This is to certify that the

dissertation entitled

LEARNING-BASED DETECTION, SEGMENTATION AND MATCHING OF OBJECTS

presented by

Nicolae Duta

has been accepted towards fulfillment
of the requirements for

Doctoral degree in Computer Science & Engineering

Anil Kumar

Major professor

Date Aug 14, 2000

Learning-Based Detection, Segmentation and Matching of Objects

By

Nicolae Duta

Dissertation

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

DOCTOR OF PHILOSOPHY

Computer Science & Engineering

2000

Abstract

Learning-Based Detection, Segmentation and Matching of Objects

By

Nicolae Duta

Object learning is an important problem in machine vision with direct implications on the ability of a computer to understand an image. Usually, an object is defined by its appearance (the pattern of gray/color values in the object of interest and its immediate neighborhood), shape, and sometimes, by its relationships to other objects in the scene. This dissertation presents appearance-based as well as shape-based methods for object learning and retrieval. The object appearance is modeled as a Markov chain that maximizes the discrimination (Kullback distance) between positive and negative examples in a training set. The learned appearance model can be used for object detection: given an arbitrary black and white image, decide if the object is present in the image and find its location(s) and size(s). Two applications will be discussed in detail: human face detection in Black and white images and heart ventricle localization in MR images. We have also developed a fully automated shape learning method which is based on clustering a set of training shapes in the

original shape space defined by the coordinates of the contour points and performing a Procrustes analysis on each cluster to obtain cluster prototypes (average objects) and statistical information about intra-cluster shape variation. The main difference from previously reported methods is that the training set is first automatically clustered and those shapes considered to be outliers are discarded. In this way, the cluster prototypes are not distorted by outlier shapes. The second difference is in the manner in which registered sets of points are extracted from each shape contour. We have proposed a flexible point matching technique that takes into account both pose/scale differences as well as non-linear shape differences between a pair of objects. The matching method is independent of the initial relative position/scale of the two objects and does not require any manually tuned parameters. Our shape learning method has been used to develop a state-of-the-art hand shape-based personal identity verification system, a shape warping-based system for segmenting the Corpus Callosum in MR images of the brain, as well as an automatic system for predicting dyslexia based on the shape of the Corpus Callosum.

To All My Teachers

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Anil Jain, for his guidance, support and encouragement during my Ph.D. years. His suggestions, questions and careful editing has contributed greatly to the quality of this dissertation. I am also grateful to the other members of my committee Dr. Marie-Pierre Jolly, Dr. Sridhar Mahadevan, Dr. R.V. Ramamoorthi and Dr. John Weng for their helpful feedback and suggestions. Many thanks to the Imaging Department at Siemens Corporate Research: Dr. Marie-Pierre Jolly, Dr. Alok Gupta, Dr. Gareth Funka-Lea and many others, for their support and assistance. They have provided data, financial support, internships and very useful advice.

I am also indebted to my coauthors and all those who kindly offered me their data and/or code: Dr. Dorin Comaniciu, Dr. Mario Figueiredo, Dr. Arvid Lundervold, Dr. Kanti Mardia, Dr. Kerstin von Plessen, Dr. Milan Sonka and Dr. Torfinn Taxt. Without their help, this research work would have not been possible.

I would also like to thank my officemates over the years, with whom I had valuable discussions and who provided helpful feedback on my research: Paul Albee, Vera Bakic, Scott Connell, Dan Gutchess, Friederike Griess, Lin Hong, Vincent Hsu, Wey

Hwang, So Hee Kim, Yatin Kulkarni, Yonghong Li, Karissa Miller, Silviu Minut, Salil Prabhakar, Arun Ross and Aditya Vailaya. And last, but not least, I am grateful to the Romanian community at Michigan State for making me feel like home.

TABLE OF CONTENTS

LIST OF TABLES x

LIST OF FIGURES xi

1 Introduction 1
1.1 An object learning approach to facilitate image understanding 3
1.2 Applications 6
1.3 Thesis outline 8

2 Literature Review 10
2.1 Object detection 10
2.1.1 Why is it difficult to detect objects? 19
2.1.2 Object representation 22
2.1.3 Classifier comparison on the face detection problem 25
2.1.4 Comparing the performances of some available face detectors 41
2.1.5 Discussion and conclusions 42
2.2 Object segmentation using deformable models 45
2.3 Point Distribution Models 50
2.4 Model design and training 55
2.5 Object matching 58
2.6 Summary 60

3 Object detection 61
3.1 Basic Methods 61
3.2 Maximum discrimination-based object detection 64
3.2.1 Introduction 65
3.2.2 Mathematical model 67
3.2.3 Most discriminant Markov chain 70
3.2.4 Classification procedure 74
3.2.5 Experimental results 75
3.3 Summary 79

4 Learning 2D shape models 81
4.1 Problem specification 82
4.2 Shape learning method 85
4.3 Shape registration 91
4.4 Shape clustering and prototype computation 97

4.5 Experimental Results 100
4.6 Discussion 104
4.7 Summary 109

5 Deformable model segmentation **110**
5.1 Top-down Active Shape segmentation 110
5.2 Warping-based segmentation 115
5.3 Summary 117

6 Object matching **119**
6.1 Deformable matching of hand shapes for user verification 120
6.1.1 Proposed Method 122
6.1.2 Experimental Results 124
6.1.3 Improving verification accuracy 127
6.2 Corpus Callosum shape analysis: a comparative study of group differences associated with dyslexia, gender and handedness 130
6.2.1 Methods 133
6.2.2 Results 136
6.3 Summary 145

7 Conclusions and Future Work **146**

BIBLIOGRAPHY **151**

List of Tables

2.1 A summary of the types of features used by different applications of object detection. 12

2.2 A survey of the learning-based object detection systems. 16

2.3 A comparison of different classifiers for the face detection problem. Legend: [*] = the classification rate does not depend on the size/distribution of the training set or the test sample, [**] = the classification rate depends on the size of the training set but not on the distribution of the training/testing samples, [* * *] = the classification rate depends on the size/distribution of the training set but not on the test example, [* * **] = the classification rate depends on the size/distribution of the training set *and* the test pattern, † The FA rate for the Maximum discrimination classifier is very small since false alarms from previous testing trials were added to the training set (as such, it should not be directly compared to the FA rates of other classifiers). The testing results show that the training set is crucial and should be constructed incrementally for each type of classifier. 33

3.1 Performance of the left ventricle detection algorithm. 79

6.1 A brief description of the five datasets used in this study, the variation factors that have been considered and a comparison between the original results and our shape analysis results. The table entries are shown in increasing order of the average age of the subjects in the datasets. *Note:* Dys = Dyslexics, Nor = Normals, Mal = Males, Fem = Females, RH = right handed, nRH = non right handed, * = only the published part of the dataset has been used. 134

List of Figures

1.1 A 256 x 256 frame from a video sequence of moving cars. a) Original image. b) Mean-shift segmentation into 4 gray levels [23]. c) The region of (b) containing the foreground car (shown in white). 2

1.2 Midsagittal MRI section of the brain. a) Original image showing the *corpus callosum* (CC) as a C-shaped central structure. b) Low-level segmentation of white matter that contains the CC (taken from [86]). c) High-level segmentation of the CC. 3

1.3 Schematic diagram of the *learning-detection-segmentation-matching* paradigm for the object learning and retrieval system developed in this thesis. The actions performed by the system are shown inside ellipsoidal boxes while the data involved in the process are shown inside rectangular boxes. The dashed arrows implying the use of appearance models in object segmentation and matching are shown only for completeness; our current system uses appearance models only for object detection. 4

1.4 Practical applications considered in this thesis. a) Detection and segmentation of the left ventricle in MR cardiac images. b) Segmentation of neuroanatomic structures in coronal-viewed MR brain images. c) Detection and segmentation of vehicles in highway video sequences. d) Segmentation of Corpus Callosum in midsagittal MR brain images. e) Hand shape-based personal identity verification. 7

2.1 What features define an object? (a) A human face is mostly defined by appearance. (b) The left ventricle in MR cardiac images is defined by a combination of appearance, shape and position relative to the lung (dark region). (a) A car is mostly defined by shape. (d) A building in aerial images is mostly defined by its shape. 11

2.2 The structure of a learning-based face detector. 17

2.3 Face-like patterns extracted from outdoor images that are typically used to generate negative examples. We believe that such patterns make the training set inconsistent, since they may very well be classified as faces if found in the right visual context. 19

2.4 The relationship between the classifier accuracy and detection performance. 21

2.5 A face pattern (left-most image) is gradually deformed into random noise (right-most image) by linear interpolation. It is very difficult to label the intermediate images as positive/negative examples for training a face detector. 24

2.6 Face detection results produced by a "relaxed" version of the maximum discrimination classifier followed by an SVM classification on the Olivetti database. 362 out of the 400 faces (90.5%) were successfully detected. 35
2.7 Face detection results produced by the maximum discrimination classifier on a group image. No arbitration has been performed; all patterns classified as faces are shown. 57 out of the 89 faces (64%) were successfully detected. About 35 background windows were misclassified as faces inducing a false accept rate of 1/49,000. 36
2.8 Face detection results produced by a "relaxed" version of the maximum discrimination classifier followed by an SVM classification on a group image. Multiple detections at nearby positions have been combined into one face rectangle. 63 out of the 89 faces (70.8%) were successfully detected. There are 34 false accepts. 37
2.9 Output of the Rowley *et al.* [111] (produced by the system demo posted at *http://www.ius.cs.cmu.edu/IUS/usrp0/har/FaceDemo/gallery.html*) (a), and Lew and Huijmans [83] (produced using the code posted at *http://www.wi.leidenuniv.nl/~mlew/face.detection.html#DMO*) (b) face detectors on a collated test image. 39
2.10 Output of the Colmenarez and Huang [22] (produced by the system demo posted at *http://troi.ifp.uiuc.edu/~antonio/section2.html*) (a), and Amit *et al.* [3] (b) face detectors on a collated test image. 40
2.11 Output of a "relaxed" version of the maximum discrimination classifier (implemented by Duta *et al.* [41]) followed by an SVM classification on a collated test image. 41
2.12 CC segmentation using *snakes*. a) Gradient magnitude of a region around CC. b) Manual snake initialization using an average CC. c) Same initialization as (b), but shifted 3 pixels upward. d)-f) Snake segmentation starting from (b). g)-i) Snake segmentation starting from (c). Segmentations in (d) and (g) used traditional snakes [76], (e) and (h) used *balloons* [21], (f) and (i) used GVF snakes [133]. Results shown in images (d)-(i) were produced using the software posted at *http://iacl.ece.jhu.edu/projects/gvf*. 47
2.13 CC segmentation using *B-spline* models [47]. a) Manual spline initialization using an average CC; b) Adaptive spline segmentation starting from (a); c) Same initialization as (a) shifted 3 pixels upward; d) Adaptive spline segmentation starting from (c); (Courtesy of Prof. Mario Figueiredo) 48
2.14 Illustration of Point Distribution Model. a) A 256 x 256 frame from a video sequence of moving cars with 55 landmark points superimposed. b) A 55 point car model. 50
2.15 A set of nine training images for the foreground car (moving left-to-right) with the car outline manually overlaid. 51
2.16 The nine aligned car contours of Fig. 2.15 52

2.17 Artificial car examples generated by simultaneously varying the model parameters corresponding to the first two largest eigenvalues on a bidimensional grid. The average car is shown in the center. The main two modes of variation determine the shape of the front and rear part of the vehicle as the global shapes in the aligned training set (Fig. 2.16) are relatively similar. Note that the original objects (Fig. 2.15) look more different because of scale and gray-level appearance. 54

3.1 Midsagittal MRI section of the brain. a) Original image. b) A *region of interest* with respect to the skull (white curve) containing the CC. c) Low-level segmentation of the white matter in the ROI in (b). 62

3.2 Example of automated car detection. a) Original image. b) Optimal left-to-right car model position. c) Optimal right-to-left car model position. d) Second best right-to-left car model position. 63

3.3 Several examples of 256×256 gradient echo cardiac MR images (short axis view) showing the left ventricle variations as a function of acquisition time, slice position, patient and imaging device. The left ventricle is the bright area inside the square. The four markers show the ventricle walls (two concentric circles). 66

3.4 The feature set defining a heart ventricle. a) The four cross sections through the ventricle and its immediate surroundings used to extract the features. b) The 100-element normalized feature vector associated with the ventricle in (a). 67

3.5 Training examples for the left ventricle detection problem. a) Positive examples. b) Negative examples. 75

3.6 The distribution of the log-likelihood ratio for left ventricle (red) and non left ventricle (blue) examples computed for the two cross-validation trials. For a decision threshold set at zero, 2,460 (2.4%) of the 101,250 positive examples and 6,854 (3.8%) of the 179,603 negative examples are misclassified. 76

3.7 Feature reordering induced by the most discriminant Markov chain (a) Positive examples. (b) Negative examples. 77

4.1 Learning the shape of the right ventricle from MR brain images. a) Manual tracing of the right ventricle performed by a neuroanatomist on three different patients. b) A ventricle model consists of a shape prototype (drawn in black) along with statistical information about shape variation. The prototype vertices (drawn as colored circles) have been obtained by averaging the coordinates of the corresponding vertices on the three ventricles (drawn as colored diamonds) *after they have been aligned in a common coordinate frame* (e.g., vertex s is the average of s_1, s_2 and s_3). The three aligned ventricle shapes are shown in dotted red, green and blue lines. This method for obtaining a shape prototype is called *Procrustes analysis* [11, 54, 34]. Note that, in order for this method to work one needs to extract sets of corresponding points of equal cardinality (in this case 16) from the three ventricle shapes. The shape variance is given by the 32×32 covariance matrix of the (x, y) coordinates of the vertices on the three ventricle shapes after alignment. 83

4.2 Expert-defined pseudo-landmarks (yellow squares) on the three shapes in Fig. 4.1 along with some obvious point correspondences (red arrows). Note that if the pseudo-landmarks are defined on each shape independently (as it was the case here) then, in most cases, it is very difficult to find corresponding points on other shapes. 84

4.3 Magnetic resonance image of the human brain, imaged in the coronal plane with in-slice resolution of 256×256 pixels. a) Original image (cropped to show only the brain region). b) Structures of interest whose contours were identified by a neuroanatomist. 86

4.4 The shape learning method (Algorithm 4.1). 89

4.5 Flexible registration of a shape approximation (b) to an original shape (a), with point correspondences drawn in green. c) Global similarity registration - Algorithm 4.2. d) Monotonic registration obtained from (c) after point reordering and inversion elimination (the point labeled O causes an inversion): Steps 1-3 of Algorithm 4.3. e) Topological neighborhood corresponding to point O: Step 4a of Algorithm 4.3. f) Similarity registration of the two topological neighborhoods in e). g) Final flexible registration. 95

4.6 Two training sets of 28 right ventricular (rows 1 and 2) and 28 globular shapes (rows 3 and 4) and a set of 11 cistern shapes (row 5) from different patients were automatically divided into clusters (main cluster (C1) drawn using multicolor dots and secondary clusters drawn in red, green and magenta). The registration of the *best fit shape* (#1047 for ventricles, #8917 for globus pallidus and #1179 for cisterns) to clusters C1 is overlaid as sets of colored points; corresponding points on different shapes are drawn using the same color. For example, corresponding bottom points on each right ventricle are drawn in red. 98

4.7 Procrustes averages (prototypes) of the shapes in the main clusters for 11 brain structures with the aligned shape examples overlaid. The clouds of consecutive points are drawn in different colors to show the accuracy of the registration. 101

4.8 Prototype of the 25 right-ventricle shapes in the main cluster (a) and prototype of the 28 right-globus pallidus shapes (b) with the aligned shape examples overlaid. The *ground truth* position for several points are shown using black circles. For each such point, we also show the *rms error* of the manual and automatic registrations. 103

5.1 Example of automatic car segmentation. a) Detected car using similarity transforms. b) Refined segmentation using piecewise similarity transforms. 112

5.2 Automatic segmentation of 11 brain structures in four different MR images using the learned models. 114

5.3 Registration of CC in the ROI image in Fig. 3.1(b). (a) The strongest 600 edges from the ROI in Fig. 3.1(b). (b) Registration of an average CC to the edge image in (a) (point correspondences are shown in green). 115

5.4 CC registration to the ROI image in Fig. 3.1(c). (a) Edges from the low level segmentation in Fig. 3.1(c). (b) Registration of an average CC to the edge image in (a) (point correspondences are shown in green). 116

5.5 High level segmentation of CC. (a) Registration of an average CC to the edge image in Fig. 5.4(a). (b) Warping the average CC onto the edge image in (a). 116

5.6 Automatic segmentation of the Corpus Callosum in nine MR images using the warping-based paradigm. 118

6.1 Hand shape acquisition system (a) and the image it captures (b) (Courtesy of Dr. Sharath Pankanti and Arun Ross). 121

6.2 Four pairs of input hand shapes: contours that form pair (a) belong to the same hand, contours that form pair (b) belong to the same hand (different from (a)), pairs (c) and (d) are formed by different hands. Based on the input contours, it is very difficult to tell which of the four pairs belongs to the same hand. None of the existing systems aligns the hand contours before extracting features, leading to inferior matching accuracy. 121

6.3 Alignment-based hand verification system. 123

6.4 Two pairs of hands before ((a),(c)) and after ((b),(d)) alignment. The hand pairs in (a) and (b) belong to the same hand while the pairs in (c) and (d) belong to different hands. These two pairs correspond to the overlapping region between the genuine and the imposter distributions in Fig. 6.5(a). 125

6.5 Hand shape-based verification performance. (a) Mean alignment error distributions for the genuine class (red) and imposter class (blue). The distributions are derived based on a total of 353 hand images of 53 persons. (b) ROC curve for the hand shape-based verification system. The annotations on the curve represent different thresholds on the MAE distance. 126

6.6 ROC curves generated by taking into account three, four and five fingers in the alignment error computation. 127

6.7 Learning a personal hand template. A hand template per person can be selected from the enrollment set based on its average alignment distance to the remaining enrollment shapes. In this case, the enrollment shape closest to the remaining ones is S_2 with an average alignment distance of 2.02. 129

6.8 (a) Midsagittal, T1-weighted MRI section of the brain (12 year old boy) showing the *Corpus Callosum* as a C-shaped structure. (b) Details showing a partition of the *Corpus Callosum* (Courtesy of Prof. Arvid Lundervold). 131

6.9 Corpus Callosum shapes belonging to normal subjects (a) and to dyslexic subjects (b). By simple visual inspection, there is no apparent difference between the two groups of shapes. 135

6.10 Comparing the normal (shown in red) and dyslexic (shown in blue) group means of Corpus Callosum shapes in our dataset. (a) *Rostrum* alignment. (b) Both *rostrum* and *splenium* alignment. The dyslexic prototype is actually cut into two parts which are aligned separately. The *posterior midbody* in the dyslexic subjects is significantly shorter than in normal subjects. 137

6.11 Two templates representing the *anterior half* and *posterior third* of the Corpus Callosum are independently aligned with a given CC instance and the distance between the aligned templates (the length of the uniting line segment) is measured. Alignment to a dyslexic CC instance (left) and to a normal instance (right). 138

6.12 Class distributions of the inter-template distances for the Bergen dataset. 138

6.13 Comparing the normal (shown in red) and dyslexic (shown in blue) group means of Corpus Callosum shapes in the Robichon and Habib [108] dataset. (a) *Rostrum* alignment. (b) Both *rostrum* and *splenium* alignment. The dyslexic prototype is actually cut into two parts which are aligned separately. The dyslexic CC prototype is more curved and significantly shorter than the normal prototype. 139

6.14 Class distributions of the inter-template distances for the Robichon and Habib [108] dataset. 139

6.15 Comparing the Right Handed (RH) (shown in magenta) and non Right Handed (nRH) (shown in green) group means for dyslexic (a) and normal (b) subjects in the Robichon and Habib [108] dataset. The dyslexic nRH prototype is a little shorter than its corresponding RH prototype. The normal nRH prototype is longer and thicker in the isthmus region than its corresponding nRH prototype. 141
6.16 (a) Comparing the male (shown in red) and female (shown in blue) group means for subjects in the Byne *et al.* [18] dataset. There are no significant differences between the two prototypes. (b) Comparing the Right Handed (RH) (shown in blue) and non Right Handed (nRH) (shown in red) group means for the male subjects in the Witelson [131] dataset. The nRH prototype is significantly longer and thicker in the isthmus region than the RH prototype. 141
6.17 Comparing the male (shown in red) and female (shown in blue) group means for dyslexic (a) and normal (b) subjects in the Bergen dataset. The female CC prototype in both dyslexic and normal subjects is shorter (though not significantly) than its corresponding male prototype. Except for the length, the male and female prototypes are practically identical. 142
6.18 Comparing the male (shown in red) and female (shown in blue) group means for subjects in the Davatzikos *et al.* [32] dataset. The female CC prototype is shorter and has a more bulbous posterior part than the male prototype. Overlay of the female (b) and male (c) group means obtained by us (shown in blue) and by Davatzikos *et al.* [32], (shown in red). The prototypes in each pair are almost identical (up to a similarity transformation) despite the fact that they have been obtained using different methods. 144

7.1 Detailed localization of the left ventricle. a) Multiple detections produced by the maximum discrimination method (Section 3.2). b) Horizontal feature profiles corresponding to each detected box in (a). An average horizontal profile is shown in red. c) The profiles in (b) aligned to the average profile using dynamic time warping. d) The feature profiles corresponding to the detected boxes in (a) *after warping to their corresponding average profiles.* A voting procedure is applied to estimate the center and the medial axis of the ventricle wall (shown in yellow). 150

Chapter 1

Introduction

Object learning is an important problem in machine vision with direct implications on the ability of a computer to understand an image. Usually, an object is defined by its appearance (the *pattern* of gray/color values in the object of interest and its immediate neighborhood), shape, and sometimes, by its relationships to other objects in the scene. For many years it has been thought that finding objects in digital images can be achieved by image segmentation (unsupervised partitioning of the image into several "homogeneous" regions based on the similarity of pixel attributes [73]). Unfortunately, unsupervised segmentation by itself has been shown to have limited applicability in object recognition. This is because most real world objects do not have homogeneous color or texture characteristics. To recognize objects, it is generally agreed that we should integrate several contextual cues and prior knowledge which are not utilized in a low-level segmentation module. Fig. 1.1 shows an example of unsupervised segmentation; the pixels in Fig. 1.1(a) have been partitioned into four different constant graylevel regions as shown in Fig. 1.1(b). Note that the

homogeneous regions in Fig. 1.1(b) do not correspond very well with the objects present in the scene. Even if we know the gray value of the region containing the foreground car, it is still not easy to segment the car from the corresponding region (see Fig. 1.1(c)). A similar situation is encountered in segmenting the MR-brain image of Fig. 1.2; an unsupervised segmentation attempting to identify the *corpus callosum* traced in Fig. 1.2(a) produces the noisy regions in Fig. 1.2(b) instead of the true object of interest in Fig. 1.2(c).

(a) (b) (c)

Figure 1.1: A 256 x 256 frame from a video sequence of moving cars. a) Original image. b) Mean-shift segmentation into 4 gray levels [23]. c) The region of (b) containing the foreground car (shown in white).

New segmentation methods have emerged that guide the image partitioning process using appearance, shape and/or contextual models. The early bottom-up segmentation paradigm grouped pixels according to their similarity in spatial and attribute (feature) space and the resulting regions were assigned object labels. This paradigm is now either being replaced by or integrated with top-down paradigms that extract objects from scenes based on the prior knowledge that the user has about the object of interest. The object models themselves have evolved from handcrafted

(a) (b) (c)

Figure 1.2: Midsagittal MRI section of the brain. a) Original image showing the *corpus callosum* (CC) as a C-shaped central structure. b) Low-level segmentation of white matter that contains the CC (taken from [86]). c) High-level segmentation of the CC.

templates that were rigidly matched to the image into learned flexible (deformable) templates that are warped (deformed) to fit the image data. This allows the segmentation method to adapt itself to the natural object variability and different viewing conditions in sensed images. However, despite all these advances, the general image understanding problem is still mostly unsolved.

1.1 An object learning approach to facilitate image understanding

The goal of this thesis is to propose and investigate an object learning and retrieval approach meant to facilitate image understanding by computers. We will demonstrate how various types of objects can be learned and subsequently retrieved from gray level images without attempting to completely partition and label the image. Our method follows a view-based paradigm: an object is defined by its shape, appearance, and

possibly other features present in several view-specific images of the object. Therefore, we do not attempt to reconstruct any $3D$ information; our belief is that it is not necessary to construct $3D$ object models and $2D$ view-specific information is enough for defining an object.

We propose the following *learning-detection-segmentation-matching* paradigm for object learning and retrieval (Fig. 1.3):

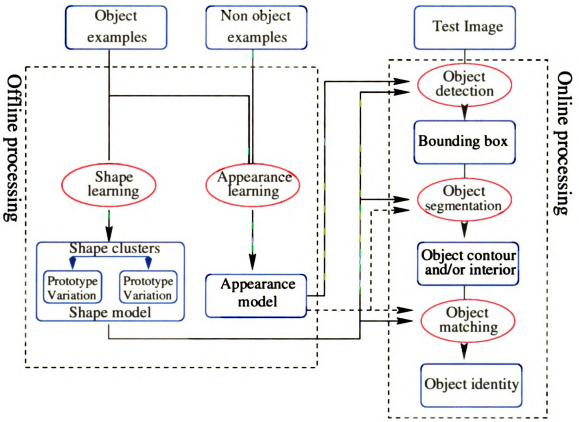


Figure 1.3: Schematic diagram of the *learning-detection-segmentation-matching* paradigm for the object learning and retrieval system developed in this thesis. The actions performed by the system are shown inside ellipsoidal boxes while the data involved in the process are shown inside rectangular boxes. The dashed arrows implying the use of appearance models in object segmentation and matching are shown only for completeness; our current system uses appearance models only for object detection.

1. *Defining and training an object model.* A shape and/or appearance model is

trained from several examples of the object of interest, and possibly, of the remaining pictorial universe (negative examples). It is still unclear how to automatically decide what sort (shape vs. appearance) of model to use, as well as how to integrate the two different types of information. In our work, we have manually determined when a shape model was better suited than an appearance model, but we are aware that a general visual learning system should integrate the two types of models.

2. *Object detection.* We believe that in order to segment and recognize an object, we must first *detect* it. In other words, we need to determine if the object we are looking for (or at least a part of it) is present in the image and if yes, then we determine a region of interest (or a bounding polygon) that contains it. We want to emphasize the difference between *object detection* and *object recognition* [94, 101]. The *object recognition* problem [94] typically assumes that a test image contains one of the objects of interest on a homogeneous background. The problem of object detection does not use this assumption and, therefore, is considered to be more difficult than the problem of isolated object recognition [101].

3. *Object segmentation.* After detection, an object may need to be segmented from the nearby background, either for computing some instance-specific properties (area, shape, etc.) or for verifying its identity. We consider that an exact segmentation is possible only if the object of interest is well distinguishable from its immediate background, that is, it has a visible boundary with a certain shape. For segmenting different objects, we have used state-of-the-art unsupervised methods (*mean shift*) as well as shape-based deformable templates (both top-down and a combination of bottom-up and top-down approaches). The "right" segmentation method to apply

seems also to be object dependent, and it remains unclear how to automatically select it.

4. *Object matching.* In certain applications, after an object belonging to a general class of objects (e.g., human faces, hands, etc.) has been detected and segmented, the final goal is to assign/verify a precise identity (out of a set of possible identities). In this work we demonstrate how the shape of the hand (hand geometry) can be reliably used to verify the identity of a person within a small size (< 100 persons) database. Although appearance information has also been used for identity recognition/verification, we have not yet incorporated it in our system.

1.2 Applications

This section describes the practical applications of our object learning system. These applications can be grouped into three main domains:

1. *Medical applications.* We have considered three different medical applications:

   (a) Detection and segmentation of the left ventricle walls in MR cardiac images (Fig. 1.4(a)). The automatic segmentation is intended to help physicians diagnose heart diseases, but automatic diagnosis can be envisioned as a long term goal.

   (b) Segmentation of neuroanatomic structures in coronal-viewed MR brain images (Fig. 1.4(b)). Certain properties of the segmented structures are needed for studying brain diseases like schizophrenia.

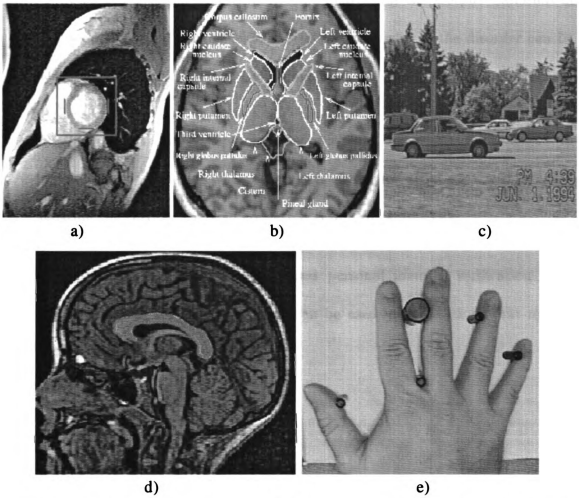


Figure 1.4: Practical applications considered in this thesis. a) Detection and segmentation of the left ventricle in MR cardiac images. b) Segmentation of neuroanatomic structures in coronal-viewed MR brain images. c) Detection and segmentation of vehicles in highway video sequences. d) Segmentation of Corpus Callosum in midsagittal MR brain images. e) Hand shape-based personal identity verification.

