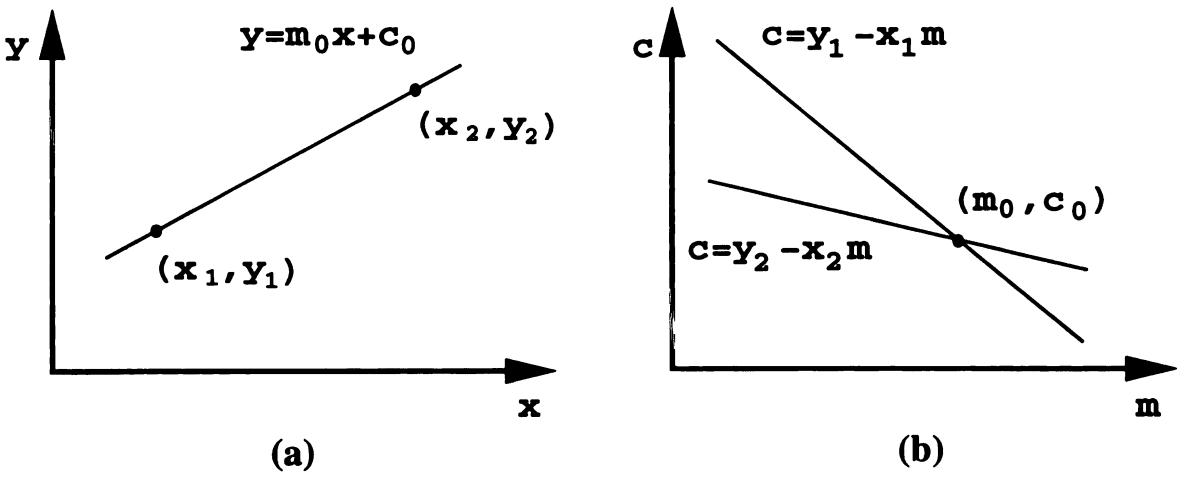
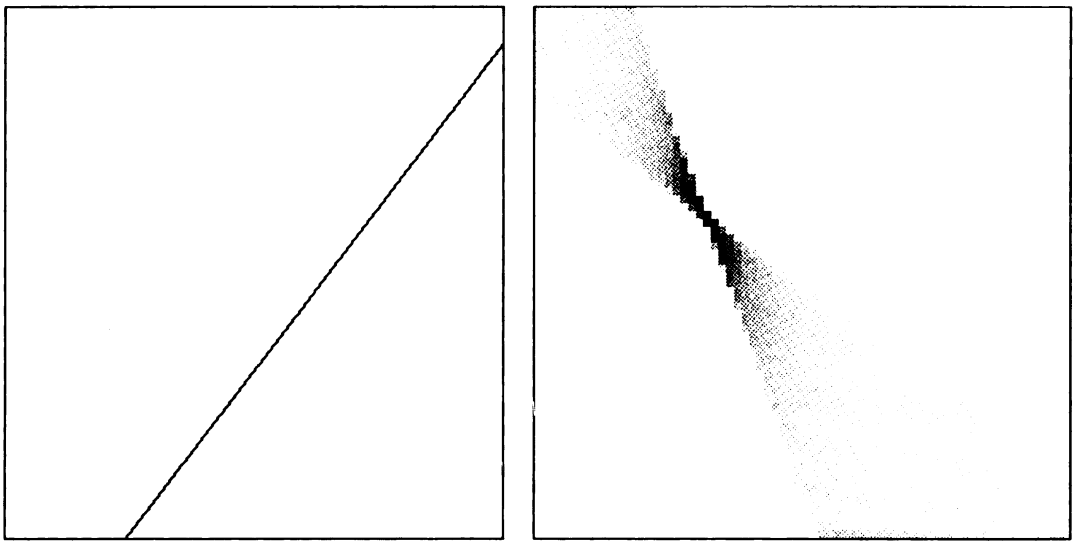


Figure 2.4: Hough Transform: From image space to parameter space. **(a)** A line in the image space $(x - y)$ described by parameters m_0, c_0: $y = c_0 + m_0x$; **(b)** transform two points (x_1, y_1) and (x_2, y_2) on the line in (a) in the $(m - c)$ parameter space.

Figure 2.5 illustrates an example of straight line detection using the Hough transform. Fig. 2.5 (a) gives an image which contains a line segment which is generated using the equation $y = 1.32x - 0.924$. We transform this image space to the parameter space and obtain the vote array in $(m - c)$ space (Fig. 2.5 (b)). The left-lower corner of the parameter array corresponds to $(1.1, -1.35)$, the bins the discretized with a width of 0.01. The votes are shown inversely related to the greyscale value so that dark bins correspond to large votes. The bin of maximum vote is located at $[22, 42]$ (with respect to the lower-left corner), which corresponds to the parameter value $(1.32, -0.93)$. This value predicts the equation of the line segment well.



(a) (b)

Figure 2.5: Hough Transform: detection of a straight line. **(a)** A line in the image space described by the equation ($y = 1.32x - 0.924$); **(b)** votes in the parameter space. The bin of the maximum count corresponds to parameter values $(1.32, 0.93)$ with a precision of 0.01.

The algorithm for detecting a general shape which cannot be represented by an analytical form using the generalized Hough transform (GHT) is given below. The general shape is represented by a tabular form (a table of the coordinates):

- Calculate the R-table based on the template: each entry is indexed by the tangent angle ϕ for points on the template; each entry contains a list of (r_{ij}, α_{ij}) values corresponding to ϕ_i, where each element in the list corresponds to a point on the template whose tangent angle has a value ϕ_i, and r_{ij} is the distance of this point to the centroid of the template (x_c, y_c), α_{ij} is the slope angle of the line connecting this point and the centroid, as described in Fig. 2.6.

- Initialize the accumulator array to zero:

  $A(x_{cmin} : x_{cmax}, y_{cmin} : y_{cmax}, s_{min} : s_{max}, \theta_{min} : \theta_{max})$.

  The parameters are center position (x, y), scale s, and rotation angle θ.

- For each edge point $\mathbf{x} = (x, y)$ in the image, do the following:

  - Compute tangent angle $\phi(\mathbf{x})$.
  - Look at the $\phi(\mathbf{x})$ entry in the R-table and obtain a list of (r, α) pairs.
  - Calculate possible center, scale and rotation angle for every pair (r, α) in the list and for every combination of (s, θ):

    $(x_c, y_c) = (x, y) + r(\phi)s(\cos(\alpha + \theta), \sin(\alpha + \theta))$

  - Increment the accumulator array (x_c, y_c, s, θ).

- Possible shape candidates are given by the maxima in array A.

Figure 2.7 shows an example of object detection using the GHT. Fig. 2.7 (a) shows the template, (b) in an input image, and (c) shows the votes in the translation parameter space $(x - y)$. The vote at each pixel in (c) is the vote when the centroid

of the template is placed at the pixel position, and is proportional to the darkness. Fig. 2.7 (d) shows the result where the detection (in green) corresponding to the maximum vote is overlapped on the input edgemap (in red).

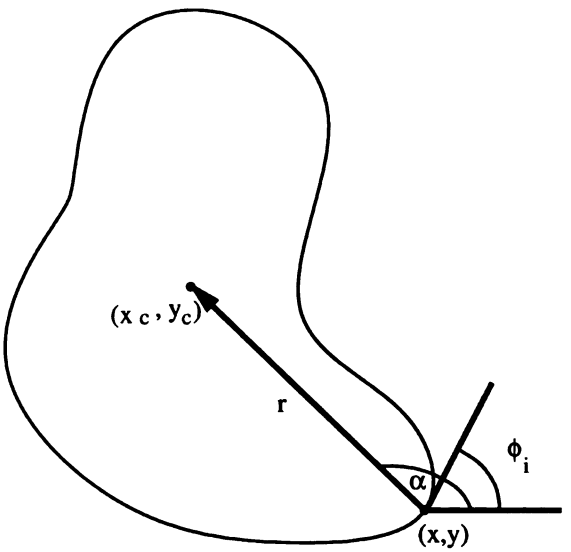


Figure 2.6: Constructing the R-table in the Generalized Hough transform.

One of the advantages of the Hough Transform (HT) method is that it is relatively insensitive to noise, minor occlusions, or gaps. But the computational requirement of HT is rather high. The storage and computation time increase exponentially with the number of parameters, making it practical only for curves with a small number of parameters. Complete surveys on different variants of the HT technique and its applications can be found in [70] and [88].

The HT method can be viewed as template matching. However, it is a rigid scheme in that it is not capable of detecting a shape which is *different* from the template by transformations other than translation, rotation or scaling. A deformable template, on the other hand, is able to "deform" itself to a certain degree to fit the data,

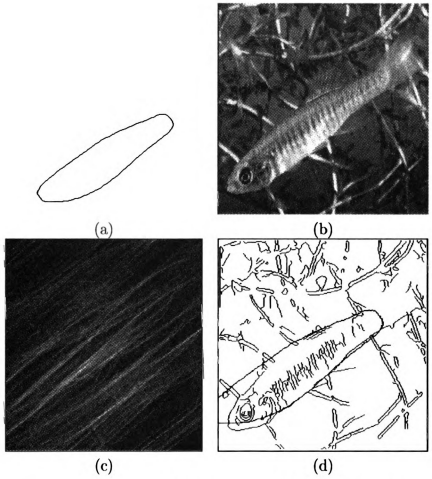


Figure 2.7: Hough Transform: detection of an arbitrary shape. **(a)** an irregular shape; **(b)** an input image; **(c)** votes in the $(x - y)$ translation parameter space. Scale and orientation are fixed for display purpose. **(d)** the detection corresponding to the maximum vote.

by transformations that are possibly more complex than translation, rotation, and scaling.

2.2 Deformable Models

The rigid template matching is effective in some domains, but it has a number of disadvantages. It would fail if the objects in the image are slightly distorted due to the imaging process, viewpoint change, or the diversity of objects shape (inter-object variance). On the contrary, a deformable model, which is "active" in the sense that it can change its shape to fit the data, is more invariant to the shape distortions. It is a promising shape model because of its flexibility, and its ability to both impose geometrical constraints on the shape and to integrate local image evidences. Deformable matching utilizes the flexibility in deformable models to match objects of similar shape (or appearance) when the deformation cannot be explained by affine transforms. In Fig. 2.8 the fish template is matched to the two fishes labeled as (a) and (b). Note that despite the global similarity, one of the matched fishes (a) is different from the template fish by local abnormality, but the similarity of the template to (a) and (b) is significantly higher than to the other objects.

There has been a substantial amount of research on deformable models in recent years. These activities can be partitioned into two classes:

- free-form models, and
- parametric models.

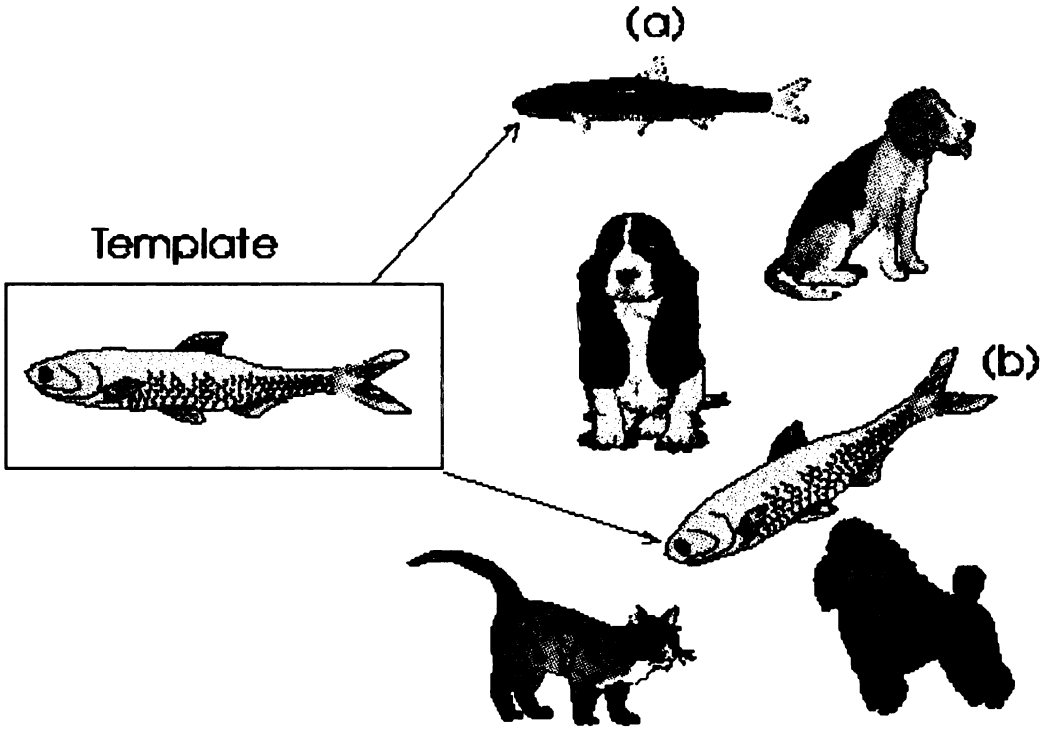


Figure 2.8: An example of deformable template matching.

By free-form deformable models, we mean that there is no global structure of the template except for some general regularization constraints; the template is constrained only by continuity and/or smoothness constraints. Such a free-form model can be deformed to match salient image features like lines and edges using potential fields (energy functions) produced by those features. Since there is no global structure for the template, it can represent an arbitrary shape as long as the continuity and smoothness constraints are satisfied. On the other hand, parametric deformable models can control the deformations using a set of parameters. The parameters are capable of encoding a specific characteristic shape and its variations. This type of model is used when more specific shape information is available, which can be described by a set of parameters. There are two ways to parameterize the shape variation. One is to handcraft a parametric formula for the curves in the shape template. Then different shapes can be obtained using different parameter values. Another method is to design a prototype for a shape class, and then apply a parametric transformation on the prototype to obtain different deformed templates. These various deformable template models are illustrated in Fig. 2.9.

2.2.1 Free-form Deformation Models

Free-form deformable models assume very little structure about the object shape.

Active Contours

Dynamic contour models have become popular after Terzopoulos, Kass and others introduced the snake model [81, 129, 131]. In their approach, an energy-minimizing

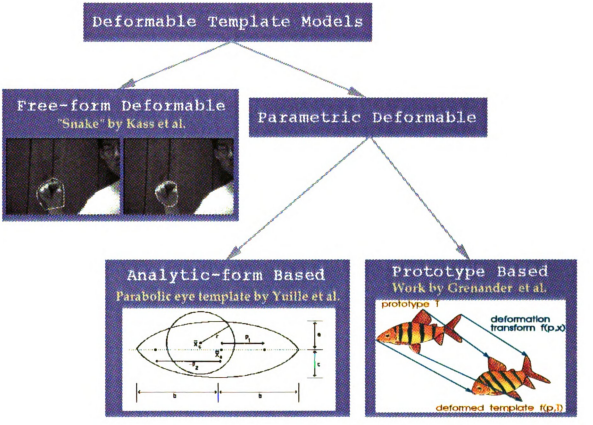


Figure 2.9: Overview of deformable template models.

spline, called a "snake", is controlled by a combination of

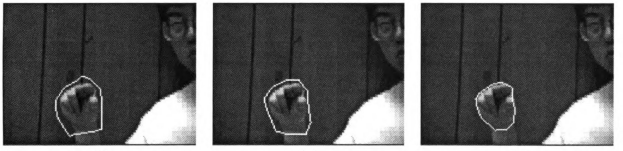
- the internal spline force which enforces the smoothness,
- the image force which attracts the spline to the desired features, and
- the external constraint force.

Each force creates its own potential field and the spline actively adjusts its position and shape until it reaches a local minimum of the potential energy:

$$\mathcal{E}_{snake} = \int_0^1 \{\mathcal{E}_{int}(v(s)) + \mathcal{E}_{image}(v(s)) + \mathcal{E}_{con}(v(s))\}ds, \tag{2.3}$$

where s is the parameterization of the contour, $v(s)$ is a point on the contour, and v_s and v_{ss} are the first and second derivatives of the contour, respectively. The internal energy of the spline, $\mathcal{E}_{int}(s) = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2$, characterizes the stretchness and smoothness of the snake. The image energy, $\mathcal{E}_{image}(v(s))$, represents the potential due to image forces, and $\mathcal{E}_{con}(v(s))$ represents the potential created by external constraint forces. The potentials are defined so that they decrease along the direction of the forces and there are low potentials near salient image features. Once an appropriate initialization of the contour is specified, the snake can quickly converge to the nearby energy minimum, using a variational method. The converged configuration is expected to give a sensible description of the object of interest. Fig. 2.10 presents an example of image segmentation using the active contour. In Fig. 2.10(a) a snake is initialized (manually) near the human hand, which is the object of interest. This snake is attracted to the salient edges of high gradient (Fig. 2.10(b)) and is

finally locked at the object boundary (Fig. 2.10(c)).



(a) (b) (c)

Figure 2.10: Image segmentation using the active contour approach. The potential is defined inversely proportional to the image gradient. **(a)** Input image: the snake is initialized around the object of interest. **(b)** Intermediate image: the snake is attracted to the salient edges by image gradient. **(c)** Final configuration: the snake converges to the object boundary of high image gradient.

This snake model provides a powerful interactive tool for image segmentation. However, the implementation of the original snake is vulnerable to image noise and the initial position. Numerous provisions have been made in the literature to improve the robustness and stability of the snakes [30, 89]. For example, Cohen [29] introduced a "balloon force" which can either inflate or deflate the contour. This force helps the snake to trespass spurious isolated weak image edges, and counters its tendency to shrink. The resulting snake is more robust to the initial position and image noise, but human intervention is needed to decide whether an inflationary or deflationary force is needed. Amini [3] and later Geiger *et al.* [48] suggested using dynamic programming to minimize the energy function. Their methods exhaustively search all the admissible solutions, and each iteration results in a locally optimum contour. As a result, this method is guaranteed to converge in a finite number of iterations. This idea of active contour has been successfully extended to perform tasks including edge and subjective contour detection, motion tracking, stereo matching and image

segmentation [149, 91, 112, 150, 155].

Optimal Active Region

While most active contour or snake approaches optimize a cost function based on an exterior boundary cost with no regard to the enclosed interior region, Cox et al. [34] examined how to apply the smoothness constraint globally and how to utilize both the boundary and interior information in image segmentation and interpretation. The implementation of their algorithm is based on a computationally efficient graph partitioning algorithm that minimizes the ratio between the exterior "boundary cost" and the interior "region benefit". For each greyscale image, they constructed a planar graph where the edges of the graph correspond to the between-pixel line processes in the image and the single faces (with 4 surrounding edges) correspond to the pixels (Fig. 2.11). In the figure, "X"s denote the pixels of the image, the blue nodes correspond to nodes of the graph, and the arcs connecting the nodes are the edges of the planar graph (they correspond to the line processes of the input image). Each arc of the graph is assigned a positive cost according to the intensity gradient. Each simple face of the graph is assigned a nonnegative "benefit" based on the greyscale value of the corresponding pixel. A closed path of the graph corresponds to a closed contour in the image. To find a good contour, i.e., a segmentation, in the image, they minimized the following objective function to obtain a partition in the graph:

$$\mathcal{L} = \frac{\sum_i cost(e_i)}{\sum_j benefit(f_j)}, \tag{2.4}$$

where $cost(e_i)$ denotes the cost of edge e_i, $benefit(f_j)$ denotes the benefit of face f_j, the summation in the numerator is over all the edges on the contour, and the summation in the denominator is over all the pixels (faces) enclosed by the contour. The edge cost on the contour is nonnegative, and inversely related to the intensity gradient, and the weight at a pixel is also nonnegative, and is assigned based on the intensity value according to different segmentation goals. For example, the "ratio regions" can be made to have an *a priori* preference for large, round objects bounded by high intensity gradients. Nevertheless, the algorithm may not find a small high contrast object in a large image if the edge costs and face benefits are incorrectly chosen. Figure 2.12 illustrates this point. Figure 2.12 (a) shows a large light rectangle on a dark background with white Gaussian noise added to the image. The black boundary shows that the white rectangle is easily segmented from the image. Figure 2.12 (b) is similar, but the white rectangle is now narrower with a correspondingly smaller area. In this circumstance, the algorithm fails to find the rectangle, but instead, finds a region with higher boundary cost but significantly larger area and correspondingly smaller ratio cost. However, we can alter the edge costs in a *non-linear* fashion in order to improve the segmentation. An example of this is shown in Figure 2.12 (c) where the edge costs have been squared. In this case, the correct segmentation has been found for the narrow rectangle.

Following are some of the characteristics of the "ratio region" approach.

- The algorithm is not iterative and finds the globally optimum closed contour. Though it is based on dynamic programming, the complexity of the algorithm

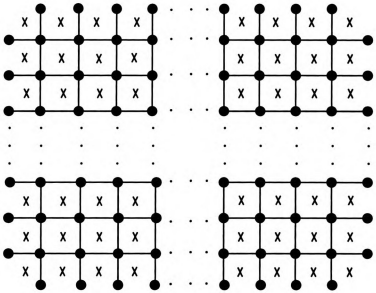
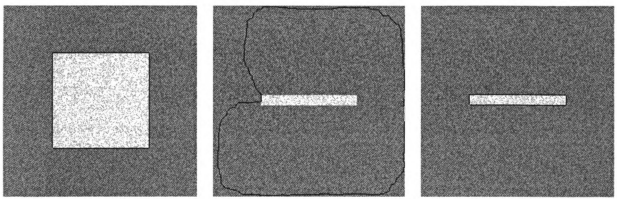


Figure 2.11: The planar graph constructed from the input image.



(a) (b) (c)

Figure 2.12: Synthetic images showing how the performance of the "ratio region" algorithm can degrade if objects with small area are desired.

is $\mathbf{O}(\mathbf{n} \log \mathbf{n})$, where $\mathbf{n}$ is the number of pixels in the search window.

- The smoothness of the contour is maintained in a novel way based on the global properties of the region's area and perimeter.

- Very little prior information is required. Only one node which passes the optimal contour is needed for the success of the approach. This condition can be relaxed by providing a node in the neighborhood of the object contour, and then use the node with the maximum gradient magnitude in its neighborhood as the root node. The process can also be fully automated by initiating the core algorithm at all salient edge pixels in the image.

- Region information can be incorporated into the objective function in a natural way. Not only does the border information contribute to the segmentation, but the internal intensity or homogeneity information is also useful.

A significant application of active contour models is an online, interactive segmentation of regions of interest in a given image. There is a clear advantage to minimize user interaction in a number of applications for various reasons including ease of use and robustness. Ratio snakes are empirically shown to allow very coarse initialization, e.g., only a single point on the desired contour or a bounding box enclosing the region of interest is needed. Figures 2.13(a)-(c) illustrate several examples of segmentation using the "ratio regions". In each of the examples, an image subwindow which contains the object of interest is specified. A point is specified (denoted by a cross) on the desired object boundary and the optimal closed contour in the subwindow which passes through the given point is then reported. Figure 2.13(a) illustrates that the

final optimum contour may be far away from the boundary of the subregion. Similarly, in Figure 2.13(c), it is unlikely that a traditional snake would find the hand based on the rectangular initialization; the presence of high contrast dark vertical lines between the boundary and the hand would almost certainly result in a strong local minimum near these structures.

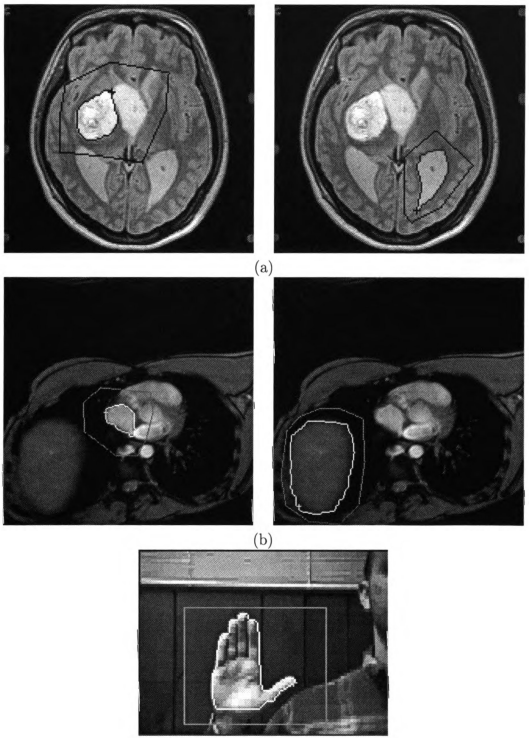
Elastic Matching

One of the early techniques for matching two deformed objects is the elastic matching method [10, 21, 44, 100]. In such approaches, the two objects are presegmented from the background. Correspondences are established for the two objects, and a match is attempted. For example, the elastic deformable model [10, 21, 100] establishes an elastic model for one of the two images to be matched. Then this image is "warped" iteratively towards the other one by some local forces (Fig. 2.14). Fischler and Eschlager [44] described a system that built up subtemplates that correspond to significant object features, and then searched for a match using a two-step process:

1. find subpart matches, and then

2. find matching configurations that satisfy relational constraints.

The applications of this model include handwritten numeral recognition, cartoon frame filling, alignment of deformed images and line drawings, motion detection, image registration [19, 20], stereo matching [99] and volume matching. Unfortunately, elastic matching has traditionally been computationally slow, has problems with correspondence, and is not robust in the presence of noise. Further, most of the



(a)

(b)

(c)

Figure 2.13: Image segmentation using the "Ratio Region" approach [34].

algorithms have not been automated.

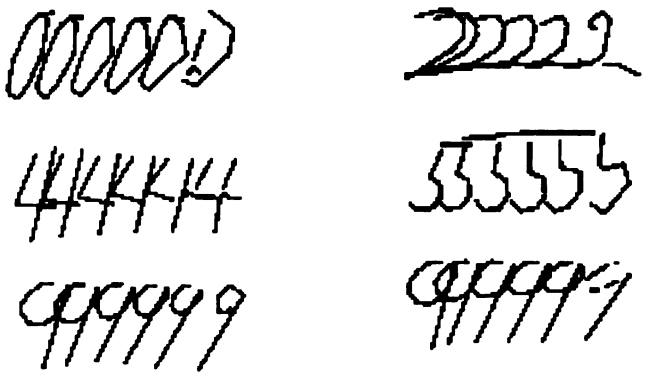


Figure 2.14: Examples of elastic matching. The reference shapes (left) are stretched onto some unknown shapes (right) iteratively by "forces" derived from local pattern matches [21].

2.2.2 Parametric Deformation Models

A parametric deformable template refers to the parametric shape model representing the *a priori* knowledge about the structural properties of a class of objects. By designing a global shape model, boundary gaps are easily bridged, and overall consistency is more likely to be achieved. By parameterizing the model, a compact description of the shape can be achieved. Furthermore, it is capable of representing a variety of shapes, and is relatively robust to image noise and distortion. Parametric deformation models are commonly used when some prior information of the geometrical shape is available, which can be encoded using preferably, a small number of parameters. There are two general ways to parameterize the shape class and its variations:

1. One can represent the shape as a collection of parameterized curves, i.e., parameterize the geometric shape directly. The template is represented by a set of

curves which is uniquely described by some parameters. The specific analytical form incorporates the prior knowledge of the shape of the objects of interest. The geometrical shape of the template can be changed by using different values of the parameters. The use of different parameter values gives rise to different shapes. Variations in the shape are determined by the distribution of the admissible parameter values. This representation requires that the geometrical shapes be well structured.

2. A so called "standard", or "prototype", or "generic" template is specified to describe the "most likely", or "average", or "characteristic" shape of a class of objects which has a global conforming structure and possibly, individual deviations. Each instance of the shape class is derived from the "prototype" via a parametric mapping. The use of different parameter values again gives rise to different shapes. Variations in the shape are also determined by the distribution of the admissible parameter values of the mapping.

In both the cases, the deformable templates interact with the image features dynamically by adjusting the parameters according to the image forces. Similar to the active contour approach, an objective function which is a weighted sum of an internal energy term and an external energy term is used to quantify how well a deformed template matches the objects in the given image. Recall that in the active contour approach, the internal energy, in terms of the stretchness and the elasticity of the spline, actually imposes a rather general and weak *a priori* distribution on the contour model, i.e., the contour should be smooth and compact. In the parametric

deformable template approaches, where the *a priori* shape preferences are explicitly encoded by the parameters, a similar internal energy term is defined based on the constraints and interactions on the geometrical structures. For example, it can be defined to penalize the deviation of the deformed template from the "expected" shape. The external energy term, which pertains to the fidelity of the deformed template to the input image, is introduced so that the template deforms according to the desired goal. It is perceived that the internal energy corresponds to a geometric measure of the fitness, and the external energy corresponds to an image fidelity measure of fitness. The two fitness measures are combined to give an overall measure of fitness, appropriately weighting both the prior knowledge and the image data. The set of parameters which optimizes the objective function gives a description of the detected or matched shape. The value of the objective function quantifies the plausibility of the detection.

Analytical Form-based Parametric Deformable Models

In some applications the geometric shapes of objects of interest can be approximated by a set of analytic curves of the same form, with different parameter values. We can handcraft the analytic model for such a shape class, and then find the description of a shape instance by determining the parameter values that best describe the shape.

One of the first instances of such a shape model is that of Widrow [148], where parameterized templates called "rubber masks" were used to describe 2D irregular shapes. The parameters were sizes and relationships between subparts of a shape.

Lakshmanan et al. [87] have used a parametric template model to locate the

airport runway boundary in radar images. In their work the *a priori* knowledge that the runway boundary consists of two straight, parallel edges are used to derive a global shape model for the runway. The runway boundary is modeled as two parallel straight line segments, parameterized by the slope and the two intercepts of the lines. The runway edge detection problem is formulated as a Bayesian estimation using a physics-based model of the radar imaging process based on the assumption that the runway boundary divides the image into three relatively homogeneous regions. The set of parameter values which maximizes the Bayesian a posteriori density determines the detected runway boundary. An alternative to the likelihood is based solely on the local image gradient on the template. It is observed that the latter typically gives a more accurate estimate of the road boundaries, but is more demanding on the initial position of the template (Fig. 2.15). The global deformable model helps in the runway detection because the model is able to integrate the local intensity homogeneity and gradient information and adjusts itself to the desired position. The use of the prior structural information contributes to its robustness to image noise. A typical edge detector does not work well here because the image is textured due to the noisy nature of the millimeter wave images.

A polygonal template is designed by Dubuisson *et al.* [79] to characterize a general model of a vehicle (Fig. 2.16 (a)). They derived an a priori probability distribution to constrain the template to be deformed within a set of allowed shapes of a typical vehicle. The likelihood function is constructed based on motion information and edge directionality so that the deformed template is contained in the motion area and its boundary coincides with salient edges in the input (highway traffic) image

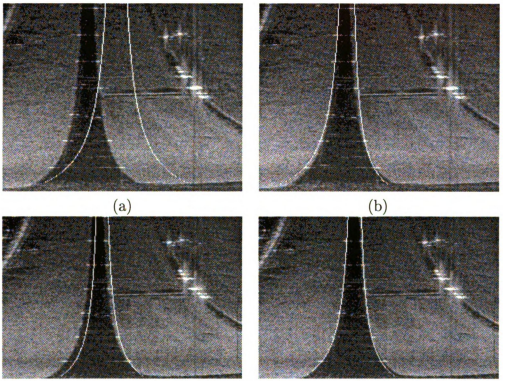
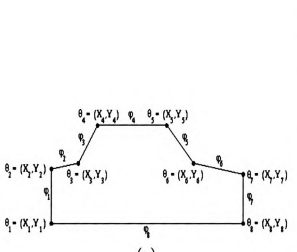
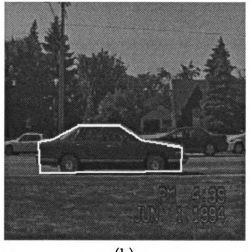


Figure 2.15: Runway detection using parametric deformable templates [87]. **(a)** initialization of the template; **(b)** the detected runway boundary using the region homogeneity assumption after the initialization of (a). **(c)** initialization of the template; **(d)** the detected runway boundary using the gradient information after the initialization of (c).

sequence. This approach was used to segment vehicles in images of traffic scenes. One segmentation result of this method is illustrated in Fig. 2.16 (b).



(a)



(b)

Figure 2.16: Vehicle segmentation using a polygonal template [79]. **(a)** the polygonal vehicle template and its parameterization; **(b)** one segmentation result using the template and motion and gradient informations.