(c) Segmentation and shape analysis of Corpus Callosum in midsagittal MR brain images (Fig. 1.4(d)). It has been reported [11] that the shape of the Corpus Callosum may indicate the presence of schizophrenia. Our present goal is the segmentation and shape analysis of the Corpus Callosum in several normal and dyslexic subjects for assessing if the disorder can be indicated by the Corpus Callosum shape.

2. *Intelligent transportation systems.* Detection and segmentation of vehicles in highway video sequences (Fig. 1.4(c)) is useful for reducing travel time by assisting the traveler to avoid congested traffic situations [36].

3. *Biometric systems.* Hand shape-based personal identity verification (Fig. 1.4(e)) is one of the biometrics that can be used in applications that require some sort of user verification [71].

1.3 Thesis outline

The remainder of the thesis is organized as follows.

Chapter 2 surveys the current state-of-the-art in *object detection*, deformable model-based *object segmentation* and *object matching*. An important goal of this chapter is to present comparative results on *common datasets* for some popular detection and segmentation methods. It also points out some of the advantages and drawbacks of the current approaches and suggests ways for improving their performance.

Chapter 3 describes our approaches to *object detection.* Objects can be located in images based on either their shape or on their gray level appearance. We present algorithms and results for each of these paradigms as well as a quantitative evaluation over a large test set.

Chapter 4 presents a novel method for learning shape models from sets of manually traced examples. We show how to cluster the shapes in the training set and eliminate the outliers. Then, a shape prototype and the main modes of variation are derived based on a highly accurate point registration algorithm. We also show results for learning several objects in the brain imaging domain.

Chapter 5 describes two methods for object segmentation that follow the detection stage. One method belongs to the top-down segmentation paradigm and is useful when the object is not well distinguishable from the background. The other method combines a bottom-up low level segmentation with a shape-based top-down matching for objects with a clearly visible boundary.

Chapter 6 examines how one can use the segmented objects for further classification or identity verification. A state-of-the-art hand shape-based personal verification system is presented and its performance is evaluated and discussed. We also demonstrate how morphometric analysis of Corpus Callosum shapes can reveal differences associated with gender, handedness or presence of dyslexia.

The thesis is concluded by Chapter 7 with a list of potential problems which should be investigated in the future.

Chapter 2

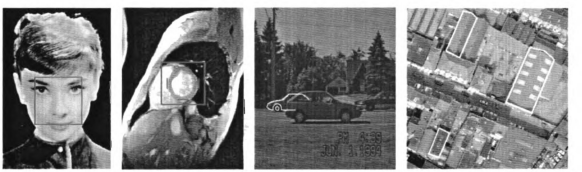
Literature Review

This section will survey the existing state-of-the-art in object detection, segmentation and matching. There exists a large amount of literature on segmentation and matching approaches. However, general object detection methods have just begun to appear and there is still a long way until they can be integrated into a practical general purpose vision system.

2.1 Object detection

Object detection is probably one of the most difficult problems in computer vision [66]. This problem can be defined as follows: given an object O and an arbitrary black and white, still image, find the pose (location and scale) of every instance of O contained in the image. In contrast to what is called "object recognition" where one needs to discriminate between well-defined object classes [94], the detection problem requires us to differentiate between an object class and its complement. Therefore, the detection

methods have to accommodate the object intra-class variability without compromising the discriminative power in distinguishing the object from "background" within cluttered[1] scenes. In most cases, one cannot assume that the test image contains an instance of the object(s) of interest, so the classification of different image patches (windows) has to be done independently and maximum likelihood type methods cannot be applied directly (one has to threshold some form of probability/distance for deciding if a pattern can be classified as the object of interest).



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

Figure 2.1: What features define an object? (a) A human face is mostly defined by appearance. (b) The left ventricle in MR cardiac images is defined by a combination of appearance, shape and position relative to the lung (dark region). (a) A car is mostly defined by shape. (d) A building in aerial images is mostly defined by its shape.

Usually, an object is defined by its appearance, shape, and sometimes, by its relationships to other objects in the scene (see Fig. 2.1). The term "appearance" was introduced in the Computer Vision literature following the seminal paper of Turk and Pentland [123]. According to Nayar *et al.* [94], "the appearance of an object is the combined effect of its shape, reflectance properties, pose in the scene, and illumination conditions". Here, we define appearance as the *pattern* of gray/color values in the

[1]Ratches *et al.* [104] define clutter as "...all signals in a scene of no interest to the observer; for example trees are of no interest to an observer looking for automobiles".

Table 2.1: A summary of the types of features used by different applications of object detection.

Feature type	Applications	Advantages	Disadvantages
Appearance	Medical imaging [41] Human face [7, 22, 83, 85, 93, 96, 97, 103, 111, 121, 126] Transportation [97, 103, 114, 115]	Very general, Implicitly include other feature types	High dimensionality Computational intensive Do not solve for 3D pose
General shape features	Robotics (i) shape indexing-based detection [6, 19] (ii) alignment-based detection [124, 67] Human face [82, 3]	Indexing handles multiple objects efficiently Allow 3D modeling and pose finding	Shape is ambiguous Sensible to occlusion, noise and feature point misdetection [6]
Color, motion and other object/domain specific features	Human face [134, 132] Transportation [69] Medical imaging [38] Automatic target/building detection [10, 104, 118]	Generally fast May be very accurate for a specific problem	Specific to an object/domain Systems have to be manually redesigned when applied to a different object

object of interest and its immediate neighborhood[2].

There is a large amount of literature on object detection methods applied to various problems (see Fig. 2.1 and Table 2.1). Earlier methods were mostly based on object/domain specific features (which may include, but are not limited to, color, motion, $3D$ models and constraints). Comprehensive reviews and evaluations can be found in [88, 38] for medical imaging, in [10, 104] for automatic target detection/recognition, and in [118] for building detection/segmentation in aerial images. Although useful (and usually fast and accurate) for their particular application, these approaches suffer from restrictive assumptions such as the presence of a single object in the scene or a stationary camera. One should also make a distinction between the algorithms handcrafted to detect a specific object and general methods that can be trained to detect an arbitrary object class in gray level images. The dedicated detection algorithms rely on the designer's knowledge about the object and domain of interest as well as on the designer's ability to code this knowledge. On the other hand, a general detection method, would necessitate very little, if any, prior knowledge about the object of interest; the specific domain information is usually replaced by a general learning mechanism based on a number of training examples of the object of interest.

Shape-based object detectors have mostly been used in robotics. There are two well known approaches to detection which use explicit shape features: (i) shape indexing [6, 19] and (ii) feature point alignment [124, 67]. There are also two recent

[2]Some shape features may be implicitly specified by this definition. However, we would like to differentiate appearance-based methods from approaches where explicit shape features are used.

methods in which the shape characteristics of the object of interest are learned from training examples [82, 3]. Indexing schemes can handle multiple objects efficiently and allow for 3D modeling and pose estimation. However, as pointed out by Beis and Lowe [6], such methods are sensitive to occlusion, noise and feature point misdetection. They are also limited to detection of rigid objects under different pose and may not work in case of non-linear variations in shape of the object of interest. The shape learning approaches[3] are able to accommodate the larger amount of variation present in an object class (e.g., a human face).

Appearance-based object detectors have recently been introduced (one of the first face detectors was that of Sung and Poggio [121]) as an alternate to shape-based detectors. They have the advantage that appearance representation is less ambiguous than the representation which uses pure shape features (see the discussions in [3, 6] and Section 2.1.2) or high level object specific features (an interesting comparison of high level features versus appearance templates for human face recognition can be found in [16]).

To our knowledge, there are relatively few appearance learning-based object detection systems, most of which are based on input gray level information [7, 22, 83, 85, 93, 96, 97, 103, 111, 121, 126] and only two approaches [3, 115] combine them with some shape features [4]. Also, many of them have only been applied to a single

[3]We would like to note that shape-learning based detection systems face the difficult problem of selecting negative examples. Appearance based approaches tackle this problem using an "active learning paradigm" [111] in which the false accepts are used for retraining the system. However, we are not aware of an equivalent algorithm for shape based systems.

[4]An object model which integrates appearance with explicit shape features (called *Active Appearance Model*) has been presented in [80, 24]. The object examples used for training are deformed to the mean shape and a *shape-free* representation of the object of interest is obtained by principal component analysis. However, the associated model fitting procedure has mainly been used for

domain (the detection of human faces in gray level images). Table 2.2 shows the main characteristics of some of the detection systems reported in the literature. It is difficult to provide a direct performance comparison of the above mentioned systems since they were not tested on a common dataset. A partial comparison based on the results reported in the original papers can be found in Rowley's and Schneiderman's theses [110, 113]. However, since all systems are based on learning, it may not be fair to directly compare their performances, if the training sets were different.

Most appearance learning-based detection systems essentially utilize the following paradigm (Fig. 2.2): several windows are placed at different positions and scales in the test image and a set of low-level features is computed from each window and fed into a classifier. Typically, the features used to describe the object of interest are the "normalized" gray-level values in the window. This generates a feature vector with a large dimensionality (of the order of a couple of hundred), whose classification is both time consuming and requires a large number of training samples to overcome the "curse of dimensionality". The main difference among these systems is the classification method: Moghaddam [93] and Lew [83] use a complex probabilistic measure (a combined distance *within and from the feature space*), Lin [85], Rowley [111], Vaillant [126] and Weng [129] use neural networks, Colmenarez [22] and Duta [41] use the log-likelihood ratio of the most discriminant Markov chain, Osuna [96] and Papageorgiou [97] use support vector machines, Amit [3] uses tree classifiers and Sung [121] uses the distance from the face and nonface prototypes.

---

segmenting (and computing the identity of) the object of interest in test images in which the object was present. No results have been reported on the method's ability to determine whether the object of interest was indeed present in the test image.

Table 2.2: A survey of the learning-based object detection systems.

System	Objects learned	Features used	Type of classifier
Amit *et al.* [3]	Human faces, handwritten characters	Edge fragments, 2D gray level pattern (20×20)	Tree classifier
Ben-Yacoob [7]	Human faces	2D gray level pattern (20×20)	Multilayer perceptrons
Colmenarez and Huang [22]	Human faces	2D gray level pattern (11×11)	Log likelihood ratio
Duta *et al.* [41]	Cardiac ventricle in MR images	1D gray level patterns (100 feat.)	Log likelihood ratio
Lew and Huijmans [83]	Human faces	2D gray level pattern (23×32)	Distance from feature space
Lin *et al.* [85]	Human faces	2D gray level pattern (12×12)	Neural network
Moghaddam and Pentland [93]	Human faces, hands	2D gray level pattern (N/A)	Distance in and from feature space
Osuna *et al.* [96]	Human faces	2D gray level pattern (19×19)	Support vector machine
Papageorgiou *et al.* [97]	Human faces, pedestrians	2D gray level pattern (19×19)	Support vector machine
Ratan *et al.* [103]	Human faces, cars	1D gray level pattern (64 feat.)	No classifier, 1D warping
Rowley *et al.* [111]	Human faces	2D gray level pattern (20×20)	Neural network
Schneiderman and Kanade [114, 115]	Human faces, cars	Wavelet coefs, Eigenvect (16×16)	Likelihood ratio
Sung and Poggio [121]	Human faces	2D gray level pattern (19×19)	Distance from prototypes
Vaillant *et al.* [126]	Human faces	2D gray level pattern (20×20)	Neural network
Weng *et al.* [129]	Human faces	2D gray level pattern (64×64)	Neural network

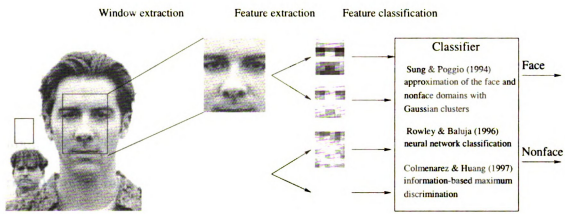


Figure 2.2: The structure of a learning-based face detector.

One of the main performance indices used to evaluate such systems is the detection time. Most detection systems are inherently slow since for each window (centered at a pixel in the test image), a feature vector with large dimensionality is extracted and classified. Two approaches have been proposed to speed up the detection process:

(i) Amit *et al.* employed an efficient focusing method during which a relatively small number of regions of interest is identified based on spatial arrangements of edge fragments. These edge fragments are invariant over a wide range of photometric and geometric transformations and those configurations that are more common in faces than in the background are learned from face examples. Only image patches centered at pixels that belong to the identified regions of interest are further classified into face or non-face. This method is one of the few attempts to incorporate shape information in an appearance-based approach.

(ii) A fast way to perform the classification (called *Information-based Maximum Discrimination*), previously employed in image registration and parameter estimation,

was introduced by Colmenarez and Huang [22] and later modified by Duta *et al.* [41] for object detection. The pattern vector is modeled by a Markov chain and its elements are rearranged such that they produce maximum discrimination between the sets of positive and negative examples. The parameters of the optimal Markov chain obtained after rearrangement are learned and a new observation is classified by thresholding its log-likelihood ratio. The main advantage of the method is that the log-likelihood ratio can be computed extremely fast, only one addition operation per feature is needed.

We are not aware of any comparative evaluation of the different learning-based approaches to object detection. This is probably due to the fact that this domain of research is relatively new and each research group uses its own proprietary data and implementations. Since the object detection domain is very broad and there already exist good comparative evaluations for some specific applications (see Table 2.1), we would like to focus our attention on appearance learning-based methods. The goal of this section is to provide a quantitative comparison of several classification methods applied to the face detection problem, as well as to investigate the difficult issues of object representation and detector training. We regard this evaluation task as very difficult and somewhat empirical and incipient because of the following reasons: (i) there are no standard datasets for training and testing[5] as are (at least for testing) for the face recognition problem (FERET database [99]), (ii) there are no commonly agreed evaluation criteria and methodologies, (iii) in most cases the code is proprietary

[5]Although some of the recent studies report testing results on Rowley's datasets [111] (part of which was taken from Sung's data [121]), about half of the studies mentioned in Table 2.2 report results on other test sets.

and cannot be accessed[6], while independent implementations are very difficult to write due to the complexity of the approaches, and (iv) the object detection problem is sufficiently difficult that empirical ROC curves obtained by simultaneously varying the parameters of the system (like those proposed for edge detector evaluation in [15]) cannot practically be computed.

2.1.1 Why is it difficult to detect objects?

There are several reasons why the object detection problem is considered to be extremely challenging.

1. *Large variability in the appearance of an object.* The sources of variability include: image plane variation (rotation, translation and scale), 3-D object pose, distance from camera and natural variations across different instances of the same object. For a detailed discussion on the variability of the human face, see Rowley's thesis [110].

Figure 2.3: Face-like patterns extracted from outdoor images that are typically used to generate negative examples. We believe that such patterns make the training set inconsistent, since they may very well be classified as faces if found in the right visual context.

2. *Limited information used for object representation.* Often, due to computing time requirements, object detection systems base their decision on a region included in the object of interest. Thus, parts of the object and its immediate context are not

[6]There are three face detection systems which can be tested through the Internet which are discussed in Section 2.1.4.

used during classification. By restricting the classified region to the central part of the face, the true class (face vs. non-face) of an instance becomes highly context dependent. This is illustrated in Fig. 2.3 where 7 face-like patterns were extracted from outdoor images that are typically used to generate negative examples [110]. We believe that the boundary between face and non-face domains is rather fuzzy since patterns like those in Fig. 2.3 may as well be classified as a face if found in the right visual context. Therefore, we think that the negative examples should not be randomly generated from images that do not contain faces since the training set may become inconsistent. The problem is amplified by histogram equalization. This not only reduces the variability generated by illumination conditions and enhances the contrast, but also increases the number of instances one can actually encounter. Consider, for example, the window extracted from the background in Fig. 2.2. It is a constant gray-level region to which noise was added. If we equalize the histogram, we obtain (the third window from top in the third column) only noise, which is seldom encountered if the histogram is not equalized[7]. This noise together with the limited information used for face detection is the most serious source of false alarms.

3. *Accuracy of the classification method.* It is obvious that the number of windows containing the object of interest we encounter in real images is a very small fraction of the total number of windows. One can balance the instance distribution when training the system, but this remains highly unbalanced towards non-objects during testing. Therefore, the classification errors should also be highly unbalanced towards

[7]See also Schneiderman's thesis [113] for an interesting discussion about pattern diversity and an ideal rote learning classifier for object detection.

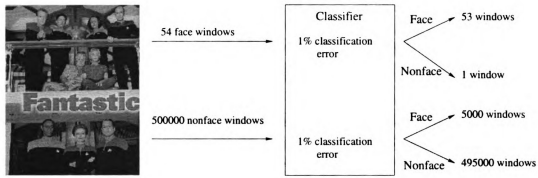


Figure 2.4: The relationship between the classifier accuracy and detection performance.

very small type II errors (non-objects classified as objects). This fact is illustrated in Fig. 2.4. Let us consider a classifier with a type I and type II error rates of 1% which is applied to an image containing 9 human faces. One usually expects that several windows placed very near the true faces be classified as faces; let us assume in this case that every face may be detected at 6 nearby positions/scales. Suppose we extract 500,000 windows at different positions and scales; only 54 of these windows should be classified as faces. With a 1% error rate, our system would correctly classify 53 of the 54 windows as faces, but at the same time, it will incorrectly classify 5,000 nonface windows as faces too. It is obvious that such a system is not useful for practical purposes. Conversely, if we are willing to accept the number of false alarms equal at most to the number of real objects, our system must classify as faces only 54 out of the 500,000 non-face windows, that is, it must have a false alarm rate of 1 in 9,259 windows classified. Unfortunately, there are not too many classifiers capable of such a performance!

2.1.2 Object representation

In order to detect an object, one should first specify the instance (feature) space from which the object examples are drawn [91]. Very few studies give any motivation for the chosen representation[8] (the feature set that describes the object of interest) [9]. Usually, an object is represented by a "normalized" $2D$ gray-value pattern that spans the central part of the object. For human faces, where the immediate background of the head is unconstrained and can take any value, most of the previously mentioned methods implicitly assume that the central part of the face (usually a square defined by the eyes, mouth and the two cheeks) reduces intra-class variation while producing a good inter-class discrimination. Sometimes, the forehead, hair and chin are also included, but the background is always masked. However, each system chooses a different number of features (between 121 and 4096, see Table 2.2) to define the face pattern, depending on the computational constraints of the classification method and personal belief. There are some type of objects, though, that cannot be defined by "a central part" (e.g., pedestrians in [97]) and whose representation has to include part of the background. In such cases, the detection performance reported is usually lower than for object representations that do not include background. On the other hand, sometimes it is useful to include the immediate background in the object description. This is the case in medical images, where immediate neighborhoods of anatomical structures are similar in different subjects (e.g., the left cardiac ventricle in [41] is always located near the lungs; see Fig. 2.1(b)). For such applications, the immedi-

[8]For a discussion of the object representation issues outside the appearance learning paradigm, see Schneiderman's thesis [113].

ate background can help to disambiguate the image interpretation and improve the detection rates.

The second important aspect of the object representation problem is the type of transformations performed on the original input (gray value) pattern in order to obtain the features defining the object. These transformations are also meant to reduce the intra-class variability due to illumination conditions (e.g., brightness and contrast) and camera position. Typical transformations are histogram equalization, lighting correction and, sometimes, feature alignment (warping) [110]. However, as mentioned above, there is a three-way trade-off between the amount of preprocessing and the total detection time of the system (note that the preprocessing has to be applied to each patch/window in the image) and the saliency (characterization power) of a feature set in describing the object. The characterization problem comes from the fact that the preprocessing /feature extraction procedures are not one-to-one, that is, background patterns may produce the same set of features as patterns that belong to the concept to be learned. Most object detection studies do not acknowledge this problem but, implicitly, reduce preprocessing to a minimum and consider as features the normalized pixel values. As such, the feature extraction procedure is close to a one-to-one mapping and the error due to overlap of class conditional densities in the feature space (which adds to the classification error) is quite small. By contrast, the representation problem is well analyzed in the face recognition literature. Craw *et al.* [30] give a comprehensive comparison of different feature sets for automatic face recognition. However, these feature sets are computationally too expensive to be used by a face detector (recall that a detector must compute the feature set for each

patch in the test image while a recognizer extracts features at one or at most a small number of locations).

Finally, it is very difficult to quantify the inherent complexity of the object detection problem. Such an attempt was made by Rowley [110] who compensates the infeasibility of an exhaustive training (training the classifier on the true distributions for object and non-object examples) with a Bayesian approach that takes into account the prior probabilities of faces and non-faces. He estimated that the prior probability that a window of arbitrary size extracted from 130 real world images contains a human face is $1/20,984$. We believe that the prior probabilities for other objects are of the same order of magnitude as well. On the other hand, it first seems that if enough information is used for object description (the object plus the immediate background) and the feature extraction procedure is one-to-one, then the class conditional densities of the object and non-object examples should not overlap (a feature vector represents either the object of interest or not[9]), that is, the Bayes error for this problem should be close to zero. However, in practice it is very difficult to establish an exact boundary between object and non-object examples (see Fig. 2.5).

Figure 2.5: A face pattern (left-most image) is gradually deformed into random noise (right-most image) by linear interpolation. It is very difficult to label the intermediate images as positive/negative examples for training a face detector.

Although the representational framework of our comparison is fixed (2D gray level

[9]Most of the time, humans use additional high level information to make this distinction.

pattern), we still have to decide on the three issues discussed above: (i) what region of the object of interest to use for feature extraction (including/excluding background), (ii) the number of features and (iii) preprocessing transformations. These are difficult choices which cannot be automatically explored given the current computing power. As Schneiderman points out [113] "Ultimately, we must make representational choices by hand... The best we can do is make educated guesses". In short, we have attempted to use a representation for faces which is common to as many of the systems described in Table 2.2 as possible. There was, however, an exception. We have determined that the square region between eyes and mouth was too ambiguous for defining a face (see the discussion in Section 2.1.1), therefore we extended it to also include the forehead (see Fig. 2.2). The rectangular 2D region was linearly subsampled to a 20×15 pattern and the only preprocessing we applied was histogram equalization on 16 gray levels[10]. The 20×15 pattern is quite close to the pattern sizes employed by previous systems and seemed to be a good compromise between representational information and computational requirements.

2.1.3 Classifier comparison on the face detection problem

As already mentioned in Section 2.1, once the representation and preprocessing of the object of interest are fixed, the detection problem becomes a two-class (object of interest vs. background) classification problem. This section presents a comparison of the results produced by six different classifiers on the face detection problem. The

[10]16 gray levels is the maximum which is currently computationally feasible for the maximum discrimination classifier.

purpose of these experiments is to attempt to identify the advantages and drawbacks of using a specific classifier[11], as well as to gain some insight into the problems associated with training. Recall from Section 2.1.2 that the representation (feature vector) used to describe a human face in all classification experiments was a 20×15 (forehead-to-mouth) pattern, and the only preprocessing applied was histogram equalization.

Classifier description

The classifier choice was driven by the following two criteria: (i) existence of relevant literature describing the classifier properties and its possible applicability to object detection (Table 2.2), and (ii) existence of either a commercial or a public domain implementation. The following classifiers were used (see also Table 2.3):

1. *1-Nearest Neighbor classifier* [37]. In a Nearest Neighbor type of classification, a training set P of points in a d-dimensional space is preprocessed into a data structure so that given any query point q, the nearest (or generally the nearest k) points of P to q can be found efficiently. We employed the "ANN" package of Arya *et al.* [5, 64] which is a system for answering nearest neighbor queries both exactly and approximately. It uses a $kd-tree$ (or $bd-tree$) to store the training data. It offers a choice among different Minkowski metrics, as well as a way to specify the amount of error to be tolerated in the search (returning a point that may not be the nearest neighbor, but is not significantly further away from the query point than the true nearest neighbor).

[11] None of reviewed object detectors attempted to combine multiple classifiers. Due to computational expenses, it is not feasible to apply several classifiers to each patch of the image. However, one can apply cascaded classification, where a pattern is fed into a more computationally demanding classifier only if it was classified as an object of interest by a previous simpler classifier [39].

We specified the $kd-tree$ data storage and the Euclidean metric and constructed an ROC curve (indexed by the error tolerance parameter ϵ, or equivalently, by the classification rate) of the training set using the "leave one out" paradigm. A "knee" in the curve was found for $\epsilon = 6$ (corresponding to a $FR = 0.08\%$ and $FA = 0.04\%$) and was kept fixed for the remaining experiments.

2. *Support Vector Machines* (SVM) [127, 17]. The main idea behind the SVM approach is to separate the classes represented as multidimensional patterns with a surface that maximizes the margin between them. This is an approximate implementation of the Structural Risk Minimization induction principle and aims at minimizing a bound (equal to the sum of the training error rate and a term which depends on the Vapnik-Chervonenkis dimension of the learning machine) on the generalization error of a model. For our experiments, we used the "SVM^{light}" package of Joachims [75, 62]. The following kernel functions were tested: first to third degree polynomials, radial basis function and sigmoid function. For linear and sigmoid kernels, the training process did not converge although we tried multiple values for the "c" parameter which regulates the trade-off between the training error and the generalization margin. For the remaining kernels, the training process took between 25 mins. (second degree polynomial) and 10 hours (radial basis function) depending on the values of "c" parameter and the width γ of the RBF kernel. The offline classification accuracy (defined in Section 2.1.3) was somewhat better (by about 3%) for the radial basis function kernel. Since results on the face detection problem using this type of kernel were also reported by Joachims, we decided to use an RBF kernel with $\gamma = 1$ for our experiments.

3. *Learning vector quantization* (LVQ) [77]. LVQ is a classification method in which the object classes are described by a relatively small number of codebook vectors, properly placed within the feature space such that the decision boundaries are approximated by the nearest neighbor rule. The accuracy of the approach is dependent on (i) the number of codebook vectors assigned to each class and their initial values, and (ii) the learning algorithm, learning rate and stopping criterion. For our evaluation, we employed the "LVQ" package of Kohonen *et al.* [78, 63]. The main parameter to be specified when using this package is the number of codebook vectors to be constructed for each class. We performed several experiments with 250, 500 and 1000 codebook vectors per class. There was an 8% increase in accuracy when going from 250 to 500 codebook vectors and no significant improvement was noticed for 1,000 codebook vectors. Therefore, all subsequent experiments were done with 500 codebook vectors/class and 20,000 learning iterations.

4. *Decision trees* [106]. Decision trees classify pattern vectors by propagating them down the tree from the root to some leaf node, which provides the class labels. Each node in the tree denotes a test of a specific feature, and each branch from that node corresponds to one of the possible values for that feature. A test pattern is classified by starting at the root node of the tree, testing the feature specified by this node, then moving down the tree branch corresponding to the value of the feature in the given pattern. This process is then repeated for the subtree rooted at the new node [91]. In our experiments, we used the $C5$ software package of Quinlan [102]. $C5$ is a program for computing classification rules in the form of decision trees and/or rulesets from a set of given training examples. Trees can optionally be converted to rulesets. Rulesets

are often simpler and easier to understand (since each rule can be interpreted) and sometimes rules are more accurate predictors than decision trees. A pattern vector is classified by a ruleset by identifying all the rules that it satisfies and resolving conflicts by (weighted) voting. $C5$ also incorporates an adaptive boosting procedure, that is, the program generates multiple trees and rulesets using different weights for each training example. The weights are adjusted after each tree or ruleset is produced to focus attention on those cases that were misclassified. For our experiments we used 3-trial boosting applied to rulesets derived from the decision trees constructed by $C5$.

5. *Neural Network* [91, 37]. Several algorithms have been designed for training a feedforward neural network (computing the network weights and biases) among which the most well known is the "backpropagation rule" (a gradient descent in the weight space). For the reported experiments, we have used the backpropagation package written by Shufelt [90]. The network architecture was fixed to one hidden layer completely connected to the input and output layers and several experiments were performed using 10, 15 and 30 hidden units. We noticed that the convergence of the training process was highly dependent on the [Learning rate, Number of training examples] pair. When using only 1,000 (500 positive and 500 negative) examples, the training converged[12] after 1,000 epochs with a learning rate of 0.1. However, when using 20,000 (half positive, half negative) examples, the training converged only after about 4,000 epochs and required a learning rate of 0.004. On the other hand, the number of hidden units (10, 15 or 30) seemed to mostly influence the actual

[12]We say that the training converged on a given training set if the classification error on that set is less than 5%.

computing time (more hidden units required more computation) and had less impact on the classification accuracy (about 5% improvement between 10 and 30 hidden units) and the learning rate. Also note that since we used the neural network as a general purpose classifier, we did not tailor the network architecture for the problem at hand as other studies did [111, 129]. This may be one of the reasons why we were not able to obtain better results as reported by others [111].

6. *Maximum discrimination* (MD) [22, 41]. A detailed description of this classifier can be found in Chapter 3. The MD classifier used for the reported experiments was implemented by us and most of the time a threshold $T = 0$ was used to generate the decision boundaries[13].

Training data and methods

The training data consisted of a combination of laboratory captured face images (Weizmann dataset: 840 pictures of 28 persons, 2 facial expressions, 3 lighting directions and 5 face orientations per person, MIT dataset: 144 images of 16 persons, all frontal views with different lighting, MSU dataset: 216 pictures of 62 persons mostly frontal view with frontal lighting) and real-world scanned photographs (390 persons with unknown lighting and frontal to semi-profile orientation). In each of the 1,590 face images, the central part of the face was manually identified using a rectangle of side ratio 4/3 which was resampled to form a 20×15 pattern and whose histogram was equalized to 16 gray levels. From each "true" face pattern, 6 synthetic patterns were

[13] In section 2.1.3 we will refer to a "relaxed" version of the MD classifier in which the threshold T was decreased in order to decrease the false reject rate (with a corresponding increase of the false accept rate).

created by symmetry and rotation in the image plane as described by Rowley [110]. Therefore, a total of 9,540 face-patterns were employed as positive examples. The set of negative examples started with a random set of 5,564 non-face patterns to which 5,013 more patterns were added from the false alarms produced by the maximum discrimination classifier during eight successive training sessions. During the eight training sessions, the system examined about half a billion patterns and attained a false alarm (FA) rate of about 1 pattern in 500,000 patterns classified on the training images, and about 1 pattern in 50,000 patterns classified on test images. After the 8th training session, many of the false alarms were face-like patterns (see Fig. 2.3) and we gave up further training since the training set could become inconsistent. Therefore, a total of 20,117 training patterns were used to bring the MD classifier to a reasonable accuracy and were subsequently used to train the other classifiers.

Evaluation results

In order to compare the discrimination power of the six classifiers, we performed the following experiments:

• *Offline assessment of the generalization ability to unseen faces and low quality images.* The set of 9,540 face patterns was split into two sets corresponding to the laboratory images (7,200 patterns) and real world images (2,340 patterns). The non-face pattern set was randomly split into two parts. A 2-way cross validation was used to assess how well a classifier would perform when tested on face images which are new and of a different quality. The results are shown in column 3 of Table 2.3 (first row corresponding to each classifier). One can notice that only the decision tree and

the maximum discrimination classifiers are sensitive to the quality of the face pattern (when trained on the scanned faces, the accuracy was much lower than when trained on the laboratory images). The FA rate did not vary too much between the two trials. On the average, the FR rate was between $18 - 32\%$ while the FA rate was in the range $3 - 20\%$. The best overall performance on this task was obtained by the SVM classifier.

• *Offline assessment of the generalization ability to rotation and symmetry.* The set of 9,540 face patterns was split into five sets (by taking every 5th pattern) such that no more than two patterns per "real face" were used for training while the system was tested on the remaining patterns (5-way cross validation). The results are shown in column 3 of Table 2.3 (second row corresponding to each classifier). The FR rate is much lower in this case, showing that symmetry and small rotation do not introduce as much variability in the face set as new face patterns. Again, the best results were attained by the SVM.

• *Online assessment of the face detector performance.* Five of the six classifiers were trained on the full set of 20,117 examples and the performance of the resulting face detector was evaluated as follows. First we examined the processing time needed for training/testing (columns 2 and 4 in Table 2.3) since this was an important parameter whose value influences the values of the remaining parameters. The training times spanned 3 orders of magnitude range (0.5 min for approximate nearest neighbor to 2,400 min for neural networks). We measured the testing time by the number of patterns classified in one second. These times also varied largely from 20,000 classifications/sec (maximum discrimination) to 4 classifications/sec (1-nearest neighbor).

Table 2.3: A comparison of different classifiers for the face detection problem. Legend: [*] = the classification rate does not depend on the size/distribution of the training set or the test sample, [**] = the classification rate depends on the size of the training set but not on the distribution of the training/testing samples, [***] = the classification rate depends on the size/distribution of the training set but not on the test example, [****] = the classification rate depends on the size/distribution of the training set *and* the test pattern, † The FA rate for the Maximum discrimination classifier is very small since false alarms from previous testing trials were added to the training set (as such, it should not be directly compared to the FA rates of other classifiers). The testing results show that the training set is crucial and should be constructed incrementally for each type of classifier.

Classifier Implementation Parameters	Train time	Offline accuracy FR / FA (%)	Testing time class/sec	Online accuracy Faces rejected/NF *patterns* accepted
1 Nearest Neighbor	–	27/8, 29/15 2.4/16	4 [**]	Not tested
Approx. 1-NN Arya *et al.* [5] $\epsilon = 6$	30 secs	26/9, 30/16 2.8/17.5	30 [****]	Not tested
SVM [127, 17] Joachims [75] RBF kernel, $\gamma = 1$	2 h	18/2, 18/4 2.5/4.7	40 [***]	FR : 5.6 – 19% FA : 0.8%
LVQ [77] Kohonen *et al.* [78] 1000 codebook vectors	10 mins	33/17, 32/15 5.8/7.6	100 [***]	FR : 7.3 – 9.5% FA : 7.4%
Decision tree [106] C5 [102] Rules, 3 trial boosting	1.5 h	11/16, 41/5 11.3/2.1	900 [****]	FR : 10.7 – 15% FA : 1.7%
Neural Network Shufelt [90] 30 hid. u. $LR = .004$	20-40 h	29/11, 32/7 10.6/10.3	2000 [*]	FR : 4.3 – 25% FA : 5.3%
Max. discr. [22] Duta *et al.* [41] $T = 0$	20 mins	15/27, 43/11 5/23.5	20000 [*]	FR : 6.5 – 35% FA : .01%†

There is one fact that we would like to emphasize: for only two of the six classifiers (neural networks and maximum discrimination), the testing time is independent of the size/structure of the training set as well as of the test pattern. As a consequence, increasing the size of the training set by including the mistakes made in the early training stages (especially on non-object patterns) will not affect the detection time. Unfortunately, the remaining four classifiers become significantly slower when the training set is increased. As an example, we attempted to retrain the SVM using about 2,000 of the false alarms it produced after the initial training session. As a result, 500 more support vectors were selected and the classifier became 25% slower. However, the FA rate decreased by a factor of 3 and the FR rate slightly increased. Therefore, it seems quite impractical to have several training sessions using the SVM's (or, in general, any sort of nearest neighbor approach). If the classifier has to go through several hundred million patterns in order to achieve a reasonable accuracy, then at 40 classifications/sec, several months of computing time would be required.

The online face detection accuracy was estimated as follows. First, the Olivetti face database (40 persons, 10 images/person with different lighting and facial expression) was grouped into one 1024 × 846 composite image (see Fig. 2.6 and *http://www.cam-orl.co.uk/facesataglance.html*) and was used to evaluate the FR rate on laboratory images. We also estimated the FR rate for real world images on a 620 × 1152 group image containing 89 faces (Figs. 2.7-2.8) downloaded from the CMU demo page (*http://www.ius.cs.cmu.edu/IUS/usrp0/har/FaceDemo/images/649/input.gif*). One should notice that during online evaluation, the FR rate can only be expressed

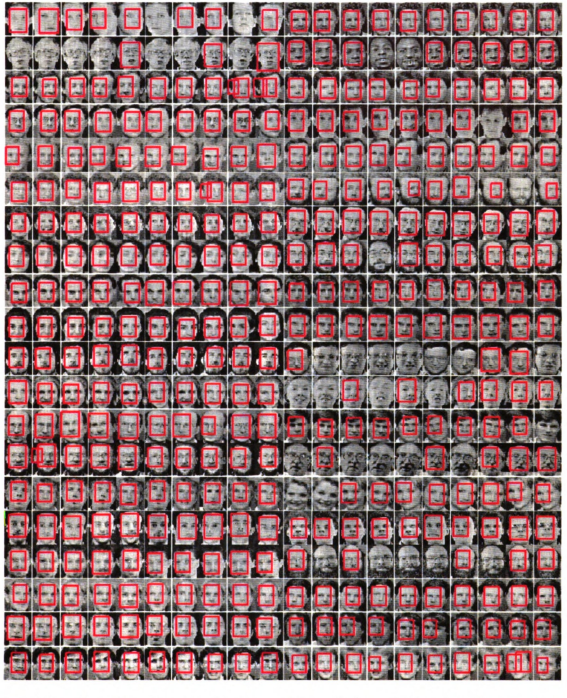


Figure 2.6: Face detection results produced by a "relaxed" version of the maximum discrimination classifier followed by an SVM classification on the Olivetti database. 362 out of the 400 faces (90.5%) were successfully detected.

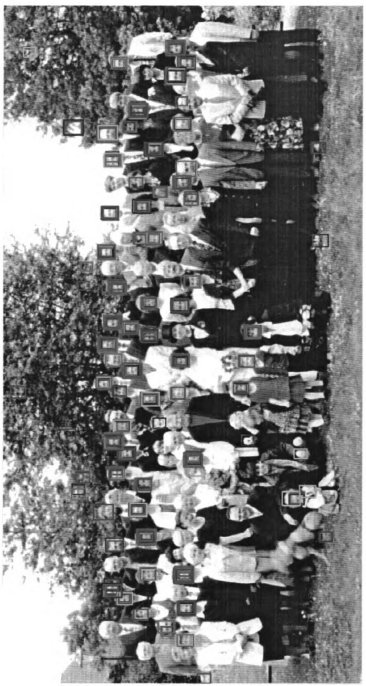


Figure 2.7: Face detection results produced by the maximum discrimination classifier on a group image. No arbitration has been performed; all patterns classified as faces are shown. 57 out of the 89 faces (64%) were successfully detected. About 35 background windows were misclassified as faces inducing a false accept rate of 1/49,000.

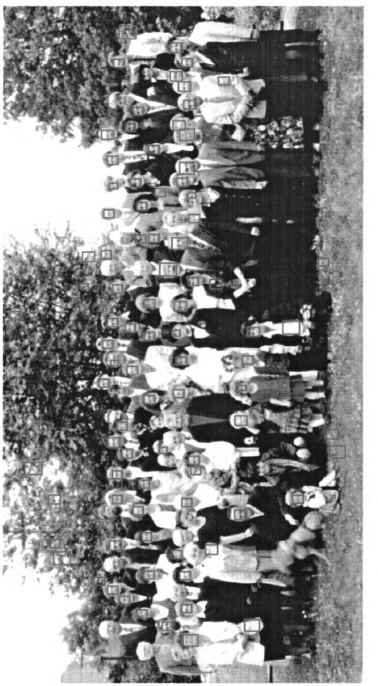
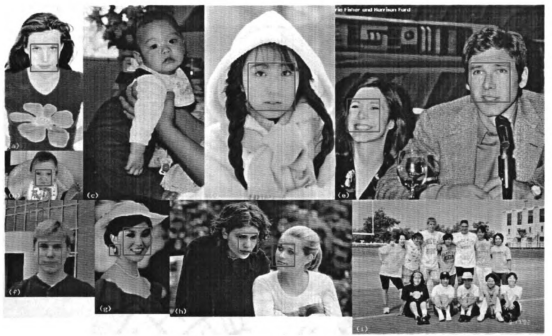
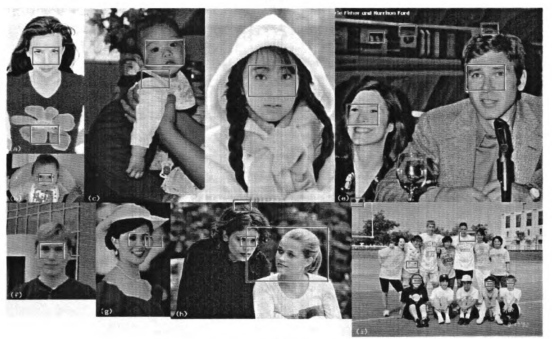


Figure 2.8: Face detection results produced by a "relaxed" version of the maximum discrimination classifier followed by an SVM classification on a group image. Multiple detections at nearby positions have been combined into one face rectangle. 63 out of the 89 faces (70.8%) were successfully detected. There are 34 false accepts.

as the percentage of *human faces* rejected as opposed to offline evaluation where the percentage of rejected *face patterns* was computed. This is due to the fact that during online testing, the set of patterns is not labeled and one usually does not know how many nearby positive responses a human face can induce (see Fig. 2.7). On the other hand, we could only indirectly assess the number of false rejects on large images. Both images described above contain over 2 million patterns (windows) to be classified which would require almost one day of computing time for the slow classifiers. Instead of classifying each pattern with a given classifier, we chose to first apply a "relaxed" version of the maximum discrimination (the log-likelihood threshold was set such that the FR rate was very small) and classify further only the positive responses. As such, the number of patterns was reduced to less than 10,000 and the testing time became acceptable. The FR estimates for the five classifiers are shown in the last column of Table 2.3. It can be noticed that all classifiers performed substantially better on the laboratory images ($FR = 7 \pm 3\%$) than on the real world image ($FR = 21 \pm 11\%$). The FA rate was evaluated on a set of 40 images that did not contain human faces (though they may have contained face-like patterns as previously discussed) from which about 1,500,000 patterns were extracted and classified. The results (also shown in the last column of Table 2.3) are not very different among the tested classifiers (except for the maximum discrimination whose FA rate is 3 orders of magnitude smaller due to the fact that it was trained in several sessions with false alarm bootstrapping).



(a)



(b)

Figure 2.9: Output of the Rowley *et al.* [111] (produced by the system demo posted at *http://www.ius.cs.cmu.edu/IUS/usrp0/har/FaceDemo/gallery.html*) (a), and Lew and Huijmans [83] (produced using the code posted at *http://www.wi.leidenuniv.nl/~mlew/face.detection.html#DMO*) (b) face detectors on a collated test image.

(a)

(b)

Figure 2.10: Output of the Colmenarez and Huang [22] (produced by the system demo posted at *http://troi.ifp.uiuc.edu/~antonio/section2.html*) (a), and Amit *et al.* [3] (b) face detectors on a collated test image.

Figure 2.11: Output of a "relaxed" version of the maximum discrimination classifier (implemented by Duta *et al.* [41]) followed by an SVM classification on a collated test image.

2.1.4 Comparing the performances of some available face detectors

It is rather difficult to quantitatively evaluate the performances of the detection systems outlined in Table 2.2. The main reason is that, except the Lew system [83] and the online demos of the Rowley [111] and Colmenarez [22] systems, the original code is not made available for public testing. Even with the above three cited systems, the user does not have access to the parameters of the method, so an empirical *ROC* curve cannot be computed. On the other hand, all the detection algorithms are quite sophisticated and do not allow for a detailed description in a conference/journal paper, so an independent implementation is not easy to achieve either. A partial comparison made on the basis of the results reported in the original papers describing some of

these methods can be found in Rowley and Schneiderman's theses [110, 113]. We also performed a "visual" comparison of the systems which we could access according to the following methodology: we picked up from Rowley's gallery [65] a fairly large (752×1152) image containing 22 mostly frontal human faces at very different scales. Since the results of Rowley's face detector on this image were already displayed, we only tested the other three mentioned systems (using their default parameter settings) as well as our implementation of the MD-based detector. The results are shown in Figs. 2.9-2.11. The detection times we obtained on this test image ranged from 25 secs for [22], 80 secs for [3], 5 mins for our implementation to about 5 hours for [83]. Since the demos were run on different machines the detection times may not be directly comparable. However, the Lew system [83] is quite slow since the classifier is based on projecting the feature vector on a relatively high-dimensional subspace. Our overall impression is that a substantial amount of work still needs to be done in designing a reliable object detection system.

2.1.5 Discussion and conclusions

In this section we have surveyed the research area of appearance-based object detection. We also attempted to identify and discuss some of the problems one encounters when designing and training an object detector. The detection problem is very important since it is the first step towards an unconstrained object recognizer. As mentioned in Section 2.1, most studies have focused on the human face detection problem, although five other object classes were successfully learned. However, the

range of choices for object representation as well as for classification methods has not been fully explored yet.

Our choice for representing a human face was determined by the trade-off between the characterization power of the feature set and detection time. We determined whether including more information (e.g., the forehead region) in the face description can reduce the ambiguity discussed in Section 2.1.2. This was indeed the case, but we still obtain several false accepts which need to be removed. Fig. 2.3 suggests that a human face is not a unique pattern, but the right pattern in the right context. That is, we do not perceive as faces face-like patterns that come from the grass or tree regions in Fig. 2.7 just because we do not perceive them at all (they cannot be visually separated from their immediate background). To our knowledge, no object detector has yet attempted to incorporate explicit context information when classifying a pattern. The relatively large number of false accepts in Fig. 2.3 suggests that such information may be required for reliable detection. However, it does not seem computationally feasible to compute contextual information for each pattern in a large image. We believe that classifying a pattern based on context should be done only if the pattern itself was classified as object of interest. If context information is going to be used, the answer to the question "What part of the object of interest should be used for feature extraction?" may not be very important.

We also performed a quantitative comparison of several classification methods which could be integrated in an object detector. Based on the results shown in Section 2.1.3, we could draw the following conclusions:

(i) The differences in accuracies of the tested classifiers were not larger than 15% while

the computation time needed for training/testing varied over a 3 order of magnitude range. Therefore, the selection of a certain classifier for object detection should also take into consideration the computational requirements not only the accuracy performance. A slow classifier may not only be impractical to test in a reasonable amount of time, but also impractical to sufficiently train it.

(ii) The choice of the training set can make a large difference in the system's accuracy and computational requirements. The results shown in Table 2.3 suggest that the training set required to obtain a good detection accuracy is classifier dependent. This statement obviously applies to the set of negative examples, that is, this set should be collected incrementally in several sessions by bootstrapping the false accept errors. However, we believe that choosing the "right" positive examples also affects the detection performance. In our case, we feel that our set of face examples was not fully representative of the face population (e.g., we had few examples of persons wearing glasses or beards, fact which impacted negatively on the detection rate). To our knowledge, no systematic strategy has been used yet for collecting positive examples in the same manner negative examples are collected. We believe though that such a strategy may be necessary.

(iii) The classifiers which have a nearest neighbor type complexity (that is, approximately linear dependence on the size of the training set) may not be well suited for object detection since both the training and testing times can become prohibitive with incremental training.

(iv) We do not believe that a single classifier can reasonably solve the general object detection problem. A sequential combination of classifiers (of which the first one

applied should be very fast) may be better suited for this problem. This fact is illustrated in Figs. 2.7-2.8 where using SVMs in a second classification stage improved the detection rate by 7% for the same false accept rate. Note that it would not have been computationally feasible to only use SVM on such a large image.

Finally, we have not explicitly addressed the problem of arbitration among multiple detections of the object of interest at nearby positions. This may require using a combination of classifiers or a voting scheme (which we actually used in our experiments). A more detailed treatment of this aspect can be found in Rowley's thesis [110].

2.2 Object segmentation using deformable models

Objects in sensed images are expected to deform due to the varying imaging conditions, sensor noise, occlusions, and natural variations among different instances of the same object. For this reason, deformable models are becoming increasingly popular. The main idea is to encode a variety of shape deformations while maintaining the inherent object structure in the model.

Deformable models have been used in a wide range of applications, including image/video database retrieval [125, 136], medical image analysis [120, 88, 27], tracking [51, 84], and restoration [4]. Here, we are specifically concerned with the use of deformable models for image segmentation. Given an object model and a set of rules about the possible aspects of the objects as seen in the real world, the goal is to locate the exact boundaries of that particular object in an image. Various approaches

have been proposed in the literature to solve this problem. The differences lie in the specification of the model, the definition of the set of rules, and the recovery process which performs the segmentation.

The study of deformable models started with an attempt to solve shape matching, where a given shape needs to be located in an image. The simplest approach to shape matching is correlation-based matching which has limited applications because objects are not always undergoing rigid transformations in many applications. Further, one might want to segment a generic class of objects such as cars, without having to model each and every type of car. Deformable models allow us to capture such sources of intra-class variability. We now categorize various deformable models that have been presented in the literature.

Active contour models were first introduced by Kass *et al.* [76] and have been intensively studied. An energy-minimizing contour, called a snake, is controlled by internal forces which enforce smoothness of the contour and external forces which attract the contour to salient features in the image. A gradient descent procedure is used to slowly bring an initial contour to the edges of the object of interest in the image. Note that these models do not encode any specific shape information about the object and are very sensitive to the initial position of the snake and to image noise (see Fig. 2.12). To overcome these limitations, more constrained energy functions with balloon forces (Cohen [21]), attractors and tangent constraints (Fua and Brechbuhler [50]), region information (Ronfard [109]) and gradient vector flow (Xu and Prince [133]) have been proposed. Others have proposed different energy minimization approaches such as dynamic programming and graph theoretic algorithms

(Amini *et al.* [2], Geiger *et al.* [51] and Cox *et al.* [28]).

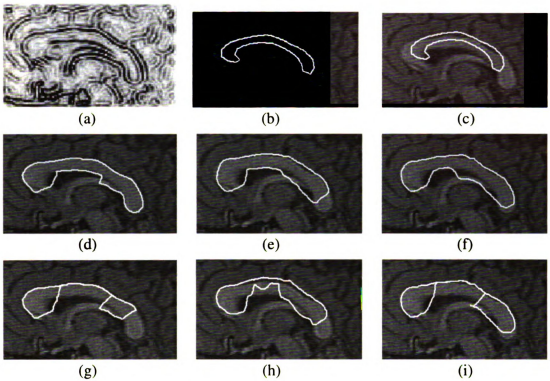


Figure 2.12: CC segmentation using *snakes*. a) Gradient magnitude of a region around CC. b) Manual snake initialization using an average CC. c) Same initialization as (b), but shifted 3 pixels upward. d)-f) Snake segmentation starting from (b). g)-i) Snake segmentation starting from (c). Segmentations in (d) and (g) used traditional snakes [76], (e) and (h) used *balloons* [21], (f) and (i) used GVF snakes [133]. Results shown in images (d)-(i) were produced using the software posted at *http://iacl.ece.jhu.edu/projects/gvf*.

Spline-based models try to impose a little more structure in the shape of the snake. These models still do not encode specific shape information but the model is expressed as a linear combination of a set of basis functions and the shape of the object is defined by the coefficients of this linear combination (Menet *et al.* [89], Figueiredo *et al.* [47]). An example of adaptive B-spline segmentation [47] is shown in Fig. 2.13. Since the segmentation method enforces the interior of the object defined by the spline to be homogeneous, the results are more accurate than those obtained

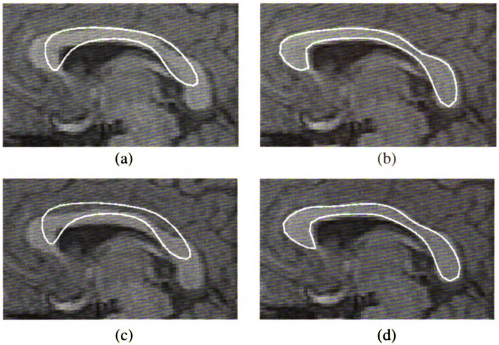


Figure 2.13: CC segmentation using *B-spline* models [47]. a) Manual spline initialization using an average CC; b) Adaptive spline segmentation starting from (a); c) Same initialization as (a) shifted 3 pixels upward; d) Adaptive spline segmentation starting from (c); (Courtesy of Prof. Mario Figueiredo)

using snakes. It also appears that the method is less sensitive to the starting position of the spline.

When some prior information about the shape of the object is known, parametric deformable models are used, where the shape of the object is encoded by a small number of parameters. The advantage of these models, also called deformable templates, is that they are better at bridging boundary gaps and they are more consistent. Since we are mostly concerned with parametric deformable models, the terms model and template will be used interchangeably. Parametric deformable models can be categorized into analytical deformable templates and prototype-based deformable templates. In both of these models, the template interacts with the image through an energy function which needs to be minimized. As in the case of active contours, the

energy function usually comprises of two terms. The internal energy term encodes the shape of the template which penalizes the deviation of the deformed template from the expected shape. The external energy term measures the fit of the template to the important features in the image.

Analytical deformable templates are defined by a set of analytical curves. The geometric shape of the template can be changed by varying the parameters in the analytical expressions. Yuille *et al.* [135] use ellipses to define the shape of the eyes and the mouth to detect facial features in images. Lakshmanan *et al.* [79] use two parallel straight lines to detect the boundary of airport runways in radar images.

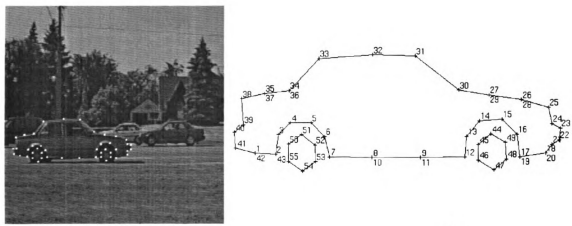
Prototype-based deformable templates are more flexible because they are derived from a set of examples of objects which are expected in the images. This approach was first presented by Grenander *et al.* [56] who described a systematic framework to represent and generate patterns from a class of shapes. A shape is represented by a set of parameters and a probability distribution on the parameters is specified to allow a flexible bias towards a particular shape.

The success of these models depends on how well the parameters and the probability distribution can be defined to accurately represent the shape class. This has lead to an increased interest in the topic of shape learning. Cootes *et al.* [27] have proposed the Point Distribution Models, where the shape class is learned from a set of example shapes. Once the shapes are aligned and properly annotated, principal component analysis is used to generate an average shape (or prototype) along with a series of modes of variation. Duta *et al.* [40] have proposed a method to automatically align and annotate the training examples. They are then able to generate an

average shape and modes of variation from a set of object outlines.

2.3 Point Distribution Models

We now describe the *Point Distribution Model* (PDM) originally designed by Cootes *et al.* [27] and improved by Duta and Sonka [44, 43]. A *PDM* represents a shape as a set of points in the Euclidean plane (space) (Fig. 2.14a)). Training shapes are obtained by (manually) tracing the object of interest in several images (see Fig. 2.15). From these tracings, an object model (Fig. 2.14b)) is constructed and a set of points corresponding to the model are extracted either manually or by automatic learning [40].



(a) (b)

Figure 2.14: Illustration of Point Distribution Model. a) A 256 x 256 frame from a video sequence of moving cars with 55 landmark points superimposed. b) A 55 point car model.

The given set of training shapes is first aligned (scaled, rotated and translated so as to minimize the sum of squared distances between corresponding points) using *Procrustes Analysis* [34] (Fig. 2.16). Each n-vertex shape instance from the training

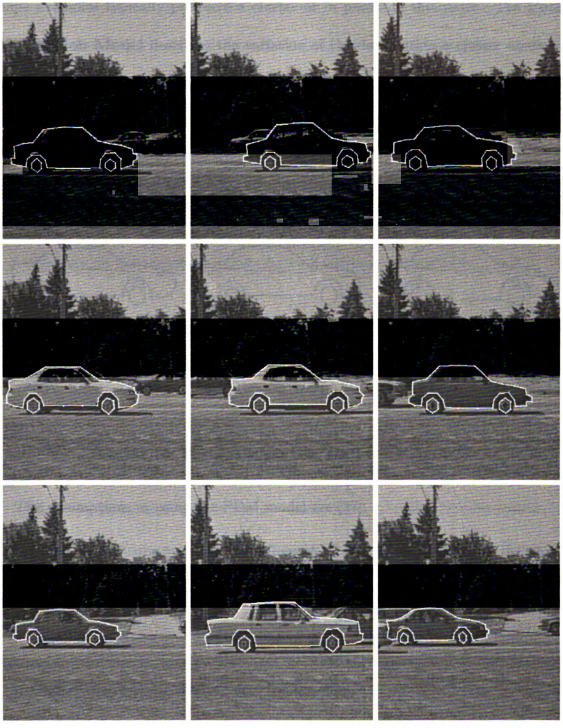


Figure 2.15: A set of nine training images for the foreground car (moving left-to-right) with the car outline manually overlaid.

set can be regarded as a single point in the $2n$-dimensional space. Thus, a set of m shape examples typically produces a cluster of m points in this space. The **Point Distribution Model** describes the variation of the data within a cluster assuming a linear dependence between point coordinates.