Yuille et al. [151] have used deformable templates to extract facial features such as the eyes and the mouth. They designed parametric models for eye and mouth templates using circles and parabolic curves. The parameters which control the shape of a template are the center and the radius of the circle, and the characteristic parameters of the parabola. Regularization constraints are imposed on the parameters in terms of the size of facial features such as the mouth and the eyes, as well as the interactions between them, e.g., the center of the mouth template should be close to the line which is at equal distances to the centers of the two eye templates. The choice of the parametric forms for the feature templates and the interactions between the parameters should reflect the known structural information about the facial features. The image (external) energy term is defined in terms of edges, peaks, and valleys in

the input intensity image based on the features of the eyes and mouth so that different parts of the template interact with different image features such as intensity peaks and valleys. By using the deformable template model, the global geometry of the shape and different local image cues are integrated to give a comprehensive goodness-of-fit of the detection. This method gives reasonable detection and tracking results of the eyes and mouths in real images when the initial positions of the templates are sufficiently close to the desired objects.

Deformable boundary templates with more degrees of freedom were proposed by Staib *et al.* [121] to detect objects in medical images. As mentioned earlier, the prior model information can range from very general such as smoothness constraints to very specific such as an exact template. Staib et al.'s prior model is between the two extremes such that some prior information about the global shape is available but it is not exact. Elliptic Fourier descriptors are used to represent open or closed boundary templates which are smooth and are continuously deformable with no obvious decomposition. The parameters of the deformable templates are the Fourier coefficients. A distribution on the Fourier coefficients is specified so that there is a flexible bias towards some particular shapes. The spread of the distribution is governed by the variability among instances of the object class. A Bayesian decision rule is then used to obtain the optimal estimate of the boundary, where the likelihood function is based on the correlation between the template and the boundary strength in the input image.

For all the techniques discussed above, a good initialization of the contour is required for meaningful solutions. The approximate translation, orientation and scale

of the object to be segmented are supposed to be known. Furthermore, the initial contour implicitly biases the converged configuration. The applicability of the parametric deformable model is limited because the shapes under investigation have to be well-defined so that they can be represented by a set of curves with a preferably small number of parameters. The parametric forms can be *ad hoc*. In fact, the parametric curves can hardly fit some actual boundaries in real images no matter how the parameters are adjusted.

Prototype-based Parametric Deformable Models

The pattern theory proposed by Grenander [58] described a systematic framework to represent shape classes that exhibit a lot of variability but also possess a characteristic structure. Grenander and Keenan [60] formulate a global, pattern-theoretic model of shape which consists of:

(1) space of generators (G);

(2) connector graph (σ);

(3) bonding relations ($R = (\rho, A)$); and

(4) transformation group ($S,\ S : G \mapsto G$).

The generators G are the basic building blocks of the structure. For example, they can be edges or nodes for polygonal shapes, or pixels for greyscale image patterns. The connector graph σ, where the nodes correspond to each of the generators, describes the interactions between the generators. The bonding relations R apply the

geometric constraints so that the resulting configurations are physically meaningful. The transformation group S maps one generator to another, and is a mechanism to produce new structures from the given ones.

This framework provides a structured method to systematically generate very general shape classes. The appropriate choice of the generators, transformation group, connector graph, and regularity conditions depend upon the specific application. In general, the above shape model can be represented by:

- a model template which describes the overall architecture of the shape, and
- a parametric statistical mapping which governs the random variations in the building blocks of the shape [4, 28, 61, 60, 97].

These factors together should control the desired global and local geometry of the shape class. Usually, the prototype template is based on the prior knowledge of the objects, which can be either specified by the high-level knowledge, or obtained from training samples. The parametric statistical mapping is chosen to reflect the particular deformations allowed in the application domain.

The shape classes described by Grenander's pattern theory can be very versatile because of different choices of the prototype template and the deformation process [60]. They can be tailored for very different applications. For example, in their work on human hands, Chow *et al.* [28] used polygons to approximate the contour of human hands (Fig. 2.17). The building blocks in the shape model are the polygonal edges which meet the regularity condition because polygons are simple and connected. Variations in different hands are described by Markov processes on the edges. Chow

et al. applied this shape model for hand synthesis and restoration. Similar approaches have also been used to model leaves, 3-D chairs, and 3-D human organs. In another paper on restoration of human hands from noisy greyscale images, Amit *et al.* [4] used an intensity image to represent a typical human hand. All instances of the class of hands are obtained by applying a number of admissible continuous transformations to the "ideal" hand image. Furthermore, this set of continuous mappings is governed by a Gaussian distribution. The observed image is assumed to be corrupted by an additive noise process. The reconstruction is obtained by maximizing the posterior distribution.

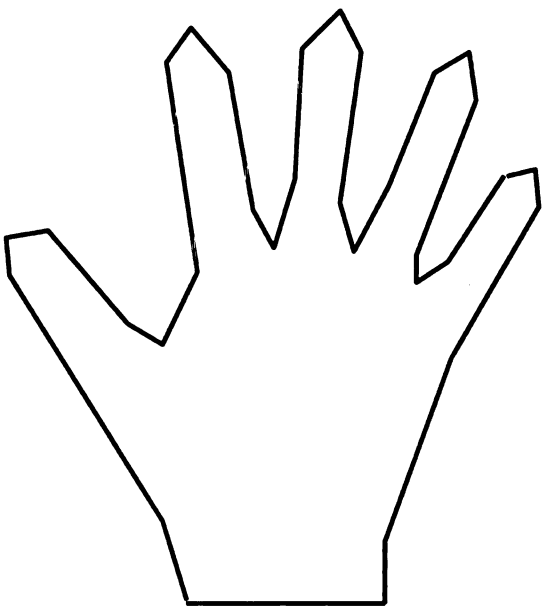


Figure 2.17: A polygonal hand template [59].

The limitations of this approach, as well as most other structured template matching methods, is that a good approximation of the location, orientation and scale of the object in the data should be provided. How to deal with the pose and the scale of the initial template is still an open problem.

Shape Modeling and Learning

The success of the structured deformable template matching approaches depends, obviously, on an accurate description of the shape class — the expected shape instances and their variations. This information, similar to the prior distribution in a Bayesian framework, can be subjective. Usually, it can be obtained from past experiences. It can also be computed from a representative set of shape instances. Some recent work on shape modeling has focused on the active learning of the shape models from training samples, influenced by the goals of "active vision". To describe a shape class, one has to learn both the "representative" shape and the "variability" in the shape class [33, 32, 82, 85, 107].

Cootes *et al.* [33, 32] proposed the "active shape models" for templates represented as line-drawings. By an "active shape model", they mean that instead of handcrafting the parametric form for the shape class, the prototype shape and its deformations are learned from a collection of correctly annotated example shapes. Basically, polygonal representations are used for shape modeling. By manually aligning the training set, i.e., establishing the correspondences between the "landmark points" (nodes) of training samples of the same class, they calculated the mean position and variation of each node from the training shapes. This mean shape is used as the generic template of the class of shapes. A number of modes of variation, i.e., the eigenvectors of the covariance matrix, are determined for describing the main factors by which the instance shapes tend to deform from the generic shape. A small set of linearly independent parameters are used to describe the deformation. In this way,

their shape model allows for considerable meaningful variability, but is still specific to the class of structures it represents. The major contribution of their work is that the active shape model is able to learn the characteristic pattern of a shape class and can deform in a way which reflects the variations in the training set. The limitations of the approach are its sensitivity to partial occlusion, and its inability to handle large scale and orientation change.

Kervrann and Heitz [82] proposed a deformable model which is very similar to Cootes *et al.*'s model. They presented an unsupervised approach to learn the structure and deformation modes of 2D polygonal objects, given long image sequences. They used a combination of both the global and local deformation modes to model a deformable shape. The global mode is the same as that in Cootes *et al.*'s work, i.e., is modeled by a generic shape plus the global deformation which is a linear combination of the variation modes obtained from principle component analysis. The local deformation, which is considered to contain additional information from the new image frame, is modeled by a Markov random process for the consecutive nodes, which takes into account interactions between the neighboring points. In the training stage, upon the processing of every new image frame, the computed local deformations are used to update the global average template and the global deformation modes. They applied this approach to object tracking. However, a good initial template is still required, and the convergence of the sequence is not guaranteed.

Given a set of representative shape examples, Pentland [105] and Pentland and Scaroff [107, 117] have proposed a novel shape modeling method using the finite element models (FEM). Although FEM is a well developed and powerful approximation

method for solutions to problems in material science and mechanical engineering, this is the first time it was used to solve a computer vision problem. They used 3-D finite element models which act like lumps of elastic clays to model 3-D shapes. They derive modes of vibration of a suitable base shape, such as an ellipsoid, and build up shapes using different modes of variation. The first few modes are the large-scale variations of the shape; the higher order modes are more localized (Fig. 2.18). A total of 30 modes were used to model human heads. They fitted models to range data by an interactive process, and compared modeled objects using fitted parameter values. The advantages of these models are that they are easy to construct and allow for a compact parametric representation of a family of shapes. Additionally, a close-form solution can be obtained for the complex 3-D shape modeling problem. However, this does not always lead to a compact description of the variability within a particular class of objects.

A 3D model which combined both local and global deformation is the deformable superquadrics proposed by Terzopoulos and Metaxas [128]. The global shape is modeled by a conventional superellipsoid with 6 parameters, which provides salient part descriptors for object recognition and database indexing purposes. The local deformation is modeled by a free-form spline, which reconstructs complex shapes that the global abstraction misses. The problem solving is physics-based, via translating visual data into external forces, and simulating the equations of motion through time to adjust the translational, rotational and deformational degrees of freedom of the model. However, a user has to handcraft parts for complex shapes. The number of degrees of freedom of the model can be large enough to make the problem in-

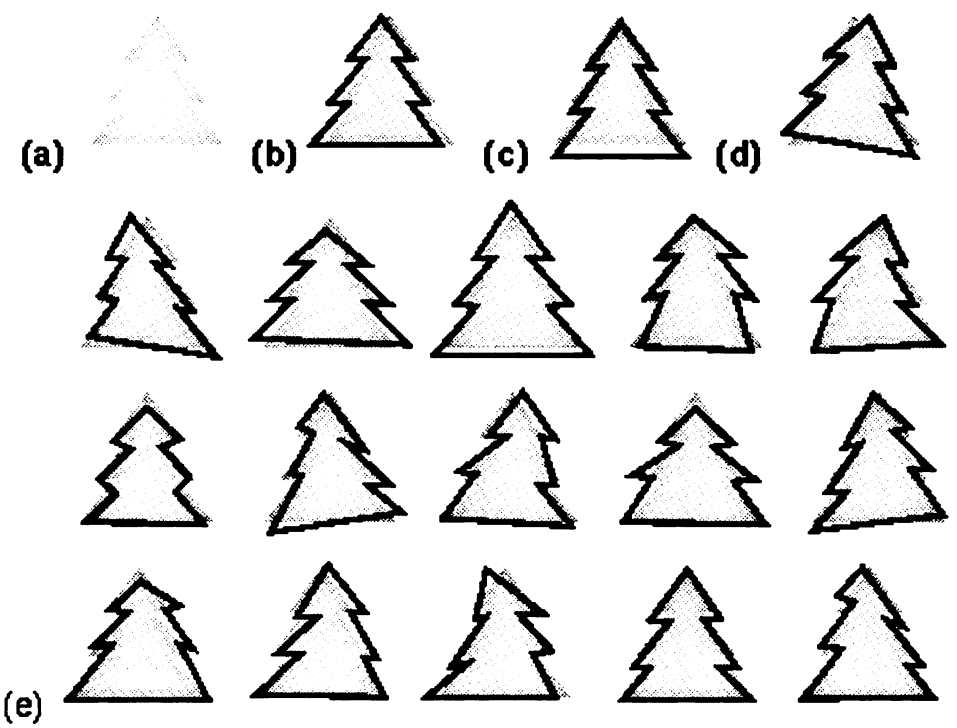


Figure 2.18: Example of the eigenmodes [117]. **(a)** an upright tree shape; **(b)-(e)** The lower 18 modes (black outline) for the shape in (a). (b) and (c) are the translation modes, (d) is the rotation mode. The rest are nonrigid variations.

feasible. This technique also suffers from the constraints present in all the existing approaches: a good initialization of the pose, scale, and deformation parameters is required to achieve reasonable results.

A general and comprehensive visual learning scheme is proposed by Weng et al. [35, 36, 126, 142, 143, 146, 144, 145]. They have proposed a system called "SHOSLIF" [143, 144] for *comprehensive* sensory learning which involves visual, auditory and other sensory information, where the term "comprehensive" here refers to both the comprehensive coverage of the sensory world and comprehensive coverage of the recognition algorithm. This system achieves a unified theory and methodology for comprehensive sensor-actuator learning with a logarithmic scalability. The SAIL project by Weng [145] is aimed at the development of the first "living machines" whose objectives include "development of a systematic theory and a practical methodology for machines to learn *autonomously* while interacting with its environment, on a daily basis, via its sensors and effectors, on-line in real time, under interactive guidance from human teachers."

2.3 Discussion

In the previous sections we have briefly surveyed recent work on deformable template modeling for 2D shapes. A summary of the discussed work is listed in Table 2.1.

The common difficulties that have been experienced in the application of existing approaches to deformable template matching are as follows:

- The algorithms need a good initialization to give meaningful results, otherwise,

Authors	Category	Prior Model	Imaging Model	Application Domain
Kass, Witkin and Terzopoulos [81]	free-form	Curves that satisfy regularity constraints	Potential field defined by salient features	Image segmentation, subjective contours
Cox, Satish and Zhong [34]	free-form	Closed contours that satisfy global regularities	Image gradient	Image segmentation
Lakshmanan, Jain and Zhong [87]	Analytical form-based parametric	Two parallel straight edges	Homogeneous regions following the lognormal distribution	Runway detection
Dubuisson, Lakshmanan, and Jain [79]	Analytical form-based parametric	A polygon that satisfies some constraints	Motion energy + edge potential	Vehicle segmentation and registration
Yuille, Hallinan and Cohen [151]	Analytical form-based parametric	Circles, lines, parabolic curves	Potential defined by salient features	Facial feature detection
Staib and Duncan [121]	Analytical form-based parametric	Elliptic Fourier descriptors	Image boundary strength	Medical image segmentation
Chow, Grenander and Keenan [28]	Prototype-based parametric	Polygon whose nodes form a Markov process	Two Gaussian distr. for pixels inside and outside the template	Hand synthesis and restoration
Cootes, Taylor and Lanitis	Prototype-based parametric	Polygon		Learning shape models
Jain, Zhong and Lakshmanan [76]	Prototype-based parametric	Bitmap	Potential field defined by edges	Object localization and matching

Table 2.1: A taxonomy of the deformable template matching approaches.

they get stuck at local minima and thus lead to incorrect results.

- The convergence of the algorithms to true solutions is slow due to the large number of parameters associated with the deformable template.

In the high-dimensional parameter space, the landscape of the objective functions is usually very complicated with many local minima. Also, because most algorithms utilize the gradient descent method in the entire space of high dimensionality, they are inevitably slow. Few reported works have dealt with the scale and orientation problems successfully. A fast, scale/orientation invariant solution is yet to be found.

Our deformation model falls in the category of prototype-based parametric deformation models. It shares some of the characteristics of the work by Amit et al. [4] and of Cootes et al. [33, 32], but has its unique properties which are appropriate for the specific application domain of interest to us. We represent the prototype template in the form of a bitmap which describes the characteristic contour/edges of an object shape. It is then deformed to fit salient edges in the input image by applying a probabilistic transformation on the prototype contour which maintains smoothness and connectedness. The matching is carried out by maximizing the *a posteriori* probability which combines both the prior shape information and the image information. A Bayesian framework was previously adopted for contour estimation [42, 124] where the prior is used to impose local smoothness and the likelihood is calculated based on edge positions. In our case, it is natural to choose a prior which reflects the global shape of the object of interest. The likelihood model that we use to fit the deformed template to the salient edges in the input image is similar to the ones used

in [42, 81, 124], but the exact functional form of our likelihood is different because it incorporates both the edge position and directional information to give a better edge localization. Details of the deformation and likelihood models are given in chapter 4.

Chapter 3

Unifying Deformable Models in a Bayesian Framework

Statistical approaches to image analysis using the Bayesian paradigm have been very popular in recent years. This paradigm has been primarily developed for situations when prior knowledge of a process is available and that knowledge needs to be combined with the sensed data to make statistical inference about the parameters of the process. This methodology has been very successful in integrating low-level image analysis and high-level tasks. Its application domain in computer vision and image processing includes image restoration [51], segmentation [27], shape modeling [108], and inference [16].

A number of researchers have noticed the links between the Bayesian framework and deformable models and tried to obtain a general solution using the Bayesian framework [95, 130, 124]. In this chapter, we investigate the relationship between deformable matching methods and the Bayesian models. We further conclude that

we can tailor the prior distribution and the likelihood in a Bayesian scenario so that the corresponding Maximum A Posteriori (MAP) solution is equivalent to the solution of the original deformable matching problem, irrespective of whether the problem is an active contour problem, or a parameterized deformable matching problem.

3.1 Bayes' Theorem

The Bayes' rule is "a system for modifying historical information on a process in the light of current data" [45]. Let the initial (prior) knowledge about a process be characterized by a density function on the parameters $\mathbf{u}$ of the process, $f(\mathbf{u})$. Let the current conditional density of observed data $\mathbf{d}$ given $\mathbf{u}$ be $f(\mathbf{d}|\mathbf{u})$. According to Bayes' rule, the posterior density function of parameters $\mathbf{u}$, given the observed data $\mathbf{d}$, is

$$f(\mathbf{u}|\mathbf{d}) = \frac{f(\mathbf{u})f(\mathbf{d}|\mathbf{u})}{\int_{\mathbf{u}} f(\mathbf{u})f(\mathbf{d}|\mathbf{u})du} \tag{3.1}$$

or, equivalently,

$$f(\mathbf{u}|\mathbf{d}) = \frac{f(\mathbf{u})f(\mathbf{d}|\mathbf{u})}{f(\mathbf{d})}, \tag{3.2}$$

where $f(d) = \int_{\mathbf{u}} f(\mathbf{u})f(\mathbf{d}|\mathbf{u})du$.

In the Bayesian model for computer vision applications, the prior model typically represents the initial knowledge about the objects in a particular scene and the likelihood function represents the joint probability distribution of the sensed data (image), conditioned on the objects in the scene. By applying the Bayes theorem, the posterior distribution of the object in the scene is obtained for inferencing purposes such as

segmentation and classification.

3.2 Bayesian Formulation for Deformable Models

We note that in almost all the deformable models, the objective function consists of two parts:

- The first term, which is referred to as the *internal energy*, *prior*, or *geometrical* information, is related only to the geometric shape of the deformed template (contour) which is an intrinsic property of the template, independent of the input image:

    1. In free-form deformation models such as active contour, this term is specified in terms of the "elasticity" and "stretchness" due to the weak regularization constraints on the contour. It is equivalent to a "prior" or "preference" for smooth and compact contours.

    2. In the first category of the parametric deformation models (Sec. 2.2.2), this term is specified in terms of the model parameters, which reflects either the choice of the parameter values, or the interactions between different parts of the template; this again determines the prior knowledge of the shape, in terms of the geometrical or contextual constraints, independent of the input image.

    3. In the prototype-based deformation model, this term is also a function of the deformation parameters. It penalizes the deviation from the expected

shape, and biases the choice of the geometric shape.

- The second term, in all the cases, pertains to the input image data. Via this term, the deformable model interacts with the image, being attracted to the desired salient image features. This term measures the fidelity, or goodness of fit, to the input image.

The deformed template which optimizes the objective function leads to an interpretation of the image. Although in some of the studies, the deformable process is performed in a Bayesian framework, we note that it can be generalized to almost all the deformable modeling problems. In fact, all the energy minimization problems which consist of an internal and an external energy term can be cast as an inference task in a probabilistic framework. To be more specific, almost all the deformation models discussed above can be interpreted in terms of Bayesian estimation in *Eq. (3.2)*, using a prior $f(\mathbf{u})$ which characterizes the intrinsic properties of the deformable template $\mathbf{u}$, and a likelihood $f(\mathbf{d}|\mathbf{u})$ which relates the template to the sensed data $\mathbf{d}$.

When the deformable template modeling is cast in the Bayesian framework using *Eq. (3.2)*, the *prior* model $f(\mathbf{u})$ is a probabilistic description of the quantity we are trying to estimate before any image data becomes available. It typically imposes the geometrical preferences of the shape model. The *imaging* model $f(\mathbf{d}|\mathbf{u})$ is a description of the noisy or stochastic process that relates the deformed template to the input image or sensor values $\mathbf{d}$. This likelihood captures the desired image cues. Bayes' rule combines these two probabilistic models to form a *posterior* model $f(\mathbf{u}|\mathbf{d})$

which describes probabilistically the best estimate of $\mathbf{u}$ given the data $\mathbf{d}$ and prior knowledge of $\mathbf{u}$. Note that $f(\mathbf{d})$ is a constant, given $\mathbf{d}$. Therefore, maximizing the posteriori density in *Eq. (3.2)* is equivalent to maximizing the product $\{f(\mathbf{d}|\mathbf{u})f(\mathbf{u})\}$.

Intuitively, the internal energy term, which is a measure of the geometrical structure on a deformed template or contour, is related to the prior model in the Bayesian formulation; the external energy term, which describes the interaction between the template and the image, corresponds to the likelihood model. The prior and sensor models are determined according to the applications and goals. In the following subsections we address the details of the Bayesian formulation for each of the deformable template models.

3.2.1 Free-form Deformable Models

The unknown quantity $\mathbf{u}$ in *Eq. (3.2)*, in the free-form deformation model, controls the relative positions of the nodes on the contour to enforce the "stretchness" and "compactness". The prior can be specified in terms of the regularity constraints of the spline, and the likelihood can be derived from the image potential energy. It has been pointed out [124, 130] that by using a Gibbs distribution for the prior model

$$p(\mathbf{u}) = \frac{1}{Z_p} \exp\left[-\mathcal{E}_{int}(\mathbf{u})\right], \tag{3.3}$$

where $\mathcal{E}_{int}$ is the internal energy of the snake as defined in *Eq. (2.3)*, and a Gibbs distribution for the sensor model

$$p(\mathbf{d}|\mathbf{u}) = \frac{1}{Z_d} \exp\left[-\mathcal{E}_{image}(\mathbf{u}, \mathbf{d})\right], \tag{3.4}$$

where $\mathcal{E}_{image}(\mathbf{u}, \mathbf{d})$ is the external energy as defined in *Eq. (2.3)*, the maximum a posteriori (MAP) estimate, i.e. the estimate of $\mathbf{u}$ which maximizes the conditional probability

$$\begin{aligned} p(\mathbf{u}|\mathbf{d}) &\propto p(\mathbf{d}|\mathbf{u})p(\mathbf{u}) \\ &\propto \frac{1}{Z_p} \exp\left[-\mathcal{E}_{int}(\mathbf{u})\right] \frac{1}{Z_d} \exp\left[-\mathcal{E}_{image}(\mathbf{u}, \mathbf{d})\right] \\ &\propto \exp\left[-(\mathcal{E}_{int}(\mathbf{u}) + \mathcal{E}_{image}(\mathbf{u}, \mathbf{d}))\right], \end{aligned} \tag{3.5}$$

is the same as the minimum energy configuration in *Eq. (2.3)* of the snake.

3.2.2 Analytical Form-based Parametric Deformation Models

The unknown quantity $\mathbf{u}$ in *Eq. (3.2)* is the set of parameters of the analytical deformation models. The Bayesian formulation for such deformable template models can be derived in a similar manner. The prior is an appropriate probabilistic distribution on the model parameters, which encodes the knowledge of the shape variations and the contextual constraints, independent of the imaging process. The most often used prior densities include the uniform distribution and the Gaussian distribution.

The sensor model depends on the imaging process, and a large diversity of stochastic models can be used according to the specific goal. For example, in detecting straight edges in millimeter-wave images [87], the analytical template edge image is formed by assuming that the observed millimeter-wave image consists of two straight and parallel edges which are determined by three parameters: the slope of the two edges, k, and the intercept of each edge, c_1 and c_2 (Fig. 3.1). These parameters are assumed to be equally likely in their domain (uniform prior density).

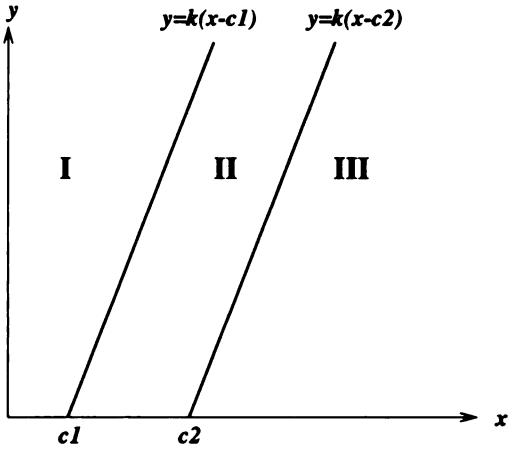


Figure 3.1: Parameterization of the runway boundary template. The runway boundary template consists of two parallel straight lines $y = k(x - c_1)$ and $y = k(x - c_2)$, with parameters k for the slope and c_1 and c_2 for the intercepts.

We assume that all deformations of the template that will keep the two edges within the confines of $\mathcal{L}$ are equally probable. In other words, the prior probability density is a uniform distribution on the set of values of k, c_1, and c_2, which satisfy the constraint that the two deformed edges must fall into the image region. The likelihood function, is determined based on the assumption that the greyscale values in three image regions (I,II and III) separated by the two straight edges follow a log-normal distribution in each region, which incorporates the essential degradations associated

with the millimeter-wave imaging process [86]. More specifically, given a hypothetical pair of underlying edges ($\{k, c_1, c_2\}$ values) we assume that the likelihood of observing the millimeter-wave image Y is given by:

$$P(Y \mid \{k, c_1, c_2\}) = \alpha e^{-E(Y,\{k,c_1,c_2\})}, \tag{3.6}$$

where α is a normalizing constant and $E(Y, \{k, c_1, c_2\})$ denotes the energy function related to the log-normal distribution:

$$E(Y, \{k, c_1, c_2\}) = \sum_{(r,\theta)\in\mathcal{L}} \log[\sigma_{r\theta}]^2 + \frac{1}{2[\sigma_{r\theta}]^2}(\log Y_{r\theta} - \frac{\mu_{r\theta}}{r^3})^2\}. \tag{3.7}$$

The quantities $\mu_{r\theta}$ and $\sigma_{r\theta}$ are assumed to be uniform over the three regions separated by the two straight edges. Their values are estimated adaptively from the data by using sample statistics over the respective regions. The r^3 divisor incorporates the deterministic range-dependent degradation in the data, and the $(\mu_{r\theta}, \sigma_{r\theta})$ pair incorporates the texture-like random variability in it.

3.2.3 Prototype-based Parametric Deformation Models

The Bayesian formulation for the prototype-based deformable template models can be derived similarly as in Section 3.2.2. The prior model is an appropriate probabilistic distribution on the model parameters, which encodes the knowledge of the shape variations and the contextual constraints, independent of the imaging process. The sensor model depends on the imaging process and again, a large diversity of stochastic

models can be used according to the specific goal.

Let the prior knowledge about the object of interest be represented by an ideal prototype $\mathcal{T}_0$. Let the possible deformations on $\mathcal{T}_0$ be described by a set of parameters $\mathcal{P}$. A probability distribution is assigned to the parameters with density $\pi(\mathcal{P})$, which models the allowed variations in deformable template $\mathcal{T}$, or equivalently, $\mathcal{P}$. This is the idea of probabilistic deformation model. It is through the prior model $\pi(\mathcal{P})$ that the intrinsic constraints and variations are expressed.

The imaging model describes the dependence of the observed image on the template. The appropriate model depends on the specific matching problem. Let $\mathcal{L}(\mathcal{I}|\mathcal{P})$ be the likelihood of observing image data $\mathcal{I}$ given a deformed template determined by deformation transform $\mathcal{P}$, the posteriori density $\pi(\mathcal{P}|\mathcal{I})$ of the deformed template given observed image $\mathcal{I}$ is proportional to

$$\mathcal{L}(\mathcal{I}|\mathcal{P})\ \pi(\mathcal{P}). \tag{3.8}$$

The solution is obtained by the maximum a posteriori (MAP) estimate of the true object scene. In chapter 4 we will formulate our approach using this methodology.

3.3 Discussion

By using the probabilistic model for the deformable template problem, we can formulate the problem in a structured fashion. The physical sensor model can be easily integrated with the prior knowledge of the configuration. Other advantage of using

the Bayesian framework is that confidence levels can be attached to the results for image interpretation and inference.

Chapter 4

A Shape-based Deformation Model

We propose a deformable model which is appropriate in situations where an inexact knowledge about the shape of the object of interest is available, and this shape information can be represented in the form of a hand-drawn sketch. Such an approach can be used in image segmentation, and object localization and detection, when the specific global shape model is given and need to be combined with the local image features. It can also be applied to content-based image database retrieval systems, where queries often include the shape of the object of interest. The user may describe the shape of an object using a sketch and ask to retrieve all the images in the database that contain such a shape. The sketch used to describe the shape does not need to match the object boundaries in the image exactly (Fig. 1.1). It is important that a retrieval system be robust to position, orientation, scale differences, and most importantly, moderate deformations of the object shape.

This problem presents several difficulties. First of all, the object to be located in the image may be different from the prespecified shape by possible deformations

which cannot be explained by a combination of translation, rotation, and scaling transforms. Secondly, the shape to be matched is not pre-segmented from the image. So, the matching process has to be integrated with segmentation. This situation is not as easy as the problem of matching two shapes based on the similarity of feature vectors. Thirdly, the approximate pose and scale of the object are unknown, as well as the number of instances of the objects in the input image. Fourthly, the shape model should be general enough to handle shapes of different appearances and connectivity. Lastly, since an inexact global shape is provided, we need to sensibly integrate this knowledge with the available image information.

To deal with the above problems, we propose a deformation model which consists of

- a prototype template which describes a representative shape of a class of objects in terms of a bitmap sketch,

- a set of parametric transformations which deform the template, and

- a probability distribution defined on the set of deformation mappings which biases the choice of possible deformed templates.

This deformation model can represent object classes of similar characteristics, and incorporate variations in the class. We will discuss all the three components in more detail in the following sections.

4.1 Representation of the Prototype Template

The prototype template consists of a set of points on the object contour, which is not necessarily closed, and can consist of several connected components. We represent such a template as a bitmap (binary image), with bright pixels (grey level of 255) lying on the contour and dark pixels (grey level of 0) elsewhere (Fig. 1.1 (a)). Such a scheme captures the global structure of a shape without specifying a parametric form for each class of shapes. This model is appropriate for general shape matching since the same approach can be applied to objects of different shapes by drawing different prototype templates. We acknowledge that this deformation model falls in the systematic framework of shape modeling described in Grenander's pattern theory [58].

4.2 Deformation Transformations

The prototype template describes only one of the possible (though most likely) instances of the object shape. Therefore, it has to be deformed to match objects in images. Deformation transform is an important component of the deformable template matching algorithm. It determines the admissible deformation space, or equivalently, the possible shapes that a deformable template can take. In theory, the deformation transform can be any function which matches a 2D point to another 2D point, as it is used to approximate the displacement in a 2D plane. But, a good deformation transform should be capable of representing a variety of shape variations, providing a concise description, preferably with a computational advantage, and preserve the

smoothness and connectivity of the template. We have used three different deformation transforms to span the displacement field, namely, the trigonometric basis in the 2D continuum, and the spline basis and wavelet basis in the curve space. Each of the deformation basis has its advantages and disadvantages. We will discuss each of them in details in the rest of this section.

4.2.1 Two-Dimensional Trigonometric Basis

This deformation basis is defined on the 2D continuum. Imagine that the template is drawn on a 2D planar rubber sheet which has a fixed boundary, but it can be deformed by stretching, squeezing, and twisting locally in the interior. As the rubber sheet deforms, the template drawn on it also changes its shape. The deformed rubber sheet can be obtained by applying a continuous mapping which maps the domain of the 2D image onto itself. The resulting 2D displacement field is represented as a continuous 2D vector function with certain boundary conditions. Without a loss of generality, we assume that the template is drawn on a unit square $\mathcal{S} = [0,1]^2$. The points in the square are mapped by the function $(x,y) \mapsto (x,y)+(\mathcal{D}^x(x,y), \mathcal{D}^y(x,y))$, where the displacement functions $\mathcal{D}^x(x,y)$ and $\mathcal{D}^y(x,y)$ are continuous and satisfy the following boundary conditions: $\mathcal{D}^x(0,y) \equiv \mathcal{D}^x(1,y) \equiv \mathcal{D}^y(x,0) \equiv \mathcal{D}^y(x,1) \equiv 0$. The space of such displacement functions is spanned by the following orthogonal bases [4]:

$$\begin{aligned} \mathbf{e}^x_{mn}(x,y) &= (2\sin(\pi n x)\cos(\pi m y), 0) \\ \mathbf{e}^y_{mn}(x,y) &= (0, 2\cos(\pi m x)\sin(\pi n y)), \end{aligned} \tag{4.1}$$

where $m, n = 1, 2, \ldots$. These basis functions, which consist of trigonometric functions of different frequencies, vary from global and smooth to local and "coarse" as m and n increase, as shown in Fig. 4.1.