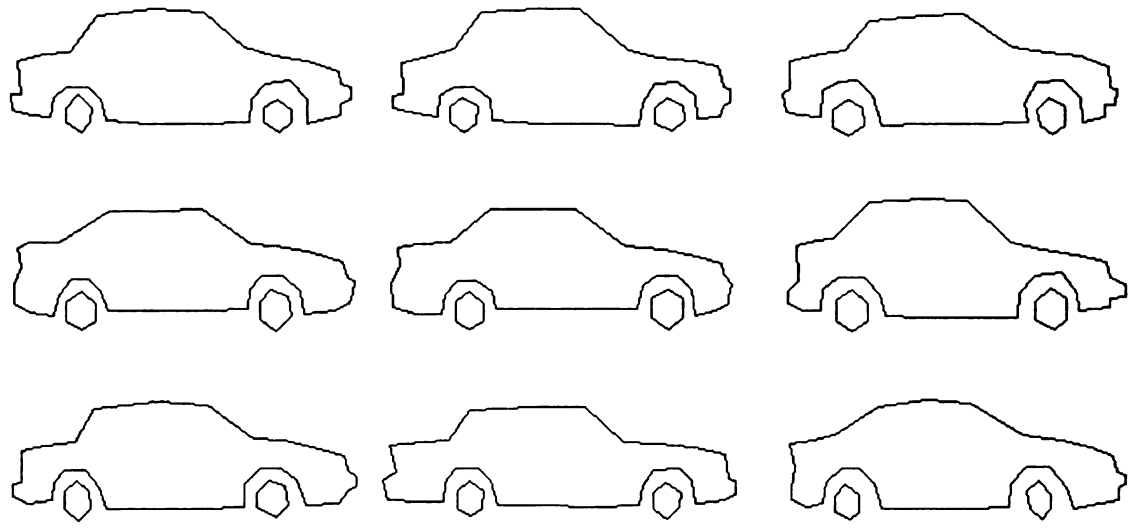


Figure 2.16: The nine aligned car contours of Fig. 2.15

The assumptions in using the PDM model are [27]:

1. Points in a cluster lie within an *Allowable Shape Domain* (ASD) of the object which is approximately ellipsoidal.

2. Training examples give an indication of the size of the ASD.

3. Cluster center is specified by an "average shape".

4. Every $2n$-dimensional point within an ASD represents a vertex sequence, whose shape is similar to the shapes of examples in the training set. Thus, by moving about the ASD, we can generate new shapes in a systematic way.

The principal axes of the $2n$-dimensional ellipsoid are described by p_k (k=1,...,2n), the unit eigenvectors of the $2n \times 2n$ covariance matrix S, corresponding to λ_k, the k-th eigenvalue of S. Each axis gives a *mode of variation*, a manner in which the landmark (object contour) points tend to move together as the shape varies. The main properties of the ASD are:

1. The eigenvectors corresponding to the largest eigenvalues describe the longest axes of the ellipsoidal ASD. Thus, they describe the most significant modes of variation in the variables used to derive the covariance matrix.

2. The variance represented by each eigenvector is equal to the corresponding eigenvalue.

3. Most of the variation in object shape can usually be represented by a small number, t, of modes. This means that the $2n$-dimensional point cluster (ellipsoid) can be approximated by a t-dimensional ellipsoid ($t \leq 2n$), where t is chosen so that

$$\sum_{j=1}^{t} \lambda_j / \sum_{j=1}^{2n} \lambda_j \geq K.$$

   where K, $0 < K \leq 1$ is specified by the user. Typical value of K is 0.95.

4. Any point in the ASD (any allowable shape of the object) can be reached by taking the mean (center) cluster point and adding a linear combination of the eigenvectors. Thus, any shape $\mathbf{x}$, (a $2n$-dimensional vector) in the training set, can be approximated using the mean shape $\bar{\mathbf{x}}$, and a weighted sum of deviations

obtained from the first t modes:

$$\mathbf{x} = \bar{\mathbf{x}} + Pb, \tag{2.1}$$

where $P = (\; p_1 \;\; p_2 \;\; \ldots \;\; p_t \;)$ is the $2n \times t$ matrix of the first t eigenvectors, and $b = (\; b_1 \;\; b_2 \;\; \ldots \;\; b_t \;)^T$ is a vector of weights.

5. New examples of shapes similar to those in the training set can be generated by varying the parameters (b_k) within suitable limits (Fig. 2.17).

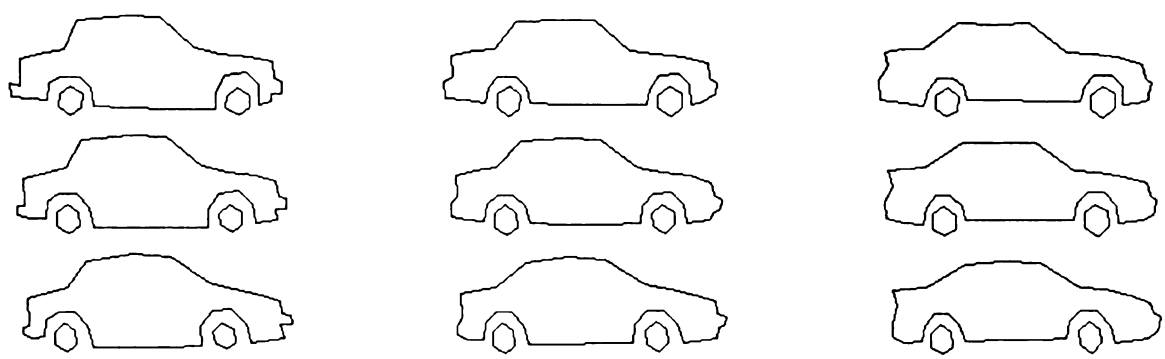


Figure 2.17: Artificial car examples generated by simultaneously varying the model parameters corresponding to the first two largest eigenvalues on a bidimensional grid. The average car is shown in the center. The main two modes of variation determine the shape of the front and rear part of the vehicle as the global shapes in the aligned training set (Fig. 2.16) are relatively similar. Note that the original objects (Fig. 2.15) look more different because of scale and gray-level appearance.

In order to take advantage of the available a priori knowledge, three additional features of the object might be included in the model: gray-level appearance, border strength, and average position of the object (for more details, see [44]). We may also use implicit knowledge about object context representing inter-relationships of several objects present in the image.

To summarize, a **knowledge-based shape model** combines generally applicable parameters of the point distribution model and the knowledge-specific parameters appropriate for the image segmentation task in question. As such, a complete model is composed of:

1. Connectivity information (the number of shapes, point ordering along contours).
2. The eigenvectors corresponding to the largest t eigenvalues of the covariance matrix describing the ASD.
3. The average border strength for corresponding border segments (if applicable).
4. The average gray level appearance values for each point of the model (if applicable).
5. The average position of the points of the average shape (if applicable).

2.4 Model design and training

A current trend in automatic image interpretation is to use model-based methods. Typically, the models are handcrafted based on the prior knowledge the user has about the object of interest. More recently, automatic model design has emerged as a powerful tool for learning object characteristics. Shape models are especially useful when the object of interest has a homogeneous appearance and can be distinguished from other objects mostly by its shape. One important application for shape-based object recognition is in medical image analysis. During the past decade there has been a lot of work in shape-based approaches for automatic segmentation of flexible

structures [120, 27, 95, 44], statistical tests to differentiate between healthy and sick patients [11] and building anatomical atlases. Among the numerous shape models that have been used, the following approaches are well known: Fourier [120], wavelet [95] and contour (eigen-shape) [27, 11]. However, regardless of the model used, the training data consists of a set of coordinates of some points along the contour of the object of interest from several images. It is usually desirable for a model to describe an *average object* (prototype), to contain information about shape variation within the training set and to be independent of the object pose. In a detailed comparison of Fourier, wavelet and eigen-shape models, Neumann and Lorenz [95] demonstrate that if one does not separate shape information from pose or parametrization information then the resulting model is unable to precisely describe the shape variation present in the training set. That is, the model parameters should be computed *after* the training shapes have been *aligned* in a common coordinate and parametrization frame.

Among the rich computer vision literature dealing with shapes, there are few studies that give a precise definition of the term *alignment*. However, most authors seem to implicitly agree that if $D(.,.)$ is a "distance" function between two sets of points, then a point set B is aligned to a point set A with respect to a transformation group G (e.g., rigid, similarity, linear, affine) if $D(A, B)$ cannot be further decreased by applying to B a transformation from G. The main difference between various alignment approaches is in the distance function used: Huttenlocher et al. [67] use the Hausdorff distance, Sclaroff and Pentland [117] use "strain energy", Ton and Jain [122] use "support functions", and Horn [61], Besl and McKay [8], Gold *et al.*

[52] and the statistical shape community [54] use a least-squares type (Procrustes[14]) distance. Other differences are the types of transformations allowed and whether point correspondences are established during the alignment process. We are not aware of any comparative study that reports alignment results on a common data set for various distance functions and neither of a common quantitative evaluation criterion. Therefore, except for some theoretical assessments, it is difficult to claim that one alignment method dominates another for a given practical problem.

Based on our literature search, there have been few attempts to automate the shape alignment/averaging process in the least-squares framework: Bookstein [11] used thin-plate splines, Hill *et al.* [60] used polygonal matching and Davatzikos *et al.* [32] used curvature registration on outlines produced by an active contour approach. In the thin-plate spline approach, the shape registration-reparametrization is only implicit and not completely automatic. Polygonal matching is based on the assumption that arc path-lengths between consecutive points are equal, which may be violated in case of severe shape differences. As pointed out by several studies, curvature is a rigid invariant of shape and its applicability is limited in case of nonlinear shape distortions. None of these methods attempt to reject a training shape if it is

[14]Procrustes was a villainous son of Poseidon in Greek mythology who robbed travelers on the road from Eulesis to Athens. He offered travelers a room for the night and fit them into his bed by stretching them if they were too short or cutting off their legs if they were too tall. One can regard by analogy one shape as the bed and the other as the person being "translated", "rotated" and "rescaled" so as to fit as close as possible to the bed (Webster dictionary). Procrustes analysis compares the differences in shape between two point sets by transforming one point set in order to match the other. The transformations allowed in a standard analysis are the ***similarity*** (***shape-preserving***) ones: scale changes, rotations and translations. One can regard by analogy one point set as the bed and the other as the person being "translated", "rotated" and "rescaled" so as to fit as close as possible to the bed. After one of the point sets has been transformed to match the other, the sum of squared differences of the coordinates between them is called the ***Procrustes distance***, while the shape instance defined by the average of their coordinates is called ***Procrustes average shape*** [119, 34, 54, 11].

significantly different from the majority in the training set.

2.5 Object matching

In many cases, after an object has been detected and segmented, one needs to classify it into one of several classes. A typical situation is *face recognition*, where an image window centered on a human face has to be assigned an "identity". Usually, a face database is searched and the most "similar" images (according to some distance function) are returned. Recently, the multiclass identity classification problem has successfully been transformed into a set of two-class problems [92]. This paradigm would present the system with two query images and ask if they represent the same person or not. In this way, the recognition task can be transformed into a set of verification subtasks. (In this context, we define the general recognition task as "Tell me what object is this?" and the general verification task as "Tell me if this is an instance of object X?" or "Tell me if these two images represent the same object?").

There is, however, a subtle and often overlooked difference between recognition and verification. This difference is of the same nature as that between *object recognition* and *object detection* we emphasized in Section 2.1; a recognition system usually reports the objects in the database closest to the query object and often it does not have an efficient strategy for rejecting objects not present in the database. This fact was evident in the September 1996 FERET (*Face Recognition Technology*) test [100] where the best performing system attained a 35% correct identification rate at a 10% false alarm rate. Unfortunately, the FERET ROC curves give only an indirect assess-

ment of the goodness of the similarity measures between two face images. In order to be able to evaluate the discriminating power of such a similarity/distance measure, one should compute the distribution of its values when comparing objects from the same class (e.g., two face images of the same person), usually called the "genuine (intra-class) distribution", versus the distribution of the values when comparing objects from different classes (e.g., two face images of different persons), usually called the "imposter (inter-class) distribution". Most verification systems currently make their classification decision based on these two distributions [70], while as far as we know, very few recognition systems have even considered them. Though the genuine and imposter distributions do not appear to us to be class dependent (e.g., the distribution of the distances between several frontal images of the same person should not depend on the identity of the person), the designers of the FERET test have not attempted to include them in their testing methodology [100].

We appreciate that there also is a somewhat "visually semantic" difference between recognition and verification methods. Most of the current verification systems (see, for example, biometric systems [71]) compare only objects with a quite close visual appearance (e.g., a human face with another human face, a fingerprint with another fingerprint, etc.), performing a very fine-scale and precise classification that many times is challenging even for a human. On the other hand, recognition systems tend to perform a rather coarse visual object classification (e.g., Nayar *et al.*'s 100-object recognition system [94] or Sclaroff's fish and tool shape-based recognition system [116]) and almost always they replace, not follow, object detection.

Since we assume that the objects of interest have already been detected and seg-

mented, our work concerns only the detailed aspects of object matching, that is, what we previously defined as the *verification paradigm*. There are broadly two domains where visual object matching systems obeying this paradigm are currently employed: biometric-based personal identification [71] and medical image-based differentiation between healthy and sick patients [11]. All such medical systems that we are aware of are based exclusively on the shape of some brain structures, while the biometric systems are based either on shape [137, 72] or on gray level appearance [31, 92]. There is one recent approach (called "Active Appearance Models" [45]) which explicitly integrates shape with appearance but it follows the "recognition" rather than the "verification" paradigm.

2.6 Summary

In this chapter, we first presented a survey of the techniques currently employed in appearance-based object detection. We have focused on a detailed description of the human face detection problem and we analyzed and compared the performances of six classifiers used for detecting faces. Next, we reviewed and compared several deformable models and we provided a detailed description of the Point Distribution Model. Segmentation results on brain and vehicle images demonstrated the advantages and drawbacks of these approaches. Finally, we discussed how object matching can find the identity of segmented objects and we provided several references to the biometric literature.

Chapter 3

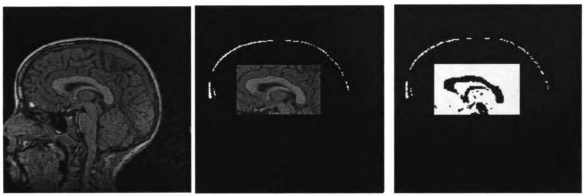
Object detection

Object detection should be the first step towards image understanding. For a long time, one has attempted to segment objects in images without first verifying if the object of interest is really present. We believe that the (many times deformable template-based) segmentation *per se* does not offer enough information (or at least not the right sort of information) to decide the presence of an object. Object detection has become a very active area of research in recent years. This chapter presents the methods we have developed for detecting objects in different applications: namely Corpus Callosum in MR brain images, cars in highway video and left ventricle in MR cardiac images.

3.1 Basic Methods

The easiest way to detect an object is to use a priori knowledge. For example, we know that the Corpus Callosum (CC) is always present at approximately the same

position/orientation with respect to the skull in the MR Images. It is relatively easy [86] to detect the skull and determine a region of interest (Fig. 3.1).



(a) (b) (c)

Figure 3.1: Midsagittal MRI section of the brain. a) Original image. b) A *region of interest* with respect to the skull (white curve) containing the CC. c) Low-level segmentation of the white matter in the ROI in (b).

When no prior information is available about object's position, one has to search for it throughout the image. A popular strategy is to quantize the continuous parameter space (translation, rotation and/or scale) into bins and perform a template matching for each bin. The problem is even more complicated if we do not know how many instances of that object are likely to be present in the image. For example, for the CC segmentation problem we know that there is *exactly* one CC per MR image, while for the car segmentation problem one does not know the number of cars that may be present. In this case one has to threshold the values of the objective function and keep only those positions where the corresponding evaluation function exceeds the above threshold value.

Moreover, sometimes the object of interest may be partially occluded, e.g. the background, right to left oriented cars in Fig. 3.2. Therefore, the matching criterion

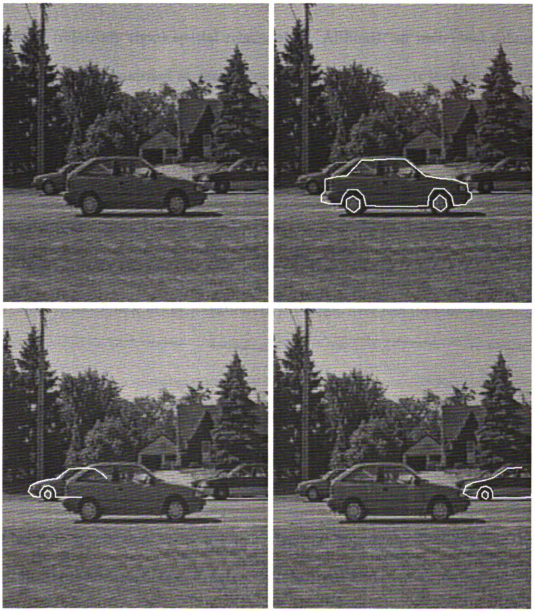


Figure 3.2: Example of automated car detection. a) Original image. b) Optimal left-to-right car model position. c) Optimal right-to-left car model position. d) Second best right-to-left car model position.

should also take into account the fact that only a part of the object may be visible. We approached this problem by splitting the car model shown in Fig. 2.14(b) into four overlapping models which are evaluated locally as if they were separate objects in a fixed (relatively rigid) spatial relationship. Although an individual submodel can be successfully matched at several locations in the image, it is quite likely that a simultaneous matching of three out of four submodels is attained only at locations where the object of interest is present. This is illustrated in Fig. 3.2, where two types of cars (moving left-to-right and right-to-left) were detected, even if the background cars are almost half occluded. Also note that, even if the model represents a sedan car (it was trained only on sedans), it is usually able to detect other types of cars as well (like the hatchback in the foreground), although with less confidence.

3.2 Maximum discrimination-based object detection

This section details one of the methods, namely "maximum discrimination-based loglikelihood ratio classification", currently (see Table 2.2) used for object detection. Though the main idea on which the classification is based has been employed for some time in image registration and parameter estimation, it has only recently [22] been adapted to object detection. We advocate this approach especially because it is by far one of the fastest: to classify an n feature pattern one has to perform n additions (decision trees may actually be faster but the decision boundaries they can represent

are much more restrictive).

3.2.1 Introduction

We propose to modify and adapt the Maximum Discrimination method for left ventricle detection in short axis cardiac MR images. There has been a substantial amount of recent work in studying the dynamic behavior of the human heart using non-invasive techniques such as magnetic resonance imaging [51, 130]. Among the domain specific methods for ventricle detection in cardiac images, one can mention Chiu and Razi's multiresolution approach for segmenting echocardiograms [20], Bosch *et al.*'s dynamic programming based approach [14], and Weng *et al.*'s algorithm based on learning an adaptive threshold and region properties [130]. In order to provide useful diagnostic information, a cardiac imaging system should perform several tasks such as segmentation of heart chambers, identification of endocardium and epicardium, measurement of the ventricular volume over different stages of the cardiac cycle, measurement of the ventricular wall motion, etc. Most approaches to segmentation and tracking of heart ventricles are based on deformable templates, which require specification of a good initial position of the boundary of interest. This is often provided manually, which is both time consuming and requires a trained operator.

Our goal is to automatically provide the approximate scale/position (given by a tight bounding box) of the left ventricle in 2-D cardiac MR images. This information is needed by most deformable template segmentation algorithms which require that a region of interest be provided by the user. This detection problem is difficult because

of the variations in shape, scale, position and gray level appearance exhibited by the cardiac images across different slice positions, time instants, patients and imaging devices (see Fig. 3.3).

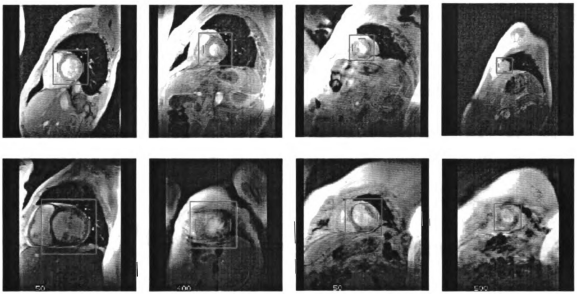
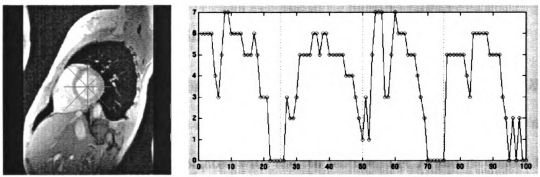


Figure 3.3: Several examples of 256 × 256 gradient echo cardiac MR images (short axis view) showing the left ventricle variations as a function of acquisition time, slice position, patient and imaging device. The left ventricle is the bright area inside the square. The four markers show the ventricle walls (two concentric circles).

Our proposed method differs from the maximum discrimination approach used by Colmenarez and Huang for face detection in two significant ways:

1. Definition of the instance space. In [22] the instance space was defined as the set of 2-bit 11 × 11 non-equalized images of human faces. In our case, the ventricle diameter ranges from 20 to 100 pixels and a drastic subsampling of the image would loose the ventricle wall (the dark ring). On the other hand, even a 20 × 20 window would generate 400 features and the system would be too slow. Therefore, we used only four profiles passing through the ventricle (see Fig. 3.4) subsampled to define a total of 100 features.

2. Solution to the optimization problem. An approximate solution to a Traveling salesman type problem is obtained in [22] using a minimum spanning tree algorithm. Since the quality of the solution is crucial for the performance of the learning algorithm, we believe simulated annealing to be a better choice for our optimization problem.



(a) (b)

Figure 3.4: The feature set defining a heart ventricle. a) The four cross sections through the ventricle and its immediate surroundings used to extract the features. b) The 100-element normalized feature vector associated with the ventricle in (a).

3.2.2 Mathematical model

In order to learn a *pattern*, one should first specify the instance (feature) space from which the pattern examples are drawn. Since the left ventricle appears as a relatively symmetric object with no elaborate texture, it was not necessary to define the heart ventricle as the entire region surrounding it (the red squares in Fig. 3.3). Instead, it was sufficient to sample four cross sections through the ventricle and its immediate neighborhood, along the four main directions (Fig. 3.4(a)). Each of the four linear cross sections was subsampled as to contain 25 points and the gray values were normalized in the range 0-7. The normalization scheme used here is a piece-wise linear

transformation that maps the average gray level of all the pixels in the cross sections to a value 3, the minimum gray level is mapped to a value 0 and the maximum gray value is mapped to 7. In this way, a heart ventricle is defined as a feature vector $x = (x_1, ..., x_{100})$, where $x_i \in 0..7$ (Fig. 3.4(b)). We denote by Ω the instance space of all such vectors.

Markov Chain-based discrimination

We regard an n-dimensional observation as the realization of a random process $X = \{X_1, X_2, .., X_n\}$, where n is the number of features defining the object of interest and X_i is the random variable associated with the ith feature. Let P and N denote the two class conditional probabilities over the feature space Ω:

$P(\mathbf{x})$ = Prob($\mathbf{X}$=$\mathbf{x}$ | $\mathbf{x}$ is an object example) and

$N(\mathbf{x})$ = Prob($\mathbf{X}$=$\mathbf{x}$ | $\mathbf{x}$ is a non object example).

For each instance $\mathbf{x} \in \Omega$, we define its log-likelihood ratio $L(\mathbf{x}) = log\frac{P(\mathbf{x})}{N(\mathbf{x})}$. Note that $L(\mathbf{x}) > 0$ if and only if $\mathbf{x}$ is more probable to be a left ventricle than a non left ventricle, while $L(\mathbf{x}) < 0$ if the converse is true.

The Kullback divergence between P and N can be regarded as the average of the log-likelihood ratio over the entire instance space [55]:

$$H_{P||N} = \sum_{\mathbf{x}\in\Omega} P(\mathbf{x}) log \frac{P(\mathbf{x})}{N(\mathbf{x})}. \tag{3.1}$$

It has been shown that the Kullback divergence is not a distance metric. However,

it is generally assumed that the larger $H_{P||N}$ is, the better one can discriminate between observations from the two classes whose distributions are P and N. It is not computationally feasible to estimate P and N taking into account all the dependencies between the features. On the other hand, assuming a complete independence of the features is not realistic because of the mismatch between the model and the data. A compromise is to consider the random process X to be a Markov chain, which can model the dependency in the data with a reasonable amount of computation.

Let us denote by S the set of feature sites with an arbitrary ordering $\{s_1, s_2, .., s_n\}$ of sites $\{1, 2, .., n\}$. Denote by $X_S = \{X_{s_1}, ..., X_{s_n}\}$ an ordering of the random variables that compose X corresponding to the site ordering $\{s_1, s_2, .., s_n\}$. If X_S is considered to be a first-order Markov chain then for $\mathbf{x} = (x_1, x_2, .., x_n) \in \Omega$ one has:

$$P(X_S = \mathbf{x}) = P(X_{s_1} = x_1, ..., X_{s_n} = x_n) = P(X_{s_n} = x_n | X_{s_{n-1}} = x_{n-1}) \times ... \times$$

$$\times P(X_{s_2} = x_2 | X_{s_1} = x_1) \times P(X_{s_1} = x_1).$$

Therefore, the log-likelihood ratio of the two distributions P and N under the Markov chain assumption can be written as follows:

$$L^S(\mathbf{x}) = log\frac{P(X_S=\mathbf{x})}{N(X_S=\mathbf{x})} = log(\frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)} \prod_{i=2}^{n} \frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}) =$$

$$= \sum_{i=2}^{n} log\frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})} + log\frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)}$$

$$= L^{s_1}(x_1) + \sum_{i=2}^{n} L^{s_i||s_{i-1}}(x_i, x_{i-1}). \qquad (3.2)$$

The Kullback divergence of the two distributions P and N under the Markov chain assumption can be computed as follows:

$$H^S_{P||N} = H_{P||N}(X_{s_1}, ..., X_{s_n}) =$$

$$= \sum_{(x_1,..,x_n)\in\Omega} P(X_{s_1} = x_1, ..., X_{s_n} = x_n) log \frac{P(X_{s_1}=x_1,...,X_{s_n}=x_n)}{N(X_{s_1}=x_1,...,X_{s_n}=x_n)}$$

$$= \sum_{(x_1,..,x_n)\in\Omega} P(X_{s_1} = x_1, ..., X_{s_n} = x_n) log(\frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)} \prod_{i=2}^{n} \frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}) =$$

$$= \sum_{(x_1,..,x_n)\in\Omega} P(X_{s_1} = x_1, .., X_{s_n} = x_n) log \frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})} +$$

$$\sum_{(x_1,..,x_n)\in\Omega} P(X_{s_1} = x_1, ..., X_{s_n} = x_n) log \frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)} =$$

$$= \sum_{i=2}^{n} (\sum_{(x_i,x_{i-1})} P(X_{s_i} = x_i, X_{s_{i-1}} = x_{i-1}) log \frac{P(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}{N(X_{s_i}=x_i|X_{s_{i-1}}=x_{i-1})}) +$$

$$\sum_{x_1} P(X_{s_1} = x_1) log \frac{P(X_{s_1}=x_1)}{N(X_{s_1}=x_1)} = H_{P||N}(X_{s_1}) + \sum_{i=2}^{n} H_{P||N}(X_{s_i}||X_{s_{i-1}}). \quad (3.3)$$

3.2.3 Most discriminant Markov chain

One can note that the divergence $H^S_{P||N}$ defined in Eq.(3.3) depends on the site ordering $\{s_1, s_2, .., s_n\}$ because each ordering produces a different Markov chain with a different distribution. The goal of the learning procedure is to find a site ordering S^* that maximizes $H^S_{P||N}$ which will result in the best discrimination between the two classes. The resulting optimization problem, although related to, is more difficult than the Traveling salesman problem since:

1. It is asymmetric (the conditional Kullback divergence is not symmetric, i.e. $H_{P||N}(X_{s_i}||X_{s_{i-1}}) \neq H_{P||N}(X_{s_{i-1}}||X_{s_i})$).

2. The salesman does not complete the tour, but remains in the last town.

3. The salesman starts from the first town with a handicap ($H_{P||N}(X_{s_1})$) which depends only on the starting point.

Therefore, the instance space of this problem is of the order of $n \times n!$, where n is the number of towns (feature sites), since for each town permutation one has n starting possibilities. It is well known that this type of problem is *NP-complete* and cannot be solved by brute-force except for a very small number of sites. Although for the symmetric Traveling salesman problem there exist strategies to find both exact and approximate solutions in a reasonable amount of time, we are not aware of any heuristic for solving the asymmetric problem involved here. However, a good approximate solution can be obtained using simulated annealing [1]. Even though there is no guarantee that an optimal solution will be found, in practice, simulated annealing does almost always find a solution which is very close to the optimal (see also the discussion in [1]). Comparing the results produced by the simulated annealing algorithm on a large number of trials with the optimal solutions (for small size problems), we found that all the solutions produced by simulated annealing were within 5% of the optimal solutions.

Once S^* is found, one can compute and store tables with the log-likelihood ratios such that, given a new observation, its log-likelihood can be obtained from $n - 1$ additions using Eq. (3.2).

The learning stage, which is described in *Algorithm 3.1*, starts by estimating the distributions P and N and the parameters of the Markov chains associated with *all* possible site permutations using the available training examples. Next, the site ordering that maximizes the Kullback distance between P and N is found using simulated annealing (*Algorithm 3.2*), and the log-likelihood ratios induced by this ordering are computed and stored.

Algorithm 3.1: Finding the most discriminating Markov Chain

• Given a set of positive/negative training examples (as preprocessed n-dimensional feature vectors).

1. For each feature site s_i, estimate $P(X_{s_i} = v)$ and $N(X_{s_i} = v)$ for $v = 0..GL-1$ (GL = number of gray levels) and compute the divergence $H_{P||N}(X_{s_i})$.