Specifically, the displacement function is specified as follows:

$$\mathcal{D}(x,y) = (\mathcal{D}^x(x,y), \mathcal{D}^y(x,y)) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{\xi^x_{mn}\cdot \mathbf{e}^x_{mn} + \xi^y_{mn}\cdot \mathbf{e}^y_{mn}}{\lambda_{mn}}, \tag{4.2}$$

where $\lambda_{mn} = \alpha\pi^2(n^2+m^2)$, $m, n = 1, 2, \ldots$ are the normalizing constants. The parameters $\underline{\boldsymbol{\xi}} = \{(\xi^x_{mn}, \xi^y_{mn}), m, n = 1, 2, \ldots\}$, which are the projections of the displacement function on the orthogonal basis, uniquely define the displacement field, and hence the deformation. We use a finite number of terms in the infinite series in *Eq. (4.2)* as the displacement function for the deformation mapping:

$$\mathcal{D}_{\underline{\boldsymbol{\xi}}}(x,y) = \sum_{m=1}^{M}\sum_{n=1}^{N} \frac{\xi^x_{mn}\cdot \mathbf{e}^x_{mn} + \xi^y_{mn}\cdot \mathbf{e}^y_{mn}}{\lambda_{mn}}. \tag{4.3}$$

Note that the dependence of the displacement $\mathcal{D}$ on the deformation parameter vector $\underline{\boldsymbol{\xi}}$ is made explicit in *Eq. (4.3)*. This continuous function preserves the connectedness of the prototype template. It also preserves the smoothness of the template when M and N are not too large (only low frequency components are used). It should be noted that the length of the deformable template varies depending on the deformation. Note also that we are only concerned with the displacements of the points on the prototype template. Figure 4.2 illustrates the deformations of a bird template sketched on a grid using the displacement functions defined in *Eq. (4.3)*. One can see that the

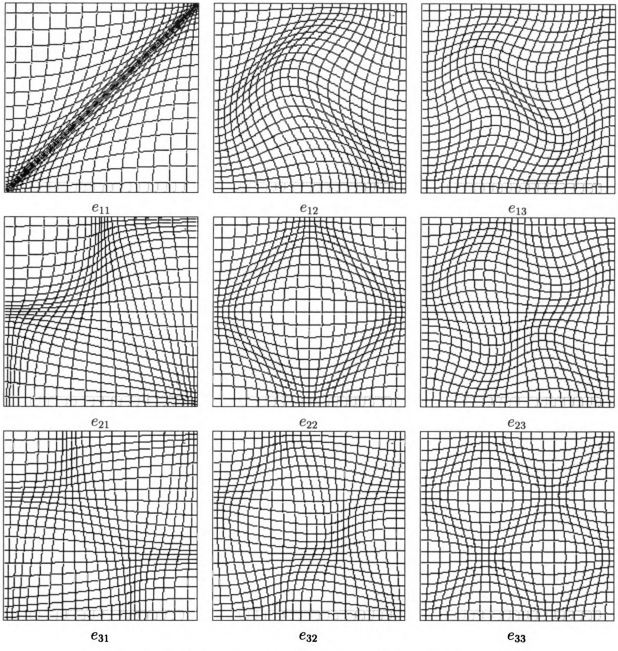


Figure 4.1: Basis functions for the deformation displacement field.

deformation becomes more complex as higher frequency components are added to the displacement function.

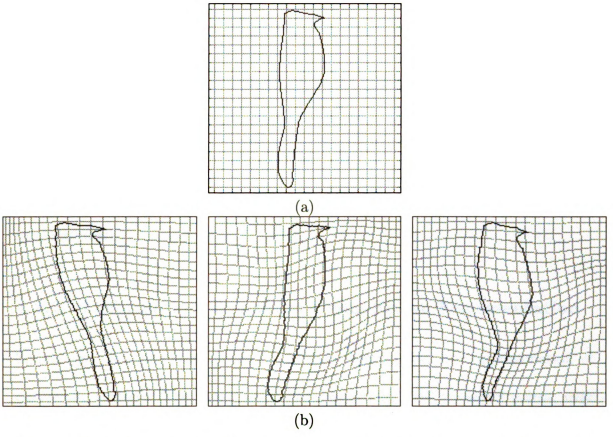


Figure 4.2: Deformation of a bird template using the 2D trigonometric basis. **(a)** the bird template with no deformation; **(b)** deformed bird template using randomly generated deformation parameter values. From left to right, the interpolation level (M, N) equals 1, 2, and 3, respectively.

This choice of the deformation transform basis has the following properties:

- The basis set in *Eq.* (4.1) is defined on the 2D continuum. So, it imposes very few constraints on the prototype template. The template can be open or closed, simply-connected or multiply-connected. Only the deformation on the template pixels are computed; we do not need to compute the displacement on the rest of the 2D domain. This deformation transform gives the most flexibility about

the template shape;

- All the basis functions, whether containing low or high frequencies, are global. As a result, they may not approximate local shape features well.

The global property of the deformation basis in *Eq.* (4.1) is not desirable in some applications because of their incapability to model local, uncorrelated changes. The other two deformation basis we define in Sections 4.2.2 and 4.2.3 have local compact support. They are expected to outperform the 2D trigonometric basis in modeling local features.

4.2.2 Deformation Transform Using Spline Representation

Splines are piecewise polynomials for efficient interpolation and approximation of curves and surfaces. They are usually characterized by a set of control points, with certain continuity requirements at the boundary. Splines have the desirable properties of continuity, bounded support, spatial uniqueness, and local controllability. Splines have been commonly used for shape modeling [43, 81, 116]. We find the spline approximation of the prototype template, and then deform it by displacing the control points. We can achieve local deformation by using a spline basis for compact local support.

B-splines

The B-spline is a well-known spline function that provides local approximations to contours/surfaces using a small set of control points. A kth order B-spline is $\mathcal{C}^{k-1}$

continuous, i.e., it is continuous up to $(k-1)$ derivatives. The degree-m B-spline representation of a curve is given as follows:

$$f(t) = \sum_{i=0}^{k-m-1} c_i \mathcal{B}_i^m(t), \quad t \in [t_m, t_{k-m}], \tag{4.4}$$

where $\mathcal{B}_i^m(t)$ are the degree-m B-spline basis, and c_i's are parameters of the representation (control points).

For a closed contour, we need to guarantee that the spline representation is periodic. This can be done by periodically extending the knot sequence $\{t_j,\ j \in \{0, 1, \ldots, k-1\}\}$ such that:

$$\tilde{t}_j = t_{j \bmod k}, j \in \mathcal{Z}, \tag{4.5}$$

and accordingly, periodically expanding the aperiodic B-spline basis $\{\mathcal{B}_i^m(t), j = 0, 1, \ldots, k-1\}$ with a period $(t_k - t_0)$ to obtain the periodic B-spline basis $\{\tilde{\mathcal{B}}_i^m(t), j = 0, 1, \ldots, k-1\}$. The closed (periodic) function can then be represented as:

$$f(t) = \sum_{i=0}^{k-m-1} c_i \tilde{\mathcal{B}}_i^m(t), \quad t \in \mathcal{R}. \tag{4.6}$$

B-spline Representation for Deformable Templates

The B-spline gives a parameterized representation of a contour, where the control points are the parameters. The B-spline representation of a contour is somewhere between a bitmap and a parametric form. A free-form bitmap has the most degrees of

freedom (each pixel location can be considered as a parameter), and the least structure (it can assume any shape). A parametric form, such as an ellipse or a parabola, can be determined by a small number of parameters, and has the most rigid structure with very few degrees of freedom. With a spline representation, we can change the parameter values to deform the contour locally. It is also more structured than a bitmap, with the shape controlled by the set of control points.

To represent a 2D contour, its x and y coordinates are fitted with a separate B-spline basis:

$$[f_x(t)\ f_y(t)] = \sum_{i=0}^{k-m-1} \mathbf{c_i} \mathcal{B}_i^m(t), \quad t \in [t_m, t_{k-m}], \tag{4.7}$$

where $\mathbf{c_i} = [c_i^x, c_i^y]$ are 2D vectors. Accordingly, a closed contour can be expressed as:

$$[f_x(t)\ f_y(t)] = \sum_{i=0}^{k-m-1} \mathbf{c_i} \tilde{\mathcal{B}}_i^m(t), \quad t \in \mathcal{R}. \tag{4.8}$$

To represent a deformable template using the splines, we should have (i) a default template (prototype) and (ii) a way to code the variation in shape. This can be achieved by imposing a probabilistic distribution on the parameters (control points). We have used an *i.i.d.* Gaussian distribution for this purpose. The set of average control points determines the prototype. If a bitmap is given to describe the default shape, we can fit a B-spline to the contour and use the estimated control points as the mean. If training samples (shapes) are available, we can learn the prototype from the samples. Let the set of default control points be $\{\mathbf{c_{0i}},\ i = 0, \ldots, k-m-1\}$, and the B-spline basis be $\{\mathcal{B}_i^m(t),\ i = 0, \ldots, k-m-1\}$, then the prototype is determined

by:

$$[x_0(t)\ y_0(t)] = \sum_{i=0}^{k-m-1} \mathbf{c_{0i}} \mathcal{B}_i^m(t), \quad t \in [t_m, t_{k-m}]. \tag{4.9}$$

The deformed template can then be derived by perturbing the control points around the default ones. The deviations are penalized so that small deformations are preferred over large ones. Let the deviations from the default control points be $\Delta \mathbf{c_i} = [\xi_i^x\ \xi_i^y]$. The deformed template is computed as:

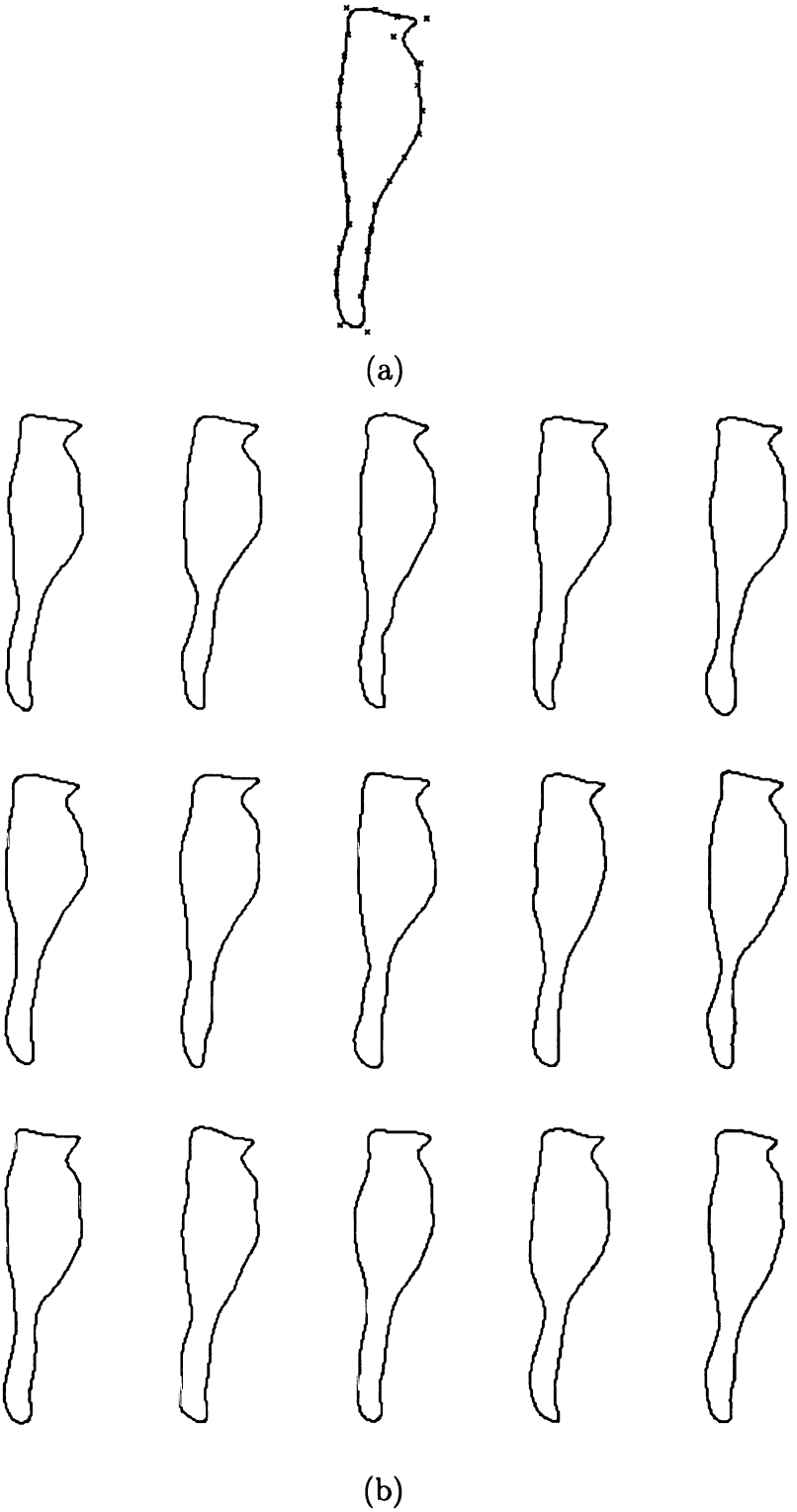
$$[x(t)\ y(t)] = \sum_{i=0}^{k-m-1} \mathbf{c_i} \mathcal{B}_i^m(t) = \sum_{i=0}^{k-m-1} (\mathbf{c_{0i}} + \Delta \mathbf{c_i}) \mathcal{B}_i^m(t), \quad t \in [t_m, t_{k-m}]. \tag{4.10}$$

We note that the B-spline representation of a deformable template is equivalent to approximating the deformation displacement using the B-spline basis functions. The displacements of the template pixels due to the disturbance $\underline{\boldsymbol{\xi}}$ in the control points are computed as follows:

$$\mathcal{D}_{\underline{\boldsymbol{\xi}}}(t) = \sum_{i=0}^{k-m-1} ([\xi_i^x\ \xi_i^y]) \mathcal{B}_i^m(t), \quad t \in [t_m, t_{k-m}], \tag{4.11}$$

where the deviations $(\xi_i^x,\ \xi_i^y)$ from the default control points are the deformation parameters.

To illustrate the deformations allowed by the spline representation, we show in Fig. 4.3 some deformed versions of a template using randomly generated control points which follow an *i.i.d.* Gaussian distribution with means equal to the default control points of the prototype. Fig. 4.3(a) is the B-spline approximation of the bird template using 30 control points. Its deformed versions are demonstrated in Fig. 4.3(b).



(a)

(b)

Figure 4.3: Deformations using the B-spline representation. **(a)** The spline representation of the prototype. The red dots are the control points. **(b)** Deformed templates obtained by randomly displacing the control points in (a) according to an *i.i.d.* Gaussian distribution.

We note that the spline-based deformation is able to model local deformations because the deformation basis functions are local. However, it requires a 1D parameterization of the template. So, it constraints the template to consist of a single component.

4.2.3 Deformation Using Wavelet Transforms

Interest in wavelets has been steadily growing over the past 15 years. It has turned out to be extremely successful in many computer vision and image processing applications involving compression, texture analysis, multiscale processing, image coding and restoration [7, 92, 115, 135, 139].

In the wavelet decomposition, fine scale features are captured by "narrow" functions with a small support, and coarse scale features are represented by "wide" functions with a large support. All these functions are either dilated or shifted versions of the so called *mother* wavelet. Each level in the wavelet decomposition corresponds to the difference (or *detail*) between two successive approximations. With such a layered structure, details at different levels can be added to obtain image representations at different resolutions. The wavelet approach fits naturally into the framework of multiscale image processing.

There are several advantages of using the wavelet transforms [37, 38] to model the shape deformations:

1. The wavelet basis is constructed in such a way that the elements with low indices have a large support and, therefore, allow for large-scale global adjustments in

the displacement field; on the other hand, the elements with higher indices have a smaller support and hence allow for local adjustments for small range abnormalities. With the hierarchy of coarse to fine wavelets, we can control the level of smoothness and locality in a coordinated way.

2. The wavelet transforms in the compact case can be computed very fast (in linear time complexity) using quadrature mirror filter (QMF) banks. This fast transform is reversible.

3. There is a relationship between the rate of decay of the coefficients in the expansion of a function and the degree of smoothness of that function.

We use the wavelet basis to span the deformation displacement field on the object contour.

Wavelet Transform

Let $\mathcal{A}_m$ be a linear, orthonormal projection on the vector space $\mathcal{V}_m$. Let $\mathcal{L}^2(\mathcal{R})$ be the vector space of measurable, square-integrable 1D functions $f(t)$. A multiresolution approximation of $\mathcal{L}^2(\mathcal{R})$ is defined as any set of vector spaces $(\{\mathcal{V}_m\},\ m \in \mathcal{Z})$ which satisfies the following properties [93]:

1. The approximation of a function $f(t) \in \mathcal{L}^2(\mathcal{R})$ at a resolution $(m+1)$ contains all the information to compute the same function at a lower resolution m, i.e., the subspaces of the multiresolution approximation are nested:

$$\mathcal{V}_m \subset \mathcal{V}_{m+1}, \quad m \in \mathcal{Z}. \tag{4.12}$$

2. The approximation $\mathcal{A}_m f(t)$ of $f(t)$ is determined by 2^m samples per unit length;

3. Information is lost when a coarser approximation is used. However, the approximation converges to the original function as the number of levels goes to infinity:

$$\lim_{m\to\infty} \mathcal{V}_m = \bigcup \mathcal{V}_m \quad \textit{is dense in } \mathcal{L}^2(\mathcal{R}), \tag{4.13}$$

and

$$\lim_{m\to-\infty} \mathcal{V}_m = \bigcap \mathcal{V}_m = \{\}. \tag{4.14}$$

For every multiresolution approximation of $\mathcal{L}^2(\mathcal{R})$, there exists a unique function $\phi(t)$, which is called a scale function, such that its dilation and translations

$$\phi_n^m(t) = 2^{-m/2}\phi(2^{-m}t - n), \quad m, n \in \mathcal{Z}, \tag{4.15}$$

form an orthonormal basis for the subspaces $\mathcal{V}_m$. The approximation of a function $f(t)$ at a resolution level m can be computed by decomposing $f(t)$ on the set of basis $\phi_n^m(t),\ n \in \mathcal{Z}$:

$$\mathcal{A}_m f(x) = 2^{-m} \sum_{n=-\infty}^{+\infty} < f(u), \phi_n^m(u) > \phi_n^m(u). \tag{4.16}$$

where $< \cdot, \cdot >$ denotes the inner product operation.

The wavelet representation is a multiresolution representation based on the differences of information between successive resolutions $\mathcal{V}_m$ and $\mathcal{V}_{m+1}$. As $\mathcal{V}_m$ is a subset of $\mathcal{V}_{m+1}$, the difference between the two sets, which is called the *detail* at level m, is

the orthogonal complement $\mathcal{O}_m$ of $\mathcal{V}_m$ in $\mathcal{V}_{m+1}$, i.e.,

$$\mathcal{O}_m \perp \mathcal{V}_m, \ and \tag{4.17}$$

$$\mathcal{O}_m \bigoplus \mathcal{V}_m = \mathcal{V}_{m+1}. \tag{4.18}$$

It has been proved that for every scale function $\phi(t)$, we can determine the corresponding mother wavelet function $\psi(t)$ such that its dilation and translations

$$\psi_n^m(t) = 2^{-m/2}\psi(2^{-m}t - n), n \in \mathcal{Z}, \tag{4.19}$$

span the orthogonal complement $\mathcal{O}_m$. The detail of $f(t)$ at level m is computed by:

$$\mathcal{A}_{m+1}f(t) - \mathcal{A}_m f(t) = 2^{-m} \sum_{n=-\infty}^{+\infty} < f(u), \psi_n^m(u) > \psi_n^m(t). \tag{4.20}$$

As a result, the dilation and translations of the wavelet $\psi(t)$,

$$\psi_n^m(t) = 2^{-m/2}\psi(2^{-m}t - n), \quad m, n \in \mathcal{Z} \tag{4.21}$$

form an orthonormal basis of $\mathcal{L}^2(\mathcal{R})$. The decomposition of a function $f(t)$ using the set of wavelet basis is called a wavelet decomposition:

$$f(t) = \sum_{m \in \mathcal{Z}} \sum_{n \in \mathcal{Z}} < f(u), \psi_n^m(u) > \psi_n^m(t). \tag{4.22}$$

The set of coefficients on these basis $< f(u), \psi_n^m(u) >$, $m, n \in \mathcal{Z}$ is the wavelet representation of $f(t)$.

The set of wavelets $\psi_n^m(t) = 2^{-m/2}\psi(2^{-m}t - n)$, $m, n \in \mathcal{Z}$ is not necessarily orthogonal. When the wavelets are not orthogonal, we can always find a function $\tilde{\psi}(t)$, which is called a dual wavelet of $\psi(t)$, such that:

$$< \psi_j^i, \tilde{\psi}_l^k >= \delta_{i,j}\delta_{k,l}. \tag{4.23}$$

In this case, ψ_n^m and $\tilde{\psi}_n^m$ form a biorthogonal pair. With the biorthogonal wavelets, the wavelet decomposition of a function $f(t)$ can be written as:

$$f(t) = \sum_{m,n=-\infty}^{\infty} d_n^m \psi_n^m(t), \tag{4.24}$$

and the wavelet coefficients d_n^m can be computed via

$$d_n^m = \int_{-\infty}^{\infty} f(t)\tilde{\psi}_n^m(t)dt. \tag{4.25}$$

Deformation Using Wavelet Transform

The advantages of the wavelet basis in flexible shape modeling which allows both global and local degrees of freedom, come at the cost of a larger number of parameters. These are the coefficients of the local and global basis. We note that in our problem, only the displacements in the proximity of the deformed template are of interest. So, to model the displacement of the template, we only need to use the subset of basis

which have a nonzero support near the template boundary.

In the particular case where the template consists of a single contour, we parameterize the contour:

$$\Omega \quad [0,1] \rightarrow \mathcal{R}^2$$

$$t \mapsto v(t) = (x(t), y(t)), \tag{4.26}$$

and then use the 1-D wavelet basis to model the displacements Δx and Δy in the two dimensions $x(t)$ and $y(t)$ separately:

$$\begin{pmatrix} \Delta x(t) \\ \Delta y(t) \end{pmatrix} = \begin{pmatrix} \sum_{m=1}^{M} x_d^m(t) \\ \sum_{m=1}^{M} y_d^m(t) \end{pmatrix}, \tag{4.27}$$

where

$$x_d^m(t) = \sum_n (\xi^x)_n^m \tilde{\psi}_n^m(t), \quad y_d^m(t) = \sum_n (\xi^y)_n^m \tilde{\psi}_n^m(t), \tag{4.28}$$

are the details at scale m with $1 \leq m \leq M$. The deformation parameters are then the wavelet coefficients $(\xi^x)_n^m$ and $(\xi^x)_n^m$. In this way, we reduce the problem of modeling the displacement in a 2-D continuum to the problem of modeling two 1-D displacements, and as a result, reduce the number of parameters needed.

We have used the B-spline wavelets [137] to span the displacement. B-spline wavelets have the desirable properties that they have compact supports, and they asymptotically converge to the Gabor functions. More details about the B-spline wavelets can be found in [136, 137]. Figure 4.4 shows the B-spline wavelets at the

first and second resolution levels.

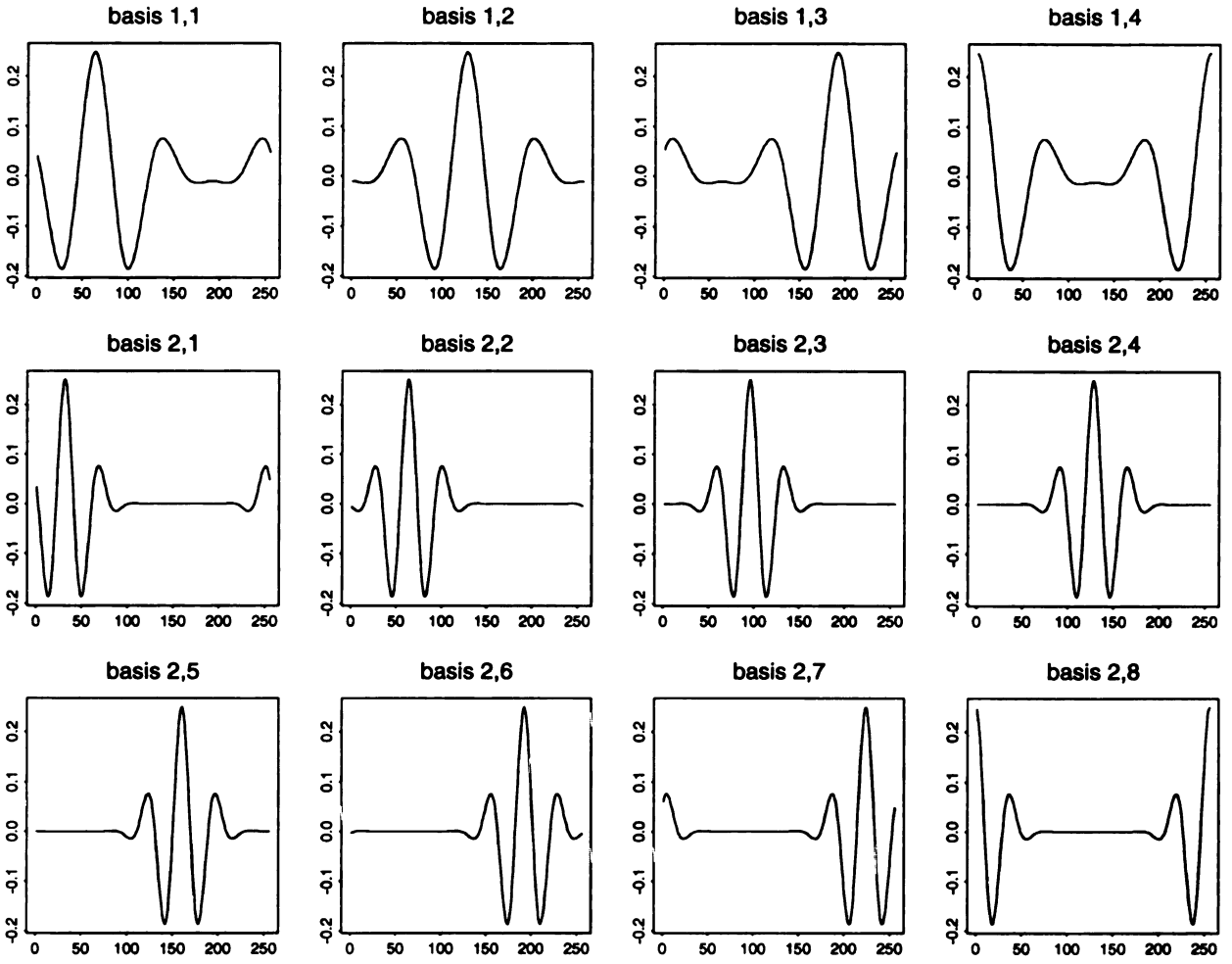
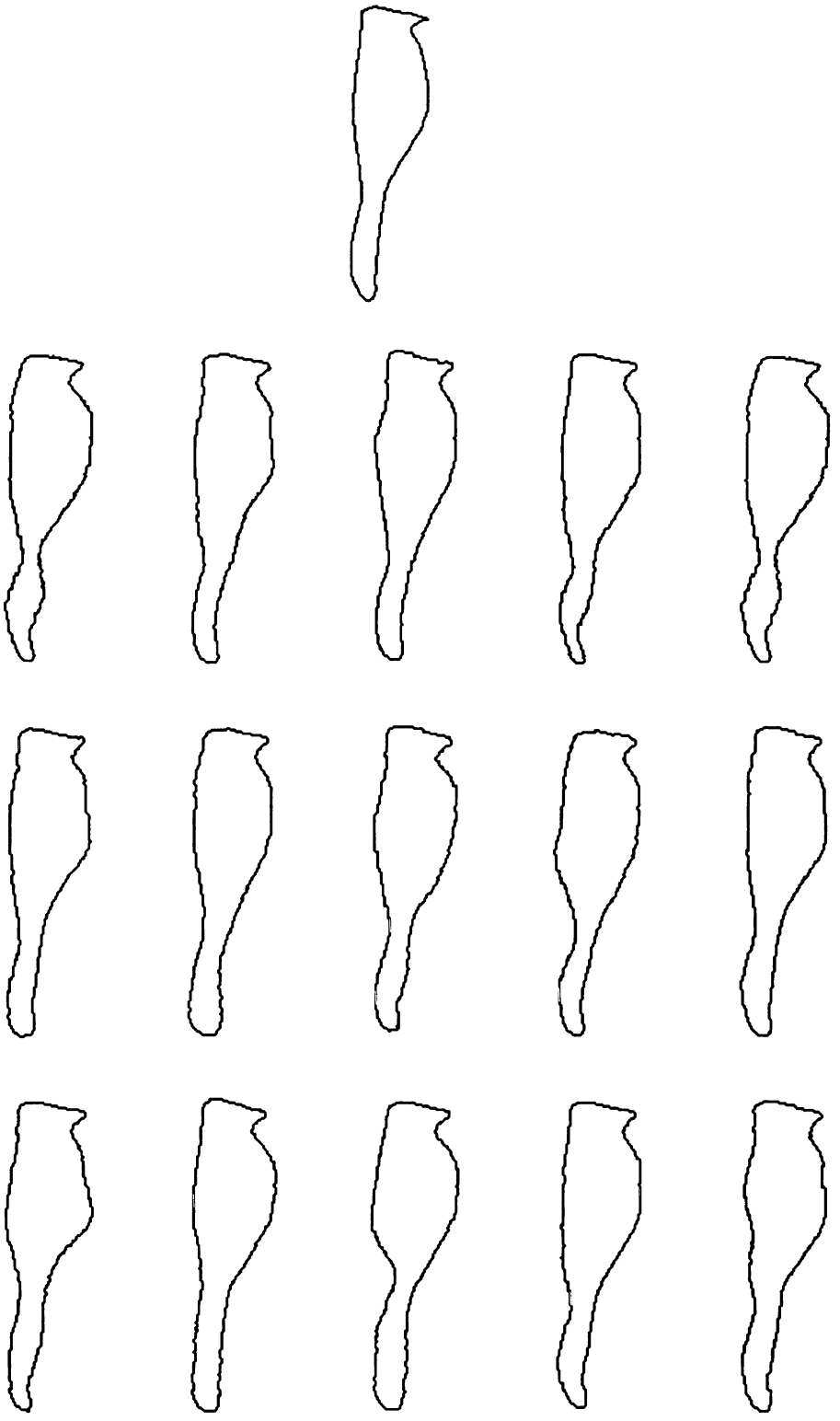


Figure 4.4: B-spline wavelet basis. They are generated by shifting and dilating a mother wavelet function.

In Fig. 4.5 we show some templates obtained by deforming a prototype using the B-spline wavelet transform. These templates are generated using deformation parameter values which come from an *i.i.d.* zero mean Gaussian distribution. Note that the deformations can be quite local.



(a)

(b)

Figure 4.5: Deformations using the B-spline wavelet basis; **(a)** prototype template; **(b)** deforming the template in (a) using the B-spline wavelet basis. Two resolution levels are used which account for a total of 24 deformation parameters. The deformation parameter values are randomly generated using a Gaussian distribution.

4.2.4 Comparison of the Three Deformation Transforms

We have proposed three different transforms to deform the template. Among them, the trigonometric function-based deformation places very few constraints on the shape of the template, but it is incapable of adequately modeling local deformations. The remaining two deformations can accommodate local changes, but assume that the template can be parameterized by the arc length. The wavelet-based deformation transform can specify the deformations in a coarse-to-fine manner. The comparison of the three transforms are summarized in Table 4.1.

4.2.5 A Probabilistic Model of Deformation

The deformation transforms described in Sec. 4.2 can represent different complex deformations by choosing the number and values of the deformation parameters $\underline{\boldsymbol{\xi}}$. However, not all the transformations result in a deformed template that visually resembles the prototype template. Usually, deformable parameters with large values result in a large deformation. As all the available prior shape information is represented in the prototype template, it is natural to assume that the prototype template exemplifies the most likely *a priori* shape of the object. Also, small deformations that leave the template similar to its original shape are more likely to be observed than large displacements. We impose a probability density on the deformation parameters $\underline{\boldsymbol{\xi}}$ to bias the possible deformed templates which can be generated. Specifically, the $\boldsymbol{\xi}$'s are assumed to be independent of each other, independent along x and y directions, and identically Gaussian distributed with mean zero and variance σ^2. This way, the

	Deformation Transform		
	Trigonometric	B-spline representation	Wavelet transform
Displacement $(\Delta x,\ \Delta y)$	$\begin{pmatrix} \sum_{m,n=1}^{M,N} \xi_{mn}^{x} \sin(\pi n x)\cos(\pi m y)/\lambda_{mn} \\ \sum_{m,n=1}^{M,N} \xi_{mn}^{y} \cos(\pi m x)\sin(\pi n y)/\lambda_{mn} \end{pmatrix}$	$\begin{pmatrix} \sum_{i=1}^{k-m} \xi_i^x \mathcal{B}_i^m(t) \\ \sum_{i=1}^{k-m} \xi_i^y \mathcal{B}_i^m(t) \end{pmatrix}$	$\begin{pmatrix} \sum_{m,n=1}^{M,N_m} (\xi^x)_n^m \tilde{\psi}_n^m(t) \\ \sum_{m,n=1}^{M,N_m} (\xi^y)_n^m \tilde{\psi}_n^m(t) \end{pmatrix}$
Basis function	global trigonometric functions with a number of frequencies	local compact support	functions with a hierarchy of global to local compact supports
Variability	deformation is global	deformation is local	can handle deformation at different resolution scales
Interaction of deformations in both the x and y axis	deformation at a pixel depends on both the x and y coordinates	no correlation; the deformation in each axis is modeled by a 1D function	no correlation; the deformation in each axis is modeled by a 1D function
Constraints on shape	no constraint using the bitmap representation	one component which can be parameterized using the arc length	one component which can be parameterized using the arc length

Table 4.1: Comparison of the three deformation schemes.

prototype, which corresponds to the zero mean, is more likely than other templates. Small deformations, which correspond to small deformation values, are more likely to happen than large deformations. The independence assumption may not hold in many situations. But this simplification does not affect the performance significantly and results in modeling and computational efficiency.

The prior distribution for each of the deformation transforms is given as follows:

2D trigonometric basis:

$$\begin{aligned} \mathcal{P}r(\underline{\xi}) &= \prod_{m,n=1}^{M,N} \mathcal{P}r(\xi_{mn}) \\ &= \prod_{m,n=1}^{M,N} \frac{1}{2\pi\sigma^2} \exp\{-\frac{{\xi_{mn}^x}^2 + {\xi_{mn}^y}^2}{2\sigma^2}\} \\ &= \frac{1}{(2\pi\sigma^2)^{MN}} \exp\{-\frac{1}{2\sigma^2} \sum_{m,n} ({\xi_{mn}^x}^2 + {\xi_{mn}^y}^2)\}. \end{aligned} \tag{4.29}$$

B-spline basis:

$$\begin{aligned} \mathcal{P}r(\underline{\xi}) &= \prod_{i=0}^{N_c-1} \mathcal{P}r(\xi_i) \\ &= \prod_{i=0}^{M,N} \frac{1}{2\pi\sigma^2} \exp\{-\frac{(\xi_i^x)^2 + (\xi_i^y)^2}{2\sigma^2}\} \\ &= \frac{1}{(2\pi\sigma^2)^{N_c}} \exp\{-\frac{1}{2\sigma^2} \sum_i ((\xi_i^x)^2 + (\xi_i^y)^2)\}. \end{aligned} \tag{4.30}$$

Wavelet basis:

$$\mathcal{P}r(\underline{\xi}) = \prod_{m,n=1}^{M,N_m} \mathcal{P}r(\xi_n^m)$$

$$= \prod_{m,n=1}^{M,N_m} \frac{1}{2\pi\sigma^2} \exp\{-\frac{((\xi^x)_n^m)^2 + ((\xi^y)_n^m)^2}{2\sigma^2}\} \tag{4.31}$$

The value of σ^2 in *Eqs. (4.29)-(4.31)* reflects the confidence about the prototype template, with a large value of σ^2 allowing more deformation. Intuitively, larger values of σ give rise to more elastic templates and smaller values of σ give rise to more rigid templates.

4.3 Bayesian Formulation and Objective Function

A Bayesian inference scheme is employed to integrate the prior shape knowledge of the template and the observed object(s) in the input image. The prior information of the object shape can be presented by a combination of

- a prototype template,
- a set of deformation transformations of the template, and
- a probability density on the set of deformation transformations.

We propose an energy function based on the image boundary strength and the deformed template in order to arrive at the likelihood. This likelihood is then combined with the prior using Bayes' rule to obtain the *a posteriori* probability density of the deformations of the template given the input image. The object is located by deforming the template so that the *a posteriori* probability density is maximized. The final shape and position of the deformed template gives a description of the object in the image.

4.3.1 Prior Distribution

We use the prior distribution to bias the global transformations (rotation, translation, and scale change) and local deformations that can be applied to a prototype template. Let $\mathcal{T}_0$ denote the prototype template, and let $\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}$ be a deformation of the prototype. This deformation is realized by rotating the prototype template by an angle Θ, locally deforming the rotation by a set of parameters $\underline{\xi}$, scaling the local deformation by a factor of s, and translating the scaled version along the x and y directions by an amount $\underline{d} = (d^x, d^y)$:

$$\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}(x,y) = \mathcal{T}_0(s \cdot [(x,y) + \mathcal{D}_{\underline{\xi}}(\mathcal{R}_\Theta(x,y))] + (d^x, d^y)), \tag{4.32}$$

where $\mathcal{D}_{\underline{\xi}}$ denotes the displacement functions given in *Eqs. (4.3), (4.11), and (4.27)*, and $\mathcal{R}_\Theta(x,y)$ is the rotation of a point (x,y) by an angle Θ. Assuming that $s, \Theta, \underline{\xi}$, and $\underline{d}$ are all independent of each other, then we can write the joint density of $s, \Theta, \underline{\xi}$, and $\underline{d}$ as the products of their marginals:

$$\mathcal{P}r(s,\Theta,\underline{\xi},\underline{d}) = \mathcal{P}r(s) \cdot \mathcal{P}r(\Theta) \cdot \mathcal{P}r(\underline{\xi}) \cdot \mathcal{P}r(\underline{d}). \tag{4.33}$$

Suppose all translations, rotations, and scale sizes are equally likely as long as the transformed template falls in the input image, then *Eq. (4.33)* reduces to:

$$\mathcal{P}r(s,\Theta,\underline{\xi},\underline{d}) = \kappa \mathcal{P}r(\underline{\xi}), \tag{4.34}$$

where κ is a normalizing constant and $\mathcal{P}r(\underline{\xi})$ is defined in *Eqs. (4.29)-(4.31).* *Eq.* (4.34) constitutes our prior density. Intuitively, a deformed template with a geometric shape similar to the prototype template is favored, regardless of its size, orientation, or location in the image.

4.3.2 Likelihood

The likelihood specifies the probability of observing the input image, given a deformed template at a specific position, orientation and scale. It is a measurement of the similarity between the deformed template and the object(s) present in the image. The likelihood which we currently use only incorporates the edge information in the input image, where both the edge positions and directions are considered. Other alternatives of the likelihood can incorporate texture, grey-scale homogeneity, or color information in more complex situations. How to effectively select a likelihood which reflects the requirement of a particular application will be an interesting topic for future work (see Chapter 8).

The deformable template is attracted and aligned to the salient edges in the input image via a directional edge potential field. This field is determined by the positions and directions of the edges in the input image. For a pixel (x, y) in the input image, its edge potential is defined as:

$$\Phi(x, y) = -\exp\{-\rho(\delta_x^2 + \delta_y^2)^{1/2}\}, \tag{4.35}$$

where (δ_x, δ_y) is the displacement to the nearest edge point in the image, and ρ is a

smoothing factor which controls the degree of smoothness of the potential field. The edge potential field can be obtained by convolving the edge map using a Gaussian mask. For pixels which are far away from the edgemap pixels, the edge potential is high. For pixels near the edge pixels, the potential is low. The potential of a pixel is zero if it happens to lie on the edgemap. The goal of the edge potential field is to attract the template to salient image features (the input edges). When we minimize the potential of a template, it is directed to the nearby edges. The edge potential of the input image in Fig. 1.1(b) computed using *Eq. (4.35)* is shown in Fig. 4.6.

Figure 4.6: Edge potential field of the input image in Fig. 1.1(b). The potential at a pixel is displayed as the greyscale value. The larger the gray scale value, the larger the potential value. The yellow pixels denote the edge map of the input image. The potential at locations far away from the edge pixels is high, and the potentials at locations near the edge pixels is low.

The edge potential field, computed from the distance to the nearest edge pixel alone, drags the template near the edge pixels. However, it is vulnerable to noisy edges, so the template may get trapped to spurious edges. We have used the gradient

direction to improve the performance so that the template agrees with the edgemap in both its position and direction. We modify this edge potential by adding to it a directional component. This new edge potential induces an energy function that relates a deformed template $\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}$ to the edges in the input image Y:

$$\mathcal{E}(\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}, Y) = \frac{1}{n_{\mathcal{T}}} \sum (1 + \Phi(x,y)|\cos(\beta(x,y))|), \tag{4.36}$$

where $n_{\mathcal{T}}$ is the number of pixels on the template, $\beta(x,y)$ is the angle between the tangent of the nearest edge and the tangent direction of the template at (x,y) (Fig. 4.7), and the constant 1 is added so that the potentials are positive and take values between 0 and 1. The summation is over all the pixels on the deformed template. The edge magnitudes and directions are computed using the Canny edge detector [22].

This definition requires that the template boundary agrees with the image edges not only in position, but also in the tangent direction. This feature is particularly useful in the presence of noisy edges. The lower this energy function, the better the deformed template matches the edges in the input image. Using this energy function, we now define a probability density that specifies the likelihood of observing the input image, given the deformations of the template:

$$\mathcal{P}r(Y \mid s, \Theta, \underline{\xi}, \underline{d}) = \alpha \exp\{-\mathcal{E}(\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}, Y)\}, \tag{4.37}$$

where α is a normalizing constant to ensure that the above function integrates to 1. The maximum likelihood is achieved when $\mathcal{E}(\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}, Y) = 0$, i.e., when the deformed

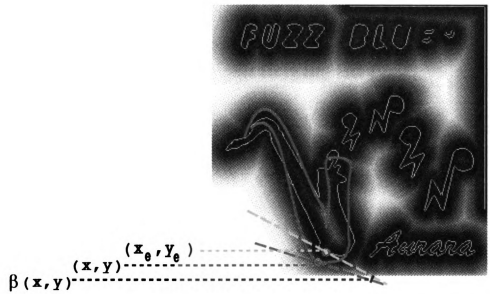


Figure 4.7: Directional edge potential field. The deformed template (in solid red) is put in the edge potential field created by the edgemap (in solid yellow). Let (x, y) be a point on the template. Let (x_e, y_e) be the nearest edge pixel to (x, y) in the image field. Then the directional potential at template pixel (x, y) is defined as the edge potential at (x, y) (caused by (x_e, y_e) according to *Eq. (4.35)* multiplied by the cosine of $\beta(x, y)$, the tangent angle between (x, y) and (x_e, y_e). The directional edge potential of the template is the average directional edge potential of all the template pixels.

template $\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}$ exactly matches the edges in the input image Y.

We note that the normalizing constant α is not necessarily independent of the deformed template. Exact computation of this constant involves an intractable summation. If this constant can be computed, then it removes the inherent bias of the energy function (and hence the likelihood) towards deformations of the template that increases its size. Since we cannot compute it, we have chosen to incorporate its effects by normalizing the energy function directly with respect to the template size - see *Eq. (4.36)*. We acknowledge that this represents a deviation from proper Bayesian inference, but many of the existing studies also suffer from the same drawback.

4.3.3 Posterior Probability Density

Using Bayes rule, the prior probability of the deformed templates in *Eq. (4.34)* and the likelihood of the input image given the deformed template in *Eq. (4.37)* can be combined to obtain the *a posteriori* probability density of the deformed template given the input image:

$$\left.\begin{aligned}
&\mathcal{P}r(s,\Theta,\underline{\xi},\underline{d} \mid Y) \\
&\quad = \mathcal{P}r(Y \mid s,\Theta,\underline{\xi},\underline{d})\ \mathcal{P}r(s,\Theta,\underline{\xi},\underline{d})/\mathcal{P}r(Y) \\
&\quad = \mathcal{C}_1 \exp\{-\mathcal{E}(\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}, Y)\} \prod_{i\in\mathcal{P}} \tfrac{1}{2\pi\sigma^2} \exp\{-\tfrac{1}{2\sigma^2}(\xi_i^{x2} + \xi_i^{y2})\} \\
&\quad = \mathcal{C}_2 \exp\{-\mathcal{E}(\mathcal{T}_{s,\Theta,\underline{\xi},\underline{d}}, Y) - \sum_{i\in\mathcal{P}} \tfrac{1}{2\sigma^2}(\xi_i^{x2} + \xi_{mi}^{y\ 2})\}
\end{aligned}\right\}, \tag{4.38}$$

where $\mathcal{C}_1$ and $\mathcal{C}_2$ are normalizing constants, and $\mathcal{P}$ denotes the set of deformation parameters.

From the Bayesian point of view, this posterior probability density reflects the distribution of the deformable template, based on the prior Gaussian distribution, and then corrected using the image edge information. The best match can be obtained by finding the Maximum A Posteriori estimate, i.e., finding the maximum of the probability function in *Eq. (4.38)*.

4.3.4 Objective Function

Our goal is to maximize the *a posteriori* density in *Eq. (4.38)* with respect to parameters $s, \Theta, \underline{\boldsymbol{\xi}}, \underline{d}$. Taking the natural logarithms on both sides of *Eq. (4.38)* results in:

$$\ln \mathcal{P}r(s, \Theta, \underline{\boldsymbol{\xi}}, \underline{d} \mid Y) = \ln \mathcal{C}_2 - \mathcal{E}(\mathcal{T}_{s,\Theta,\underline{\boldsymbol{\xi}},\underline{d}}, Y) - \sum_{i \in \mathcal{P}} (\xi_i^{x2} + \xi_i^{y2})/2\sigma^2. \tag{4.39}$$

Equivalently, we seek to minimize the following objective function with respect to $s, \Theta, \underline{\boldsymbol{\xi}}, \underline{d}$:

$$\mathcal{L}(\mathcal{T}_{s,\Theta,\underline{\boldsymbol{\xi}},\underline{d}}, Y) = \mathcal{E}(\mathcal{T}_{s,\Theta,\underline{\boldsymbol{\xi}},\underline{d}}, Y) + \gamma \sum_{i \in \mathcal{P}} (\xi_i^{x2} + \xi_i^{y2}), \tag{4.40}$$

where $\gamma = 1/2\sigma^2$. This objective function consists of two parts:

- a model-based term which measures the deviation of the deformed template from the prototype template, and

- a data-driven term which describes the fitness of the deformed template contour to the boundary in the image.

The parameter γ can be interpreted as providing a relative weighting of the two penalty measures; a larger value of γ implies a lower variance of the deformation parameters, and as a result, a more rigid template.

The objective function $\mathcal{L}$ in *Eq. (4.40)* is a function of the deformation parameters and the object pose parameters. We can find the set of parameters which gives the best objective function value. The optimal solution that maximizes *Eq. (4.40)* can be obtained, in principle, by an exhaustive search of the parameter space of $s, \Theta, \underline{\xi}$, and $\underline{d}$. However, this is not feasible due to the high dimensionality of the search space. We find the minimum by first roughly estimating the pose of the object, and then using the gradient descent method to find the nearby local minimum of the objective function surface.

4.4 Discussion

The proposed deformation model (*Eq. (4.32)*) is a more powerful tool than the affine transformations for object matching and localization. An affine transformation can be expressed as the product of an arbitrary sequence of rotation, translation, and scale matrices and has the property of preserving parallelism of lines, but not the magnitude of angles and lengths. Our deformation model is more general than the affine transformations, in that it (i) includes linear scaling, rotation, and translation as a special case when we set the deformation parameters to *zeros*, and (ii) can allow nonlinear deformation of the shape (the deformation basis functions are nonlinear, see Fig. 4.1.) which cannot be handled by the affine transforms. Figure 4.8 gives some

examples of the deformation using the affine transform and the nonrigid deformation transform. In Fig. 4.8(a), the parallelism of the opposite edges of the rectangle is maintained after the affine transform, while the linearity is lost after applying the nonrigid deformation transform (Fig. 4.8(b)).

The proposed deformation model also has advantages over the feature-based matching methods such as invariant moments. First, the objects do not need to be segmented or grouped to apply the deformable matching process. The proposed model accumulates local, fragmental evidence and combines it with the global model to establish a match. Secondly, the invariant moments (typically only low-order moments are used), are necessary but not sufficient for shape matching. In other words, two shapes can have very similar lower order invariant moments, but with dramatically different appearances. For the deformation matching process, the objective function is minimized only if the template matches a subpart of the input image exactly; the objective function value provides a measure of dissimilarity.

We now show that the invariant moment features and deformable template matching method do give different similarity measures. We generate in Fig. 4.9 some random samples of deformed templates using the deformation transform defined in *Eq. (4.1)*. We compute the deformation between the deformed templates and the prototype template. In Fig. 4.10 we arrange the deformed templates in Fig. 4.9 so that the amount of deformation from the prototype increases from top to bottom, left to right. The template in the upper-left corner is the prototype template with no deformation, and the template in the lower-right corner has the most deformation. We give in Fig. 4.11 shows the same set of deformed templates but now they are ranked using

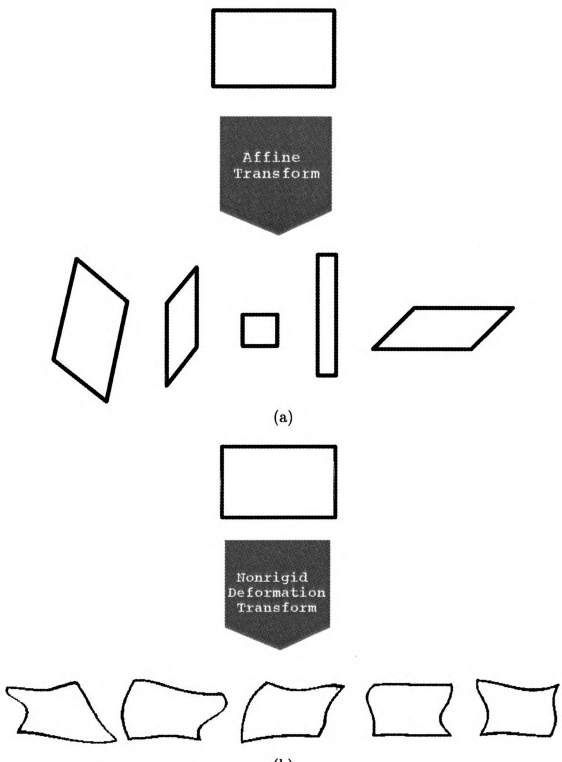


Figure 4.8: Comparison of the affine transform and nonrigid deformation transform. **(a)** deforming a rectangle using the affine transform; **(b)** deforming a rectangle using the nonrigid deformation transform defined in *Eq. (4.1)*.

the Euclidean distance between the invariant moment features (defined in *Eqs. (7.2, 7.3)*) of the template and deformed shapes. In Sec. 7.1 we will further show how the deformable template matching method can be used to improve the matching results obtained by using simple shape features including the invariant moments.

In Fig. 4.12 we show the results of applying the deformable template model to a fish image using different deformation basis. It is observed that with comparable numbers of degree's of freedom, both the spline basis and the wavelet basis can model local deformations better than the global trigonometric basis. However, a template with an excessive number of degrees of freedom may be attracted to spurious image features which are not desired.

We have proposed to use three different deformation basis to model the deformation field. The selected set of basis determines the allowed deformation space and the template space. It is desirable to use that deformation basis which facilitates both the representation and computation in the following ways:

- From the representation perspective, the deformation basis, as a means for approximation, should be both accurate and efficient. Accuracy means that the set of basis should approximate the natural deformation with precision. Efficiency means that the set of basis should approximate the natural deformation well using, preferably, a small set of parameters.

- From the computational perspective, the selected deformation basis should help to facilitate the computation for the optimal solution. It is favorable to have a smooth objective function in the deformation parameter space, so that the

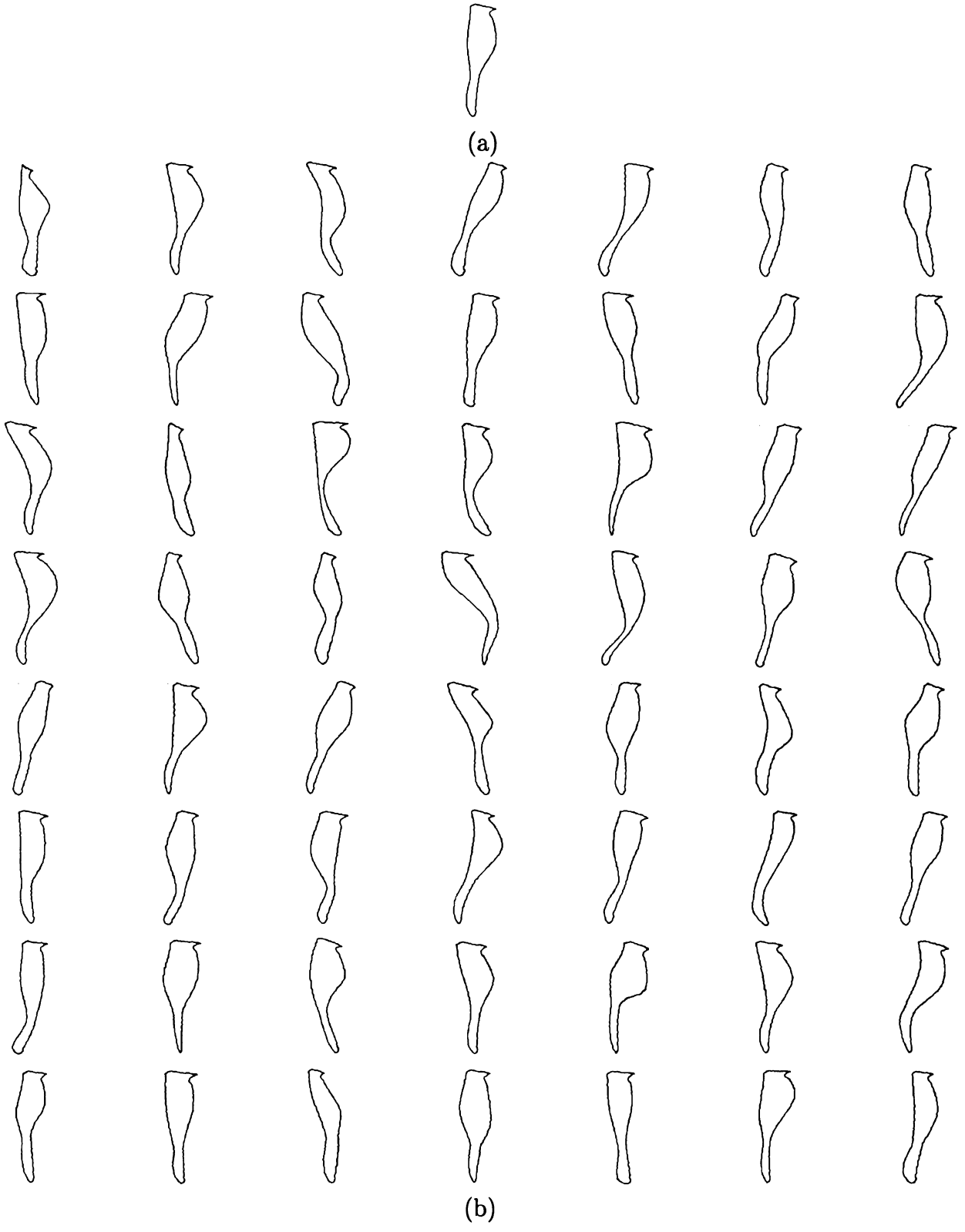


Figure 4.9: Deformed templates derived from a prototype. **(a)** the prototype; **(b)** randomly generated deformed templates.

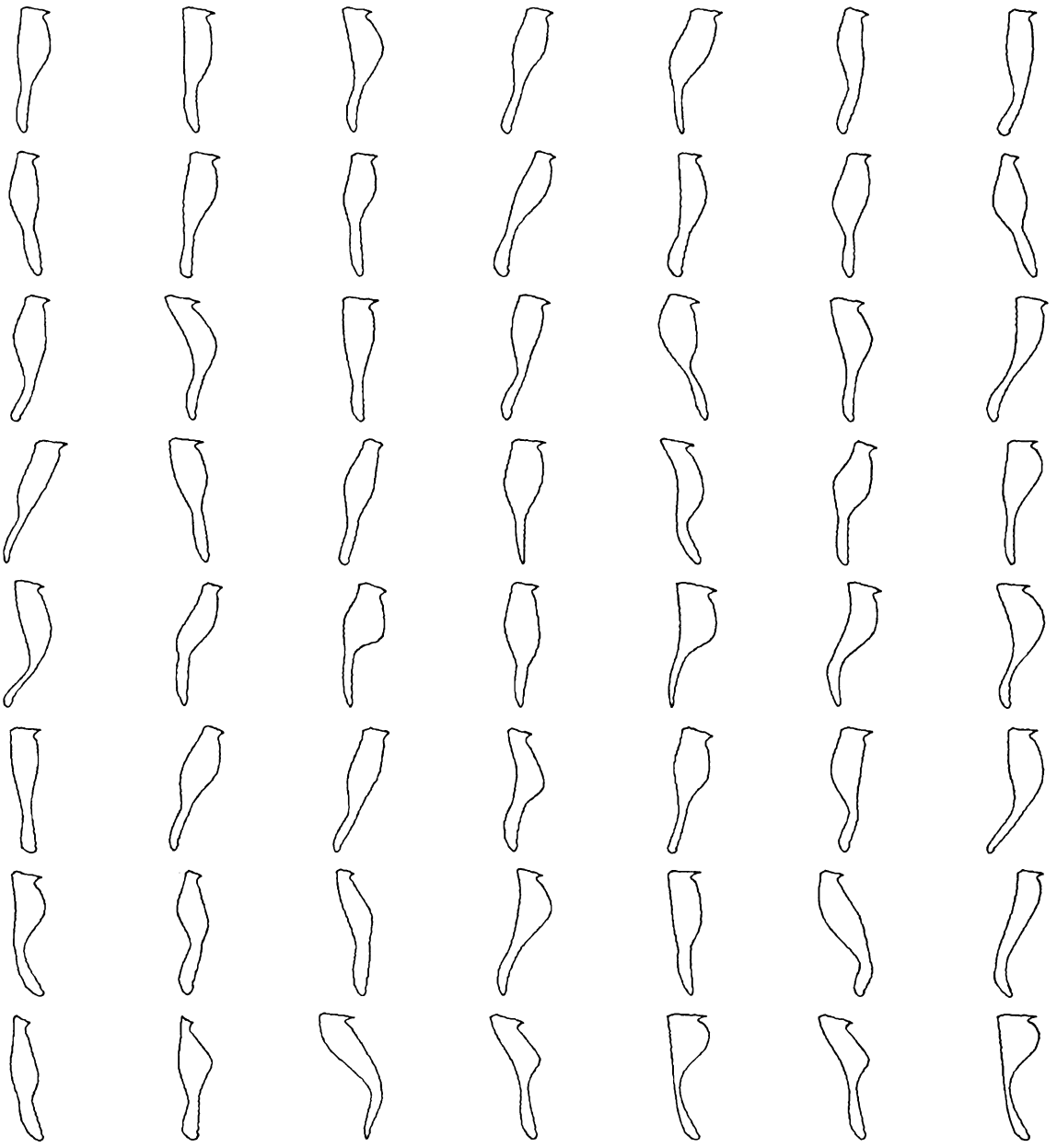


Figure 4.10: Ranking of the deformed templates derived from the prototype. They are arranged with decreasing similarity from top to bottom, left to right, using deformable template matching method.

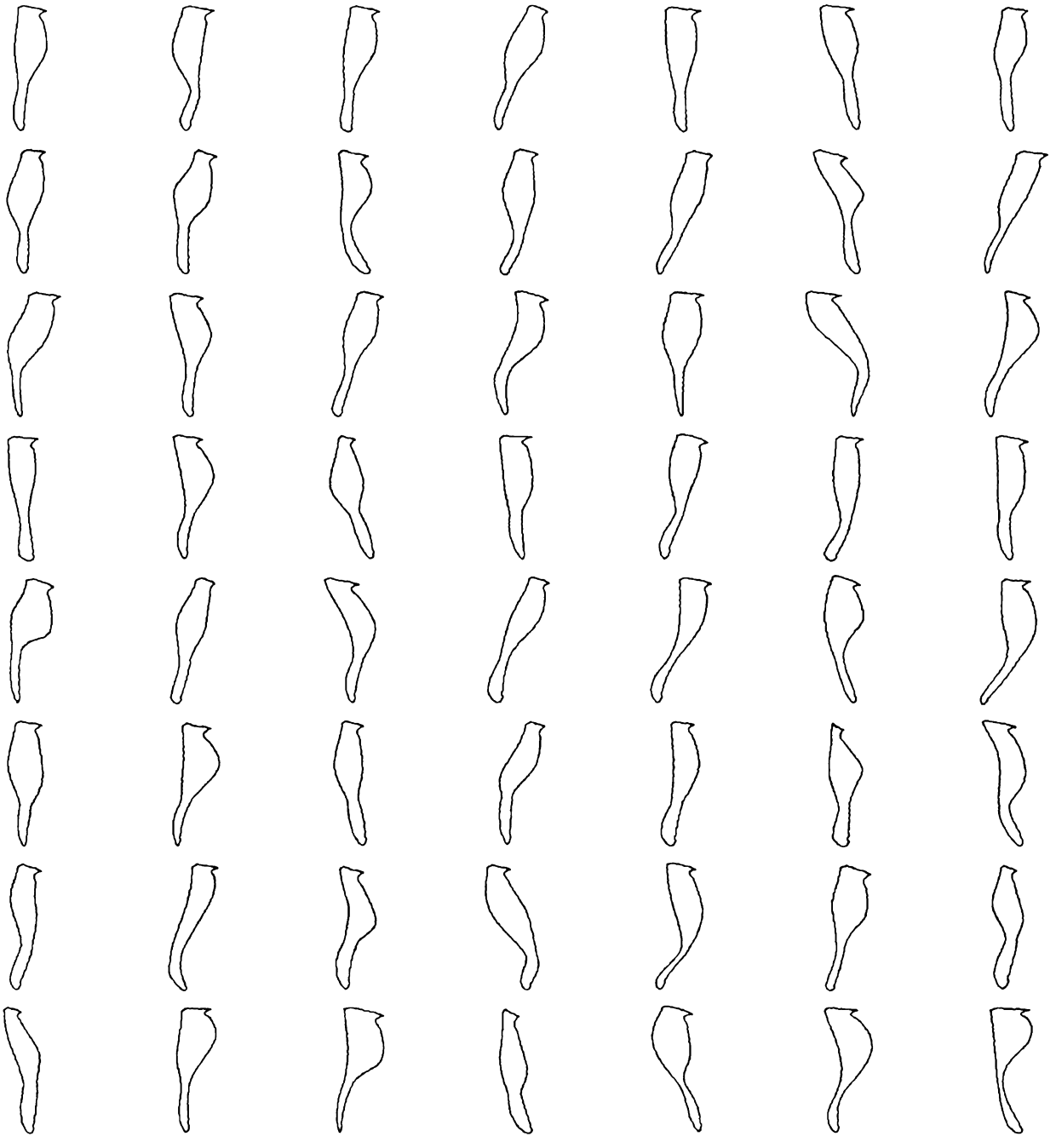
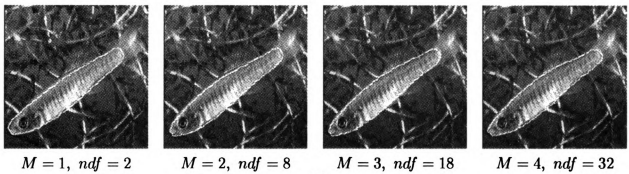
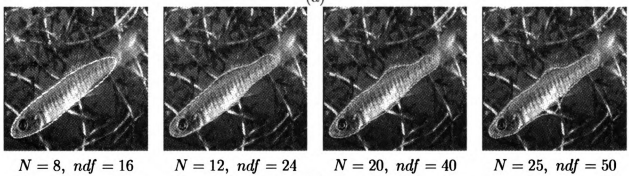


Figure 4.11: Ranking of the deformed templates using invariant moments. They are arranged with decreasing similarity from top to bottom, left to right, using invariant moments.