2. For each site pair (s_i,s_j), estimate $P(X_{s_i} = v_1, X_{s_j} = v_2)$, $N(X_{s_i} = v_1, X_{s_j} = v_2), P(X_{s_i} = v_1|X_{s_j} = v_2)$ and $N(X_{s_i} = v_1|X_{s_j} = v_2)$ for $v1, v2 \in 0..GL-1$ and compute $H_{P||N}(X_{s_i}||X_{s_j}) = \sum_{v_1,v_2=0}^{GL-1} P_X(X_{s_i} = v_1, X_{s_j} = v_2) ln \frac{P_X(X_{s_i}=v_1|X_{s_j}=v_2)}{N_X(X_{s_i}=v_1|X_{s_j}=v_2)}$.

3. Solve a traveling salesman type problem over the sites S to find $S^* = \{s_1^*, ..., s_n^*\}$ that maximizes $H_{P||N}(X_S)$.

4. Compute and store $L(X_{s_1^*} = v) = ln \frac{P(X_{s_1^*}=v)}{N(X_{s_1^*}=v)}$ and $L(X_{s_i^*} = v_1||X_{s_{i-1}^*} = v_2) = ln \frac{P(X_{s_i^*}=v1|X_{s_{i-1}^*}=v2)}{N(X_{s_i^*}=v1|X_{s_{i-1}^*}=v2)}$ for $v, v_1, v_2 \in \{0..GL-1\}$.

Algorithm 3.2: Maximizing $H_{P||N}$ using Simulated Annealing

(We are actually minimizing $-H_{P||N}$)

- Input: The sets of parameters for the positive and negative Markov chain models.
- Output: A permutation S^* of the sites $S = \{1..n\}$ that maximizes $H_{P||N}(S^*)$.

For $i = 1$ to n do

Initialize S_i^* with the cyclic permutation $\{i, i+1, ...n, 1, ..., i-1\}$.

Compute the Kullback distance $H_{P||N}(X_{S_i^*})$ (system energy) associated with S_i^*.

For $Temp = 2$ to 0.0005 do

For $l = 1$ to N_{iter} do

Randomly select two non-consecutive sites p and q different from $s_{i,1}^*$.

Let S_i^{**} be the permutation obtained from S_i^* by interchanging p and q.

Set $\Delta M = H_{P||N}(X_{S_i^{**}}) - H_{P||N}(X_{S_i^*})$.

If $\Delta M < 0$, accept the transition from S_i^* to S_i^{**} (set $S_i^* = S_i^{**}$);

otherwise accept this transition only with probability $e^{-(\Delta M)/Temp)}$;

end

$Temp = Temp * 0.97$

end

end

Choose S^* as the permutation S_i^{**} that minimizes the energy over the n starting sites.

3.2.4 Classification procedure

The detection (testing) stage consists of scanning the test image at different scales with a constant size window from which a feature vector is extracted and classified. The classification procedure using the most discriminant Markov chain, detailed in Algorithm 3.3, is very simple: the log-likelihood ratio for a window is computed as the sum of conditional log-likelihood ratios associated with the Markov chain ordering (Eq.(3.2)). The total number of additions used is at most equal to the number of features.

Algorithm 3.3: Classification

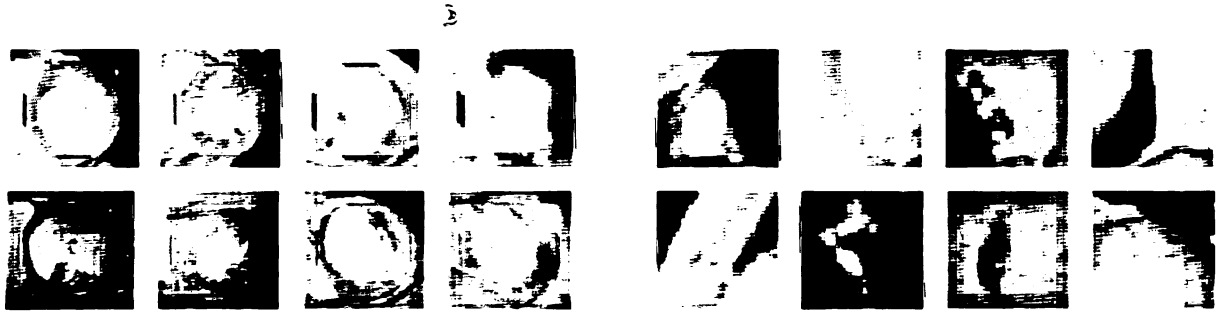
- Given S^*, the best Markov chain structure and the learned likelihoods $L(X_{s_1^*} = v)$ and $L(X_{s_i^*} = v_1 || X_{s_{i-1}^*} = v_2)$.
- Given a test example $O = (o_1, ...o_n)$ (as preprocessed n-dimensional feature vector).

1. Compute the likelihood $L_O = L(X_{s_1^*} = o_{s_1^*}) + \sum_{i=2}^{n} L(X_{s_i^*} = o_{s_i^*} || X_{s_{i-1}^*} = o_{s_{i-1}^*})$.
2. If $L_O > T$ then classify O as left ventricle else classify it as non left ventricle.

Here T is a threshold to be learned from the ROC curve of the training set depending on the desired (correct detection - false alarm) trade-off.

3.2.5 Experimental results

A collection of 1,350 MR cardiac images from 14 patients was used to test our system. The images were acquired using a Siemens Magnetom MRI system. For each patient, a number of slices (4 to 10) were acquired at different time instances (5 to 15) of the heart beat, thus producing a matrix of $2D$ images (in Fig. 3.8, slices are shown vertically and time instances are shown horizontally). As the heart is beating, the left ventricle is changing its size, but the scale factor between the end of diastolic and the end of systolic periods is negligible compared to the scale factor between slices at the base and the apex of the heart.



(a) (b)

Figure 3.5: Training examples for the left ventricle detection problem. a) Positive examples. b) Negative examples.

In each image, a tight bounding box (defined by the center coordinates and scale) containing the left ventricle was manually identified (Fig. 3.5(a)). From each cardiac image, 75 positive examples were produced by translating the manually defined box up to 2 pixels in each coordinate and scaling it up or down 5%. In this way, a total of 101,250 ventricle patterns (positive examples) were generated. We also produced a total of 179,603 non-ventricle patterns (negative examples) by uniformly subsampling a subset of the 1,350 available images at 8 different scales. Several non left ventricle

patterns are shown in Fig. 3.5(b).

We trained and tested our algorithm using two-way cross-validation. The 14 subjects were randomly divided into two subsets of 7 subjects each such that the total number of images in the two subsets were approximately equal. The distributions of the log-likelihood values for the sets of positive and negative examples over the two cross-validation trials are shown in Fig. 3.6. They are very well separated, and by setting the decision threshold at 0, the resubstitution detection rate is 97.6% with a false alarm rate of 3.8%. It is also interesting to note how the representation of positive and negative examples look after their features have been reordered according to the most discriminant Markov chain (Fig. 3.7). Although the positive examples are not well registered when extracted from the data, after feature reordering they look pretty much aligned.

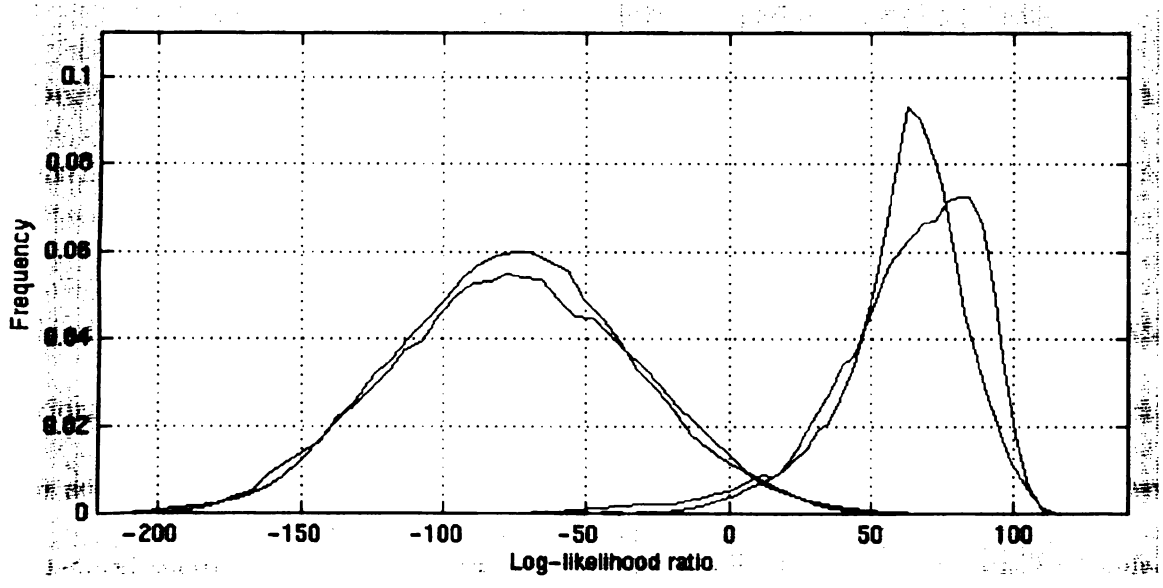
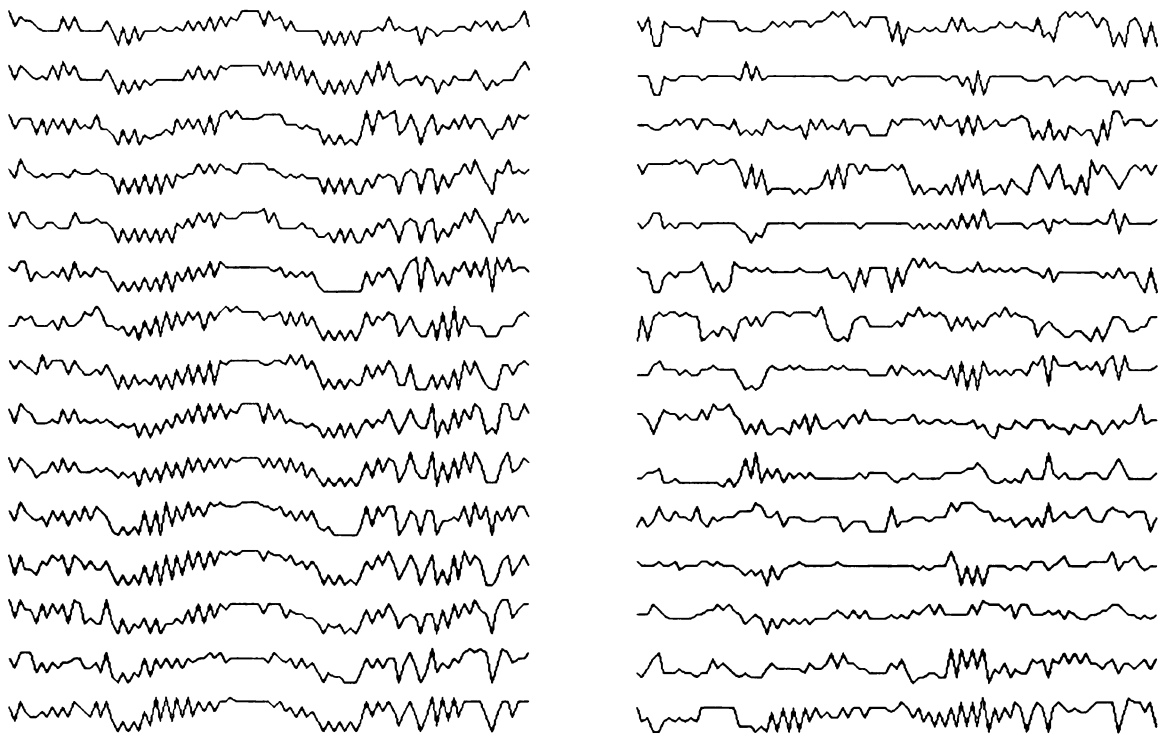


Figure 3.6: The distribution of the log-likelihood ratio for left ventricle (red) and non left ventricle (blue) examples computed for the two cross-validation trials. For a decision threshold set at zero, 2,460 (2.4%) of the 101,250 positive examples and 6,854 (3.8%) of the 179,603 negative examples are misclassified.



(a) (b)

Figure 3.7: Feature reordering induced by the most discriminant Markov chain (a) Positive examples. (b) Negative examples.

During testing, each image was subsampled at 8 different scales and scanned with a constant 25×25 pixel window using a step of 2 pixels along rows and columns. This means that, at each scale, a number of windows equal to a quarter of the number of pixels of the image at that scale was used for feature extraction and classification. All positions that produced a positive *log-likelihood ratio* were classified as left ventricles. Since several neighboring positions might have been classified as left ventricle, we partitioned them into clusters (a cluster was considered to be a set of image positions classified as left ventricles that had a distance smaller than 25 pixels to its centroid). At each scale, only the cluster centroids were reported, together with the *log-likelihood ratio* value for that cluster (a weighted average of the *log-likelihood ratio* values in the

cluster).

It was not possible to choose the best scale/position combination based on the *log-likelihood* value of a cluster. That is, values of the *log-likelihood* criterion obtained at different scales are *not comparable*: in about 25% of the cases, the largest *log-likelihood* value failed to represent the real scale/position combination. Therefore, we report *all* cluster positions generated at different scales (an average of 7 clusters are generated per image by combining all responses at different scales). Even if we could not obtain a single scale/position combination per image using this method, the real combination was among those 7 clusters reported in 95.5% of the cases. Moreover, the 4.5% failure cases came only from the bottom most slice, where the left ventricle is very small (15-20 pixels in diameter) and looks like a homogeneous grey disk. We suspect that these situations were rarely encountered in the training set, so they could not be learned very well. The quantitative results of the detection task are summarized in Table 3.1. The false alarm rate has been greatly reduced by reporting only cluster centroids.

We could select the best hypothesis by performing a consistency check along all the images that represent the same slice: our prior knowledge states that, in time, one heart slice does not modify its scale/position too much, while consecutive spatial slices tend to be smaller. By enforcing these conditions, we could obtain complete spatio-temporal hypotheses about the heart location. A typical detection result on a complete spatio-temporal (8 slice positions, 15 sampling times) sequence of one patient is shown in Fig. 3.8).

Table 3.1: Performance of the left ventricle detection algorithm.

Resubstitution detection rate	97.6%
Resubstitution false alarm rate	3.8%
Test set detection rate	95.5%
Test set false alarms per image	6
Test set false alarm rate/windows analyzed	0.03%
Detection time/image (Sun Ultra 10)	2 sec

3.3 Summary

We addressed the problem of object detection in black and white images. Simple techniques based on prior knowledge and/or shape cues were shown to produce a good localization of the object of interest in some particular cases. For situations where the object of interest is mostly described by texture properties and the test image may contain several/none instances of it, we have developed an appearance based detection method. We modified a fast classifier, previously employed in the speech processing domain, and demonstrated it on left ventricle detection in MR cardiac images. A detailed formulation of the mathematical model (first order Markov chain) as well as the algorithms required for deriving the chain order that best discriminates positive examples from negative examples have been provided. Finally, results on a spatio-temporal MR cardiac study are shown and discussed.

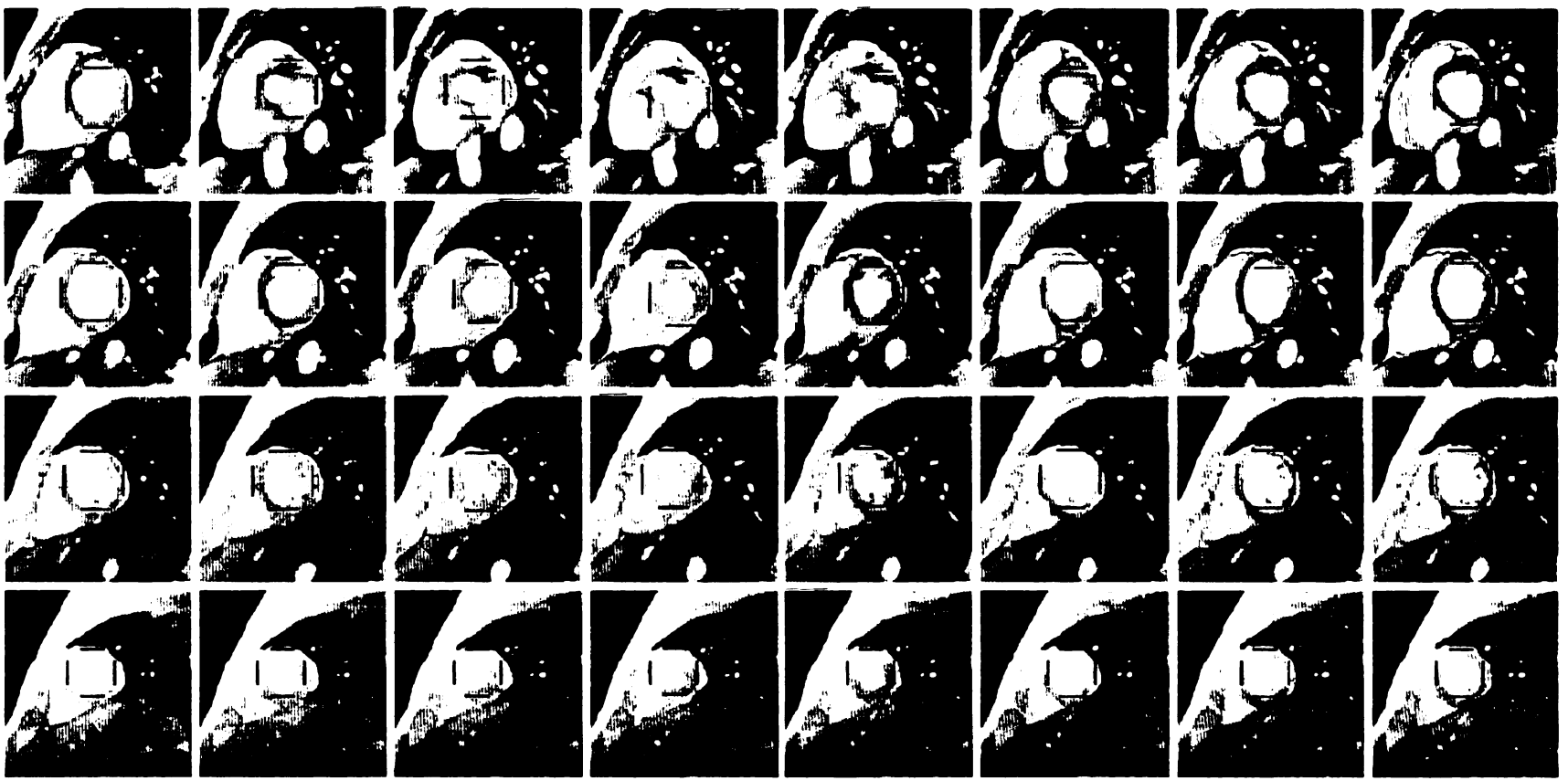


Fig. 3.8. Results of the detection algorithm on a complete spatio-temporal study

Chapter 4

Learning 2D shape models

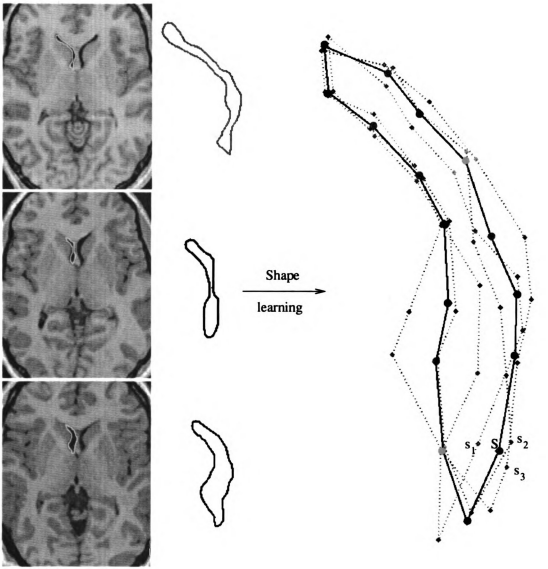
This chapter describes a novel approach towards automatic design of 2D shape models. Our method is based on clustering a set of training shapes in the original shape space defined by the coordinates of the contour points and performing a Procrustes analysis [11, 54, 34] on each cluster to obtain cluster prototypes (average objects) and statistical information about intra-cluster shape variation. The main difference from previously reported methods is that the training set is first automatically clustered and those shapes considered to be outliers are discarded (Section 4.4). In this way, the cluster prototypes are not distorted by outlier shapes. The second difference is in the manner in which registered sets of points are extracted from each shape contour. We propose a flexible point matching technique (Section 4.3) that takes into account both pose/scale differences as well as non-linear shape differences between a pair of objects. The matching method is independent of the initial relative position/scale of the two objects and does not require any manually tuned parameters. A quantitative analysis of our shape registration approach within the main cluster of each object

(brain structure), demonstrated results that compare very well to those achieved by manual registration; achieving an average rms error of about 1 pixel (Section 4.5). Our approach can serve as a fully automated substitute to the tedious and time-consuming manual shape registration and analysis.

4.1 Problem specification

We will concentrate on learning $2D$ shape models: given a number of 2D shapes, a model consists of a shape prototype along with statistical information about shape variation around the prototype (Fig. 4.1). In some cases, more than one prototype may be necessary to capture the variability in the training set.

We use a least-squares type (Procrustes) distance between two shapes whose choice was motivated by the following facts: (i) it provides a convenient way to compute a prototype (average shape) from a set of simultaneously aligned shapes (Procrustes analysis [34]), (ii) once the point correspondences are found, there exists an analytical (exact) solution to the alignment problem [61, 26] and (iii) it has been quite frequently used in medical image analysis [11, 25, 26, 52]. Unfortunately, least-squares alignment methods do not deal with parametrization, and can be applied only to sets of *corresponding* points. In practice, such sets of points have been obtained by a painstakingly manual inspection of the data of interest. This fact is illustrated in Fig. 4.2 where an expert defined the points (called pseudo-landmarks) that characterize each of the 3 ventricle shapes in Fig. 4.1. When these points are defined independently for each shape, it may be very difficult to exactly define point correspondences in the absence



(a) (b)

Figure 4.1: Learning the shape of the right ventricle from MR brain images. a) Manual tracing of the right ventricle performed by a neuroanatomist on three different patients. b) A ventricle model consists of a shape prototype (drawn in black) along with statistical information about shape variation. The prototype vertices (drawn as colored circles) have been obtained by averaging the coordinates of the corresponding vertices on the three ventricles (drawn as colored diamonds) *after they have been aligned in a common coordinate frame* (e.g., vertex s is the average of s_1, s_2 and s_3). The three aligned ventricle shapes are shown in dotted red, green and blue lines. This method for obtaining a shape prototype is called *Procrustes analysis* [11, 54, 34]. Note that, in order for this method to work one needs to extract sets of corresponding points of equal cardinality (in this case 16) from the three ventricle shapes. The shape variance is given by the 32×32 covariance matrix of the (x, y) coordinates of the vertices on the three ventricle shapes after alignment.

of anatomical landmarks (points for which correspondences can be defined based on prior knowledge). Other problems are human bias and lack of reproducibility, that is, different experts may extract different numbers of pseudo-landmarks and even different point correspondences.

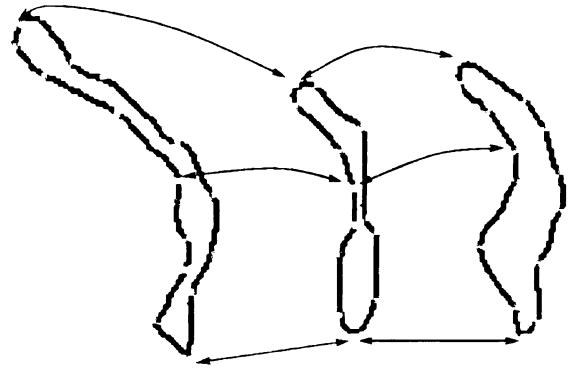


Figure 4.2: Expert-defined pseudo-landmarks (yellow squares) on the three shapes in Fig. 4.1 along with some obvious point correspondences (red arrows). Note that if the pseudo-landmarks are defined on each shape independently (as it was the case here) then, in most cases, it is very difficult to find corresponding points on other shapes.

We present an alternate solution to the problem of shape reparametrization-alignment-averaging problem. One difference from most previous methods is that it does not necessarily compute a single average shape from the given training set, but it rather detects shape clusters in the data and provides a shape average and *variation* for each cluster. The idea of detecting shape clusters in the training set has been employed before by Cootes and Taylor [26] and by Gold *et al.* [53]. However, in [26] the clusters were not detected in the initial shape space but in the parameter space obtained by Principal Component Analysis, while in [53] clustering is formulated as a complex optimization problem whose solution is computationally expensive.

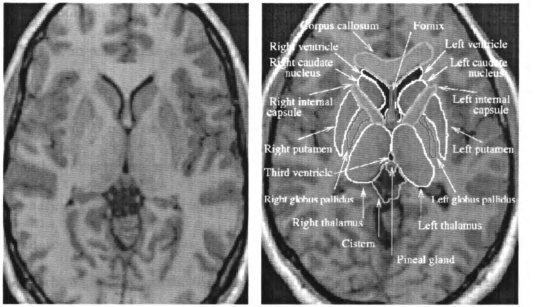
Mathematically speaking, we are attempting to solve the following problem: Given

a set of m shape instances $S_k = \{(x_i^k, y_i^k)\}_{i=1..n_k}^{k=1..m}$ represented by a set of boundary points (shape S_k represented by n_k boundary points), partition it into a set of clusters and, for each shape cluster, compute a *prototype* (mean shape). The set of prototypes will be used as models for detection of object instances in new images by means of deformable template matching. Working with average templates learned from examples results in a faster and more reliable segmentation. As a direct application of our shape learning method, we use it to learn the shapes of 11 different structures in MR brain images and demonstrate how the learnt shapes facilitate subsequent image segmentation. Figure 4.3 shows an example of an MR brain image taken in the coronal plane. Several structures of interest whose contours were identified by a neuroanatomist are shown in Figure 4.3(b). Note that parts of the contours of some structures cannot be distinguished from the background based on gray level information alone. In such cases, only a model-based segmentation that uses prior knowledge about the shape of a structure of interest is able to identify the structures [44].

4.2 Shape learning method

In this section we first provide a short introduction to the shape related terminology that will be used throughout the chapter. Then, we present an outline of the proposed shape learning method and the ideas that stand behind it.

An *n-point* shape instance $B = \{s_i^B\}_{i=1..n} = \{(x_i^B, y_i^B)\}_{i=1..n}$ is said to be *aligned* to $A = \{(x_i^A, y_i^A)\}_{i=1..n}$ if the *sum-of-squares* $SS(A, B) =$



(a) (b)

Figure 4.3: Magnetic resonance image of the human brain, imaged in the coronal plane with in-slice resolution of 256×256 pixels. a) Original image (cropped to show only the brain region). b) Structures of interest whose contours were identified by a neuroanatomist.

$\sum_{i=1}^{n} \left[\left(x_i^A - x_i^B \right)^2 + \left(y_i^A - y_i^B \right)^2 \right]$ cannot be decreased by scaling, rotating or translating B. In this case, the quantity $SS(A,B)/n$ is called the *Mean Alignment Error* ($MAE(A,B)$). Practical algorithms for shape alignment can be found in [26, 34]. In general, the alignment procedure is not symmetric and, if $|A| = |B| \leq 2$, A and B can be aligned exactly (the alignment error is 0).

Let $A = \{(x_j^A, y_j^A)\}_{j=1..p}$ and $B = \{(x_k^B, y_k^B)\}_{k=1..r}$ be two shape instances defined by p and r contour points, respectively. A *match matrix* $M = \{M_{j,k}\}_{k=1..r}^{j=1..p}$ [52] is defined by:

$$M_{j,k} = \begin{cases} 1 & \text{if point } A_j \text{ corresponds to point } B_k \\ 0 & \text{otherwise.} \end{cases}$$

We consider 0-1 match matrices M corresponding to symmetric one-to-one links

(point correspondences); that is, a point $A_j \in A$ can have at most one corresponding point $B_k \in B$, in which case the correspondence is symmetric. The points from both sets that have no correspondence are called *outliers*. We denote by A_M and B_M the subsets of A and B matched by M and by $MAE(M) = MAE(A_M, B_M)$. Note that the definition of a match matrix M *does not assume the existence of a similarity transformation that would overlay the subsets* A_M *and* B_M *exactly* (this would be equivalent to $MAE(A_M, B_M) = 0$). However, in practice it makes sense to use a match matrix M only when the distances between corresponding points matched by M are reasonably small. For a discussion of this issue, see Section 4.6.

The *Procrustes average* of a set of shapes $\{A_k\}_{k=1..m}$ is a shape instance near the center of the empirical distribution of A_k's in the shape space defined by the coordinates of the contour points. The computation of Procrustes average is graphically illustrated in Fig. 4.1(b). From the three ventricle shapes (dense sets of $2D$ points), sparser subsets of corresponding points (in this case 16 points) were extracted. These subsets are aligned into a common coordinate frame and their coordinates are averaged in order to obtain the Procrustes average shape. For a detailed definition, properties and methods of computing an average shape, see [11, 26, 34].