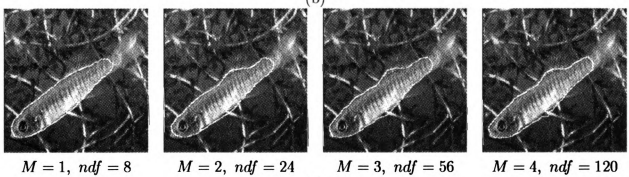
$M = 1,\ ndf = 2$ $M = 2,\ ndf = 8$ $M = 3,\ ndf = 18$ $M = 4,\ ndf = 32$

(a)



$N = 8,\ ndf = 16$ $N = 12,\ ndf = 24$ $N = 20,\ ndf = 40$ $N = 25,\ ndf = 50$

(b)



$M = 1,\ ndf = 8$ $M = 2,\ ndf = 24$ $M = 3,\ ndf = 56$ $M = 4,\ ndf = 120$

(c)

Figure 4.12: Locating a fish using the deformable templates with different deformation basis. **(a)** Using 2D trigonometric basis. M is the number of approximation levels, ndf is the number of basis used (number of degree's of freedom). **(b)** B-spline basis. N is the number of control points used. **(c)** B-spline wavelet basis. M is the number of resolution levels.

deformable matching process can quickly converge to the desired optimum;

All the three deformation basis are commonly used in approximation and modeling. We would, ideally, select the set of deformation basis which satisfies the above criteria. But for a specific applications, it is difficult to determine this. For the representation, we may adopt a measure which combines the description efficiency (say, coding length) and the deformation matching score (say, the directional edge energy of the template) to compare the various deformation transforms for a particular matching problem. It is more difficult to evaluate, from the computational perspective, which set of deformation basis renders a smooth energy surface. Further, it is not possible to express analytically the energy function using the deformation parameters because the edge potential field itself is empirical. A straight forward measure of the smoothness of a function is high order derivatives. If we express the objective function explicitly in terms of the deformation parameters, we can evaluate its high order derivatives. However, this process is complicated because of the high dimensionality of the deformation parameter space. Condition number is one way to measure the ill-conditionness of a function. Let the single value decomposition (SVD) of a matrix A be $A = UWV^t$. The condition number of A is defined as the ratio of the largest (in magnitude) of the w_i's to the smallest of the w_i's, where w_i's are the diagonal elements of matrix W. We can compute the condition number for the Jacobian matrix of the objective function to see if it has good computational attributes. But, they also have to be computed empirically to evaluate the objective function surface. Further, the condition number is also a local measure of smoothness.

An alternative might be the entropy of the posterior distribution. The entropy of a distribution measures its randomness, or predictivity. A smooth probability density function is difficult to predict and possesses a large amount of randomness. Assuming that a set of pdf's all come from the same family, we can compare the smoothness by examining the entropy of each of the pdf. The larger the entropy, the smoother the pdf. The entropy provides a global and overall smoothness measure of a function, given appropriate assumptions. However, the difficulty of applying the entropy to measure the smoothness of the posterior distributions in our case is that the exact posterior pdf is difficult to compute. We have to empirically evaluate the likelihood for each point in the parameter space and we have to normalize the generalized pdf over the whole parameter space.

Theoretically, it is difficult to contrast the smoothness of the objective functions using various deformation basis. However, we can make some intuitive observations. The 2D-trigonometric basis provides a smoother objective function surface because the basis is global. This agrees with the empirical observations where the matching process converges more consistently and faster.

Chapter 5

Image Segmentation and Object Tracking

An important application of the deformable template matching method is image segmentation. As indicated earlier, the deformable template is attracted to the salient image features of the input image. When the objective function value is minimized, the template attempts to align with the edge map of the input image. Suppose we manually initialize the template near the object of interest, then the converged configuration gives a description of the object boundary.

The deformable template can also track objects in image sequences. The same object present in successive frames retains its global shape, though it varies from frame to frame. The prototype can specify the common shape characteristics, whereas the deformation can specify the changes between frames. Since the object shape in two successive frames does not change much, the converged configuration in the current frame provides a good initialization for the deformable template in the next frame.

In the rest of this chapter, we present some experimental results on image segmentation and object tracking using the deformable template matching algorithm.

5.1 Preprocessing

Given an input image, we start out by sketching a prototype template which resembles an object of interest. Then the input image is processed to obtain the corresponding edge potential image(s). First, we use the Canny edge detector ($\sigma = 1$, mask size is 9×9, and the lower and upper thresholds are 0.5 and 0.9, respectively) to calculate the edge map of the input image as well as the gradient direction at each edge point. Then, the edge potential field is calculated from the output of the Canny edge detector. Fig. 5.1 illustrates the intermediate results (images) of the preprocessing stage. Figure 5.1(a) is the input CD cover image which contains a saxophone. Its corresponding edge map is shown is Fig. 5.1(b), obtained using the canny edge detector with a σ [1] of 3 pixels. Note that the boundary of the saxophone is well preserved, but there are many noisy edges near the outer contour. The edge potential magnitude (*Eq. (4.35)*) is illustrated in (c), where a dark grey value denotes low potential energy. Fig. 5.1(d) shows the edge directional field (*Eq. (4.36)*), where the direction at a point is defined as the tangent direction of the closest edge pixel. The direction is not calculated for pixels in the "white" regions because they are so far away from the edge pixels, and the potential magnitude is so small that the potential fields (both the magnitude and the direction) are negligible.

[1] σ is a real number that determines the width of the detected edges. Larger values of σ will tend to find edges that are more widely spaced.

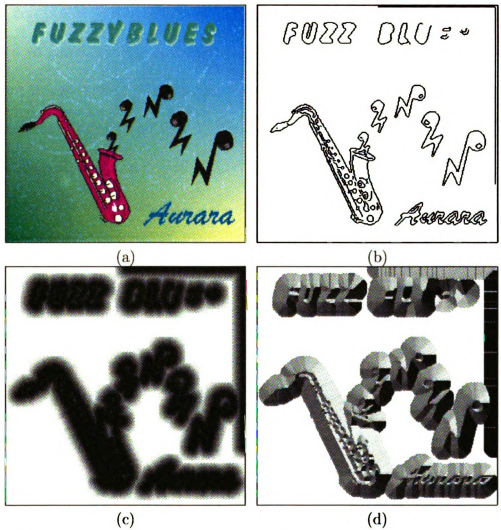


Figure 5.1: Preprocessing. **(a)** input image (256 × 256); **(b)** edge map using the Canny edge detector; **(c)** magnitude of the edge potential field; **(d)** direction field;

5.2 Image Segmentation

The proposed deformable model can be used to segment desired objects from their background in an input image. When it is used in this way, the given prototype template is first placed in the neighborhood of the desired objects. The template is then deformed to match the object contours in the input images. The gradient descent algorithm is used to find the local minimum of the objective function in *Eq. (4.40)*. The minimum energy configuration gives the segmentation result. The final deformed template as well as the intermediate sequences of the deformable template are presented to show how the deformable template evolves to match the salient edges in the input image.

Figure 5.2 shows the segmentation process when we (manually) place a hand-drawn template (Fig. 5.2(a)) in the neighborhood of a saxophone in a CD cover image (Fig. 5.2(b)). Figure 5.2(c) shows the manually chosen initial position of the template, and Figs. 5.2(d) and 5.2(e) are the intermediate snapshots of the deformation process. The final converged shape of the template is shown in Fig. 5.2(f), which gives a segmentation of the desired object.

In Fig. 5.3 we show the same evolution sequence when we apply a fish template (Fig. 5.3 (a)) to an image containing a fish (Fig. 5.3 (b)). In both the cases, the deformable templates converge to the correct object contour by decreasing the objective function value $\mathcal{L}$.

Figure 5.4 shows an example of the sensitivity of the matching process in Fig. 5.3 to the initial placement of the template. As long as we initialize the template

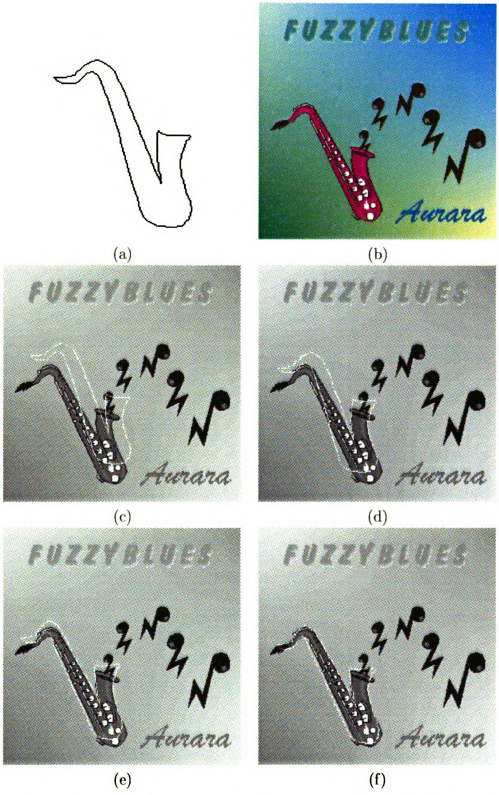


Figure 5.2: Localization of a saxophone using manually chosen initial template position. The input image size is 285×286. **(a)** The prototype template; **(b)** input image; **(c)** initial position, $\mathcal{L} = 0.603$; **(d)** 10 iterations, $\mathcal{L} = 0.327$; **(e)** 16 iterations, $\mathcal{L} = 0.186$; **(f)** 30 iterations, $\mathcal{L} = 0.123$.

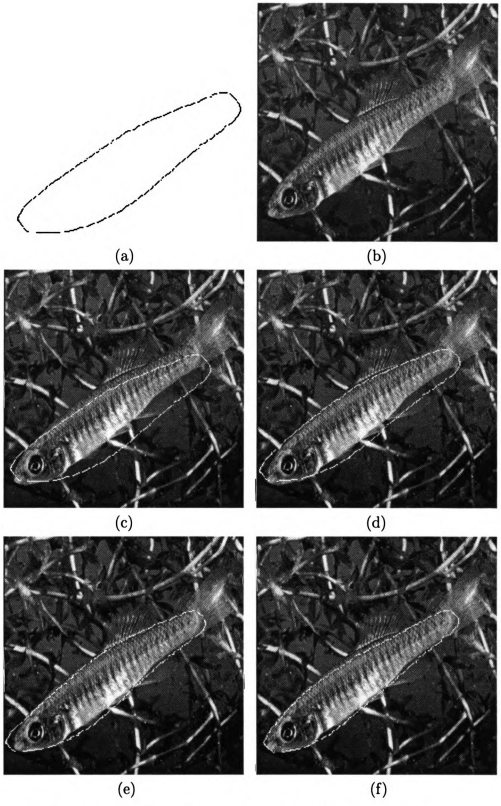


Figure 5.3: Localization of a fish using manually chosen initial template position. Image size is 256×256. **(a)** Prototype template; **(b)** input image; **(c)** initial position, $\mathcal{L} = 0.432$; **(d)** 4 iterations, $\mathcal{L} = 0.308$; **(e)** 7 iterations, $\mathcal{L} = 0.221$; **(f)** 40 iterations, $\mathcal{L} = 0.157$.

(Fig. 5.3(a)) so that its centroid falls in the black region in Fig. 5.4, the template converges to the correct configuration. The value of σ which we used to obtain the potential energy field is 11 pixels. The extent of the convergence region is about 40 pixels.

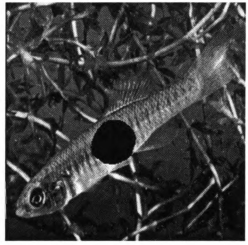


Figure 5.4: Sensitivity of the matching algorithm to the initial position.

5.3 Object Tracking

Object tracking is a challenging and important problem in computer vision. Tracking can be of interest both over time (video sequences) or through space (2D slices of a 3D structure). Tracking over time provides useful information about scene changes and motion parameters, while tracking in space helps in recovering the 3D structure from 2D projections.

Deformable contours such as snakes [81, 131] have been applied to tracking tasks, including image-based tracking of rigid and nonrigid objects [8, 14, 17, 71, 81]. The force-driven snake model can easily incorporate the dynamics derived from time-

varying images. Kass et al. [81] have used snakes to track facial features such as lips in an image sequence. The estimated motion of these features are then used to explain facial expressions, etc. Multiple snakes were later used by Terzopoulos and Waters [132] to track more articulate facial features. Leymarie and Levine [89] have used the snake model to track cells in biological image sequences. Bascle and Deriche [15] have combined deformable region models and deformable contours in a sequential way to track moving objects, where a correlation-based region matching method was used in the first stage to roughly locate the objects, and a gradient-based contour models was then used to refine the tracking result. A deformable stochastic model was proposed by Kervrann and Heitz [83] to track objects in long image sequences, where a point distribution is used to characterize the structure and variations in the object shape.

The snake model interacts with and is attracted to salient image features under local smoothness and stretchness constraints. It is a flexible shape model for image segmentation and feature extraction, and there have been many successful applications. However, it does not inherently incorporate any global structure information except for the local regularization constraints. The configuration of a converged snake depends largely on its initialization. When a snake is used to track features in an image sequence, the converged snake in the current frame is often used to initialize the snake in the next frame. The global structure of the object shape is indirectly carried out via the initialization. However, when the image features are very weak or absent because of occlusion, the snake, due to its lack of global structure, may fail to track the shape and get distracted by spurious image features.

Unlike the "free-form" active contour model, our prototype-based deformable template explicitly encodes the global structure in shape modeling. It has potential in object tracking in image sequences due to the following reasons:

- The object of interest in the image sequence can vary from frame to frame due to a change in the view point, the motion of the object, or the non-rigid nature of the object. These shape variations can be captured by the deformable shape model;

- Although the object shape varies from frame to frame, the overall structure of the object is generally maintained. The deformable shape model can capture this overall structure by using an appropriate prototype;

- The motion or deformation between two successive frames is not significantly large so the converged configuration in the current frame can be used to provide a reasonable initialization for the next frame.

In the rest of this section, we describe how to apply the prototype-based template model to track objects in an image sequence. We investigate all the possible cues that can be used to improve the tracking results. In particular, we use image gradient, inter-frame motion and region correspondences to track the objects.

5.3.1 Tracking Criteria

Many object tracking applications share the following properties:

- The inter-frame motion of the object is small so that the object in the next frame is in the neighborhood of the object in the current frame;
- The same point on the object has a consistent color/greyscale in all the frames of the sequence;
- The boundary of the moving object has a relatively large motion field.
- The boundary of the object has a large image gradient value.

Based on the above observations, we track the boundary of an object using the following criteria:

- Shape similarity: object shapes are similar in two successive frames;
- Region similarity: the properties (color, texture) of a region in the object remain constant throughout the sequence;
- Motion cue: the object boundary should be attracted to pixels with a large amount of motion.
- Gradient cue: the object boundary should be attracted to pixels with large image gradient.

These criteria are explained in more detail in the following subsections:

Matching Regions

Suppose the deformable template delineates the object boundary accurately in the first frame. The segmented region (object) is used as a reference object $\mathcal{O}_{ref}$ for

color/texture matching for the rest of the sequence, i.e., a point on the object in each frame exhibits similar region statistics as the corresponding point on the object in the first frame.

Suppose we have successfully tracked the object up to the ith frame. Since the inter-frame motion of the object is assumed to be small, we can assume that the object boundary in the $(i+1)$th frame is enclosed in a band (shaded region in frame $(i+1)$ in Fig. 5.5) centered at the object boundary in the ith frame. An alternate way to state this is that when the object evolve from the current frame to the next frame, the corresponding point of a current boundary point in the next frame is enclosed in a disc centered at its current position. The radius of the disc depends on the inter-frame object motion. The larger the inter-frame motion, the larger the disc. This radius determines the width of the band. As the deformable template model provides the correspondence of the boundary points in different frames, we can employ the region correspondence to help tracking the boundary. When we track the object in the $(i+1)$th frame, we can first predict a radius for each boundary point based on the tracking result in the ith frame. Each point in this disc is compared to the object region around the corresponding boundary point in the reference object in terms of color/greyscale. We compute for each point in the band a color/greyscale distance to the likely corresponding points on the reference object. The mathematical description comes as follows.

Assume that the boundary of the object in the first frame consists of a linked list of n_0 points $p_0^0, p_1^0, p_2^0, \ldots, p_{n_0-1}^0$, where the superscript indicates the frame number, and the subscript indicates which point it is on the object boundary. We define the

neighborhood $\mathcal{N}^0(k)$ of the kth boundary point in the 0th frame to be the intersection of a disc centered at p_k^0 and the object region. We define the neighborhood $N(l)$ of an image pixel l to be the disc centered at l (See Fig. 5.5). For each point l in the band in the $(i+1)$th frame, we compute a matching score which measures the color/greyscale similarity to possible corresponding points on the reference object.

$$distance(l) = \min_{k, p_k^0 \in N(l)} Dist(l, \mathcal{N}^0(k)), \tag{5.1}$$

where $Dist(l, \mathcal{N}^0(k))$, the distance of pixel l to region $\mathcal{N}^0(k)$, is defined using the order statistics as follows. Let the Euclidean distance between the greyscale/color of pixel l to the greyscale/color of pixels in $\mathcal{N}^0(k)$ be $d_{lk_0}, d_{lk_1}, d_{lk_2}, \ldots, d_{lk_{N_k-1}}$, in the order of increasing distance. $Dist(l, \mathcal{N}^0(k))$ is defined as the average of the distance between the 10th percentile and the 40th percentile, that is,

$$Dist(l, \mathcal{N}^0(k)) = \frac{\sum_{i=k_m}^{k_n} d_{lk_i}}{\sum_{i=k_m}^{k_n} 1}, \tag{5.2}$$

where k_m and k_n are the 10th percentile and 40th percentile points. This statistics is used because it is robust to noise.

Given the converged deformable template in the ith frame, we can compute a distance map in the $(i+1)$th frame, where for each point in the band around the template position in the ith frame, a distance is assigned to reflect its degree of resemblance to the potential object boundary. If this pixel happens to lie on the object boundary, the distance would be very small. If it is a background pixel which

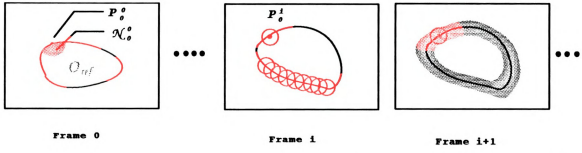


Figure 5.5: Computing color/greyscale distance. The detected object in the first frame is used as the reference object. For frame i+1, color/greyscale distance is computed for the band (shaded region) around the detected object boundary in frame i.

is quite different from the object pixels in color, the distance would be large. As a result, when we search for the possible object boundary in the new frame, it is not likely to be located in regions with large distance values. Such an example is illustrated in Fig. 5.6, where Fig. 5.6(a) shows the reference object (a human hand) segmented from the first frame, Fig. 5.6(b) shows the localized hand in the 3rd frame, Fig. 5.6(c) is the 4th input frame, and Fig. 5.6(d) shows the negated color distance map computed for the 4th frame based on the template position in the previous frame (frame 3) and the reference object in the first frame (frame 0). We can tell the bright region (small color distance) which corresponds to the hand region in the 4th frame.

Motion Cues

A moving object creates a large amount of motion at its boundary. When we subtract two successive frames and compute the absolute values of the frame difference, the boundary of the moving object is highlighted. Image subtraction has been commonly used to segment objects from their background. We use it to provide a supplemental cue for object tracking.

Figure 5.6: Computing color distance for an input frame. **(a)** The detected object in the first frame is used as the reference object (image size: 288 × 352). **(b)** The tracking result for the third frame; **(c)** The fourth input frame; **(d)** The computed color distance map (negated) for the fourth frame. The greyscale value is inversely related to the color distance. The bright region in the map indicates a potential object with a matching color composition.

The motion field is obtained by computing the absolute values of the inter-frame differences, and then smoothed using a 2D Gaussian mask.

Image Gradient

Object boundary (either static or moving) is often characterized by discontinuities in color/greyscale, which is indicated by a large image gradient. This criterion is most commonly used in image segmentation. We compute the image gradient for each image frame as the sum of squares of the color/greyscale differences along the $x-$ and $y-$ axes, and then smooth it for each frame using a 2D Gaussian mask.

5.3.2 Objective Function

To track an object in an image sequence, we deform the template so that:

1. Small deformations are preferred.

2. The template is attracted to image pixels with large gradient;

3. The template is attracted to image pixels with large motion (inter-frame distances).

4. The template deforms itself to minimize the average color/greyscale distance of the enclosed pixels;

The first goal is achieved by penalizing large deformation parameters. The remaining three goals are achieved using the following definition of image potential field.

Image Potential Field

We compute an image potential field to find a match between the deformable template and the object boundary in the image frame. The potential field for the tracking problem incorporates image gradient, color/greyscale cues, and motion cues. Let the image gradient plane for the ith frame be $\mathcal{G}_i$, the absolute difference between two frames $(i-1)$ and i be $\mathcal{M}_i$, and the color distance map for the ith frame be $\mathcal{C}_i$, then the potential $\mathcal{P}_i$ is computed as:

$$\mathcal{P}_i = -(\mathcal{G}_i \odot \mathcal{M}_i) \odot (C_{max} - \mathcal{C}_i), \tag{5.3}$$

where $\odot$ denotes pixel-wise multiplication, and C_{max} is the maximum color distance in $\mathcal{C}_i$.

The potential $\mathcal{P}_i(\mathcal{T})$ of a template $\mathcal{T}$, when placed in the tracking potential field, is the average of potentials of all the template pixels. When the template minimizes its potential, it is doing at least one of the following: (i) increasing its gradient (attracted to image edges), (ii) increasing its motion (attracted to moving boundaries), and (iii) increasing its color/greyscale similarity to the reference object.

The deviation of the template from the prototype is measured by the sum of squares of the deformation parameters. We usually prefer small deformations to large deformations. The objective function, which incorporates both the amount of

deformation and the goodness of matching, is defined as:

$$\mathcal{L}(\underline{\boldsymbol{\xi}}, x, y, s) = \omega(\sum_{\xi_i \in \underline{\boldsymbol{\xi}}} \xi_i^2) + \mathcal{P}(\mathcal{T}). \tag{5.4}$$

This objective function is minimized to match a template model to the object of interest in a sequence. We use the gradient descent method to minimize the objective function (*Eq.* (5.4)) w.r.t. the deformation parameters $\underline{\boldsymbol{\xi}}$, the translation parameters x, y and the scale parameter s.

Figure 5.7 illustrate an example of the fusion of image gradient, color consistency, and motion. After the integration of the multiple image cues, an image potential field is obtained which highlights the desired salient image features.

5.3.3 Tracking Algorithm

The tracking proceeds as follows:

1. Initialize a template in the proximity of the object in the *first* frame. Currently, this initialization is done manually. Apply the deformable template matching process until it converges. Record the converged template and the color/greyscale information on the segmented object boundary and its neighboring pixels on the object. As motion and region similarity information is not available for the first frame, only the gradient is used for the potential field.

2. Update the prototype to the converged deformable template in the current frame; Compute the color/greyscale distance map using the next frame and the

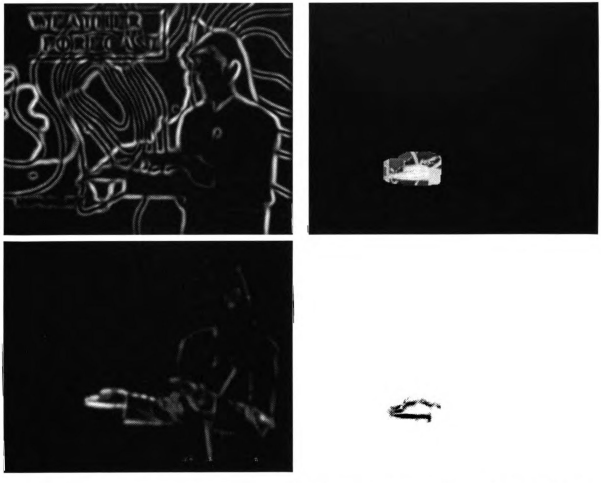


Figure 5.7: Integrating image gradient, color consistency, and inter-frame motion. **(a)** The image gradient for the input frame in Fig. 5.6(c); **(b)** The color distant map (negated) for this frame; **(c)** The inter-frame motion for this frame; **(d)** The integrated image potential field using *Eq. 5.3*.

reference object (segmented object in the first frame); initialize the template in the next frame using the pose (translation, scale) of the converged configuration in the current frame; perform the deformable template matching using the potential which combines image gradient, inter-frame motion, and color/greyscale distance map, until it converges;

3. Repeat step 2 for all the subsequent frames.

5.3.4 Experimental Results

We have applied the proposed tracking algorithm to track objects in a number of image sequences including an MRI image sequence, a home video, and a TV program.

Figure 5.8 shows the tracking result of applying the deformable template matching method to track the left ventrical in a cardiac image (MRI) sequence using image gradient information alone. The time evolution of a particular region can be an important evidence for diagnostic purposes. The difficulty for this sequence is that for frames 3, 4, and 5, the inner wall of the ventrical is hardly detectable in the bottom-right corner (Fig. 5.8(a)). The true edges of the heart wall are barely visible. The edges shown in the image do not correspond to true physiological boundaries, they are due to inhomogeneity in the acquisition and floating papillary muscles. This is also indicated by the image gradient image, where there is a very weak gradient at the locations of the inner wall, and there are spurious regions with strong gradients above the inner walls. For these several frames, the structure contained in deformable template overcomes the missing and spurious image features, and the template main-

tains its structure rather than being attracted to the spurious features. As shown, the deformable template is able to correctly follow the contractions and expansions of the left ventricle through time. In each frame, the inner wall of the left ventricle is detected. For this sequence, it takes 0.46 sec. to compute the potential field for the whole image sequence, and another 0.36 sec. to track the entire sequence on a SGI Indigo 2.

Human face tracking can be of significant importance in a number of applications, including video conferencing where the face region is located and coded at a higher bit rate than the background for transmission and storage. Figure 5.9 shows the tracking results of a human face using only image gradient information. The sequence consists of 35 frames from a video which lasts 12 secs, which was sampled at 3 frames per second. Note that the template is capable of handling partial occlusions in the frames in the bottom row. Despite the shift and rotation of the face, the deformable template can reasonably delineate the contour of the head through the whole sequence. It takes 5.058 sec to compute the potential image for the 35 frames, and another 2.532 sec to track the face for the 12 sec video clip.

Figure 5.10 shows the tracking of a human hand in a weather forecast video clip using image gradient, color region constancy, and inter-frame motion. The map in the background contains curves with very strong image gradient. The edge force in the background curves is stronger than that at the boundary of the hand. However, with the help of color and motion information, we can reasonably track the moving hand through the 15 frames.

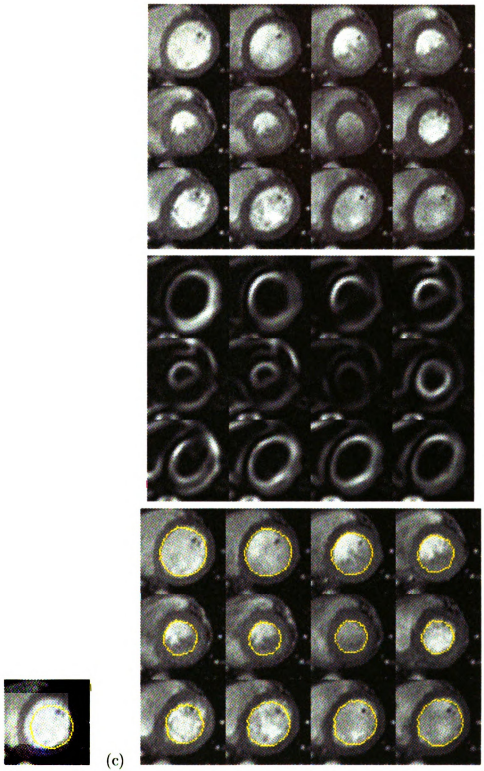


Figure 5.8: Tracking a heart in a medical image (MRI) sequence (each frame size is 64×64) using the spline representation. **(a)** Input image sequence; **(b)** the image gradient of the input sequence; **(c)** template initialization in the first frame; **(d)** tracking results for the sequence: the deformable template is able to capture the contractions and the expansions as the heart beats.

(a)

(b)

Figure 5.9: Tracking a human face in an image sequence (each frame size is 120×160). (a) template initialization in the first frame; (b) tracking results for the sequence.

5.3.5 Summary

We have presented an application of the prototype-based deformable template to the tracking of objects in image sequences from different sources. The major advantage of the prototype-based deformable model has advantage over the widely used "snake model" in tracking applications is that it inherently contains global structural information about the object shape. The inherent structure makes it less sensitive to weak or missing image features.

We have combined image gradient, region color/greyscale, and motion cues to facilitate the tracking process. In particular, we have introduced a region-based matching criterion which takes advantage of the color/greyscale constancy of corresponding object pixels throughout the sequence. We have applied the algorithm to a number of image sequences from different sources. The experimental results are promising. The inherent structure in the deformable template, together with the region, motion, and image gradient cues, make the algorithm relatively insensitive to the adverse effects of weak image features and partial occlusion.

The proposed framework is quite general and can be applied to a number of tracking tasks. Future work will incorporate temporal prediction to improve the tracking results.

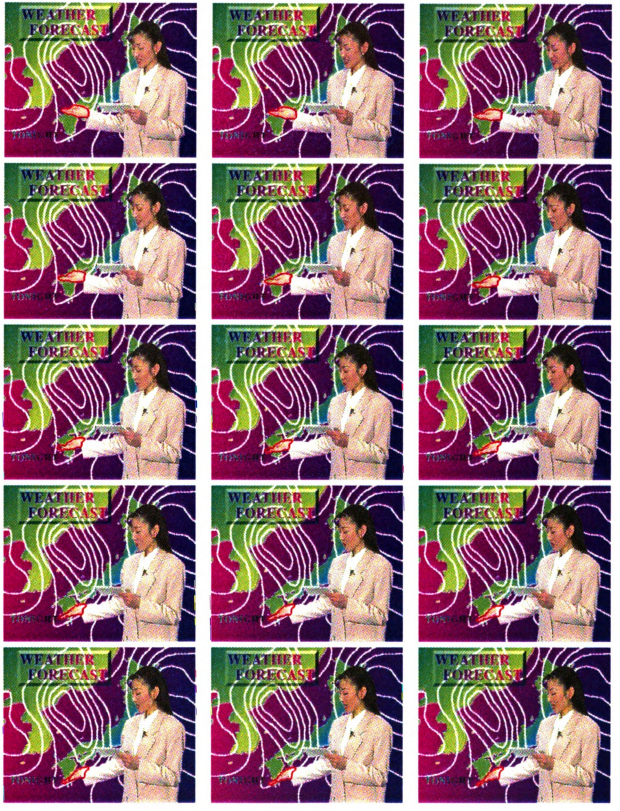


Figure 5.10: Tracking a human hand in a weather forecast TV program (each frame size is 288×352).

Chapter 6

Multi-resolution Algorithm for Localization

The localization and identification of objects in the input image involves optimizing the objective function in *Eq. (4.40)* in a high-dimensional parameter space of $s, \Theta, \underline{\xi}$, and $\underline{d}$. However, this objective function to be minimized is not unimodal. In fact, it is a rather complex function over the parameter space. The minima for this function can, in principle, be obtained by using Monte Carlo relaxation algorithms such as the Gibbs sampler [51], the Metropolis algorithm [50, 96], or the stochastic diffusion algorithms [53, 52, 61]. In all such algorithms, the minima are obtained by constructing an ergodic Markov chain whose limiting (stationary) distribution has support over only the modes of the *a posteriori* probability density function. However, because of the characteristics of stochastic sampling, these approaches achieve the optimal solution at the cost of excessive computing time.

We have implemented a multiresolution algorithm [119] to locate an object more

efficiently. The search is initiated from a smoother and subsampled energy surface to a progressively finer one, in a manner similar to searching in a scale space [13, 147, 149]. The basic idea is that by using a smoother and subsampled energy surface, the algorithm can quickly converge to the neighborhood of the optimal solution, though there is no guarantee of convergence to the correct location; by using a more accurate energy function at the finer stage, a better localization can be achieved. Note that the smoothness of the energy surface is controlled by the parameter ρ in *Eq. (4.35)*. The larger the value of ρ, the smoother the energy function.