The outline of our shape learning method is as follows (see also Fig. 4.4):

Algorithm 4.1: Shape Learning Outline

Input: A set of m shapes instances S_1,...,S_m, each represented by a sequence of boundary points.

1. *Polygonal approximation*: For each shape S_k in the training set, compute a polygonal approximation S'_k.

2. *Global and local similarity registration*: For each $j, k = 1..m$, perform a flexible one-to-one registration (mapping) of S'_k to S_j. If the registration succeeds, define $T_{j,k}$ as the subset of S_j that corresponds (was matched) to the points of S'_k, otherwise set $T_{j,k} = \emptyset$.

3. *Inter-shape distance matrix computation*: Compute a *pairwise mean alignment error* matrix $\mathcal{D} = \{d_{j,k}\}_{j,k=1..m}$, where $d_{j,k} = MAE(T_{j,k}, S'_k)$ if $T_{j,k} \neq \emptyset$ or $d_{j,k} = \infty$, otherwise.

4. *Shape clustering and prototype computation*: Set the current training set equal to the original set of m shapes: $CTS = \{S_k\}_{k=1..m}$. While $CTS \neq \emptyset$ do

   (a) Find the shape approximation S'_{i_0} that has the least average distance to the shapes $S_j \in CTS$ (the *best fit shape* to the current training set).

   (b) Extract from CTS and put in a cluster all the shapes $S_{i_1},..,S_{i_p}$ to which S'_{i_0} could be fit (see Sections 4.4 and 4.6.3).

   (c) The cluster prototype is defined as the *Procrustes average* of $T_{i_1,i_0},..., T_{i_p,i_0}$. The shape variance inside the cluster is defined as the covariance matrix of the aligned sets $\{T_{i_k,i_0}\}_{k=1..p}$. The size of the covariance matrix is equal to twice the number of points in S'_{i_0}.

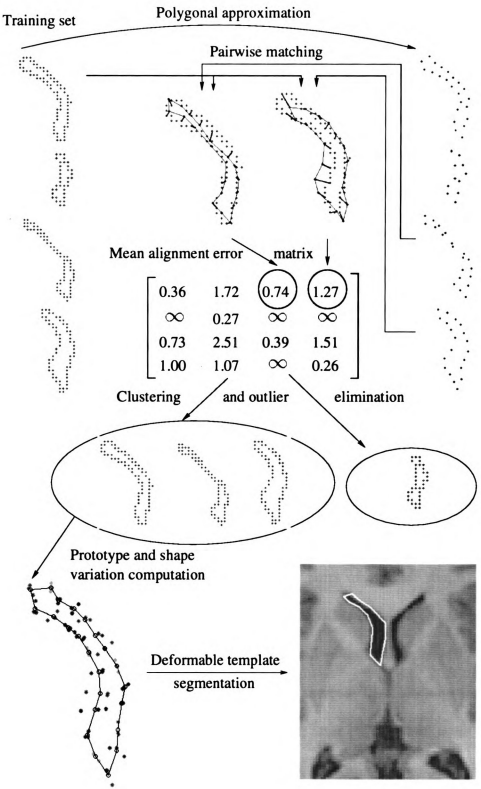


Figure 4.4: The shape learning method (Algorithm 4.1).

Step 1 of the learning algorithm finds a polygonal approximation of each shape using a method described in [47]. The distance between consecutive vertices of the polygonal approximations is about 2-3 pixels in order to smooth small shape artifacts, noise and quantization effects. The polygonal approximation is only used to extract subsets of *corresponding* points from the *original* shapes. This makes the registration task *easier* and *implicitly* brings together the extracted subsets into a *common parametrization frame*. Indeed, if a point s_{i_0} on a polygonal approximation S' is registered to $s_{i_1} \in S_1$, $s_{i_2} \in S_2$, ..., $s_{i_m} \in S_m$ ($S_1, ..., S_m$ are original shapes that form a cluster) then, by transitivity, $s_{i_1}, s_{i_2}, .., s_{i_m}$ are correspondents on $S_1, .., S_m$ of *one vertex* of an average shape (the idea of registration by transitivity was previously used by Sclaroff and Pentland [117], though in a different context). For example, in Fig. 4.1(b), s_1, s_2 and s_3 are the corresponding vertices on the three example shapes of the prototype vertex s. It is not advisable to register two polygonal approximations because one can dilute information about the original point variation and encounter contention problems caused by the one-to-one mapping requirement. On the other hand, one should not attempt to directly register pairs of original shapes since it is more difficult to define and register local topological neighborhoods (see section 4.3) because of the local noise (jagged contours) and point contention. By using a smoother and sparser approximation of a shape, we increase the likelihood that *every* point on it will eventually have a correspondent. If, after registration, there exists a point on S'_k that does not have a correspondent on S_j then we say that the registration between S'_k and S_j has failed and we set $d_{j,k} = \infty$.

4.3 Shape registration

In order to ensure that the shape variation present in the original data is *preserved*, one needs a precise automatic registration method. Hill et al. [59] reported that the mean-square-error (from a *ground truth*) of their registration method is about twice as large as the error in manual registration. They also reported that other methods were even less robust. Therefore, we have decided to combine several ideas from the literature [52, 48, 57, 117, 46] with new ideas in order to obtain a more precise registration method.

Our shape registration method consists of two stages: (i) global similarity registration of two arbitrary sets of points and (ii) non-linear registration based on local similarity of two curves (ordered sets of points).

The global registration method attempts to find a similarity transformation corresponding to a one-to-one mapping of a subset A' of a shape instance A onto a subset B' of a shape instance B. (Ideally, *the entire shape instance* A should be mapped one-to-one by a similarity transformation onto *the entire shape instance* B, but most of the time the number of vertices on the two shapes are different, and moreover, some vertices on one shape have no corresponding vertex on the other shape). This mapping is required to simultaneously fulfill two contradicting requirements: (i) the size n of the matched subsets is as large as possible, and (ii) the *mean-alignment-error* between A' and B' is as small as possible.

There is a plethora of point-based registration methods, see [87] for a recent survey. As noted in [52], when solving for best alignment transformation and its

associated match matrix (set of point correspondences), one has to search one of the two complementary spaces: either the space of correspondence functions which is finite but exponential in the number of points from the two sets, e.g., Softassign [52] or the space of similarity transformations which is infinite, and can be only partially explored, e.g., ICP [8] or Chamfer Matching [13]. We propose a registration procedure (Algorithm 4.2) based on a polynomial quasi-exhaustive exploration of the correspondence functions (match matrices) space. Its main novelty compared to techniques previously used in the literature [48, 124, 57, 46, 52] is the way it resolves the *shrinking* effect [46]: an unconstrained linear registration of two sets of points tends to "shrink" one set with respect to the other since, theoretically, the "best" alignment is obtained when one point set is rescaled to become a single point. Our problem formulation requires a small MAE between the two chosen subsets, using as many point correspondences as possible. Unfortunately, if we have less than 3 correspondences, the MAE is 0 and this should be compensated for. Therefore, we want to explicitly specify in the search criterion that a $q\%$ increase in MAE with a $p\%$ increase in the number of correspondences is accepted as long as no individual distance between a pair of corresponding points exceeds a given threshold. One of the simplest functionals that captures this trade-off is the ratio between a compensated MAE and the number of correspondences:

$$f(M) = [MAE(M) + K]/n, \tag{4.1}$$

where K is a constant depending on the percentages p, q and the scale of the object

(see Section 4.6 for the properties of this functional and how to choose K). If we also impose the constraint that the mapping is one-to-one, we implicitly solve the shrinking problem. With a large number of one-to-one correspondences (and the assumption that the two shapes are not sampled at very different rates), there can be no shrinking of one shape with respect to the other.

Algorithm 4.2 (Global similarity registration)

1. Set $V_{min} = \infty$.

2. For every pair of points $(a_{j1}, a_{j2}) \in A \times A$

   For every pair of points $(b_{k1}, b_{k2}) \in B \times B$ do steps (i) through (v)

   i. Find the similarity transformation ψ that aligns the sets $\{a_{j1}, a_{j2}\}$ and $\{b_{k1}, b_{k2}\}$.

   ii. Apply ψ to all the points in B to obtain B'.

   iii. For every point b_k of B', find its nearest neighbor $NN(b_k)$ in A. If the distance between b_k and $NN(b_k)$ is smaller than a threshold T then set a correspondence between the two points. A match matrix M between A and B is constructed in this way. Since two points from B' can have the same nearest neighbor in A, we enforce the one-to-one correspondence requirement, that is, allow a point to be linked to its second to fifth nearest neighbor if the first one can be assigned to a closer point in B', and the length of the link does not exceed

T. Recompute the transformation ψ that aligns the sets A and B according to the match matrix M.

iv. Compute $f(M)$.

v. If $f(M) < V_{min}$ then $V_{min} = f(M)$, $\psi_{min} = \psi$.

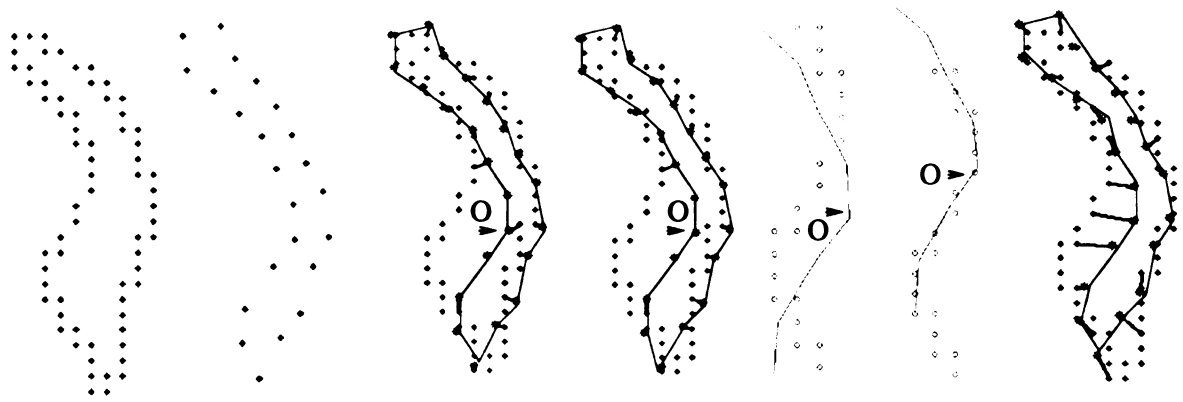
3. Apply ψ_{min} to all the points in B to obtain B'.

4. For every point b_k of B', find its nearest neighbor in A. If the distance between b_k and its nearest neighbor is smaller than T then set a correspondence between the two. A match matrix M' between A and B is constructed in this way and enforced to correspond to one-to-one links.

5. Find the linear transformation ψ_{final} that aligns the corresponding sets $A_{M'}$ and $B_{M'}$.

Figure 4.5(c) shows an example of similarity registration of a ventricle shape approximation to a full ventricle shape. Although globally the registration is quite good, some point correspondences are wrong (e.g., the point labeled O) or have been missed (the two neighboring vertices of O).

We are interested not only in computing an average shape (which is robust to slight misregistrations) but also the shape variation present in the data set which is best described by the set of high curvature points. Since a *global* linear registration does not necessarily perform a good local registration (see [46] and Figure 4.5(c)), we need to locally refine the results of the global registration such that the corresponding



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

Figure 4.5: Flexible registration of a shape approximation (b) to an original shape (a), with point correspondences drawn in green. c) Global similarity registration - Algorithm 4.2. d) Monotonic registration obtained from (c) after point reordering and inversion elimination (the point labeled O causes an inversion): Steps 1-3 of Algorithm 4.3. e) Topological neighborhood corresponding to point O: Step 4a of Algorithm 4.3. f) Similarity registration of the two topological neighborhoods in e). g) Final flexible registration.

points of high curvature from the two data sets are matched together. However, some high-curvature points in A may not correspond to high curvature points in B, therefore we do not enforce this requirement explicitly, but rather through *local similarity registration* and *monotonicity*. We define the term "local" in a topological sense according to the natural point ordering along curves defined by A and B. A good registration should be *monotonic*, that is, preserve the topologies (point ordering) on the two shapes.

A registration of two curves (sets of points along the contour) A and B (which can be regarded as a partial function f from A to B) is called *monotonic* if:

1. The points of A are cyclicly reordered such that point a_1 corresponds to point b_1.

2. There are no *inversions*, that is, if a_i and a_j correspond to b_k and b_l (in this order) and $i < j$ then $k < l$ (for example, an inversion is caused by point O in Figure 4.5(c)).

Note that a monotonic registration of two sets of distinct points is one-to-one, therefore we can perform a local registration and obtain one-to-one links by looking for a monotonic registration.

Algorithm 4.3 (Monotonic, local similarity-based registration)

Input: two sets of points A and B and a set $\mathcal{M}$ of one-to-one links between a subset A' of A and a subset B' of B obtained by global similarity registration.

1. Cyclicly reorder the points of A and B and the links in $\mathcal{M}$ such that a_1 corresponds to b_1.

2. If the number of inversions exceeds $|\mathcal{M}|/2$, reverse the ordering of the points in A.

3. Break the smallest number of links in $\mathcal{M}$ such that there are no more inversions (Figure 4.5(d)). Note that we are left with a monotonic registration.

4. For $i = 1..|B|$ do

   (a) Find a topological neighborhood of b_i, $[b_l, b_{l+1}, ..., b_i, ...b_{r-1}, b_r]$ (the actual size of the neighborhood depends on the curvature at b_i; the larger the curvature, the smaller the neighborhood) such that both b_l and b_r have correspondences in A, let them be $a_{l'}$ and $a_{r'}$ with $l' < r'$ (Figure 4.5(e)).

(b) Perform a similarity registration between the sets $[a_{l'}, a_{l'+1}, .., a_{r'}]$ and $[b_l, b_{l+1}, .., b_r]$ (Figure 4.5(f)).

(c) If B_i is linked to a different point in A than it was before, then record this change in $\mathcal{M}$.

5. Break the smallest number of links in $\mathcal{M}$ such that there are no more inversions.

4.4 Shape clustering and prototype computation

Since the objects we deal with are complex manifolds of different dimensionality (depending on the number of points along the contour), shape clustering proved to be a difficult problem. We are aware of only two type of approaches to shape clustering: (i) projecting all shape instances to a common subspace (e.g., PCA, Fourier, modal space, etc.) and treating the projection coordinates as points in an Euclidean space [26] and (ii) specifying a "distance" between 2 shapes and performing a distance-based clustering [53]. As such, shape clustering reduces to the general clustering problem for which numerous solutions have been proposed. One limitation of projection-based methods is that they exhibit an inherent loss of information due to the fact that the projection transformation is not one-to-one. That is, one point in the transformed space may correspond to several shapes whose visual appearance may be quite different (see the discussion in [44]).

Here, we propose to solve the shape parameterization/correspondence/clustering

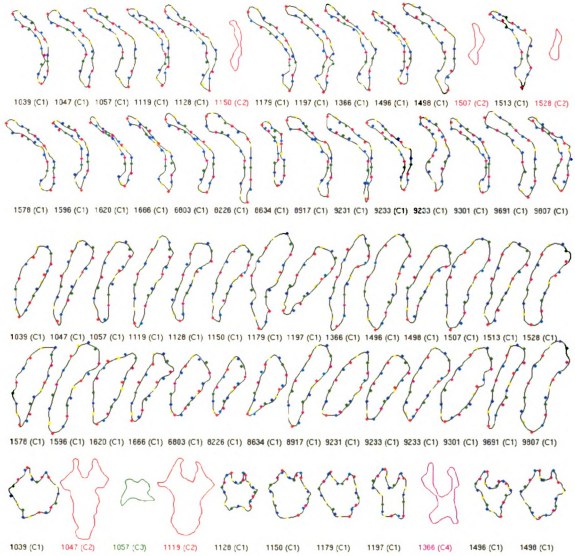


Figure 4.6: Two training sets of 28 right ventricular (rows 1 and 2) and 28 globular shapes (rows 3 and 4) and a set of 11 cistern shapes (row 5) from different patients were automatically divided into clusters (main cluster (C1) drawn using multicolor dots and secondary clusters drawn in red, green and magenta). The registration of the *best fit shape* (#1047 for ventricles, #8917 for globus pallidus and #1179 for cisterns) to clusters C1 is overlaid as sets of colored points; corresponding points on different shapes are drawn using the same color. For example, corresponding bottom points on each right ventricle are drawn in red.

problem in a unitary framework based on the MAE distance. The third step of Algorithm 4.1 defines a pseudo-distance matrix $\mathcal{D}$ of *mean alignment errors* between a *polygonal approximation of a shape* and an *original shape* from the training set. A natural way for obtaining shape clusters based on $\mathcal{D}$ which is also helpful for cluster prototype (average) computation is a greedy, divide-and-conquer strategy related to *histogram mode seeking* [58]:

1. Find a seed which is closest to the data (analogue of a mode of a histogram). This is done in Step 4a of Algorithm 4.1 by finding the *shape approximation* S'_{i_0} that *best fits* the current training set (based on the average distance to the rest of the shapes). S'_{i_0} will be used as a reference set for extracting corresponding sets of points of the same size from as many training shapes as possible, as discussed in Section 4.2.

2. Find all shapes S_j in the current training set that "fit" (are close enough to) S'_{i_0} (Step 4b of Algorithm 4.1). These shapes are removed from the training set and will form a new cluster (analogue to finding the two valleys adjacent to the histogram mode and grouping in one cluster all points between the valleys). For the mathematical details of what we mean by "*close enough*", see Section 4.6.3. We mention that it is very important that the cluster seed fits the shapes in the cluster as well as possible. Though there is no theoretical guarantee that all the points of the seed are perfectly registered to the shapes in the cluster, a poor fit can be due to one of the following two causes: either the training shapes are very different or the registration results are not accurate.

This cluster extraction procedure continues using a smaller current training set until all shapes have been assigned to a cluster. The clustering process makes only

one pass through the data since each cluster starts from a shape (seed) which is close to the cluster center. We noticed that this seed is very similar to the cluster average, therefore we do not believe that adding more passes (cluster reassignments) would modify the cluster membership (at least for well separated clusters).

For each cluster, the cluster prototype is defined as the *Procrustes Average* of the subsets of registered points extracted from each shape in the cluster. The cluster variation is defined as the $2n \times 2n$ covariance matrix of the subsets of points used to compute the prototype (n is the number of points on the cluster prototype). This variation can be used by a segmentation method to reject shape deformations that have not been seen in the training set (see [44, 25]).

4.5 Experimental Results

The shape learning method presented above was employed to design a shape model for 11 brain structures and its performance was assessed by a quantitative comparison to a "ground truth" model obtained manually. The training set consisted of observer-defined contours identified by a neuroanatomist in 28 individual T1-weighted contiguous MR images of the human brain, imaged in the coronal plane with in-slice resolution of 256×256 pixels. Figure 4.5 shows the following five registration stages: global similarity registration - Algorithm 4.2 (Fig. 4.5(c)), point reordering and inversion elimination - Algorithm 4.3.1-3 (Fig. 4.5(d)), defining topological neighborhoods in Algorithm 4.3.4a (Fig. 4.5(e)) and their registration (Fig. 4.5(f)) and final registration (Fig. 4.5(g)). Figure 4.6 shows the original manual tracings and clustering

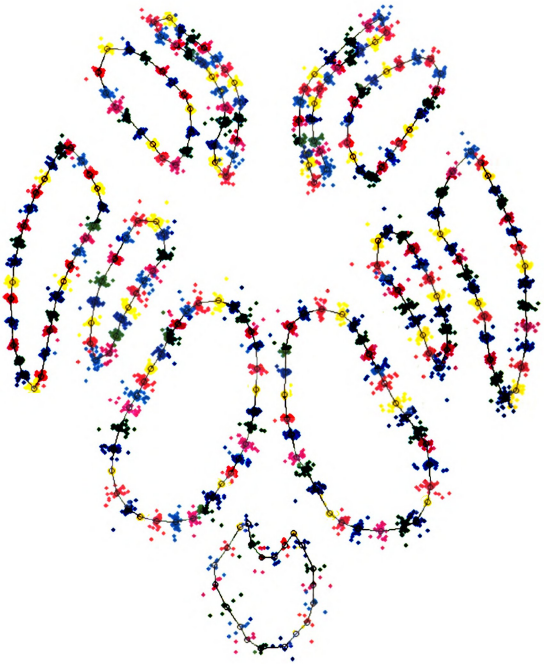
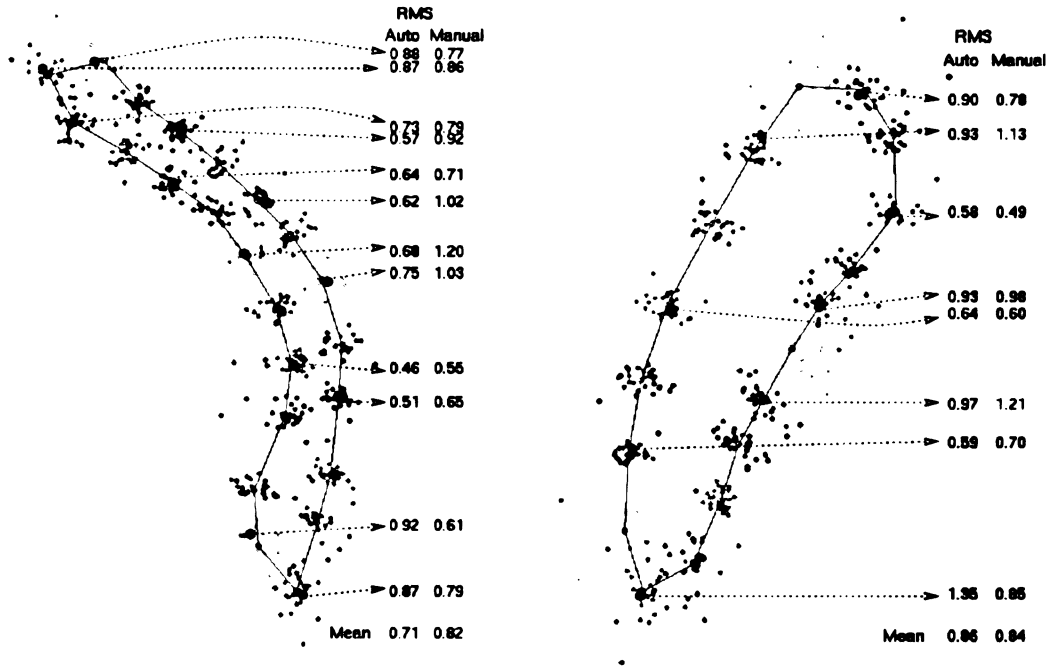


Figure 4.7: Procrustes averages (prototypes) of the shapes in the main clusters for 11 brain structures with the aligned shape examples overlaid. The clouds of consecutive points are drawn in different colors to show the accuracy of the registration.

results for three structures together with the *best fit shape* registration to the main cluster (the sets $T_{i_1,i_0}, ..., T_{i_p,i_0}$ as defined in Algorithm 4.1). The main cluster is drawn using multicolor dots, while the secondary clusters (which can be considered as outlier shapes) are drawn in red, green and magenta. To emphasize the role of a good registration for extraction of $\{T_{i_k,i_0}\}_{k=1..p}$, we drew consecutive points on T_{i_k,i_0} using different colors, the same color for corresponding points on each shape. For example, corresponding bottom points on each right ventricle are drawn in red. Figure 4.7 shows the main cluster prototypes for 11 structures with the aligned shape examples overlaid. Consecutive point clouds are drawn in different colors to show that the clouds are non-overlapping; the registration appears to be very precise.

In order to obtain a quantitative validation of our results, we used the method employed in [59]. From each structure prototype, we manually selected several points which we considered most important in defining its shape (points with the highest curvature) and we manually registered them to the training images. We defined the *ground truth* position of these points as the Procrustes average of the manually registered points (these point positions are shown for the right ventricle and globus pallidus in Fig. 4.8 as black circles). We computed and compared the *root-mean-square* (rms) distance of manually placed points from the ground truth and the rms distance of the automatically registered points from this ground truth, respectively. The *rms* distances for the right ventricle and globus pallidus are also shown in Fig. 4.8: for every point selected on each shape, each distance is displayed on the same y coordinate as the ground truth point it corresponds to. The average rms distances for the selected points are similar, though, on individual points they may be quite



(a) (b)

Figure 4.8: Prototype of the 25 right-ventricle shapes in the main cluster (a) and prototype of the 28 right-globus pallidus shapes (b) with the aligned shape examples overlaid. The *ground truth* position for several points are shown using black circles. For each such point, we also show the *rms error* of the manual and automatic registrations.

different. The very high curvature points (the extreme upper or lower points) are somewhat better registered manually while the intermediate points are better placed automatically. This was expected, since it is very difficult for a human to exactly place a point (identify landmarks) if there are no curvature or other anatomical cues. The average rms error for each of the 11 structures is between $0.7 - 1.2$ pixels. For some structures, the automatic method produced a slightly smaller *rms* distance than the manual one, while for others the *rms* of the automatic method is slightly larger.

4.6 Discussion

We will provide a brief discussion on the choice of parameters, criterion function, computational complexity, failure cases and range of applicability of our algorithm:

1. *Upper bound on the distance between corresponding points.* Based on several thousand registration experiments using various real data sets for which there exists a ground truth match matrix, we found that *almost all* the distances between valid corresponding points are smaller than 10% of the *object scale*. The *scale* of a shape instance $A = \{(x_i, y_i)\}_{i=1..n}$ is defined as

$$Scale(A) = \sqrt{(\max_{i=1..n}(x_i) - \min_{i=1..n}(x_i) + 1).(\max_{i=1..n}(y_i) - \min_{i=1..n}(y_i) + 1)}. \quad [1]$$

Therefore, we enforce this *upper bound* when we set a point correspondence in

[1] This definition of scale is different from that employed by most previous studies [119, 34, 54, 11] where $Scale(A) = \sqrt{\sum_{i=1}^{n}((x_i - \bar{x})^2 + (y_i - \bar{y})^2)}$ since we wanted it to be independent of the number of points and the sampling rate. Consider, for example, two circles centered at the origin of radius 1 and 0.001, respectively. Our definition of scale does not depend on whether the point set consists of exactly 10 points on the outer circle and 1000 on the inner circle or vice versa. By contrast, the Euclidean norm-based definition of scale is highly dependent on the sampling rate which, for some applications, may be an undesirable property.

Step 2.3 of Algorithm 4.2. In practice, if there exists a proper matching between two sets, most of the links are actually shorter than half of this upper bound. We also found it to be convenient to work with scale independent objects, therefore, before registering two point sets, we rescale them such that the largest one has a scale equal to 10. In this case, the threshold T in Step 2.3 of Algorithm 4.2 is set to 1.

2. *Properties of the evaluation function $f(M)$ (Eq. (4.1)).* Since the distance between two corresponding points is not larger than 1 (see above), we have that

$$K/n \leq f(M) \leq (K+1)/n, \forall \text{ match matrix } M \text{ and } \forall\, n.$$

If a match matrix M' has $p\%$ more links than M, one has

$$K/[(1+p)n] \leq f(M') \leq (K+1)/[(1+p)n]$$

and M' is preferred to M if

$$(K+1)/[(1+p)n] < K/n \Leftrightarrow Kp > 1.$$

On the other hand, the choice between an n-link matrix and one with less than $(1+p)n$ links (if $p < 1/K$) is determined by the MAE: the $p\%$ increase in the number of links will be accepted iff

$$K/n + MAE(M)/n \geq K/[(1+p)n] + (1+q)MAE(M)/[(1+p)n] \Leftrightarrow$$

$$q \leq p + Kp/MAE(M).$$

Since $MAE(M) \leq 1$, one has that, if $q \leq p(1+K)$ then $q \leq p + Kp/MAE(M)$. In particular, if $K = 2$ a 50% increase in the number of links will always be accepted no matter what the increase of the MAE is, while a 25% is accepted if the MAE increase is less than 75% (this does not mean that if $q > 75\%$ then M' is always rejected; in practice it might be accepted until q becomes about 100%).

3. *Constructing a shape cluster (Step 4 of Algorithm 4.1).* One should note that a match matrix M obtained by global similarity registration satisfies $MAE(M) \leq 10\%Scale$ and this does not usually hold after the local registration. After performing all 8,687 possible pairwise registrations (between examples of the same structure), we found that in almost all the cases, a $MAE \geq 15\%Scale$ corresponds to a wrong matching. This is due to either large shape differences or cases when least-squares is not a good matching philosophy (for example, the hand shapes in [59], or handwritten characters). Our method always found a good least squares matching if it was present. For the shape learning application reported here, we expected to find one main shape cluster and a few other small clusters with outlier shapes. Therefore, when computing the best fit to the training set in Step 4.a, we did not compute the average *Mean Alignment Error* to all the shapes but only to the closest 70% (expecting that the furthest 30% might be outliers). On the other hand, two shapes were considered to be "*close enough*" in Step 4.b if their $MAE \leq 15\%Scale$. We would also like to mention that the manner in which all the above parameters are chosen seems to be application independent. We obtained similar results for learning shapes coming from different applications like fingerprint matching or hand shape-based person verification, with no changes in the implementation.