6.1 Multiresolution Algorithm

We have employed a multiresolution approach to quickly find a good match. A global search for plausible candidate positions is performed at the coarsest stage: a smooth potential field is used with a large value of ρ in *Eq. (4.35)*. This smooth potential field has fewer spurious local minima, which helps the deformable template to find the global valleys. This stage attempts to roughly locate the global optima efficiently without regard to localization accuracy. Therefore, it can be performed at a reduced resolution. A subsampled deformable template is placed at a set of regular positions, and a set of discretized orientations, in the input image using the smooth edge potential field. The spacings between the template positions are chosen to be one fourth the size of the template so that all the significant local minima of the energy surface are covered. This computation can be done efficiently because

1. We work only with a subsampled template,

2. We use coarse step-sizes for the deformation parameters,

3. Initial positions with considerably high energy can be discarded immediately,

4. Fewer numbers of deformation parameters and iteration steps are needed since we are only interested in an approximate match, and

5. A larger step size can be used at the coarse matching level to speed up the convergence.

Finer level matchings are initialized using the promising candidates screened from the previous stage. The deformed templates obtained at stage $(l-1)$ with low energy (below a threshold) are used as initial templates for the matching at stage l. A larger number of deformation parameters and finer step sizes are used to obtain more detailed matches. We have used three stages altogether. In all the stages, the local minimization is performed by using a deterministic gradient descent algorithm. The hierarchy of step sizes from coarse to fine allows the multi-stage process to escape local minima in the deformation parameter space. This coarse-to-fine matching can automatically find acceptable solutions to the minimization problem at an affordable computational cost.

The algorithm for object localization is summarized as follows:

Begin

- *Preprocessing*
  - *Calculate the edge map and gradient direction of the input image using the Canny edge detector;*
  - *Compute the directional edge potential images at three different resolutions (coarse to fine) according to the edge map;*
- *Automatic Localization*

1. *Perform the coarsest-level matching, initialized at evenly spaced positions, and over a discretized set of orientations, using a coarse step size and a smooth edge potential field. For each of these initial positions, the matching process is as follows:*

   loop: *Calculate the objective function and the partial derivatives* w.r.t. *parameters* $s, \Theta, \underline{\xi}, \underline{d}$
   - *If the objective function is less than a threshold, goto step 2 (finer-level matching).*
   - *Else if the number of iterations exceeds a threshold, report no object of interest and stop;*
   - *Else use the gradient descent method to update the deformation parameters. Go to* loop.
2. *Perform finer-level matching initialized by the candidate templates generated by the coarsest-level search of step 1. If the objective function is less than a threshold, goto step 3 (finest-level matching). The deformation procedure is the same as described in step 1, but a finer step size and a coarser edge potential field are used.*
3. *Perform finest-level matching initialized by the candidate templates generated by the finer-level search. If the objective function is less than a threshold, output the retrieved object. Here, an even finer step-size and a less smooth edge potential field than step 2 are used.*

End

In the above algorithm, the partial derivatives with respect to the displacements d^x and d^y are approximated by finite differences. The partial derivatives with respect to the remaining parameters are obtained by the chain rule, and by using the partial derivatives of d^x and d^y with respect to those other parameters.

6.2 Experimental Results

We present the experimental results on automatically locating objects using (i) shape cues, and (ii) both the shape and texture cues. In these experiments, both the presence and the number of desired objects in the image, their position, pose and scale [1] are unknown. Localized objects, along with their poses and scales are reported. A quantitative value is also associated which each located object which describes the confidence in its matching.

The multiresolution algorithm described in Sec. 6.1 is applied to automatically search an input image for desired objects given the shape description, regardless of the size, orientation and location of the presented objects. Ideally, a global search in the parameter space is needed. However, in order to achieve a reasonable level of efficiency, we use the multiresolution algorithm to find a suboptimum solution.

In the multiresolution implementation, we discretized the template orientation into a number of different orientations (currently, we use 12 different directions) which uniformly cover the interval $[0°, 360°]$. The deformed template is initialized at each orientation. Templates with orientations within each of these direction intervals are expected to be recovered by the deformation process itself. Unlike other parameters, we do not perform a global minimization of the objective function over all the values of the scale parameter s. Instead, we settle for a minimum of the objective function when the value of s is in the local neighborhood of 1.0 (0.7 to 1.3). We initialize the

[1] For moderate scale range from 0.7 to 1.3 of the size of the template. In principle, objects of all scales can be taken care of by an exhaustive search for all scales, but this is computationally prohibitive.

coarse template at different positions. The spacing between these positions is about one fourth the size of the template. We have used three different values (12, 8 and 3) of the smoothness parameter ρ in *Eq. (4.35)* when calculating the directional edge potential fields at different resolutions. The parameter γ in *Eq. (4.40)* which reflects the relative weighting between the prior (template) and the likelihood (data) was set to 0.1. In our experiments, we found that the results were not very sensitive to the the choice of γ as long as it is in the range 0.1 to 1.0. Finally, in order to increase the speed of the likelihood-energy calculations, we pre-compute some images, namely, the Canny edge image and three potential field images corresponding to the edge potential fields at the three resolutions.

In the following we show the experimental results using the three-stage coarse-to-fine deformable template matching scheme to *automatically* search the input image for the desired shapes. Fig. 6.1 illustrates the three stages for locating a fish. The parameter space is searched at the coarse level, with the most subsampled potential field and template, fewer approximation levels and smoother potential surface for low computational cost. Only a few locations are passed to the next level to initialize the matching at a higher accuracy. At the next (finer) level, a smaller sampling rate is used, and only 6 locations are obtained and passed to the finest level, where one localization is reported when the objective function value is thresholded.

We also illustrate that our matching scheme can localize objects independent of their location, and orientation in the image. Objects of different shapes are retrieved using different prototype templates. In Fig. 6.2 we show the localizations of a guitar and a star separately. Each of the localizations is illustrated by the initial hand-

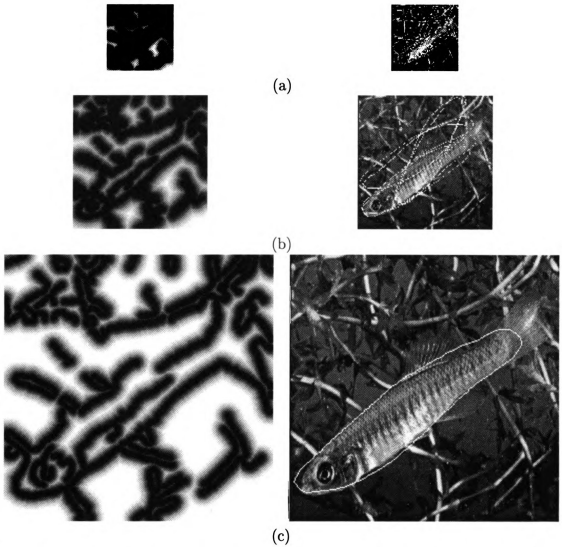
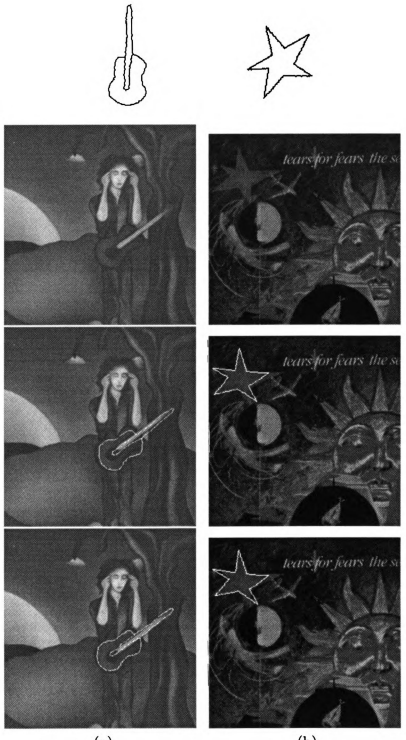


Figure 6.1: Locating a fish using the coarse-to-fine multiresolution algorithm. From left to right: edge potential field, output. **(a)** coarse level: the template and potential field are subsampled $1:4$ in each dimension, a few configurations are obtained and passed to the next level (finer); **(b)** finer level: the template and potential field are subsampled $1:2$ in x and y direction, 6 configurations are obtained and passed to the next level (finest); **(c)** finest level, 1 configuration is obtained $\mathcal{L} = 0.22$.

drawn template, the input image, the resulting template at the coarsest level (initial template for the finer level), and the retrieved object image.

In the following experiments, we demonstrate that our localization scheme is able to retrieve all the objects in an input image that resemble the prototype template. Note that these objects may be of different sizes, different orientations, and may have local variations. In [61], multiple objects resembling the same prototype template were localized by using jump-diffusion algorithms. The addition of jumps further slows down the convergence of computationally demanding diffusion algorithms. In contrast, the multi-resolution minimization method adopted in this dissertation localizes multiple objects without any additional overhead. (Each object resembling the prototype template corresponds to a different minima of the objective function. The objective function value is below the threshold at each of those minima, and hence all these objects are retrieved.) In Fig. 6.3, we have applied a "seed" template to the image of the cross-section of an orange using our multi-stage deformation algorithm. Fig. 6.3(a) shows the "seed" template and Fig. 6.3(b) shows the input image. All the resulting templates with objective function values below a threshold are presented in Fig. 6.3(c). All the six "seeds" with different orientations are retrieved correctly, and there is no false retrieval.

Classification of microbial cells in terms of morphotype has been of substantial interest to microbiologists. Microbiologists have discovered a few hundred morphotypes which cover a small portion of the bacteria in the world. Our deformable template matching algorithm may be used for locating and identifying specific microbial cells without first segmenting the image. Figure 6.4 shows the localization result for a



(a) (b)

Figure 6.2: Automatic localization of desired objects using the coarse-to-fine multiresolution matching. **(a)** retrieval of a guitar using multiresolution deformable template matching (320×304), $\mathcal{L} = 0.186$; **(b)** retrieval of a star using multiresolution deformable template matching (256×256), $\mathcal{L} = 0.157$; (From top to bottom: hand-drawn template, input image, retrieved deformed template at the coarsest level, retrieved deformed template at the finest level.)

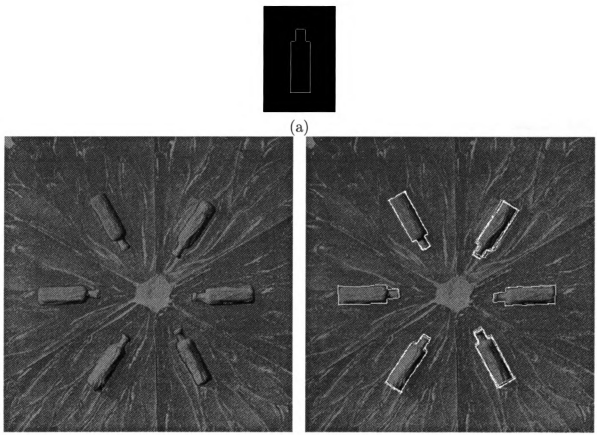


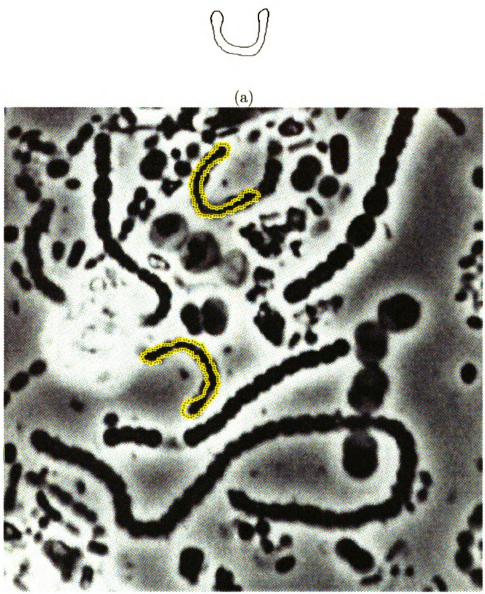
Figure 6.3: Automatic localization of "seeds" using the coarse-to-fine multiresolution matching. **(a)** a "seed" template; **(b)** the input image of the cross-section of an orange (453×436); **(c)** retrieved objects when the objective function is thresholded at 0.160.

bacteria image.

Our deformable template algorithm can also handle prototype templates that are not closed. We have applied a prototype hand template which consists of an open curve to five different hand images. Note that the hand in each of the five input images is slightly different from the others both in shape and orientation. The localization result is shown in Fig. 6.5, where 6.5(a) shows the hand-drawn template, 6.5(b) shows the five different input images, and 6.5(c) shows the corresponding retrieval and localization. Note that although the hand instances have different shapes, the deformable template is able to accommodate these variations.

In Fig. 6.6 we show the retrieval of two tower images using the same prototype tower template. The input images are pictures of the Washington monument taken at different times, and from different view points (Figs. 6.6(b) and (c)). The difference in the imaging process produces two different appearances of the tower. We have used the tower template as shown in Fig. 6.6(a) to localize the towers in the two pictures. Both the towers are correctly retrieved even though they are of different scales, orientations, and appearances.

The energy function cannot only be used to retrieve objects in an input image that resemble a prototype template, but the same function can also be used to reject the hypothesis that a certain object is present in an image. Figures 6.2-6.6 showed that when the image contains an object resembling the prototype template, then it can be retrieved and localized accurately by our approach. In the following experiments, we demonstrate that if we apply a deformable template to an image which does not contain an object of similar shape, then the resulting objective function value will



(a)

(b)

Figure 6.4: Automatic localization of microscopic forms using the coarse-to-fine multiresolution algorithm. **(a)** a morphotype template; **(b)** two localized forms ($\mathcal{L} = 0.18, 0.21$).

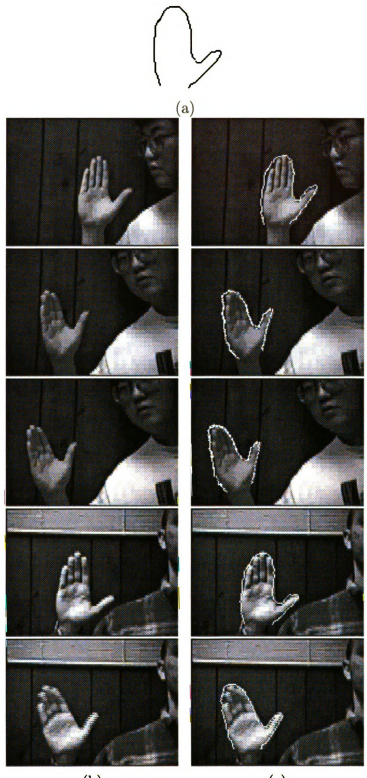


Figure 6.5: Automatic localization of human hand using coarse-to-fine algorithm. **(a)** the hand template; **(b)** input images which contain a hand (121×160); **(c)** retrieved hands ($\mathcal{L} \in [0.191, 0.267]$).

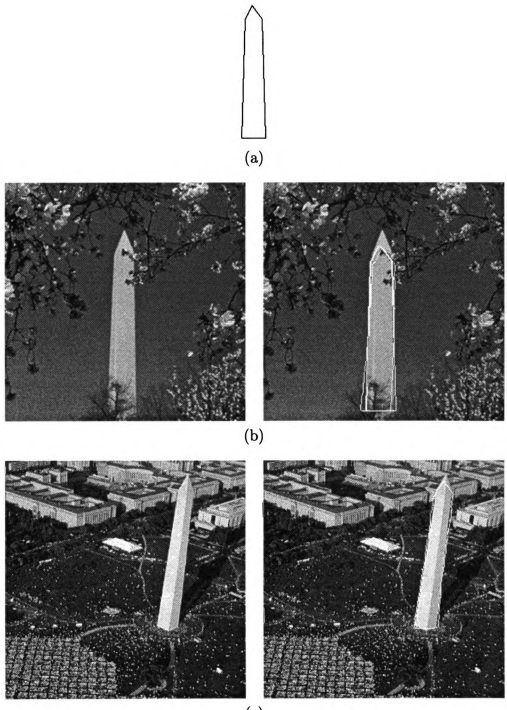
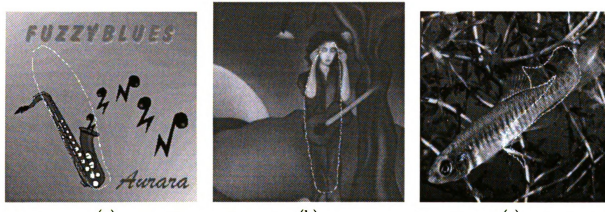


Figure 6.6: Applying a tower template. **(a)** the template; **(b)** retrieval of tower 1 (280×280), $\mathcal{L} = 0.227$; **(c)** retrieval of tower 2 (280×280), $\mathcal{L} = 0.243$.

be sufficiently large so that we can reject the hypothesis that the specified objects are present in the image. In Figure 6.7(a) a fish template is applied to the image containing a saxophone, in Fig. 6.7(b) we apply the same template to a CD cover image, and in Fig. 6.7(c) we apply a saxophone template to the fish image. In these cases, the final objective value $\mathcal{L}$ is significantly higher than the matching scores when there is a correct match.



(a) (b) (c)

Figure 6.7: What if the template is not present in the image? **(a)** applying a fish template to a saxophone image, $\mathcal{L} = 0.587$ ($\mathcal{L} = 0.142$ for the saxophone template); **(b)** applying the fish template to a guitar image, $\mathcal{L} = 0.230$ ($\mathcal{L} = 0.142$ for the guitar template); **(c)** applying a saxophone template to a fish image, $\mathcal{L} = 0.430$ ($\mathcal{L} = 0.170$ for the fish template).

The computation time for the multiresolution matching algorithm depends both on the size and complexity of the input image and the template. Usually more search time is required if (i) the image size is large, or (ii) the template size is large, or (iii) the image is complex with cluttered or textured content. Excluding the computation of the edge map and the directional edge potential field, it takes about 8.8 seconds to retrieve the hand template from an image of size 121×160, 7.8 seconds to retrieve the saxophone (image size: 285×286), and 9.2 seconds to retrieve the fish (image

size 256×256) on a Sun Sparc 20 workstation.

Chapter 7

Retrieval From Image Databases

We are now living in the age of multimedia, where digital libraries are beginning to play a more and more important role. In contrast to traditional databases which are mainly accessed by textual queries, digital libraries, including image and video databases, require representation and management using visual or pictorial cues. The current trend in image and video database research reflects this need. A number of content-based image database retrieval systems have been designed and built. Among them, QBIC (Querying by Image Content) [102] can query large on-line image databases using image content (color, texture, shape, geometric composition). It uses both semantic and statistical features to describe the image content. Photobook [106] is a set of interactive tools for browsing and searching image databases. It uses both semantic-preserving content-based features and text annotations for querying. Vinod and Murase [140] proposed to locate an object by matching the corresponding DCT coefficients in the transformed domain [140]. A similar approach has been applied to computational videos [152]. Color, texture and shape features have also been applied

to index and browse digital video databases [152]. The compression standard for videos, MPEG-4, is quite different from the former standards MPEG-1 and MPEG-2, in that it is object-oriented and allows object-based interactivity and scalability. In MPEG-4, each video frame is a composition of video objects (VO), and each VO is associated with shape, texture and motion attributes. The most recent standard under development, MPEG-7, is aimed to facilitate representation and retrieval of multimedia databases including image, video and audio. For all these applications, shape, as an important visual cue for human perception, plays a significant role. Queries typically involve a set of curves which need to be located in the images or video frames of the database.

Speed and accuracy are two important issues in the design of a database retrieval system. The retrieval accuracy can be defined in terms of *precision* and *recall* rates. The precision rate is the percentage of retrieved items which are similar to the query among the total number of retrieved items. It measures the amount of correctly retrieved items. The recall rate is the percent of retrieved items which are similar to the query among the total number of database items which are similar to the query. A good retrieval system should have high precision and recall rates, although, in reality, we have to compromise between the two.

In most of the existing retrieval systems, the challenge is to extract appropriate features such that they are representative of a specific image/object attribute and at the same time, are able to discriminate images/objects with different attributes. Once features have been extracted to characterize the image property of interest, the matching and retrieval problem is reduced to computing the similarity in the feature

space and finding database images which are most similar to the query image.

However, it is not always clear whether a given set of features is appropriate for a specific application. Feature-based methods cannot be applied when the objects of interest have not been segmented from the background. The proposed deformable template matching algorithm does not compute any specific shape features. It provides an appealing solution for the retrieval tasks because of its capability to (i) model an overall shape, (ii) accommodate shape variations, and (iii) easily handle different shapes. However, the generality of the approach and avoidance of segmentation are achieved at the cost of expensive computation. As a result, the DTM method is more suited for off-line retrieval tasks rather than online retrievals.

In order to make the DTM method feasible for online retrievals, we have adopted a hierarchical retrieval scheme. In the first (screening) stage, the database is browsed using some simple and efficient matching criteria; in the second stage, the DTM is applied to the small set of choices obtained in the first stage. This hierarchical mechanism can improve both efficiency and accuracy.

We have designed two image database retrieval systems using the hierarchical architecture. One is a shape-based retrieval system for binary trademark image databases. Another is a two-stage retrieval system for general image retrieval using color, texture and shape. We will introduce each of the systems in the rest of the chapter.

7.1 A Shape-based Retrieval System for Trademark Image Databases

A trademark identifies and distinguishes the source of goods or services of one party from those of others. There are over a million registered trademarks in the U.S. alone, and they represent a number of goods and products which are sold by different manufacturers and service organizations. Most of the trademarks take the form of a symbol or a design, like an abstract drawing of an animal, or a natural object (Sun, Moon, etc.).

When a trademark applicant registers for a new trademark, there must be an assurance that the new design does not coincide or create confusion with any of the already registered logos. Resembling trademarks can result in disputes over copyright, etc., and determining the possibility of such a conflict is based on the similarity in the binary shapes according to the USPTO (U.S. Patent and Trademark Office). It is tedious and expensive to compare the new design to the thousands of registered trademarks manually. A shape-based retrieval system can facilitate the searching process by browsing for a small set of matching candidates.

We have incorporated the deformable template matching algorithm with multiple simple shape cues into a retrieval system for a binary trademark image database, where the user provides a hand-drawn trademark, and similar trademarks in the database are to be retrieved [138]. The trademark image database retrieval system achieves both the desired efficiency and accuracy using a two-stage hierarchy:

- in the first stage, simple and easily computable statistical shape features are used to quickly browse through the database to generate a moderate number of plausible retrievals;

- in the second stage, the outputs from the first stage are screened using the proposed deformable template matching process to discard spurious matches.

This system is outlined in Fig 7.1. The first stage of the hierarchy acts as a quick browser where the database images are pruned using easily calculated shape indices, including the invariant moments and the edge direction histogram [138]. The output of this stage is a relatively small set of candidate logos which have similar shape indices as the query trademark. Those candidates can, however, be quite different from the query shape visually because the simple shape features used in the pruning stage are not "information preserving". The deformable matching model described in Sec. 4.3, then acts as a screener in the second stage to discard the spurious candidates, where the objective function values are used as a dissimilarity measure between images. In principle, we would like to apply the deformable matching process to every database image. This is prohibited by the relatively high computational cost of the deformable matching method. That is why the hierarchical architecture is used to give both a high *precision* and a high *recall* rate.

7.1.1 Browser Using Simple Shape Features

In the trademark image database, each trademark image is an object which needs to be matched against the others. In other words, there is no segmentation problem

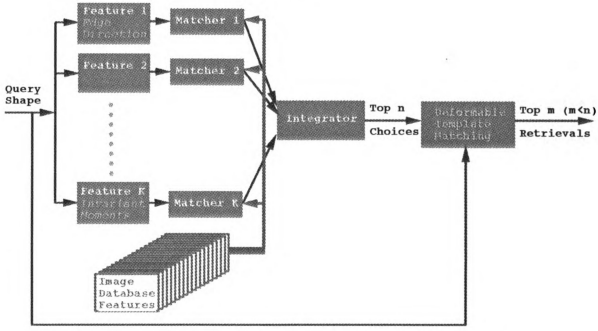


Figure 7.1: The hierarchical trademark database retrieval model.

here, and we can compare one logo image directly to another. So, we can match two trademark images by projecting them into a feature space and computing the distance between them. A large distance would imply that the two trademarks are quite dissimilar.

Speed is a crucial requirement for matching, so we use shape features that are efficient to compute. Invariant moments which are invariant to scaling, translation, and rotation are well-known shape features which have been demonstrated to be effective for matching and recognition. Edge direction histogram is another shape feature which is invariant to translation and scaling. We have used these two features since they were demonstrated to be effective for trademark shapes [74].

Edge Direction Histogram

We use the histogram of the tangent direction of the edge pixels as a shape feature. For example, a circle has a uniformly distributed histogram, and a n-polygon corresponds to a histogram with n nonzero bins. The edge direction histogram has the following desirable properties:

- it is invariant to translation;
- the normalized histogram is invariant to scale change;
- a circular shift in the histogram corresponds to a rotation in the image.

Once the normalized histogram is computed for a shape, we use the Euclidean distance as the dissimilarity measure.

Invariant Moments

For a 2-D image, $f(x, y)$, the central moment of order $(p + q)$ is given by [40]:

$$\mu_{pq} = \sum_x \sum_y (x - \overline{x})^p (y - \overline{y})^q f(x, y), \quad p, q = 0, 1, \ldots \tag{7.1}$$

where $(\overline{x}, \overline{y})$ is the centroid of $f(x, y)$. Seven moment invariants based on the 2nd- and 3rd-order moments are given as follows:

$$\begin{aligned} M_1 &= (\mu_{20} + \mu_{02}), \\ M_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, \end{aligned}$$

$$
\begin{aligned}
M_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2, \\
M_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2, \\
M_5 &= (\mu_{30} + \mu_{12})(\mu_{30} - 3\mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\
&\quad +(3\mu_{21} - \mu_{03})(\mu_{21} + 3\mu_{03})[3(\mu_{03} + \mu_{21})^2 - (\mu_{21} - \mu_{03})^2], \\
M_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2 \\
&\quad +4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}), \\
M_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\
&\quad -(\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{03} + \mu_{21})^2 - (\mu_{21} - \mu_{03})^2]. \qquad (7.2)
\end{aligned}
$$

Moments M_1 through M_6 are invariant under rotation and reflection, but M_7 is invariant only in its absolute magnitude under a reflection. Scale invariance is achieved through the following transformations:

$$
\begin{aligned}
&M_1' = M_1/n, \quad M_2' = M_2/r^4, \quad M_3' = M_3/r^6, \\
&M_4' = M_4/r^6, \quad M_5' = M_5/r^{12}, \quad M_6' = M_6/r^8, \\
&M_7' = M_7/r^{12}, \qquad (7.3)
\end{aligned}
$$

where n is the number of pixels in $f(x, y)$ and r is the radius of gyration of the object:

$$r = (\mu_{20} + \mu_{02})^{1/2}.$$

Again, the Euclidean distance is used for finding the similarity between the two

shapes represented as moment vectors.

Combining Simple Shape Features

The edge direction histogram captures local boundary information, but it is a poor descriptor for the overall object shape. On the other-hand, invariant moments offer a global description about the object shape. In theory, if an arbitrary large number of invariant moments are kept, we can reconstruct the shape exactly. But in practice, we only use a small set of invariant moments for computational efficiency and noise insensitivity. Each of the two feature sets describes different shape attributes. We combine both of them to give a more accurate and complete description about the object shape. Let P and Q be two shapes under comparison. Let D_e be the dissimilarity measure between P and Q using edge direction histograms and D_m be the dissimilarity measure between P and Q using invariant moments. Jain and Vailaya [74] define an integrated distance measure D_i between P and Q as

$$D_i = \frac{w_e * D_e + w_m * D_m}{w_e + w_m}, \tag{7.4}$$

where w_e and w_m are the weights assigned to the edge direction histogram-based similarity and the invariant moment-based similarity, respectively. We have used equal weights ($w_c = w_s = 1$) in our experiments.

In the browsing stage, the edge direction histogram and invariant moment features are computed for the query trademark shape, then the feature vector is compared to the precomputed features of the trademark images in the database. The distances are

sorted and the top N retrieved images are returned based on the integrated distance. Note that a large value for N decreases the miss rate, but increases the computational cost. When we choose a value for N, we need to compromise between the two. Of course, a good choice of N also depends on the size of the database, and the number of matching items in the database. We have used $N = 10$ in the experiments.

7.1.2 Refinement Using Deformable Template Matching

Both the edge direction histogram and the invariant moments used in the previous section are necessary but not sufficient measures for shape matching. In other words, two dramatically different shapes can have very similar edge direction histograms and invariant moment feature vectors. It is, however, observed that, using the above features, database images which are similar to the query in shape are likely to be among the top N retrievals. But, some of the retrieved images also contain trademarks that seem to be perceptually very different than the query image. To further refine the retrievals and guarantee that only visually similar shapes are reported to the user, we employ a more elaborate matching technique based on deformable templates [76]. This matching scheme, which attempts to establish a point to point correspondence between the query image and the database image and to quantify the cost of the correspondence, provides a better similarity description at a higher computational cost. During the refined matching stage, the edge map of the query trademark is compared to the edge maps of only the top N retrieved trademark images. The query trademark is used as the prototype template.

The generalized Hough transform is an elegant technique to detect the presence of an arbitrary shape, given the corresponding tabular form [11] (See Sec. 2.1.2). It also determines the pose and scale of the detected shape by searching for the peaks in the accumulator array of the transformed parameter space. However, the major disadvantages of this method are as follows:

1. It is computationally expensive; the complexity increases exponentially with the number of parameters, and

2. It is a rigid template matching method, so it is very sensitive to the deviations from the tabulated shape.

We use the generalized HT to calculate a set of poses and scales to initialize the deformable template matching process, using the tabular form of the prototype template.

Whether a trademark image matches the query or not is determined by the final objective value returned by the deformable template matching algorithm. The final retrieved images are ranked according to the objective function value.

7.1.3 Experimental Results

We have applied the hierarchical retrieval method to a trademark image database to evaluate the performance of the system. The database contains 1,100 binary images scanned from several books [68, 69, 1] using an HP flatbed scanner (See Fig. 7.2).

A query consists of a hand drawn image of a shape, which may be disconnected, or may contain holes. Figure 7.3 shows five hand drawn query trademarks used in

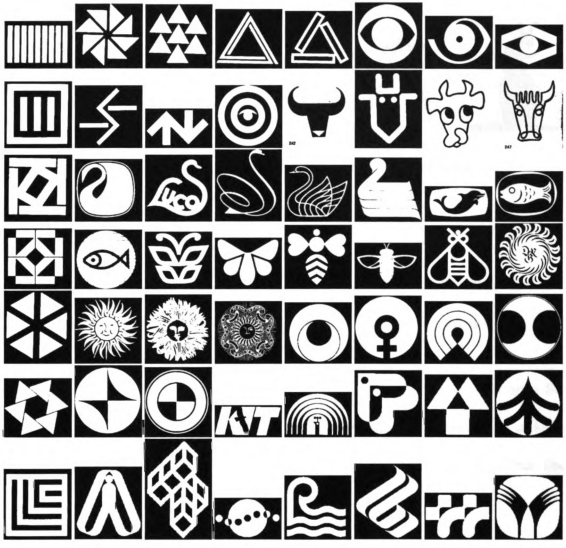


Figure 7.2: Some images from the trademark image database.

the experiments.

The corresponding edgemaps of the queries, which are used as the prototype template are depicted in Fig. 7.4.

Fast pruning

The query image is compared to the database images based on the edge direction histogram and invariant moment shape features using the integrated dissimilarity

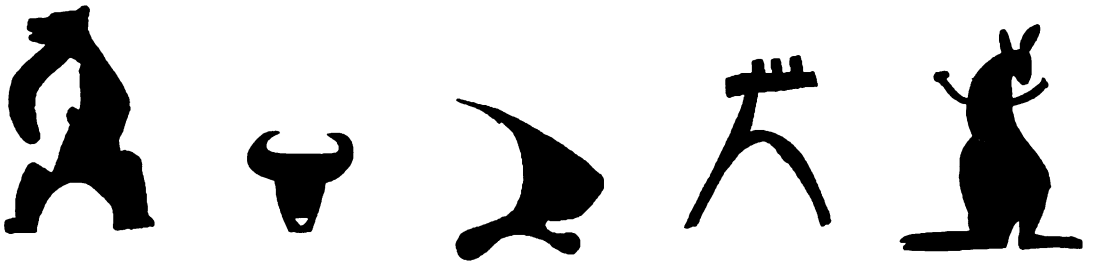


Figure 7.3: Examples of hand drawn query trademarks.

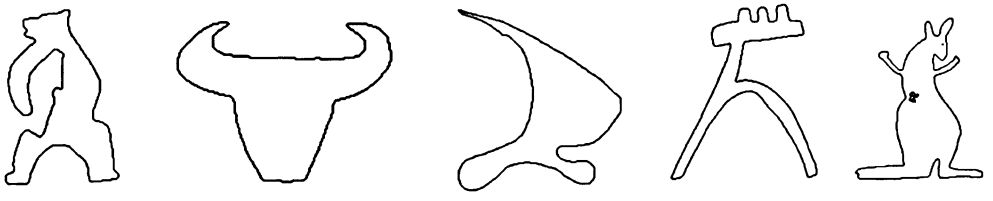


Figure 7.4: Examples of hand drawn query trademark templates.

index D_t (Eq. (7.4)). For each query, this stage takes about 45 seconds on a Sun Sparc 20 workstation. Fig. 7.5 shows the top 10 retrieved images in the order of increasing dissimilarity for a query containing a hand-drawn bull sketch. Note that the correct database image has the smallest dissimilarity value.

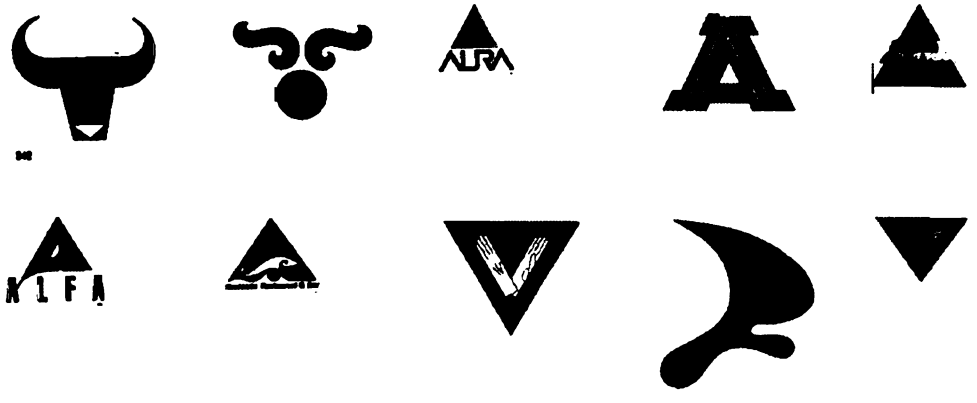
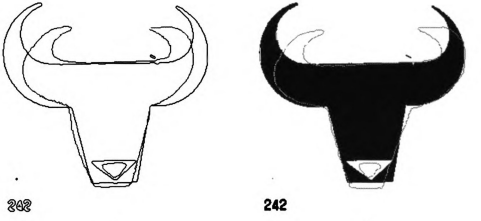


Figure 7.5: Database pruning results for the hand-drawn bull sketch as shown in Fig. 7.3: the top 10 retrievals given in the increasing order of dissimilarity.

Figure 7.6: Database pruning results for the hand drawn kangaroo shown in Fig. 7.3. The top 10 retrievals are given in the increasing order of dissimilarity.

Deformable template matching

Under the assumption that all plausible candidates for a query logo are contained in the top 10 retrievals in the fast pruning stage, we apply the deformable matching scheme on these candidates only to further refine the results. The initial pose parameters of the deformable template (position, scale, and orientation) are estimated using the generalized Hough transform. Figs. 7.7- 7.11 illustrate the initial and final configurations of the deformable template match for some trademarks.

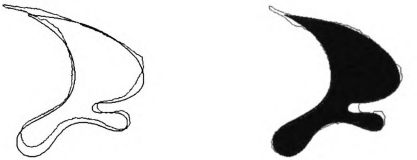


(a) (b)

Figure 7.7: Deformable template matching; **(a)** initial position of the bull template overlaid on the edge map of a bull logo, **(b)** final match for the bull logo.

template	*Top* 10 *retrievals from the fast pruning stage*									
	1	2	3	4	5	6	7	8	9	10
bull	.372	.670	.959	.856	.847	.862	.803	.784	.820	.913
boomerang	.137	.596	.731	.820	.628	.785	.794	.857	.771	.804
bear	.425	.639	.504	.509	.705	.688	.640	.669	.574	.609
kangaroo	.751	.422	.521	.630	.877	.725	.639	.628	.645	.559
deer	.392	.457	.662	.857	.677	.665	.488	.787	.686	.425

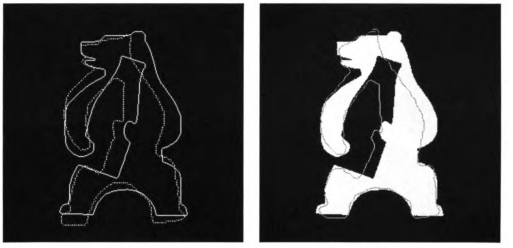
Table 7.1: Dissimilarity values for the five query images when the deformable template matching is applied to the top 10 retrieved images from the fast pruning stage.



(a) (b)

Figure 7.8: Deformable template matching; **(a)** initial position of the boomerang template overlaid on the edge map of a boomerang logo using the generalized Hough transform, **(b)** final match for the boomerang logo.

Table 7.1 presents the dissimilarity measures of the five hand-drawn logos (Fig. 7.3) to the top 10 retrieved images by the pruning stage. In four out of the five queries, the simple integrated shape dissimilarity index ranks the correct logo in the first place, and in one case, the correct logo is ranked in the second place. The dissimilarity score using the deformable matching ranks the desired images (underlined) in the first place for all the five queries. An incorrect match should result in a large dissimilarity measure (Fig. 7.12) so that the matching hypothesis is rejected. It



(a) (b)

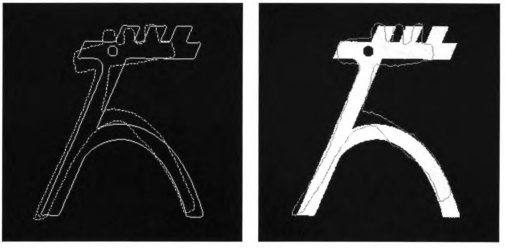
Figure 7.9: Deformable template matching; **(a)** initial position of the bear template overlaid on the edge map of a bear logo using the generalized Hough transform, **(b)** final match for the bear logo.

typically takes $5-8$ seconds to calculate the initial configuration using the generalized Hough transform. The iterative deformable matching process takes about 6 seconds on a Sun Sparc 20 workstation.

7.1.4 Machine Perception versus Human Perception

Content-based image database retrieval [41, 106] [41, 106] has been receiving increasing interest in recent years as an innovation to save human labor. Humans use color, shape, and texture to understand the contents of an image, therefore it is natural to use features based on these attributes for image retrieval. It is important that such features capture the human perception of the image content. Therefore, image features which simulate human perception are utilized to retrieve images based on their content.

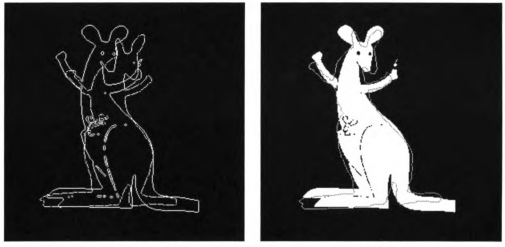
The perception and interpretation of a scene by humans is a very complex process



(a) (b)

Figure 7.10: Deformable template matching; **(a)** initial position of the deer template overlaid on the edge map of a deer logo using the generalized Hough transform, **(b)** final match for the deer logo.

which is not at all understood by vision researchers. No mathematical definition of shape can match the human's perception of shapes, which involves context, and previous knowledge and experiences. The many existing features and distance measures give only a very crude representation of the measure of shape similarity perceived by humans. There is no machine vision system currently available that can simulate the human perception. However, we do want our matching and retrieval results to agree with those of human subjects as much as possible. It is then instructive to compare the matching results with those obtained by human subjects. We have performed a limited experiment, in which human subjects were asked to retrieve images which are similar to the input query image from the trademark image database of about 1000 images. We have collected responses from five human subjects. Although a large number of human subjects would have given us more confidence in the experimental results, the retrieval process is so dull, stressful, and tiring, that we could find only a



(a) (b)

Figure 7.11: Deformable template matching; **(a)** initial position of the kangaroo template overlaid on the edge map of a kangaroo logo using the generalized Hough transform, **(b)** final match for the kangaroo logo.

few volunteers. (This is one of the reasons that an automatic image database retrieval system is desired.) Fig. 7.13 summarizes the retrieval results using Fig. 7.13(a) as the query logo. Each human subject was asked to retrieve the top ten most similar logos from the database. Those logos which received the most ballots are displayed in Fig. 7.13(b), where under each logo we list two numbers: the first is the number of votes out of the five human subjects, the second is the dissimilarity score calculated from the deformable matching process. We note a negative correlation between the dissimilarity value of the algorithm and the similarity ranking by the human subjects. Note that the dissimilarity value is reasonably small for the good retrievals. Most human respondents retrieved all the trademark images in the database which contain a bull head, even though the shape of the bull in the retrieved images is quite different from the query shape. This indicates that human subjects tend to abstract the query image for some conceptual information.

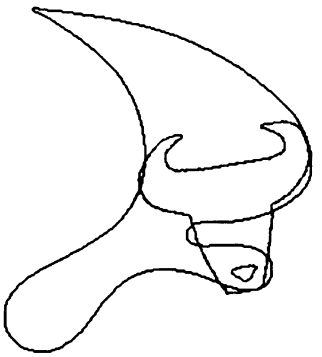
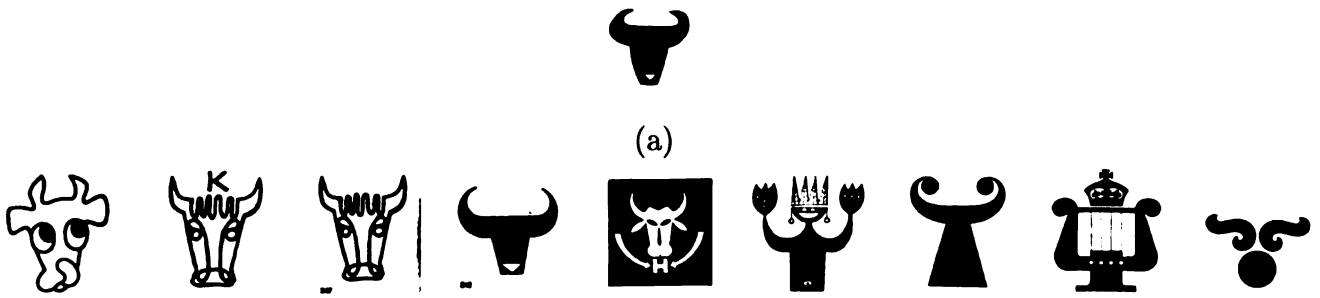


Figure 7.12: Deformable template matching result of the boomerang image using the bull template.

From the above experiments, we note that a human's perception of shape can be rather subjective and as a result, the representation of the desired object tends to vary a lot. Furthermore, human subjects tend to abstract semantic content from the scene and use it for interpretation. But, semantically similar images may actually be visually very different from each other. Image retrieval based on user-provided information such as hand-drawn sketches remains a challenging problem in multimedia applications.

7.1.5 Summary

We have presented a shape-based trademark image retrieval algorithm. Efficiency and accuracy are achieved by designing a two-stage hierarchical retrieval system: *(i)* a simple statistical feature-based process quickly browses through a database for a moderate number of plausible retrievals; and *(ii)* a deformable matching process screens



(a)

3(0.237) 4(0.324) 4(0.325) 5(0.372) 4(0.372) 4(0.409) 3(0.557) 3(0.609) 2(0.693)

(b)

Figure 7.13: Human perception versus the deformable template matching algorithm. **(a)** a hand-drawn trademark; **(b)** the top retrievals from a logo image database for the query in (a) by human subjects. The first number under each retrieved image is the number of ballots from the five human respondents; the second number is the dissimilarity measure given by the deformable matching algorithm.

the candidate set for the best matches. Preliminary results on a trademark image database show that this is a promising technique for content-based image database retrieval. The technique is robust under rotated, scaled and noisy versions of the database images.

We note that image retrieval based on user provided information such as hand drawn sketches is a challenging problem in multimedia applications. A human's perception of shape can be rather subjective and as a result, the representation of the desired object tends to vary a lot. Our goal is to extract images having similar semantic content. Figure 7.13 shows that semantically similar images may actually be visually very different from each other. In order to retrieve these images in the fast pruning stage, we need to somehow extract semantic meaning from the images. One way to extract the semantic content is through a semi-automatic (requiring manual intervention in preprocessing) scheme using textual description of the trademark images. This highlights the importance of text-based search which can then be incor-

porated prior to the shape-based retrieval of images. A more automatic and objective approach to preserving the semantic content is to extract components that represent different entities in the trademark images, and to query on the basis of these component entities.

7.2 Image Database Retrieval Using Color, Texture and Shape

We have incorporated the DTM algorithm into a content-based retrieval system. The advantage of DTM is that it does not require specific shape features, and no segmentation of the input image is necessary. However, the ability to deform the template is achieved at the cost of a search in a high-dimensional parameter space. Object matching requires either a global optimization of a non-concave (usually with many local extrema) [61] objective function or a good initialization of the template near the true location of the object in the image domain [81, 151].

We have designed a database retrieval system which integrates the three important content cues: shape, texture, and color. In particular, texture and color features are used as supplemental clues to help locate promising regions in the image which are likely to contain the desired objects. This eliminates a large portion of the database images from further screening. Once a small set of candidate regions is obtained, we then use the deformable template matching method to localize the objects in the proximity of these regions. A diagram of this system is given in Fig. 7.14.

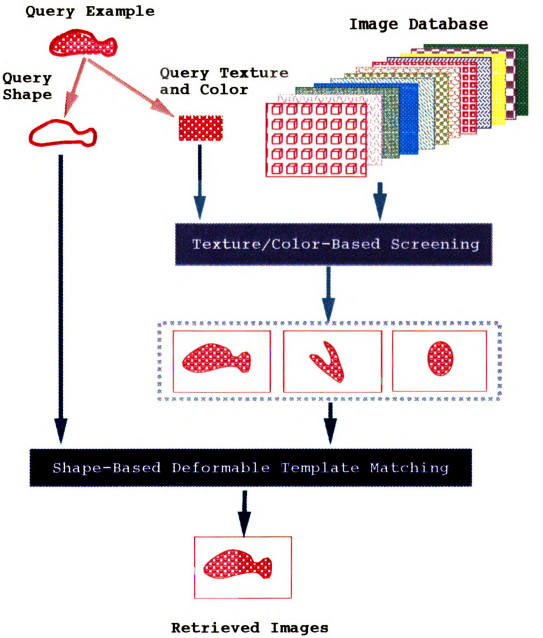


Figure 7.14: Diagram of the image retrieval system using color, texture, and shape.

The motivation of this work is threefold: (i) the region cues (texture and color) may come naturally as a constraint in the retrieval task, (ii) the region cues may be used to expedite the localization process: the deformable template matching process need not be executed where the region cues are quite different from the desired ones, and (iii) region-based matching methods are more robust to misalignment and position shift than edge-based methods. We use the region information to obtain some good yet coarse initializations. The contributions of this work are as follows: (i) we extract the color texture features directly from the compressed image data, (ii) we use the region attributes to direct the shape-based search to save computational costs, and (iii) we sensibly fuse multiple content cues to efficiently retrieve images from a non-annotated image database where the only information available is the bit stream of the images.

7.2.1 Matching Using Color and Texture

Texture and color features have been used in several content-based image database systems to retrieve objects or images of a specific texture and color composition [55, 106, 110, 109, 125]. We use texture and color cues in addition to shape information to localize objects. For example, one may be interested in finding a golden fish, with a particular shape, color, and texture. The texture and color information can be specified in terms of a sample pattern, as in the case "I want to retrieve all fish images with the same color and texture as the fish in this picture". When such image region information is available, we use these features to quickly screen the input

image for a small set of candidate positions where we can initialize the deformable template-based shape matching process.

As the color and texture cues are used as supplemental tools for examining an image for the presence of a candidate object, we need to use features which are easy to compute and at the same time, characterize the desired color and texture properties. For this purpose, we extract the features from the block DCT coefficients of an image. These coefficients can be obtained directly from DCT compressed images and videos (JPEG [141], MPEG [47]) without first decompressing them. This is very appealing since more and more images and videos are stored in a compressed format for efficient storage and transfer [118, 152].

DCT Compressed Images

DCT-based image compression techniques encode a two-dimensional image by the block DCT coefficients. To compress an image, the DCT coefficients of each $N \times N$ image block (macroblock) are computed and quantized. These compression techniques take advantage of the fact that most of the high frequency components of the transformed image are close to zero. The low-order coefficients are quantized to save the bits, and then further compressed using either the Huffman coding or the arithmetic coding method. The JPEG images and Intra frames of MPEG videos are compressed this way, where the value of N is set to 8.

The DCT coefficients $\{c_{uv}\}$ of an $N \times N$ (N is usually a power of 2) image region

$\{I_{xy},\ 0 \leq x < N, 0 \leq y < N\}$ are computed as follows:

$$c_{uv} = \frac{1}{N}\mathcal{K}_u\mathcal{K}_v \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I_{xy} \cos\frac{\pi u(2x+1)}{2N} \cos\frac{\pi v(2y+1)}{2N} \tag{7.5}$$

where u and v denote the horizontal and vertical frequencies $(u, v = 0, 1, \ldots, N-1)$, and $\mathcal{K}_w = \frac{1}{\sqrt{2}}$ for $w = 0$ and $\mathcal{K}_w = 1$, otherwise. The DC component (c_{00}) of the transformed coefficients represents the average of the spatial domain signals I_{xy} in the macroblock, and the AC components ($c_{uv},\ u \neq 0$ or $v \neq 0$) capture the frequency (characterized by u and v) and directionality (by tuning the u and v values) properties of the $N \times N$ image block.

One property of the Discrete Cosine Transform is that for a typical image, its energy is dominant at the low frequency components. This means that the coefficients of the high frequency components are close to zero, and therefore negligible in most cases. Most of the image information is contained in the low frequency components, which represent a "coarse" or "blurred" version of the spatial image. We will now show how we extract texture and color features from DCT coefficients.

Texture Features

An image region is textured if it contains some repetitive gray level pattern. Texture is usually characterized by the spatial variation, randomness, contrast, directionality, and coarseness in the image [6, 120]. Textured images provide rich information about the image content. Numerous texture features have been proposed in literature which include local extremas [80], co-occurance statistics [12], fractal dimensions

[25], Fourier transform energy and multichannel filters. The multichannel filtering approach has been used extensively in texture analysis. This includes the SAR model (Simultaneous Moving Average) and its variations [84, 94, 133], the Gabor filter based approach by Jain and Farrokhnia [73], the wavelet transform model by Chang and Kuo [24], and the subband approach by Jernigan and D'Astous [78], to name a few.

As the Discrete Cosine Transform converts the spatial image information into the spatial frequency domain, we define texture features as the energies in different channels of a local macroblock. The absolute values of the AC components of the quantized DCT coefficients of each macroblock indexes the channel spectrum. We use them as the texture features which are expected to capture the spatial variation and directionality of the image texture. The DC component, which is the average greyscale value of the macroblock, is not considered a texture measure. This is reasonable because we usually subtract the mean or normalize the image before extracting texture features.

Color Features

The **YCrCb** color model is widely used to encode color images in TV and video and in compression standards, including JPEG and MPEG. This color space is obtained by applying a linear transformation to the **RGB** color space where the **Y** plane represents the luminance information, and the **Cr** and **Cb** planes encode the chrominance differences. The advantage of this color model is that human eyes are usually more sensitive to the luminance changes than to the chrominance changes. As a result, the chrominance frames can be encoded at a lower bit rate than the luminance frame

for compression purposes, without significantly affecting the quality of the perceived image.

In line with the JPEG and MPEG standards, we use the **YCrCb** model for representing color images. We use the DC components of the DCT coefficients of the three frames **Y**, **Cr** and **Cb** to represent the color for a macroblock. We note that although the intensity (the **Y** plane) is subject to lighting conditions, the **Cr** and **Cb** components are more robust indicators of the color attribute. However, for image retrieval tasks, people do distinguish between bright red and dark red. So, the intensity also plays a role in color perception.

We should note that although we use the DC component of DCT for representing the color attribute and AC components for texture, we believe that texture and color properties are mingled together. A variation in color results in color texture. It is often difficult to draw a clear boundary between color and texture.

Feature Selection

There are N^2 DCT coefficients for an $N \times N$ image block; for an 8×8 macroblock, there are 64 coefficients. Not all the coefficients contain useful information for image similarity computation. As mentioned earlier, for a typical image a large portion of the high frequency components have negligible coefficients. We use the following two different criteria to choose only M features out of the N^2 total number of features, $M << N^2$:

1. Take the M lowest frequency components. That is, we pick $|c_{10}|$, $|c_{01}|$, $|c_{20}|$, $|c_{11}|$, $|c_{02}|$, ... and so on, until

we have selected M features;

2. Find the M features which maximize the energy for the query image. This criteria adapts to the query image and proceeds as follows:

   (a) obtain the quantized DCT coefficients for all the DCT blocks for the query object region;

   (b) compute the absolute values of the AC components as features;

   (c) sum up the energies for each frequency component over all the DCT blocks in the region;

   (d) select those M features that have the most energy over all the blocks.

The texture features are extracted separately for each of the three color frames (Y, Cr, Cb). It turns out that for most cases, the two criteria select the same set of features. When the query image presents very fine texture, the second criteria selects a set of features which outperforms the first one.

Representing the Query Image Region

The query image is represented by a set of feature vectors. Each vector corresponds to an $N \times N$ block in the query image region. We allow the overlapping of the macroblocks so that the blocks densely cover the query region, and all the $N \times N$ configurations in the query region are covered. The DCT coefficients of a non-aligned block can be computed from the DCT coefficients of its four overlapping, aligned macroblocks using the algorithm proposed by Chang and Messerschmitt [23].

Each feature vector consists of the color and texture features which are extracted as specified in sections 7.2.1 and 7.2.1. If the number of features is large then we cluster all the feature vectors, and only keep the features corresponding to the cluster centers to maintain a small set of representative features.

Similarity Computation

We have represented the query region attributes using a set of feature vectors (Sec. 7.2.1) which characterize color and texture. In the same manner, we can also extract a set of feature vectors to represent a region in the test image, one vector for each macroblock in this region. Then we can match the query region to an arbitrary region in the database image by comparing the two characteristic feature vector sets. We have derived a symmetric distance measure between query feature set Q and a test region feature set R. First, we define the color and texture distances of the ith feature vector in set R to vector set Q as the distance to the vector in Q which gives a minimum distance taken over all vectors in Q:

$$\mathbf{dist}_{text}(R_i, Q) = \mathbf{Min}_{j \in Q} \frac{1}{N} \sum_{k=0}^{N-1} \frac{(ftext_{ik} - ftext_{jk})^2}{vartext_k} \tag{7.6}$$

$$\mathbf{dist}_{color}(R_i, Q) = \mathbf{Min}_{j \in Q} \frac{1}{3} \sum_{k=1}^{3} \frac{(fcolor_{ik} - fcolor_{jk})^2}{varcolor_k}, \tag{7.7}$$

where R_i denotes the ith feature vector in R, $ftext_{ik}$ ($fcolor_{ik}$) denotes the texture (color) feature k for vector i, and $vartext_k$ ($varcolor_k$) denotes the variance of texture (color) feature k in the database. The weighted distance measure is used because the

DC component usually has a very large variation, the low frequency AC features have a smaller variation, and the high frequency AC components have the least variation. We weigh the contribution of each feature by the variance of each feature component computed from all the macroblocks in the database images. (This is equivalent to the Mahalanobios distance with a diagonal covariance matrix.) The distance of the *ith* vector in R to the query set Q is the summation of the distances in color and texture space:

$$\mathbf{Dist}(R_i, Q) = \mathbf{dist}_{text}(R_i, Q) + \mathbf{dist}_{color}(R_i, Q). \tag{7.8}$$

The distance of set R to set Q is defined as the average distance of vectors in R to Q:

$$\overline{\mathbf{Dist}(R, Q)} = \sum_{i=1}^{N_R} Dist(R_i, Q)/N_R. \tag{7.9}$$

where N_R is the number of feature vectors in R. Note that this distance is asymmetric. We define a symmetric distance measure between R and Q as follows:

$$\mathbf{DIST}(R, Q) = \frac{1}{2}(\overline{\mathbf{Dist}(R, Q)} + \overline{\mathbf{Dist}(Q, R)}). \tag{7.10}$$

Note that the color/texture features, which are the spectrum energies of local blocks, are not necessarily invariant to scaling, rotation, and translations which are not in the multiples of the block size. However, because the distance measure described in *Eq. (7.10)* is based on the Euclidean distances between blocks from the two regions, and the integration of the individual block distance does not depend on the geometric composition of the blocks in a region, it is relatively invariant to trans-

lation, moderate rotation, and rearrangement of the blocks in a region, as long as the applied transformation maintains the block integrity. It is not invariant to transformations which break the block integrity, such as scaling. Wavelet transforms are also used to compress images and represent texture [24, 135, 154], where the coefficients of the wavelets are used to characterize a texture. Compared to our method, the wavelet representation captures texture attributes at different scales. Vector Quantization (VQ) is another related technique [103] with similar performance. Our choice is a convenient and quick solution for DCT compressed images, though it may not be very sophisticated.

7.2.2 Integrating Texture, Color and Shape

We have integrated texture, color, and shape cues to improve the performance of the retrieval process. The integrated system operates in two stages. Since region-based matching methods are relatively robust to minor displacements as long as the two matching regions substantially overlap, we browse the database using color and texture in the first stage, so that only a small set of images, and a small number of locations in the candidate images are obtained. In particular, image regions that are very different from the query region in terms of the defined texture and color measures are excluded from the second stage which uses the more elaborate deformable template matching method. Only the configurations that most resemble the query regions in texture and color will trigger a deformable template matching process. In this way, we prune the possible initializations of the deformable template using texture and

color. In the second stage, the identified regions with the desired texture and color are used to direct the shape-based search, so that the iterative matching process is only performed in the proximity of those candidate locations.

The integrated matching algorithm is described as follows:

Region-based screening:

- Compute feature vectors for the query region:
  - Extract the quantized DCT coefficients for the macroblocks in the sample region;
  - Compute DCT coefficients for the other displaced 8×8 blocks from the DCT coefficients of the 4 overlapping macroblocks;
  - Form the color and texture feature vectors for each block, as described in Section 7.2.1;
  - If the number of sample blocks exceeds a threshold, cluster the sample feature vectors; keep the cluster centers as the representative sample feature vectors;
- Find similar images in the database:

  for each database image,

  for each macroblock in the database image:

  - compute the color and texture feature vectors;

  place the masked query shape at evenly spaced positions, and over a discretized set of orientations; compute the distance between the query texture and color attributes and the masked input image region as described in section 7.2.1. If the distance is less then a threshold, initialize shape-based matching.

Shape-based matching:

- initialize the query template at the computed configurations from the previous stage for M iterations; if the final objective function value is less than a threshold, report the detection;

7.2.3 Experimental Results

We have applied the integrated retrieval algorithm to an image database of 592 color images containing people, animals, birds, fish, flowers, outdoor and indoor scenes,

etc. These images are of varying size from 256×384 to 420×562. They have been collected from different sources including the Kodak Photo CD, web sites (Electronic Zoo/Net Vet - Animal Image Collection URL: http://netvet/wusti.edu/pix.htm), and HP Labs. Some sample images from the database are illustrated in Fig. 7.15.

Figure 7.15: Some sample images from the database. They have been "scaled" for display purposes.

To gain some insight into the DCT spectrums which we have used as texture and color features, Fig. 7.17 shows the absolute value of block DCT coefficients of a color image of houses (Fig. 7.17(a)). Figures 7.17(b)-(d) show the absolute values of the DCT coefficients for the three color components separately. Each small image (block) corresponds to the spectrum of a specific channel, that is, one feature for all the macroblocks in the image. The x-axis (across the features) indicates horizontal variations, and the y-axis (across the features) indicates vertical variations, with

increasing frequencies from left to right, top to bottom. So, the block at the top left corner corresponds to the DC component, which is the averaged and subsampled version of the input image, and the small images on the top row, from left to right, correspond to channels of zero vertical frequency, and increasing horizontal frequencies. This figure shows that the top left channels, which represent the low frequency components, contain most of the energy, while the high frequency channels, which are located at the bottom right corner of each figure, are mostly blank. It also indicates that the channel spectrums capture the directionality and coarseness of the spatial image; for all the vertical edges in the input image, there is a corresponding high frequency component in the horizontal frequencies, and vice versa. Furthermore, diagonal variations are captured by the channel energies around the diagonal line. This example illustrates that the DCT domain features do characterize the texture and color attributes.

The user specifies the query texture and/or color by example, in a similar manner as the other content-based image database retrieval systems [102, 106]. A user interface for specifying query texture and color cues is shown in Fig. 7.16, where the user can load an input image and specify a subwindow or a convex polygon in the image using the mouse. The texture and color features of the macroblocks in this specified region are then computed as the reference feature vectors for the texture/color cues.

We now show the retrieval results using only texture and color, as described by the first stage of the integrated algorithm. Figure 7.18 shows one example of color matching, where the image in the subwindow in Fig. 7.18(a) is the query sample, and Fig. 7.18(b) gives the top-4 retrieved images from the database. The three DC

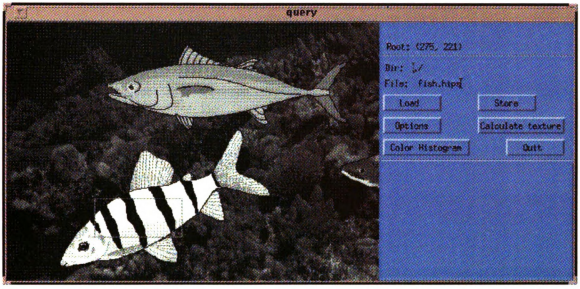


Figure 7.16: Interface for specifying reference texture/color.

components of the color frames are used as the color features.

Figure 7.19 shows one matching result using the texture features. Five features are selected from each of the **Y**, **Cr**, and **Cb** frames, so that a total of 15 features are used. Figure 7.19(a) specifies the query textured region, Fig. 7.19(b) shows the matching macroblocks in the same image, and Fig. 7.19(c) shows the top-10 retrieved regions with similar texture.

One example of object localization using color and shape is illustrated in Fig. 7.20, where the rectangular region in Fig. 7.20(a) specifies the sample color. Matching macroblocks in the same images are identified by 'x', as shown in Fig. 7.20(c). Note that almost all the blocks on the fish where the query is extracted are marked. So is part of another fish with a similar blueish color. No blocks in the background pass the color matching test. Shape matching using the hand-drawn sketch in Fig. 7.20(b) is then processed around the two detected regions. The final matched result is shown in Fig. 7.20(d). The final configuration of the deformed templates agrees in most part

	images retrieved	computation time [0]
Stage 1	11%	0.1 sec
stage 2	1.2%	1.76 sec

Table 7.2: Performance of the two-stage algorithm; the database contains 592 color images.

with the fish boundaries. The deviations from the fish boundary are due to the edges extracted in the textured background. Note that although there is another striped fish in the image, it is not localized due to its different color.

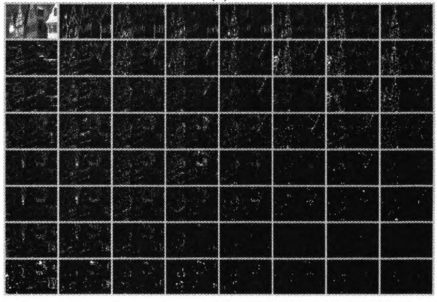
We show another example of the integrated retrieval in Fig. 7.21. One region is extracted from a cardinal to specify the query color and texture, as shown in Fig. 7.21(a). A sketch of a side view of a bird is used as the shape template (Fig. 7.21(b)). One cardinal image is retrieved from the database using the combined shape and region information (Fig. 7.21(c)).

The performance of the system is summarized in Table 7.2. Using texture and color, we can eliminate a large portion of the database images. A total of 18 color and texture features are used. Given a query image, it typically takes about 180 sec. to perform a retrieval on our database containing 592 images on a SGI Indigo 2 workstation. Query images are successfully retrieved.