4. *Time complexity.* There are two aspects regarding the time complexity of our method. The first one concerns the shape registration complexity (Algorithm 4.2). It is easy to see that if the two sets to be registered have n points each, then the complexity of Algorithm 4.2 is $O(n^5)$. This can be reduced to $O(n^4)$ if one data set is formed of ordered, relatively evenly sampled points along a continuous curve

(and even further reduced to $O(n^3)$ if both data sets are like that) and one does not expect large scale differences between the two objects or in the percentage of outliers found in the two data sets. The heuristic is based on the observation that an initial 2-point pairing hypothesis in which the two points from A (or those from B) are very close together produces a worse estimate of the transformation between the two sets than a hypothesis where the selected points are far away. In this case we could form the initial hypothesis based only on the "diagonal" of the data set(s): $(s_i, s_{i+|A|/2})$. Though it might be argued that this complexity is still high, for applications using 2D contours extracted from images that usually have less than a few hundred points, it is sufficiently fast and gives the result in order of seconds (our $O(n^4)$ implementation needs about 10 seconds to register a 30-point approximation to a 100-point original contour on a Sun Ultrasparc (296 MHz processor)).

The second complexity aspect concerns the shape learning itself. Algorithm 4.1 performs m^2 pairwise matchings, where m is the number of shapes in the training set. We mention that it is neither possible nor necessary to apply it *directly* to a large shape set. The idea behind trying all m^2 matchings is to find a shape in the training set that fits well with the remaining shape examples, and compute a mean shape based on it. The resulting mean shape is sufficiently smooth (see Fig. 4.7) and retains enough shape characteristics so that it can be used for direct matching to new shape examples. Therefore, we only need to apply the quadratic pair-wise matching to a relatively small training set (a few tens of elements). After that, one can classify the remaining shapes only by registration to the estimates of the current clusters prototypes. Each time a shape does not fit any of the current prototypes,

a new cluster is started. In this way, the learning process becomes incremental, and linear in the size of the training set.

5. *Failure cases.* Algorithm 4.2 may fail to find the real point matchings between two point sets in cases where the proportion of outliers in at least one of the sets is very large (e.g., when matching one object outline to the edge points in a cluttered scene) and no additional information (constraints on the allowed similarity transformations or edge direction at each point) is used. However, this is not a failure of the search method in finding an "optimal" matching according to the optimality criterion in Eq. 4.1, but rather a failure of the data points in satisfying this optimality criterion. Since we used Algorithm 4.2 as a filter (declaring failure whenever the proportion of outliers resulted was high and stopping further processing) we have not noticed any failure of Algorithm 4.3. In the current application, Algorithm 4.4 used the assumption that we expect to have one main cluster along with some outlier shapes. This seems to be a reasonable assumption for shapes which describe natural variations across different subjects in medical or biometric images. However, we do not consider that this algorithm will necessarily fail on true multimodal distributions. If the 70% threshold defined in Section 4.6.3 is decreased, we believe that the method could accommodate multimodal distributions as well.

6. *Range of applicability.* As experimental results, we showed how to identify clusters and compute prototypes for 11 different neuroanatomic brain structures from training sets of 28 manual tracings per structure in MR images. However, this method has a larger range of applicability (see Chapters 5-6 and [49, 42] for more applications). It can also be employed for designing shape models for open contours (Section 6.1)

as well as for sets of points containing no connectivity information (e.g., fingerprint minutiae or palm feature points [42]). In the latter case, only Algorithm 4.2 is applied for data registration, and the amount of non-linear deformation that can be handled is smaller. In general, there is a tradeoff (regulated by the threshold T described in Section 4.6.1) between the amount of non-linear deformation accepted and the proportion of outlier points present in the data sets. If one does not expect to have many outliers, then the threshold T can be increased and more non-linear deformation (equivalent to a larger average distance between corresponding points) is accepted.

4.7 Summary

This chapter demonstrated an automatic approach to 2D shape model design. A set of training shapes is divided into several clusters for which shape prototypes and modes of variation are derived. The method is based on finding homologous points on a set of shapes using a two-stage approach. The first stage aligns two shapes using similarity transformations and produces an initial set of homologous points. This set is further refined by a second stage that handles nonlinear shape deformations. The method has been employed to learn shape models for 11 brain structures. A quantitative analysis of the registration accuracy shows that the automatic results are as precise as manual registration. We have concluded the chapter with a discussion concerning the choice of the method parameters and computational time requirements.

Chapter 5

Deformable model segmentation

Once the object of interest has been detected, the segmentation problem can be approached either using a top-down strategy (as described in [44]) or by a combination of bottom-up and top-down methods [40].

5.1 Top-down Active Shape segmentation

The top-down search procedure is based on the model fitting strategy. At each step of the fitting process, several model location hypotheses are considered and evaluated. During the hypothesis generation, the actual image data play no role (no image preprocessing is done). Also, an outlier detection and replacement procedure has been developed to detect misplaced points and infer their new positions. The outlier detection improves robustness and accuracy of the shape model fitting process. The searching procedure consists of the following steps [44]: 1) Model fitting using similarity transforms, 2) model fitting using piecewise similarity transforms, 3) outlier

removal, 4) final point adjustment and 5) final outlier removal.

Model Fitting Function As a result of the hypotheses generation processes, shape model locations are sequentially hypothesized. In order to evaluate the model location hypotheses, a *fitness function* is needed to assess the agreement between the image data and the particular model instance. We have designed a *fitness function* $F = F_B/(F_{GA})^2$ that consists of two components:

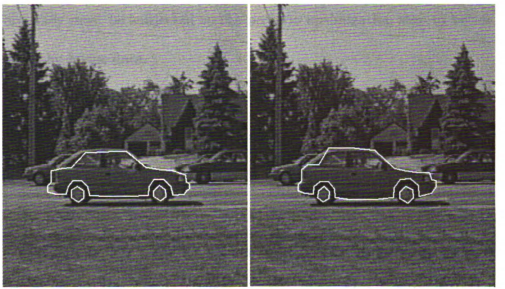
1. *Fitness of the gray level appearance* F_{GA} is determined as the average squared Euclidean distance between the actual gray level appearance and the mean gray level profile incorporated in the shape model.

2. *Fitness of the border* F_B is calculated as the ratio between the aggregate response of all four-point cliques along the contour and the maximum possible response (twice the number of cliques).

Model fitting using similarity transforms. Shape instance hypotheses specify the locations of all model points within the analyzed image. The hypotheses are generated using similarity transformations and are applied to the model average position. All generated hypotheses are sequentially evaluated using the model fitting function F and the best fit is determined (Fig. 5.1(a)).

Model fitting using piecewise similarity transforms. Since non-rigid objects or objects with inter-subject variability are discussed here, similarity transforms do not account for any potential deformations of the expected shape. Therefore, similarity transforms are applied to subsets of 5–7 consecutive model points. This accounts for global non-linear deformations of the prototype (stretching or shrinkage), while

maintaining the expected shape locally (Fig.5.1(b)).



(a) (b)

Figure 5.1: Example of automatic car segmentation. a) Detected car using similarity transforms. b) Refined segmentation using piecewise similarity transforms.

Outlier removal. Under unfavorable circumstances, the previous step may introduce incorrectly determined vertices – **outliers**. This may happen if a subshape fitted by the previous step exhibits weak edges or if there exists another border of similar properties in the neighborhood. Typically, when using PDM's, shapes that do not correspond to the allowed shape at any stage of the detection process are rejected or approximated (projected onto) with allowable ones. To treat the problem of outliers in a systematic fashion, we developed an approach to outlier detection and position adjustment [44]. The misplaced points are identified using the information about the relative positions of the shape model vertices that are implicitly included in the shape model.

Final point adjustment. Some of the shape model points may have been declared outliers in the previous step. Consequently, their position may have been adjusted

solely considering the average shape appearance and not considering the image data. Therefore, they must be subjected to the position optimization step to better correspond with the image data.

Final outlier removal. Outliers may be introduced during the final point adjustment. Following the same outlier detection procedure applied earlier, the outliers are identified and removed. No adjustment is attempted in this final step of model fitting.

This paradigm is very useful for segmenting objects with low contrast from background (e.g., the *thalami* in MR brain images [44]) where a bottom-up approach would not be able to segment them. On the other hand, if the objects have a good contrast with the background a low-level segmentation or edge detection could be a good starting step for a high level procedure.

We employed the automatically extracted prototypes to train a knowledge-based Point Distribution Model (PDM) [44, 25] and segment 11 neuroanatomical structures. Note that Algorithm 1 not only gives an average shape and variation for each cluster, but also the registration of the cluster prototype to all the shapes in the cluster. Therefore, PDM training becomes very fast and does not require any kind of human intervention once the shapes in the training set have been clustered.

We show the results of the automatic segmentation on four different MR images in Fig. 5.2. Since the automatically derived models are almost identical to the manually traced models, the segmentation results are as good as those reported in [44] with an average boundary positioning error of 0.8 ± 0.1 pixels with respect to the manual tracings, and maximum boundary positioning error of 4.3 ± 1.2 pixels.

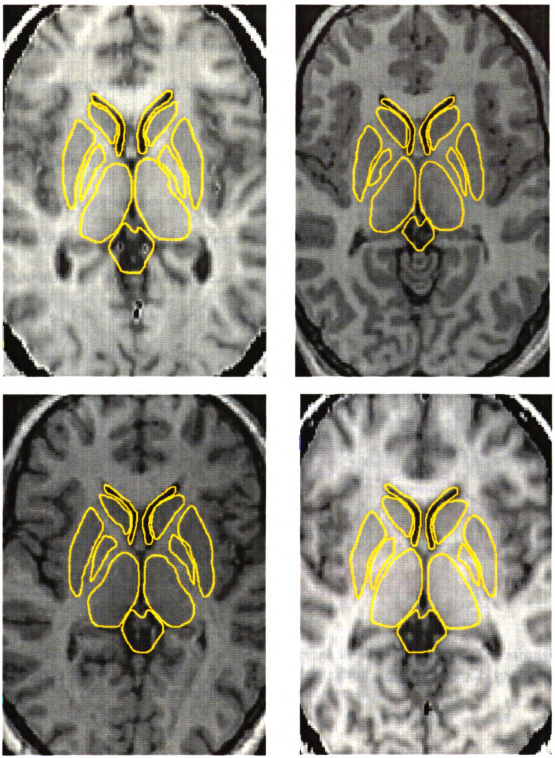


Figure 5.2: Automatic segmentation of 11 brain structures in four different MR images using the learned models.

5.2 Warping-based segmentation

When the object of interest is homogeneous and has a good contrast with respect to the immediate background, one can combine low-level with high-level segmentation methods. Low-level methods include edge detection (Fig.5.3(a)) and unsupervised segmentation followed by edge detection (Fig.5.4(a)). High-level methods learn the shape and/or appearance of the object of interest and refine the low-level results using this additional information. For example, the main characteristic that makes the CC distinguishable from other structures in the brain is its shape. The CC shape can be learned from manual tracings using a PDM and can be used in a model-based segmentation guided by a low-level edge map. The first stage of the warping-based segmentation is the alignment of a learned CC prototype to the edge image.

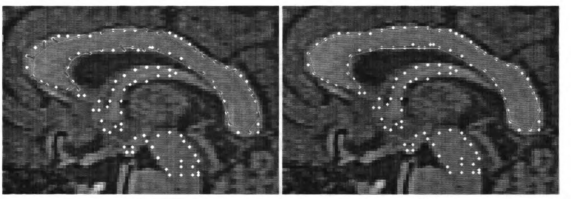
(a) (b)

Figure 5.3: Registration of CC in the ROI image in Fig. 3.1(b). (a) The strongest 600 edges from the ROI in Fig. 3.1(b). (b) Registration of an average CC to the edge image in (a) (point correspondences are shown in green).

Figure 5.3(a) shows the alignment of a CC prototype to the edge map computed from the ROI in Fig.3.1(b) while Figure 5.4(a) shows the alignment to an edge map obtained following a low-level pixel classification. The second stage of the segmenta-

(a) (b)

Figure 5.4: CC registration to the ROI image in Fig. 3.1(c). (a) Edges from the low level segmentation in Fig. 3.1(c). (b) Registration of an average CC to the edge image in (a) (point correspondences are shown in green).



(a) (b)

Figure 5.5: High level segmentation of CC. (a) Registration of an average CC to the edge image in Fig. 5.4(a). (b) Warping the average CC onto the edge image in (a).

tion warps the CC prototype to the edge image: each vertex of the prototype moves towards its corresponding edge (Figs. 5.3(b) and 5.4(b)). If a vertex has no corresponding image edge, then there is a disagreement between the edge map and the expected shape. This can be caused either by a large shape difference between the template and the real object, or more often, by the vagaries of the edge detection procedure. In this case, one can either keep the position of the model vertices with no corresponding image evidence after the alignment (if we are confident that the

object of interest does not deviate significantly from the shape template) or one can simply exclude them from the template. In both situations one could also apply an outlier removal step (Section 3.2.1) in order to enforce the shape variation present in the training set. Since we usually use a small number of points for the object prototype, the warping result may not be very smooth (Fig.5.4(b)). Therefore, one can conclude the segmentation by applying some smoothing operators like a morphological opening. Results of the automatic segmentation of the Corpus Callosum in nine MR images using the warping-based paradigm are shown in Fig. 5.6.

Although, in principle, the warping could be applied to the edge set obtained directly from the gray-level image as well as from a low level segmentation, we found that the low-level classification eliminates most of the noisy edges belonging to structures other than CC.

5.3 Summary

This chapter addressed the problem of shape-based object segmentation. First, we described a top-down approach based on an active shape fitting paradigm. This method may be quite computationally demanding but it has the advantage to work for objects that are not easily separable from the immediate background. We also demonstrated an alternate approach which is based on a combination of bottom-up and top-down methods. This is able to perform a fast and accurate segmentation provided that the object of interest is visually separable from the background.

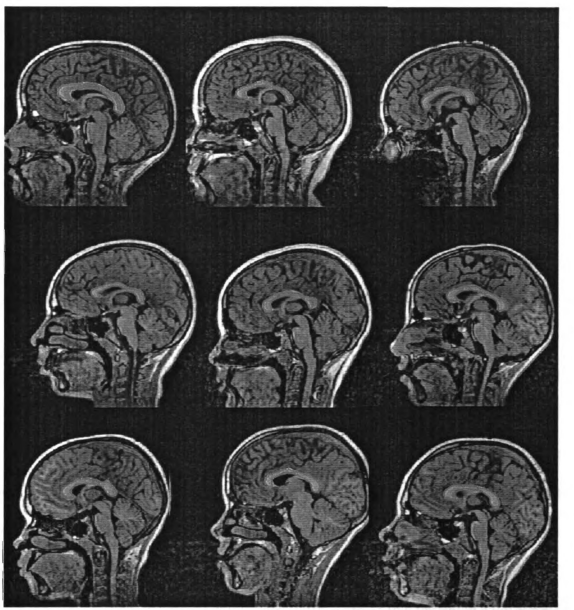


Figure 5.6: Automatic segmentation of the Corpus Callosum in nine MR images using the warping-based paradigm.

Chapter 6

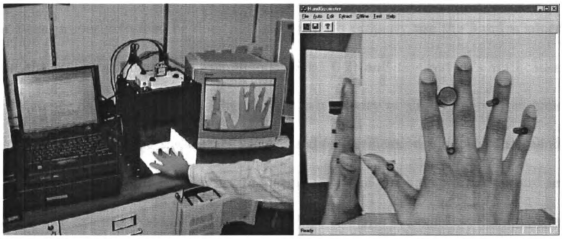
Object matching

This chapter presents the last stage of the *detection-segmentation-matching* paradigm for object learning and retrieval. As discussed in Section 2.5, an exact matching of two objects is essential for systems in the personal identification domain. We will demonstrate how the shape of a human hand (though not necessarily unique within a large population) can be used to reliably verify the identity of a person in a database of 53 persons. We will next attempt to identify, using morphometric analysis of Corpus Callosum shapes extracted from MR images, the extent to which the CC shape is correlated to (or moreover, can predict) the gender, handedness (left handed or right handed) or presence of dyslexia.

6.1 Deformable matching of hand shapes for user verification

Automatic human identification has become an important issue in today's information and network-based society. The techniques for automatically identifying an individual based on his physical or behavioral characteristics are called biometrics. Biometric systems are already employed in domains that require some sort of user verification (e.g., for access control or welfare disbursement programs). Numerous distinguishing traits that have been used for personal identification include fingerprints, face, voice, iris and hand geometry. It is generally accepted that fingerprint and iris patterns can uniquely define each member of an extremely large population which makes them suitable for large-scale recognition (establishing a subject's identity). However, in many applications, because of privacy or limited resources, we only need to authenticate a person (confirm or deny the person's claimed identity). In these situations, we can use traits with less discriminating power such as voice or hand shape.

Hand geometry-based verification systems have been available for almost three decades. Still, their technical descriptions are scarce and the available information is based mostly on patents (see [137, 74] and the references therein). However, the problem of matching hand shapes is not only important for biometric systems, but it is part of a more general, shape-based object learning and recognition topic (see, for example, studies by Grenander *et al.* [56] and Hill *et al.* [59]). We propose to approach the practical problem of person verification based on hand geometry using the powerful tools of deformable shape analysis. This is motivated by the limited



(a) (b)

Figure 6.1: Hand shape acquisition system (a) and the image it captures (b) (Courtesy of Dr. Sharath Pankanti and Arun Ross).

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

Figure 6.2: Four pairs of input hand shapes: contours that form pair (a) belong to the same hand, contours that form pair (b) belong to the same hand (different from (a)), pairs (c) and (d) are formed by different hands. Based on the input contours, it is very difficult to tell which of the four pairs belongs to the same hand. None of the existing systems aligns the hand contours before extracting features, leading to inferior matching accuracy.

ability of the hand shape acquisition system to implicitly register different hand images using the rigid pegs on the hand scanner platen (Fig. 6.1(b)). If the user has not been properly trained or does not cooperate to properly use the hand scanner, then the resulting images are not aligned (Fig. 6.2) and the system's verification performance degrades [137, 74]. Therefore, it is necessary to align the acquired hand shapes before extracting the feature vector used for verification. On the other hand, it is also useful to compare the discriminating power of the handcrafted feature set used by the existing systems to that of the *shape distance* between two hand shapes which is a byproduct of our alignment procedure.

6.1.1 Proposed Method

Given a pair of top views of hand images acquired by a hand scanner (Fig. 6.1) similar to those described in [137, 74], we propose the following hand shape matching paradigm (Fig. 6.3):

1. *Peg removal.* A mask containing the known positions of the five pegs is used to replace the pegs with a background like color.

2. *Contour extraction.* A *mean-shift* unsupervised segmentation [23] is applied to each image, followed by a morphological smoothing and hole filling. Then a contour following algorithm is used to compute the shape of the hand.

3. *Finger extraction and alignment.* The five pairs of corresponding fingers are extracted from each contour and aligned separately with respect to the rigid transformations group as described in Section 4.3. We chose to align pairs of fingers as

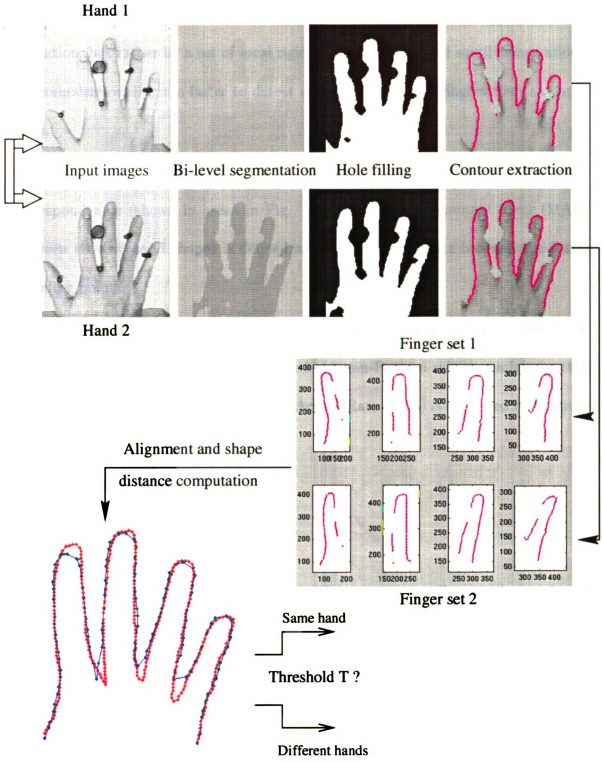


Figure 6.3: Alignment-based hand verification system.

opposed to the entire hand because of the following reasons: (i) a human hand is an articulated object and the motion of one finger cannot be described by a linear transformation, but rather by a set of local rigid transformations and small deformations, (ii) computationally, it is faster to detect and align individual fingers than an entire hand.

4. *Pairwise distance computation.* Each alignment in Step 3 produces a set of point correspondences (shown in green in Fig. 6.3). The *Mean Alignment Error* (MAE) between the two hand shapes is defined as the average distance between the corresponding points.

5. *Verification.* A pair of hand shapes is said to belong to the same hand if their MAE is smaller than a threshold T. Usually, the Neymann-Pearson rule that minimizes the False Reject Rate (FRR) for a fixed False Accept Rate (FAR) is employed to compute T.

6.1.2 Experimental Results

A data set of 353 hand images belonging to 53 persons was collected (the number of images per person varied between 2 and 15). We show the alignment of two hand shape pairs in Fig. 6.4; pair in (a) belongs to the same hand, while pair in (c) is formed by different hands. In each pair, one of the hand shapes contains about $120 - 130$ points, while the other contains about $300 - 350$ points (see Section 4.6 for details about the number of points to be used). To each pair of images of the same hand, we applied Steps 1-5 of the algorithm in Section 6.1.1 and obtained a

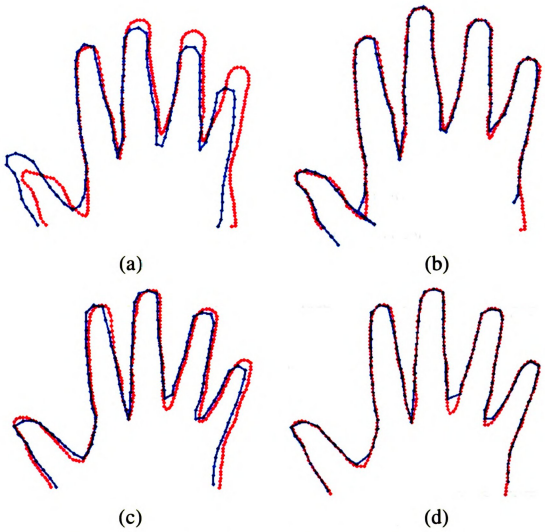
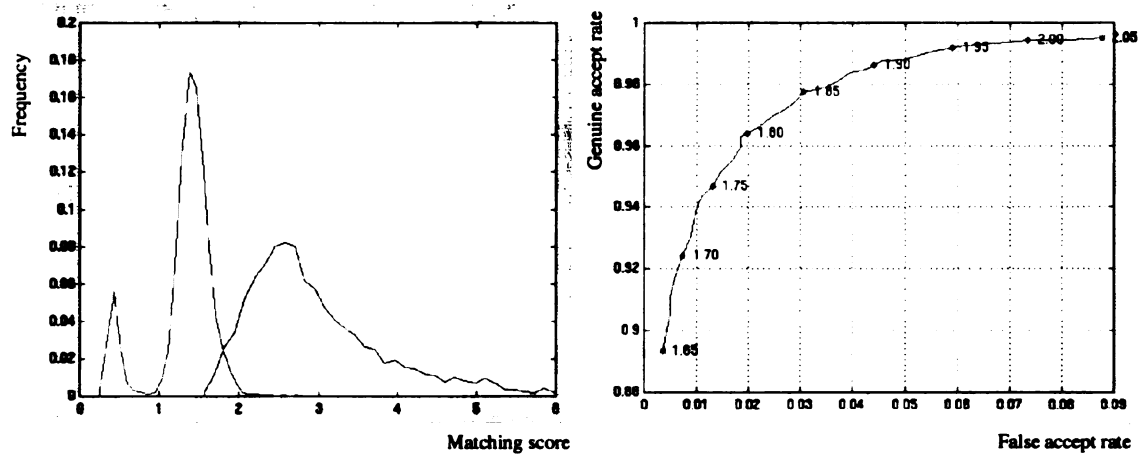


Figure 6.4: Two pairs of hands before ((a),(c)) and after ((b),(d)) alignment. The hand pairs in (a) and (b) belong to the same hand while the pairs in (c) and (d) belong to different hands. These two pairs correspond to the overlapping region between the genuine and the imposter distributions in Fig. 6.5(a).

complete set of 3,002 intra-class pairwise distances. We also randomly chose a set of 3,992 pairs of images of different hands and obtained a set of inter-class distances. Based on these distance sets, we computed genuine and imposter distributions (Fig. 6.5(a)). Note that the smaller peak in the genuine distribution is only an artifact of the alignment method. When we match two hands, the contours to be aligned are sampled differently (one of them has about three times more points than the other). Therefore, when a hand shape is matched to itself, the alignment error is not zero due to the different point sampling along the contour. It appears that matching two different samplings of the same contour produces an alignment error of about 0.5 pixels while when matching two different contours of the same hand the error is about 1.5 pixels.



(a) (b)

Figure 6.5: Hand shape-based verification performance. (a) Mean alignment error distributions for the genuine class (red) and imposter class (blue). The distributions are derived based on a total of 353 hand images of 53 persons. (b) ROC curve for the hand shape-based verification system. The annotations on the curve represent different thresholds on the MAE distance.

The *ROC* curve associated with the two distributions is shown in Fig. 6.5(b). One can see that the classification system is very accurate: e.g, for a threshold of

$T = 1.80$, the genuine accept rate is 96.5% for a 2% false accept rate. Although the 2% false accept rate may seem high, in practice, it is much smaller, since a user of the system does not know the identity of which other user he can assume such that their hand shapes match.

6.1.3 Improving verification accuracy

One can improve the accuracy of the verification method by analyzing the causes of the overlap between the genuine and the imposter distributions shown in Fig. 6.5(a).

There are mainly two situations that produce the overlap:

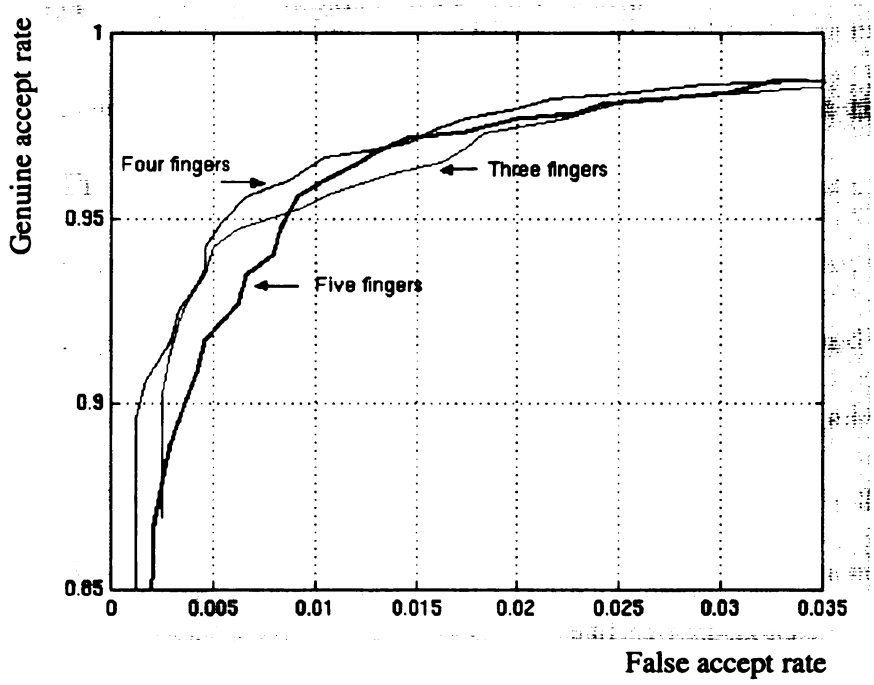


Figure 6.6: ROC curves generated by taking into account three, four and five fingers in the alignment error computation.

1. The right tail of the genuine distribution (that overlaps the imposter distribution) is generated by images where the subjects did not properly place their hand on the scanner. Such an example is shown in Fig. 6.7 (scans S_2 and S_3). We noticed

that the spatial arrangement of the pegs on the scanner allows quite a large flexibility in placement of the subject's thumb (see Fig. 6.7). Since a bent thumb cannot be linearly aligned to an unbent thumb, the alignment error will be larger in this case. This fact is quantitatively confirmed by excluding the user's thumb from the computation of the alignment error. Figure 6.6 shows three ROC curves computed by taking into account three (red curve), four (green curve) and five (blue curve) fingers of the hands in the matching. The best performance is attained when four finger are used (the thumb is excluded). This is the reason why Fig. 6.3 shows only four pairs of fingers extracted from each hand shape.