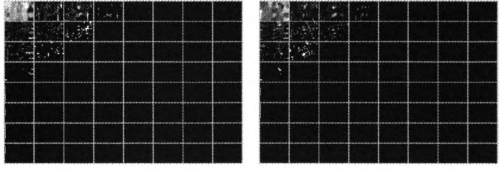
7.2.4 Discussion

We have proposed an algorithm for object localization using shape, color, and texture. Shape-based deformable template matching methods have potential in object retrieval because of their versatility and generalizability in handling different classes of objects and different instances of objects belonging to the same shape class. But,

(a)



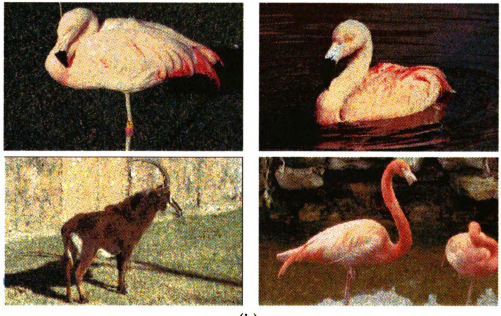
(b)



(c) (d)

Figure 7.17: Features extracted from the block DCT coefficients. **(a)** 250×384 input color image; **(b)** DCT features for the **Y** frame (intensity); **(c)** DCT features for the **Cr** frame (chrominance); **(d)** DCT features for the **Cb** frame (chrominance);

(a)



(b)

Figure 7.18: Retrieval based on color. **(a)** query example is given by the rectangular region; **(b)** top-4 retrieved images from the database which contain blocks of similar color.

one disadvantage in adopting them in content-based image retrieval systems is their large computational cost. We have proposed efficient methods to compute texture and color features to direct the initialization of the shape-based deformable template matching method. These texture and color features can be directly extracted from compressed images. This filtering stage allows the deformable template matching to be applied to a very small subset of database images, and only to a few specific posi-

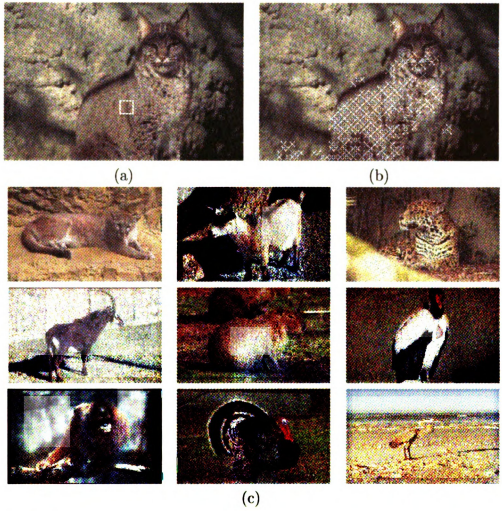


Figure 7.19: Retrieval based on texture. **(a)** query example is specified by the rectangular region; **(b)** matching macroblocks are marked with crosses in the query image; **(c)** other nine retrieved images from the database which contain regions of similar texture.

tions in the candidate images. Preliminary experimental results show computational gains using these supplemental features.

The proposed method assumes no preprocessing of the database. The input is the raw image data. We believe that our system can be used as an auxiliary tool to annotate, organize, and index the database using color, texture, and shape attributes off-line, where features (shape, color and texture) of retrieved items are computed

and stored as database indexes.

7.3 Summary

In this chapter we have applied the DTM method to image database retrieval applications and proposed two hierarchical retrieval algorithms.

Shape-based deformable template matching methods have potential in object retrieval because of their versatility and generalizability in handling different classes of objects and different instances of objects belonging to the same shape class. But, one disadvantage in adopting them in content-based image retrieval systems is their large computational cost. The hierarchical two-stage algorithm is an attempt to achieve both efficiency and accuracy, where simple and quickly computed features are used to browse the database for a small set of candidates which are further screened using the deformable template matching method.

The popularity and abundance of digital database retrieval systems have been increasing at a fast pace in recent years. Good representation and search methods are very important for multimedia systems. We are investigating whether we can use the deformable template matching method to obtain some automatic and flexible shape-based indexing scheme for image databases. Such an indexing system can be very useful for digital libraries.

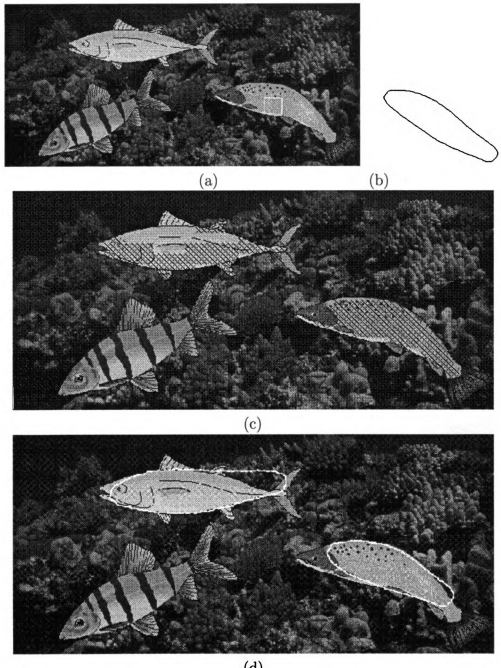


Figure 7.20: Retrieval based on color and shape. **(a)** query color example is specified by the rectangular region; **(b)** sketch for the shape; **(c)** matching macroblocks are marked with crosses in the query image; **(d)** retrieved shapes.

(a)

(b)

(c)

Figure 7.21: Retrieval based on color, texture, and shape. **(a)** query region example is given by the rectangular region; **(b)** sketch for the shape; **(c)** retrieved shape.

Chapter 8

Summary and Future Work

We have proposed and implemented a deformable template based algorithm for general object matching. The systematic paradigm consists of the following three components:

- a representative template for a class of objects;
- a deformation process on the prototype template in the form of a parameterized statistical transformation;
- an imaging process describing the fit of the deformed template to the image data.

We have primarily used a bitmap representation for characterizing the shape template. We have also implemented a spline representation which has more structure and fewer parameter than the bitmap representation [1]. We have proposed three different

[1] In the bitmap representation, the coordinates of each foreground pixel can be considered as parameters.

kinds of deformation transforms to deform the prototype template, each with its own advantages and deficiencies. Once the deformation space is determined, a probabilistic distribution is imposed on it to control the variations. The deformation model, which consists of the hand-drawn prototype template and the probabilistic transformation on it, is used as the prior in a Bayesian scheme to reflect the prior knowledge of the shape. The likelihood function is based on the edge map (both the position and the orientation) of the input image. The object is localized by maximizing the *a posteriori* probability. The matching scheme is general since it can be applied to objects with an arbitrary shape as specified by the prototype template.

We have applied the deformable template matching algorithm to successfully

- localize objects in images of complex background;
- retrieve images from image databases; and
- track objects in image sequences.

In the following sections, we propose some future research topics which are related to the work described in this dissertation.

8.1 Learning in Deformable Template Matching

The shape-based deformable template model consists of three components: (i) a prototype template of the representative contour/edges which describes the prior knowledge of an object shape; (ii) a deformation transformation on the template which determines the space of deformed templates; and (iii) a probabilistic distribution on

the template space to govern the variations. Due to the capability of both representing a characteristic shape and accommodating variations in the shape, deformable templates make a very flexible and versatile shape model. It has shown great potential in applications including medical image segmentation, feature extraction [75], object matching and tracking [76], and image database retrieval [138, 153].

In our scheme, the prototype is prespecified and the deformation transform and distribution of the parameters are predetermined. However, in some circumstances, it is desirable to learn each of these components from input data so that the model is more efficient and effective.

Learning the prototype template A challenge in deformable template matching methods is to learn, from training samples, the representative prototype(s). That is, given some instances from a common shape class, how to learn from these instances a template which best describes the shape class. A straight forward solution to this is to align the sample shapes and use the average shape as the prototype. A related and more complex problem is to learn a small set of representative prototypes for a complex shape class. The construction of this set of prototypes should consider a minimal representation to describe the shape class which may be multimodal. How to represent the shape class in terms of multiple prototypes and learn the most representative, dissimilar, and efficient prototypes remains an open problem.

Another related problem is to learn the shape template given the greyscale/color image of an object. To find the shape description using curves, we need to extract the perceptually salient contours from the object image.

Learning the deformation transformation The deformation transform along with the prototype template determines the space of deformable templates. Different choices of the transformation would result in different sets of deformable templates. It is desirable to select transforms which best span/approximate the deformation space of a shape class. If we have a set of training samples which provides a good coverage of the shape class(es), we may estimate the deformation space and select the transform adaptively. One possible solution is the eigenspace approach. We can compute the covariance matrix and obtain the set of eigenvectors which correspond to large variations. This basis would provide a succinct description of the deformations.

Learning the distribution of the deformations The distribution of the deformations is related to the variations in shape classes. Given a set of training samples and the deformation transforms, we can compute the statistics of the deformation parameters for the training samples and then use them to derive the distribution of the deformations.

8.2 Image Database Annotation and Indexing

We expect that the deformable template matching algorithm can be applied to annotate and index digital databases offline. To do this, we need to design a set of "media" templates which may roughly span the space of all possible query shapes. This set of templates is applied to each database image using the deformable template matching algorithm. The good matches and the corresponding scores and parameter values

are stored for indexing purpose. Given a query, its template is first compared to the database templates to obtain the matching scores. These scores combined with the prestored scores can give some information about a potential match of the query with the database images.

8.3 Incorporating Region Information

In addition to object shape, region information (greyscale, color, texture) gives important visual cues. Region-based features are more robust to misalignment than shape-based features. It is desirable to incorporate region information in the deformable template matching scheme. We have used color/texture to localize initial positions for the deformable template model (Sec. 7.2). However, once an initialization is obtained near regions of matching color/texture, the region cues do not participate in the matching process anymore. The matching result could be improved if we incorporate the region statistics in the objective function for the deformable template matching process, so that the template can be modified by the texture/color features. There are three ways the region features can be used:

- use the region statistics inside and outside of the closed contour. For example, if we assume the inside region follows a homogeneous model and the outside region follows another homogeneous model, we can penalize the deviations from the two models in the objective function;

- use only the region statistics inside the closed contour. The inside region corresponds to the object of interest. It is generally easier to establish a model for

the object of interest than the background;

- use the region statistics in a narrow band around the contour.

When we incorporate the region cues, they should be used in an efficient way so that the computational cost for the deformable template is still affordable.

8.4 Shape Matching in the Compressed Domain

We have been able to extract useful color and texture features in the compressed domain. We are currently investigating whether shape matching can also be performed in the compressed domain, which may be feasible now that the edge detectors are available for compressed data. An integrated and efficient content-based retrieval system for compressed digital library will offer a great potential with the rapid accumulation of image and video data, which are typically compressed for storage and transmission efficiency. We will also look into extracting more reliable texture features, which capture texture structure that go beyond the size of a DCT macroblock.

Bibliography

[1] *Collection of Trademarks and Logotypes in Japan.* Tokyo: Graphic-sha, 1973.

[2] J. Altmann and H.J.P. Reitbock. A fast correlation method for scale- and translation-invariant pattern recognition. *IEEE Trans. Pattern Anal. and Machine Intell.*, 6(1):46–57, January 1984.

[3] A. Amini, S. Tehrani, and T. Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. *Proc. 2nd Inter. Conf. on Computer Vision (ICCV)*, pages 95–99, 1988.

[4] Y. Amit, U. Grenander, and M. Piccioni. Structural image restoration through deformable template. *J. American Statistical Association*, 86(414):376–387, June 1991.

[5] H.C. Andrews and B.R. Hunt. *Digital Image Restoration.* Englewood Cliffs, NJ:Prentice-Hall, 1977.

[6] H. Tamura ans S. Mori and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Trans. Syst. Man Cybern.*, 6(4):460–473, April 1976.

[7] A. Averbuch, D. Lazar, and M. Israeli. Image compression using wavelet transform and multiresolution decomposition. *IEEE Trans. Image Processing*, 5(1):4–15, 1996.

[8] N. Ayache, I. Cohen, and I. Herlin. Medical image tracking. In A. Blake and A. Yuille, editors, *Active vision*, chapter 17, pages 285–301. Cambridge, Mass.: MIT Press, 1992.

[9] H.H. Bailey. *Image Correlation.* Santa Monica : Rand, 1976.

[10] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Comput. Vision Graphics Image Process.*, 46:1–21, 1989.

[11] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[12] A. Baraldi and F. Parmiggiani. An investigation of the textural characteristics associated with grey level coocurrence matrix statistical parameters. *IEEE Trans. Geoscience and remote sensing*, 33(2):293–304, 1995.

[13] S. T. Barnard. Stochastic stereo matching over scale. *Comput. Vision Graphics Image Process.*, 3:17–32, 1989.

[14] B. Bascle, P. Bouthemy, R. Deriche, and F. Meyer. Tracking complex primitives in an image sequence. *Proc. of the 12th Int. Conf. on Pattern Recognition, Jerusalem, Israel*, I:426–431, October 1994.

[15] B. Bascle and R. Deriche. Region tracking through image sequences. In *Proc. 5th Int. Conf. on Computer Vision (ICCV), Cambridge Massachusetts*, pages 302–307, June 1995.

[16] T. O. Binford, T. S. Levitt, and W. B. Wann. Bayesian inference in model-based machine vision. In Levitt, Kanal, and Lemmer, editors, *Uncertainty in AI.* North Holland, 1989.

[17] A. Blake, R. Curwen, and A. A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *Int. J. Comput. Vision*, 11(2):127–145, October 1993.

[18] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[19] C. Broit. *Optimal registration of deformed images.* PhD thesis, Univ. of Pennsylvania, 1981.

[20] D.J. Burr. A dynamic model for image registration. *Computer Graphics and Image Processing*, 15:102–112, 1981.

[21] D.J. Burr. Elastic matching of line drawings. *IEEE Trans. Pattern Anal. and Machine Intell.*, 3(6):708–713, November 1981.

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

[23] S.F. Chang and D.G. Messerschmitt. A new approach to decoding and compositing motion compensated DCT-based images. In *Proc IEEE Int. Conf. Acoust. Speech Signal Proc.*, pages 421–424, 1993, Minneapolis, MN.

[24] T. Chang and C.J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Trans. Image Processing*, 2(4):429–441, October 1994.

[25] B.B. Chaudhuri and N. Sakar. Texture segmentation using fractal dimension. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(1):72–77, 1995.

[26] R.T. Chin and C.R. Dyer. Model-based recognition in robot vision. *Comput. Surv.*, 18:67–108, 1986.

[27] P. B. Chou and C. M. Brown. The theory and practice of Bayesian image labeling. *Int. J. Comput. Vision*, 4(3):185–210, 1990.

[28] Y. S. Chow, U. Grenander, and D. M. Keenan. *HANDS. A Pattern-theoretic Study of Biological Shapes.* New York, Springer-Verlag, 1991.

[29] L. Cohen. Note on active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[30] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Trans. Pattern Anal. and Machine Intell.*, 15(11):1131–1147, November 1993.

[31] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, pages 297–301, April 1965.

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

[33] T. F. Cootes, C. J. Taylor, and A. Lanitis. Active shape models: Evaluation of a multi-resolution method for improving image search. *Proc. British Machine Vision Conference*, pages 327–336, 1994.

[34] I.J. Cox, S.B. Rao, and Y. Zhong. "Ratio regions", a technique for image segmentation. In *Proc. 13th Int. Conf. on Pattern Recognition (ICPR)*, pages 557–564, Vienna, Austria, 1996.

[35] Y. Cui, D. Swets, and J. Weng. Learning-based hand sign recognition using SHOSLIF-M. *Proc. 5th Int. Conf. on Computer Vision (ICCV)*, pages 631–636, 1995.

[36] Y. cui and J. Weng. Hand segmentation using learning-based prediction and verification for hand-sign recognition. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition (CVPR)*, pages 88–93, 1996.

[37] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. on Information Theory*, (5):961–1005, Sept. 1990.

[38] R. A. DeVore, B. Jawerth, and B. J. Lucier. Image compression through wavelet transform coding. *IEEE Trans. on Information Theory*, (2):719–746, March 1992.

[39] R. O. Duda and P. E. Hart. Use of the Hough transforms to detect lines and curves in pictures. *Comm. ACM*, 15(1):11–15, 1972.

[40] S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft identification by moment invariants. *IEEE Transactions on Computers*, 26(1):39–45, January 1977.

[41] C. Faloutsos, R. Barber, M. Flicker, J. Hafner, W. Niblack, and W. Equitz. Efficient and effective querying by image content. *J. Intell. Inform. Syst.*, 3:231–262, 1994.

[42] M. Figueiredo and J. Leitao. Bayesian estimation of ventricular contours in angiographic images. *IEEE Trans. Med. Imaging*, 11(3):416–429, September 1992.

[43] M. Figueiredo, J. Leitao, and A. K. Jain. Adaptive B-splines and boundary estimation. To appear in CVPR'97.

[44] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Computers*, 22(1):67–92, January 1973.

[45] H. Freeman. *Introduction to Statistical Inference.* Addison-Wesley Publishing Company, Inc., 1963.

[46] P. Gader, B. Forester, M. Ganzberger, A. Gillies, B. Mitchell, M. Whalen, and T. Yocum. Recognition of handwritten digits using template and model matching. *Pattern Recognition*, 24(5):421–432, 1991.

[47] D. L. Gall. MPEG: a video compression standard for multimedia applications. *Communications of the ACM*, 34(4):47–58, 1991.

[48] D. Geiger, A. Gupta, L. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(3):294–302, March 1993.

[49] D. Geiger and A. Yuille. A common framework for image segmentation. *Int. J. Comput. Vision*, 6(3):227–243, 1990.

[50] S. B. Gelfand and S. K. Mitter. Metropolis-type annealing algorithms for global optimization in $\mathcal{R}$. *SIAM J. Control and Optimization*, 31(1):111–131, January 1993.

[51] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. and Machine Intell.*, 6(6):721–742, November 1984.

[52] S. Geman and C. Hwang. Diffusions for global optimization. *SIAM J. Control and Optimization*, 24:1031–1043, September 1986.

[53] B. Gidas. Global optimization via the Langevin equation. *Proc. of the IEEE Conf. on Decision and Control, Fort Lauderdale*, pages 774–778, 1985.

[54] R.C. Gonzalez and P. Wintz. *Digital Image Processing.* Reading, MA: Addison-Wesley, 1977.

[55] M. M. Gorkani and R. W. Picard. Texture orientation for sorting photos "at a glance". *Proc. of the 12th Int. Conf. on Pattern Recognition, Jerusalem, Israel,* pages A459–A464, October 1994.

[56] A. Goshtasby. Description and discrimination of planar shapes using shape matrices. *IEEE Trans. Pattern Anal. and Machine Intell.*, 7(6):738–743, November 1985.

[57] A. Goshtasby. Template matching in rotated images. *IEEE Trans. Pattern Anal. and Machine Intell.*, 7(3):338–344, May 1985.

[58] U. Grenander. *Pattern Synthesis: Lectures in Pattern Theory,* volume 18 of *Applied Mathematical Sciences.* Springer-Verlag, 1976.

[59] U. Grenander, Y. Chow, and D.M. Keenan. *Hands: A pattern theoretic study of biological shapes,* volume 2 of *Research Notes in Neural Computing.* Springer-Verlag, 1991.

[60] U. Grenander and D. M. Keenan. Towards automated image understanding. In K. V. Mardia and G. K. Kanji, editors, *Advances in Applied Statistics: Statistics and Images (1),* chapter 6, pages 89–103. Carfax Publishing Company, 1993.

[61] U. Grenander and M. I. Miller. Representation of knowledge in complex systems. *J. of Royal Statistical Society (B),* 56(3):1–33, 1994.

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

[63] K. Hirata and F. Kato. Query by visual example. In *Advances in Database Tech. EDBT '92.* Vienna: Springer-Verlag, March 1991.

[64] B. Holt and L. Hartwick. Visual image retrieval for applications in art and art history. *Proc. of SPIE, Storage and Retrieval for Image and Video Databases,* 2185:70–81, February 1994.

[65] T. Y. Hou, A. Hsu, P. Liu, and M. Y. Chiu. A content-based indexing technique using relative geometry features. *Proc. of SPIE, Image Storage and Retrieval Systems,* 1662:29–68, 1992.

[66] P. V. C. Hough. *Method and Means for Recognizing Complex Patterns,* 1962. U.S. Patent 3069654.

[67] M.K. Hu. Visual pattern recognition by moment invariants. In J.K. Aggarwal, R.O. Duda, and A. Rosenfeld, editors, *Computer Methods in Image Analysis.* IEEE Computer Society, Los Angeles, CA, 1977.

[68] T. Igarashi. *World Trademarks and Logotypes.* Tokyo: Graphic-sha, 1983.

[69] T. Igarashi. *World Trademarks and Logotypes II: A Collection of International Symbols and their Applications.* Tokyo: Graphic-sha, 1987.

[70] J. Illingworth and J. Kittler. A survey of Hough transform. *Comput. Vision Graphics Image Process.,* 44:87–116, 1988.

[71] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. Computer Vision'96*, pages I:343–356, 1996.

[72] A. K. Jain and C. Dorai. Challenges in computer vision: integration, evaluation and applications. To appear in Pattern Recognition, 1997.

[73] A. K. Jain and F. Farrokhnia. Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[74] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.

[75] A. K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. To appear in IEEE Trans. PAMI, 1997.

[76] A.K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *IEEE Trans. Pattern Anal. and Machine Intell.*, 18(3):267–278, March 1996.

[77] R. C. Jain and T. O. Binford. Dialogue: Ignorance, Myopia, and Naivete in Computer Vision. *Comput. Vision Graphics Image Process.*, 53:112–117, January 1991.

[78] M.E. Jernigan and F. D'Astous. Entropy-based texture analysis in the spatial frequency domain. *IEEE Trans. Pattern Anal. and Machine Intell.*, 6(2), March 1984.

[79] M.P. Dubuisson Jolly, S. Lakshmanan, and A. K. Jain. Vehicle segmentation using deformable templates. *IEEE Trans. Pattern Anal. and Machine Intell.*, 18(3):293–308, March 1996.

[80] K. Karu, A. K. Jain, and R. M. Bolle. Is there any texture in the image? *Pattern Recognition*, 29(9):1437–1446, 1996.

[81] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vision*, 1(4):321–331, 1988.

[82] C. Kervrann and F. Heitz. A hierarchical statistical framework for the segmentation of deformable objects in image sequences. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 724–728, Seattle, WA, June 1994.

[83] C. Kervrann and F. Heitz. Robust tracking of stochastic deformable models in long image sequences. In *Proc. Inter. Conf. on Image Processing (ICIP)*, volume 3, pages 88–92, Austin, TX, 1994.

[84] A. Khotanzad and R.L. Kashyap. Feature selection for texture recognition based on image synthesis. *IEEE Trans. Syst. Man Cybern.*, 17(6):1087–1095, November 1987.

[85] K.F. Lai and R.T. Chin. Deformable contours: modeling and extraction. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(11):1084–1090, November 1995.

[86] S. Lakshmanan and H. Grimmer. Detecting straight edges in radar images using deformable templates. *IEEE Trans. Pattern Anal. and Machine Intell.*, 1995. to appear.

[87] S. Lakshmanan, A.K. Jain, and Y. Zhong. Detecting straight edges in millimeter wave images. *Proc. IEEE International Conference on Image Processing*, pages 258–261, Washington, DC, October 1995.

[88] V. F. Leavers. Survey: Which Hough transform? *CVGIP: Image Understanding*, 58(2):250–264, September 1993.

[89] F. Leymarie and M. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Anal. and Machine Intell.*, 15(6):617–634, June 1993.

[90] Jae S. Lim. *Two-Dimensional Signal and Image Processing.* Englewood-Cliffs, NJ: Prentice-Hall, 1990.

[91] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(2):158–175, February 1995.

[92] S. Mallat. Wavelet for vision. *Proceedings IEEE*, 84(4):604–614, April 1996.

[93] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. and Machine Intell.*, 11(7):674–693, July 1989.

[94] J. Mao and A.K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, 1992.

[95] K.V. Mardia. Recent advances in shape statistics and image analysis. *Proc. 2nd Asian Conf. on Computer Vision*, pages 174–178, Singapore, Dec. 1995.

[96] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Reller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[97] M. I. Miller, G. E. Christensen, Y. Amit, and U. Grenander. Mathematical textbook of deformable neuroanatomies. *Proc. Natl. Acad. Sci. USA*, 90:11944–11948, December 1993.

[98] M. I. Miller, A. Srivastava, and U. Grenander. Conditional-Mean estimation via Jump-Diffusion process in multiple target tracking/recognition. *IEEE Trans. Signal Processing*, 43(11):2678–2690, 1995.

[99] K. Mori, M. Kidodi, and H. Asada. An iterative predictor correction method for automatic stereo comparison. *Comput. Vision Graphics Image Process.*, 2:393–401, 1973.

[100] M. Moshfeghi, S. Ranganath, and K. Nawyn. Three-Dimensional elastic matching of volumes. *IEEE Trans. Image Processing*, 3(2):128–138, March 1994.

[101] S. Negahdaripour and A. K. Jain. Challenges in computer vision research: future directions of research. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 189–199, Urbana, IL 1992.

[102] W. Niblack, R. Barber, and W. Equitz. The QBIC project: Querying images by content using color, texture, and shape. *Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases*, 1908:173–187, 1993.

[103] K.L. Oehler and R.M Gray. Combining image compression and classification using vector quantization. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(5):461–473, May 1995.

[104] D. C. W. Pao, H. F. Li, and R. Jayakumar. Shape recognition using the straight line Hough transform: Theory and generalization. *IEEE Trans. Pattern Anal. and Machine Intell.*, 14(11):1076–1089, November 1992.

[105] A. Pentland. Automatic extraction of deformable part models. *Int. J. Comput. Vision*, 13(2):107–126, 1990.

[106] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: tools for content-based manipulation of image databases. *Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases II, 2185-05*, February 1994.

[107] A. Pentland and S. Sclaroff. Close-form solutions for physically based modeling and recognition. *IEEE Trans. Pattern Anal. and Machine Intell.*, 13(7):703–714, 1991.

[108] D.B. Phillips and A.F.M. Smith. Dynamic image analysis using Bayesian shape and texture models. In K. V. Mardia, editor, *Advances in Applied Statistics: Statistics and Images (1)*, pages 299–322. Carfax Publishing Company, 1994.

[109] R.W. Picard. The society of models for video and image libraries. Technical Report 360, MIT Media Lab, Perceptual Computing Section, 1995.

[110] R.W. Picard and F. Liu. A new world ordering for images. In *Proc. IEEE Conf. Acoustic, Speech, and Signal Processing*, April Adelaide Australia, 1994.

[111] W.K. Pratt. *Digital Image Processing.* New York: Wiley-Interscience, 1978.

[112] R. Ronfard. Region-based strategies for active contour models. *Int. J. Comput. Vision*, pages 229–251, 1993.

[113] A. Rosenfeld and A. C. Kak. *Digital Picture Processing.* Academic Press, 1982.

[114] A. Rosenfeld and G.J. Vanderburg. Coarse-fine template matching. *IEEE Trans. Syst. Man Cybern.*, 7:104–107, February 1977.

[115] T. W. Ryan, L. D. Sanders, H. D. Fisher, and A. E. Iverson. Image compression by texture modeling in the wavelet domain. *IEEE Trans. Image Processing*, 5(1):26–36, 1996.

[116] P. Saint-Marc, H. Rom, and G. Medioni. B-spline contour representation and symmetry detection. *IEEE Trans. Pattern Anal. and Machine Intell.*, 15(11):1191–1197, 1993.

[117] S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(6):545–561, June 1995.

[118] B. Shen and I.K. Sethi. Direct feature extraction from compressed images. In *Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases IV*, volume 2670, 1995.

[119] T. Simchony, R. Chellappa, and Z. Lichtenstein. Pyramid implementation of optimal-step conjugate-search algorithms for some low-level vision problems. *IEEE Trans. Syst. Man Cybern.*, 19(6):1408–1425, Nov./Dec. 1989.

[120] R. Sriram, J.M. Francos, and W.A. Pearlman. Texture coding using a Wold decomposition model. In *Proceedings of the 12th Int'l Conf. on Pattern Recognition*, volume III, pages 35–39, October 1994, Jerusalem, Israel.

[121] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. Pattern Anal. and Machine Intell.*, 14(11):1061–1075, November 1992.

[122] G. C. Stockman. Object recognition. In R. C. Jain and A. K. Jain, editors, *Analysis and Interpretation of Range Images*, pages 225–253. Springer-Verlag, 1990.

[123] G. C. Stockman and A. K. Agrawala. Equivalence of Hough curve detection to template matching. *Communications of the ACM*, pages 820–822, 1977.

[124] G. Storvik. A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Trans. Pattern Anal. and Machine Intell.*, 16(10):976–986, October 1994.

[125] M.J. Swain and D.H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, 1991.

[126] D. Swets and J. Weng. The self-organizing hierarchical optimal subspace learning and inference framework for object recognition. In *Proc. Int'l Conf. Neural Networks and Signal Processing*, Nanjing, China, December 1995.

[127] R. Szeliski and J. Coughlan. Hierarchical spline-based registration. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 194–201, Seattle, WA, June 1994.

[128] D. Terzopolous and D. Metaxas. Dynamic 3-d models with local and global deformations: Deformable superquadrics. *IEEE Trans. Pattern Anal. and Machine Intell.*, 13(7):703–714, July 1991.

[129] D. Terzopolous, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Comput. Graphic*, 21(4):205–214, 1987.

[130] D. Terzopolous and R. Szeliski. Tracking with Kalman snakes. In A. Blake and A. Yuille, editors, *Active vision*, chapter 1, pages 1–19. Cambridge, Mass.:MIT Press, 1992.

[131] D. Terzopolous, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, (36):91–123, 1988.

[132] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Trans. Pattern Anal. and Machine Intell.*, 15(6):569–579, June 1993.

[133] M. Tuceryan and A.K. Jain. Texture analysis. In C.H. Chen, L.F. Pau, and P.S.P. Wang, editors, *The Handbook of Pattern Recognition and Computer Vision*, pages 235–276. World Scientific Publishing Co., 1993.

[134] G. Turin. An introduction to matched filtering. *IEEE Trans. Information Theory*, pages 311–329, 1960.

[135] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Processing*, 4(11):1549–1570, 1995.

[136] M. Unser, A. Aldroubi, and M. Eden. On the asymptotic convergence of B-spline wavelets to Gabor functions. *IEEE Trans. Information Theory*, 38(2):864–872, March 1992.

[137] M. Unser, A. Aldroubi, and M. Eden. A family of polynomial spline wavelet transforms. *Signal Processing*, pages 141–162, 1993.

[138] A. Vailaya, Y. Zhong, and A. K. Jain. A hierarchical system for efficient image retrieval. In *Proc. 13th Int. Conf. on Pattern Recognition (ICPR)*, pages 356–360, Vienna, Austria, 1996.

[139] B.C. Vemuri and A. Radisavljevic. From global to local, a continuum of shape models with fractal priors. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 307–313–627, New York City, NY, June 1993.

[140] V.V. Vinod and H. Murase. Object location using complementary color features: histogram and DCT. In *Proc. 13th Int. Conf. on Patter Recognition (ICPR)*, pages 554–559, Vienna, Austria, 1996.

[141] G.K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):31–44, 1991.

[142] J. Weng. On comprehensive visual learning. In *Proc. NSF/ARPA Workshop on Performance vs. Methodology in Computer Vision*, pages 152–166, Seattle, WA, June 1994.

[143] J. Weng. SHOSLIF: A framework for sensor-based learning for high-dimensional complex systems. In *Proc. IEEE Workshop on Architectures for Semiotic Modeling and situation analysis in Large Complex Systems*, Monterey, CA, August 1995.

[144] J. Weng. Cresceptron and SHOSLIF: Toward comprehensive visual learning. In S.K. Nayar and T. Poggio, editors, *Early Visual Learning*. Oxford University Press, New York, 1996.

[145] J. Weng. The living machine initiative. Technical Report TP96-60, Computer Science Dept., Michigan State Univ., December 1996.

[146] J. Weng and S. Chen. Incremental learning for visual-based navigation. In *Proc. Int'l Conf. on Pattern Recognition*, Vienna, Austria, August 1996.

[147] G. Whitten. Scale space tracking and deformable sheet models for computational vision. *IEEE Trans. Pattern Anal. and Machine Intell.*, 15(7):697–706, July 1993.

[148] B. Widrow. The rubber mask technique, Parts I and II. *Pattern Recognition*, 5:175–211, 1973.

[149] A. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. *Int. J. Comput. Vision*, pages 133–144, 1987.

[150] G. Xu, E. Segawa, and S. Tsuji. Robust active contours with insensitive parameters. *Pattern Recognition*, 27(7):879–884, 1994.

[151] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *Int. J. Comput. Vision*, 8(2):133–144, 1992.

[152] H. J. Zhang, C. Y. Low, and S. W. Smoliar. Video parsing and browsing using compressed data. *Multimedia Tools and Applications*, pages 89–111, 1995.

[153] Y. Zhong and A. K. Jain. Object localization using color, texture and shape. Technical Report TP96-36, Computer Science Dept., Michigan State Univ., 1996.

[154] B. Zhu, A.H. Tewfik, M.A. Colestock, O. N. Gerek, and A.E. Cetin. Image coding with wavelet representations, edge information and visual masking. In *Proc.*

Inter. Conf. on Image Processing (ICIP), volume 1, pages 582–585, Washington DC, 1995.

[155] S. C. Zhu, T. S. Lee, and A. L. Yuille. Region competition: unifying snakes, region growing, energy/Bayes/MDL for Multi-band image segmentation. In *Proc. 5th Inter. Conf. on Computer Vision (ICCV)*, pages 416–423, Boston, 1995.