A second way to improve system's accuracy with respect to the genuine distribution is by *learning a personal hand template* (Fig. 6.7). Note that the genuine distribution in Figure 6.5(a) was computed using *all* the enrollment data. However, since after alignment, multiple contours of the same hand are very similar, in practice it is enough to keep *only one* hand template per user. This template should be the one that *best matches the hand contours belonging to that user* (has the smallest average distance to the remaining enrollment shapes of that user). After selecting one hand template for each user as described above, the genuine accept rate increases by about 2% for the same false accept rate.

2. We also noticed that the left tail of the imposter distribution (that overlaps the genuine distribution) is generated by the images where different subjects have almost identical handshapes. Such an example is shown in Fig. 6.4(c-d). Although in this case nothing can be done from a pattern recognition viewpoint, this problem is not likely to be frequently encountered since a user of the system does not know

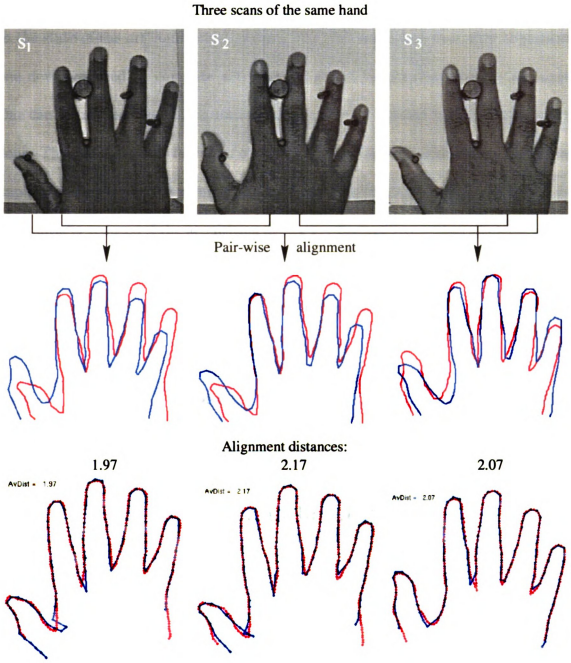


Figure 6.7: Learning a personal hand template. A hand template per person can be selected from the enrollment set based on its average alignment distance to the remaining enrollment shapes. In this case, the enrollment shape closest to the remaining ones is S_2 with an average alignment distance of 2.02.

the identity of which other user he can assume such that their hand shapes match.

Finally, we believe that the performance of our verification system is comparable to the state-of-the-art commercial systems. We also emphasize that after alignment, one does not need to compute a traditional set of handcrafted features anymore. One can simply use the MAE criterion whose value is available once the alignment is done. The matching time for a pair of hand images using our general shape alignment implementation is about 8 seconds on a 250 MHz Sun Ultrasparc. It can, however, be substantially reduced if the acquisition device is redesigned such that the hand image can be segmented from the background by simple thresholding and the alignment procedure is reimplemented specifically for this application.

6.2 Corpus Callosum shape analysis: a comparative study of group differences associated with dyslexia, gender and handedness

Corpus Callosum is the main interhemispheric commisure of the brain consisting of approximately 180 million fibers, most of which connect homologous cortical areas (Fig. 6.8). Corpus Callosum is a structure that is of interest when searching for the morphological bases of the brain's laterality. It is also thought to play an important role in organizing auditory stimuli and in language perception.

Unfortunately, the influences of age, gender, handedness, presence of dyslexia, and other factors on the variability of the Corpus Callosum size and shape are still con-

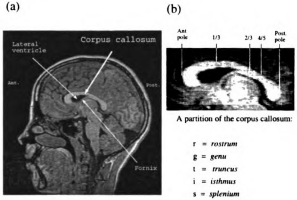


Figure 6.8: (a) Midsagittal, T1-weighted MRI section of the brain (12 year old boy) showing the *Corpus Callosum* as a C-shaped structure. (b) Details showing a partition of the *Corpus Callosum* (Courtesy of Prof. Arvid Lundervold).

troversial. Previous studies have reported inconsistent and many times contradicting results concerning the CC size differences. A larger splenium in women compared to men was found by de Lacoste and Holloway [33] and Davatzikos *et al.* [32] while Byne *et al.* [18] and Witelson [131] found no significant difference. Witelson [131] and Robichon & Habib [108] reported significantly larger and thicker isthmus regions in non right handed men compared with right handers (no handedness related difference was found in women though). In the same way, reports of differences in callosal size between dyslexic and normal subjects have produced inconsistent results. Duara *et al.* [35] found a larger splenial (posterior) area of the midsagittal CC in 12 male and 9 female dyslexic adults, as compared to 19 controls. Rumsey *et al.* [112] reported a larger area of the posterior third part of the CC in a group of 21 dyslexic men as compared to a control group. Hynd *et al.* [68], who examined 11 male and 5 female children with dyslexia, found the region of the genu (anterior part) to be smaller in the dyslexic group. Larsen *et al.* [81], comparing the callosal area of 15 female and 4 female adolescent dyslexics to 17 controls matched for age, gender and IQ, did not

find area differences between the groups. Robichon and Habib [108] compared the CC shape and size in a group of 16 adult dyslexics and 12 controls. Overall, the dyslexics had larger callosal area than controls. In contrast to previous studies, Robichon and Habib could not find differences in the anterior (genu) nor in the posterior (splenial) areas between the groups. However, they detected significant differences in the isthmus area, with a larger isthmus in the dyslexic group. Using a "circularity index", they found the CC to be thicker and more round-shaped in the dyslexic group, whereas the CC in non-dyslexic subjects were more flat and thinner along their antero-posterior axis. The most recent study known to us was done by Pennington *et al.* [98] who found no significant callosal differences between dyslexics and normals.

Some methodological problems that may have influenced the results in previous studies are determination of a proper midsagittal section and the possible influence of the total brain volume on estimated CC area. Methods for segmentation and subdivision of the Corpus Callosum have also differed among studies. Four of the studies have used a subdivision of the CC into five subregions, but since the methods for subdivision are different, subregion results cannot be compared directly. Even more important, as Davatzikos *et al.* [32] have noted, the quite variable CC curvature, which can highly affect the computed areas, has never been taken into consideration. We therefore believe that CC shape analysis is more informative than area analysis since it is a continuous global measurement which does not depend on size or pose.

The main goal of this study is to investigate to what extent the shape of the Corpus Callosum is correlated to (or can even predict) the gender, handedness or the presence of dyslexia. A second goal is to apply a common shape analysis framework

to several datasets previously employed in the literature as an attempt to mitigate the partly contradicting results that have been reported.

6.2.1 Methods

Data

For this analysis, we have used five different datasets: [128, 108, 18, 131, 32]. Except for the Bergen dataset [128] (Fig. 6.9), the remaining CC shapes were scanned from the original papers, therefore we could use only the male population of Witelson [131] and a random half of the Byne *et al.* [18] dataset. The total number of CC shapes we processed was 132, however each dataset was analyzed independently since the tracing procedures for each set of shapes were probably different [107]. We do not believe that scanning the CC shapes from previously published papers has introduced any major methodological problem: the shape analysis is independent of the absolute scale information which is unknown for the scanned data. A brief comparative description of the data is shown in Table 6.1, a more detailed description of the Bergen dataset follows while the other four sets are completely described in the original papers.

Imaging data from 54 subjects, 28 (22 males, 6 females) dyslexic children and 26 (21 males, 5 females) controls matched for age (12 ± 1 years) and handedness (all right handed), used in this study were part of a series of 3D MRI examinations on a Siemens Impact 1.0 T scanner with whole head, ear-to-ear, multispectral 3D gradient echo acquisitions (T1W **FLASH** TR=22ms, TE=6ms, FA=30°; T2W **DESS** TR=26, TE=9,45, FA=40; PDW **FISP** TR=23, TE=10, FA=15; FOV=256mm, 3Dslab=160mm, with a

Table 6.1: A brief description of the five datasets used in this study, the variation factors that have been considered and a comparison between the original results and our shape analysis results. The table entries are shown in increasing order of the average age of the subjects in the datasets. *Note:* Dys = Dyslexics, Nor = Normals, Mal = Males, Fem = Females, RH = right handed, nRH = non right handed, * = only the published part of the dataset has been used.

Study	Data description	Comparisons	Original results	Our shape analysis results
Plessen *et al.* [128]	22 dysl. males, 6 dysl. females 21 normal males, 5 normal fem Age: 12±1, all right handed	Dys vs. Nor Mal vs. Fem	No significant area differences associated with dyslexia Male vs. Female comparison not performed	Dyslexic posterior mid body significantly shorter No significant sex related difference
Robichon and Habib [108]	9 dyslexic RH, 7 dyslexic nRH 10 normal RH, 2 normal nRH Age: early twenties, all males	Dys vs. Nor RH vs. nRH	Dyslexic CCs thicker and more circular RH dyslexics and nRH normals have larger isthmus	Dyslexic CCs more curved, post. midbody significantly shorter. nRH dyslexic CC shorter than RH dyslexic. nRH normal post. midbody thicker and longer than RH.
Byne *et al.* [18]*	8 males, 11 females Various ages, average around 40 Handedness unknown, all normal	Mal vs. Fem	No significant sex related difference	No significant sex related difference
Witelson [131]*	9 RH, 6 nRH Age (death): 50, all males All normal	RH vs. nRH	Total CC, post. midbody and isthmus areas significantly larger in nRH	Posterior midbody thicker and significantly longer in nRH
Davatzikos *et al.* [32]	8 males, 8 females Age: 70 ± 6, all right handed All normal	Mal vs. Fem	Posterior CC part more bulbous in females	Posterior CC part more bulbous and shorter in females

voxel size of $1 \times 1 \times 1.25$ mm^3).

From each multispectral data set, the T1-weighted 3D FLASH channel was selected for CC analysis because of good signal-to-noise ratio and gray matter/white matter contrast. To eliminate the variability in CC shape and cross-sectional area attributable to differences in head position in the scanner and the orientation of the scan plane used to generate the mid-sagittal image [105], each 3D data set was subject to AC-PC alignment using the AFNI software package [29].

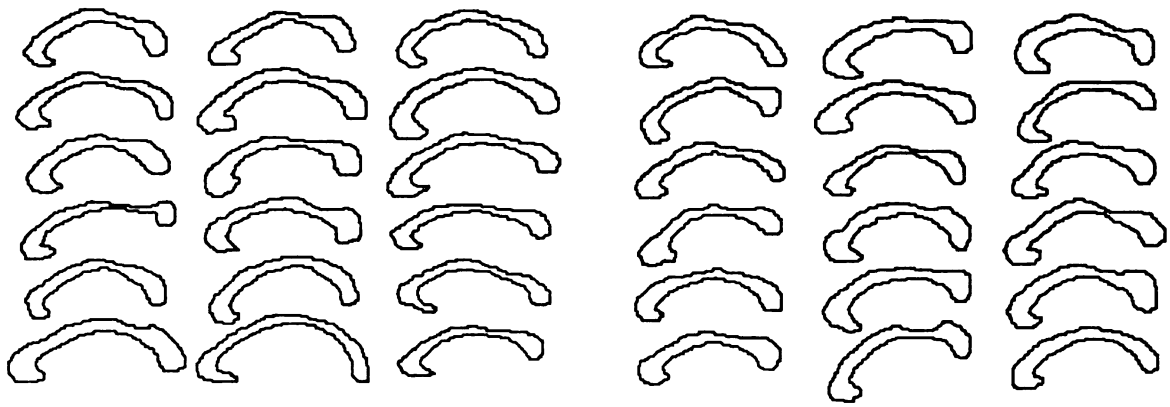


Figure 6.9: Corpus Callosum shapes belonging to normal subjects (a) and to dyslexic subjects (b). By simple visual inspection, there is no apparent difference between the two groups of shapes.

Shape analysis

We have investigated the possible differences in the Corpus Callosum shape associated with gender, handedness and presence of dyslexia by comparing the group means (prototypes) of CC shapes in each data set. The choice of the group mean shapes was motivated by the fact that a simple visual inspection or set of features (e.g., perimeter, area, bounding box, etc.) may not reveal any significant difference between the CC shapes belonging to different classes (e.g., dyslexic vs. normal, see Fig. 6.9). On

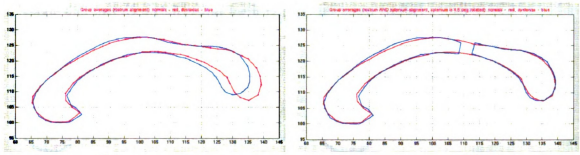
the other hand, several studies reported shape differences associated with sex [32] and schizophrenia [11] groups. We believe that in order to find any differences, one has to compare some representative shapes of the two groups, which have been aligned with respect to the similarity group of planar transformations (scaling, rotations and translations). The problem of defining shape prototypes has attracted considerable interest [34, 12], but practical methods for computing these prototypes are somewhat scarce. This is mainly due to the fact that the statistical shape theory [34, 12] can only be applied to sets of points of equal cardinality between which point correspondences have been established. However, almost all the time, the data that must be processed consists of a set of contours with different point counts and no known point correspondences (Fig. 6.9).

6.2.2 Results

Influence of dyslexia over callosal shape

We computed shape prototypes for the normal and dyslexic groups in the Bergen dataset using the shape learning method presented in Chapter 4. After aligning the *rostrum* of the two prototype shapes, one can notice a four-pixel length difference between them (Fig. 6.10(a)). Moreover, if the dyslexic group prototype is cut into two pieces in the isthmus region, and the two parts are aligned separately to the normal prototype, then there is an almost perfect matching of the *rostrum* and *splenium* parts of the two CC prototype (Fig. 6.10(b)).

Is it only by chance that the dyslexic prototype is 4 pixels shorter than the normal



(a) (b)

Figure 6.10: Comparing the normal (shown in red) and dyslexic (shown in blue) group means of Corpus Callosum shapes in our dataset. (a) *Rostrum* alignment. (b) Both *rostrum* and *splenium* alignment. The dyslexic prototype is actually cut into two parts which are aligned separately. The *posterior midbody* in the dyslexic subjects is significantly shorter than in normal subjects.

prototype, or are the two classes really separable by the *posterior midbody* length? In order to answer this question, we designed a classification system which, given a CC shape instance, computes the length of the *posterior midbody* and assigns it to one of the two groups (normal/dyslexic). Since there are no real anatomical landmarks delimiting the isthmus, we decided to measure its length as follows. Two templates representing the *anterior half* and *posterior third* are computed from the normal Corpus Callosum prototype in Fig. 6.10 by eliminating what we assume to be the posterior midbody. These two templates, shown in blue in Fig. 6.11, are separately aligned to a new shape instance (we actually do not allow a scale factor larger than 5% in order to preserve the relative size ratios of the templates). Subsequently, we compute the distance that separates the aligned templates as the length of the cyan line segment shown in Fig. 6.11. We would like to mention that the actual templates employed are not important for classification purposes, as long as they can be well aligned to the new CC instance. What really matters is the distance that separates them after the alignment. Fig. 6.11 shows the distance computation for a

dyslexic CC instance (left), and for a normal instance (right).

Figure 6.11: Two templates representing the *anterior half* and *posterior third* of the Corpus Callosum are independently aligned with a given CC instance and the distance between the aligned templates (the length of the uniting line segment) is measured. Alignment to a dyslexic CC instance (left) and to a normal instance (right).

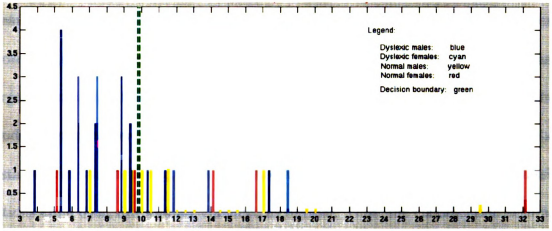
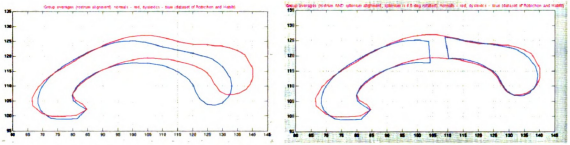


Figure 6.12: Class distributions of the inter-template distances for the Bergen dataset.

The distributions of the inter-template distances for the Bergen set of 26 normal and 28 dyslexic subjects are shown in Fig. 6.12. The two distributions are well separated though some overlap is present. If a threshold is set at 9.8 (the green vertical segment in Fig. 6.12) and all shapes with an inter-template distance smaller than 9.8 are classified as dyslexic while the ones with a greater distance are classified as normal, then there are 12 misclassifications (6 normals classified as dyslexic and 6 dyslexics classified as normal) in the 54-subject data set. Therefore, about 78% of the CC shapes in our dataset can be accurately classified by this method.

We obtained similar results on the dataset of Robichon and Habib [108]. The



(a) (b)

Figure 6.13: Comparing the normal (shown in red) and dyslexic (shown in blue) group means of Corpus Callosum shapes in the Robichon and Habib [108] dataset. (a) *Rostrum* alignment. (b) Both *rostrum* and *splenium* alignment. The dyslexic prototype is actually cut into two parts which are aligned separately. The dyslexic CC prototype is more curved and significantly shorter than the normal prototype.

means of the normal and dyslexic groups are shown in Fig. 6.10. The dyslexic prototype is again about 6 pixels shorter than the normal prototype. It also has a larger curvature, a fact noticed in the original study, too. The classification method produces even better results on this dataset; the distributions of the inter-template distances are shown in Fig. 6.14. If a threshold is set at 8, then the only misclassifications are 4 dyslexics who are classified as normal. Thus, the classification accuracy on this dataset is 86%.

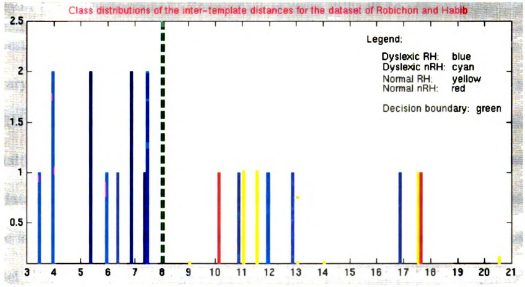


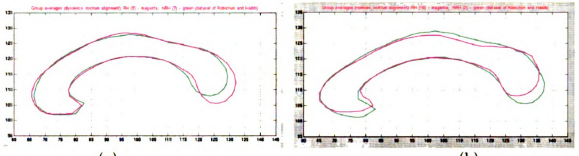
Figure 6.14: Class distributions of the inter-template distances for the Robichon and Habib [108] dataset.

In conclusion, it seems that the length of the posterior midbody can distinguish a dyslexic shape example from a normal one about 80% of the time. Although this accuracy may not be considered high enough for an automatic diagnosis system, we believe it to be a sufficient evidence that the dyslexic CC shapes are indeed shorter than the normal ones.

Callosal shape and hand preference

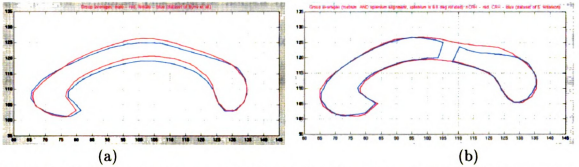
We have tested the influence of the handedness on the CC shape on the following two datasets: Robichon and Habib [108] and Witelson [131] (the part of the dataset which has been published). Both sets contain only males, the former in their twenties, while the later were analyzed after their death (around fifties). The group averages for the Robichon and Habib dataset are shown in Fig. 6.15. The nRH dyslexic prototype is 2-3 pixels shorter than the RH dyslexic prototype (Fig. 6.15(a)). This is confirmed by the inter-template distance distributions shown in Fig. 6.14 (compare the blue (RH) and cyan (nRH) histograms). However, it is more difficult to separate the classes, the best decision threshold (6.3) produces a classification accuracy of 69%. On the contrary, the nRH normal prototype is longer and thicker in the isthmus region than the RH normal prototype (Fig. 6.15(b)). Since the data was insufficient (only 2 normal nRH subjects), one cannot tell if this finding is statistically significant on this dataset. However, the fact that the nRH normal prototype is longer and thicker in the isthmus region than the RH normal prototype remains true for the Witelson dataset (Fig. 6.16(b)). Although we have not attempted to classify the shapes in this set, we believe that the two shape classes are quite separable by the same template

method.



(a) (b)

Figure 6.15: Comparing the Right Handed (RH) (shown in magenta) and non Right Handed (nRH) (shown in green) group means for dyslexic (a) and normal (b) subjects in the Robichon and Habib [108] dataset. The dyslexic nRH prototype is a little shorter than its corresponding RH prototype. The normal nRH prototype is longer and thicker in the isthmus region than its corresponding nRH prototype.



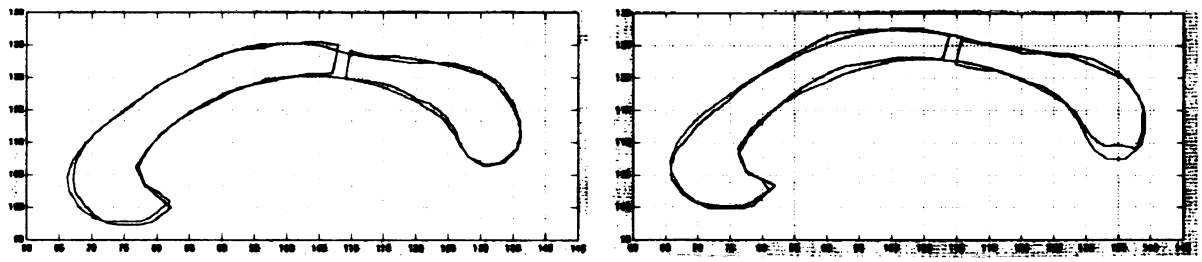
(a) (b)

Figure 6.16: (a) Comparing the male (shown in red) and female (shown in blue) group means for subjects in the Byne *et al.* [18] dataset. There are no significant differences between the two prototypes. (b) Comparing the Right Handed (RH) (shown in blue) and non Right Handed (nRH) (shown in red) group means for the male subjects in the Witelson [131] dataset. The nRH prototype is significantly longer and thicker in the isthmus region than the RH prototype.

Gender differences in callosal shape

Three datasets have been used to investigate gender related differences in callosal shape: our dataset, Byne *et al.* [18] (the part of the dataset which has been published) and Davatzikos *et al.* [32]. This collection of data has much more variance than the data used for hand preference or dyslexia related analysis as far as the age or subject handedness are concerned. Our dataset contains only children (12 ± 1 years old),

that of Byne *et al.* contains adults of various ages (with an average age of about 35 years), while the subjects in the Davatzikos *et al.* are quite old (70 ± 6 years old). Moreover, subject handedness for the Byne *et al.* dataset is unknown. The group averages with respect to gender for our dataset are shown in Fig. 6.17. Both the dyslexic (Fig. 6.17(a)) and normal (Fig. 6.17(b)) female prototypes are about 2 pixels shorter than their corresponding male prototypes. An analysis of the posterior midbody length distributions in Fig. 6.12 reveals that the variance of the female population is substantially larger than the variance of the male population, therefore we do not consider the 2-pixel length difference in group means to be significant. Except for the length difference, the prototypes in the two pairs are almost identical. A shape analysis of the Byne *et al.* dataset (Fig. 6.16(a)) does not reveal any major difference between the male and female prototypes. The only significant difference was found in the dataset of Davatzikos *et al.* (Fig. 6.18(a)). The posterior part of the female prototype is shorter and more bulbous than the male prototype. Since this dataset is small, we have not checked how well these differences can separate the two classes.



(a) (b)

Figure 6.17: Comparing the male (shown in red) and female (shown in blue) group means for dyslexic (a) and normal (b) subjects in the Bergen dataset. The female CC prototype in both dyslexic and normal subjects is shorter (though not significantly) than its corresponding male prototype. Except for the length, the male and female prototypes are practically identical.

Discussion and conclusions

We would like to briefly compare the results produced by our shape analysis method with those in the original studies [128, 108, 18, 131, 32]. On the datasets of Byne *et al.* and Davatzikos *et al.* our results are identical to those reported in [18, 32]. A comparison between the group means produced by our shape learning method and that of Davatzikos *et al.* [32] (Fig. 11) is shown in Figs. 6.18(b) and (c). The prototypes in each pair are almost identical (up to a similarity transformation) despite the fact that they were produced using different methods. The shape analysis not only confirmed but also further clarified the influence of handedness on Corpus Callosum reported by Robichon and Habib [108] and Witelson [131]. And finally, the most important result of this work, stating that the dyslexic population can be relatively well separated from the normal population based on the length of the posterior midbody, was not reached (and we believe it cannot be obtained based only on area measurements) neither by Robichon and Habib [108] nor by our previous study [128].

This study leaves unresolved the contradicting results concerning gender influences on Corpus Callosum. However, it seems to us that most analyses that reported a more bulbous splenium in females were done on elderly while those that could not find any differences had younger subjects (see Table 1). The contradicting findings may become consistent if "a different or earlier ageing process may occur in the brains of men than in women" as Witelson noted in [131].

Finally, a short note on the computational aspects of our method. The prototype

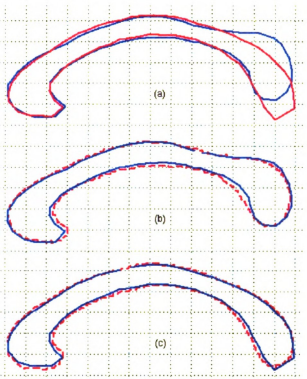


Figure 6.18: Comparing the male (shown in red) and female (shown in blue) group means for subjects in the Davatzikos *et al.* [32] dataset. The female CC prototype is shorter and has a more bulbous posterior part than the male prototype. Overlay of the female (b) and male (c) group means obtained by us (shown in blue) and by Davatzikos *et al.* [32], (shown in red). The prototypes in each pair are almost identical (up to a similarity transformation) despite the fact that they have been obtained using different methods.

computation does not require human intervention once the contours have been extracted from images; several automatic methods for tracing the Corpus Callosum in MR images have been reported. The computation time required for computing the two class prototypes is (on an average) 90 seconds per CC shape in the dataset on a Sun UltraSparc 10 computer.

6.3 Summary

In this chapter we have demonstrated how the shape of segmented objects can be used for further object classification or identity verification. A hand shape-based personal verification system has been presented and its performance has been evaluated and discussed. We also have demonstrated how morphometric analysis of Corpus Callosum shapes can reveal differences associated with gender, handedness or presence of dyslexia.

Chapter 7

Conclusions and Future Work

As we have discussed in Chapter 2, reliable detection, segmentation and subsequently matching of objects in images is still an open problem. Although several object recognition systems have been reported in the literature, they are rarely able to handle arbitrary backgrounds or occlusion, not to mention images that do not contain any of the objects these systems have been trained on. In his thesis, Rowley [110] estimated that the prior probability that a window of arbitrary size extracted from 130 real world images contains a human face is $1/20,984$. We believe that the prior probabilities for other objects are of the same order of magnitude as well. This makes the object detection problem one of the most difficult ones which computer vision and pattern recognition has to face.

The main purpose of this dissertation is to integrate some of the most recent advances in computer vision and pattern recognition together with new ideas in designing a global object learning and retrieval system. We would like to point out that integration is a very difficult task; almost all methods employed or at least evaluated

throughout this thesis are the result of recent Ph.D. theses (e.g., the *mean-shift unsupervised segmentation* [23], the *maximum discrimination* object detection [22]) or other type of academic research work. None of these methods is easy to implement, and many times one cannot implement them without the cooperation of their authors.

There are several important ideas beginning to develop in the vision community which we advocate in this thesis:

1. *A general object detection/recognition/matching system should be based on learning.* Here, learning denotes not only the estimation of the parameters of an (often sophisticated) model from training examples, but also the strategy of selecting the training examples according to their discrimination power. We have developed a learning-based system that is able to detect, segment and match objects in several diverse applications and we have evaluated its performance on large test sets.

2. *The necessity for an automatic model design and training.* We believe that a general mechanism for object modeling is needed; this mechanism should be general enough to be applied to most objects, and yet it should be able to distinguish not only between the objects learned, but between each object learned and the rest of the world. The current principal component analysis-based recognition systems are able to encompass any type of object, but they cannot accurately reject patterns representing non-objects. It is not yet clear how appearance information can be combined with shape information in developing a global model for object detection. The well known "Active appearance model" [24] that combines the two types of clues, also based on principal component analysis, is not yet reliable enough for object detection. For the time being we have developed separate appearance and shape

models but we detect objects based only on one of them (depending on whether the object of interest is better defined by shape or appearance). In any case, no matter what the model is, one should be able to automatically train it. This requires the training examples be automatically aligned in order to reduce the amount of variation between instances of the same object [110]. For appearance-based models, the alignment is currently done manually. One main advance provided by our work is an automatic method for aligning shape examples and training shape models.

3. *The necessity for an adequate object similarity measure.* We have introduced and evaluated a shape-based object similarity measure (mean alignment error). A related measure applying to appearance models was designed by Moghaddam *et al.* [92]. Both measures are based on some sort of warping (either shape or appearance), and work very well in case of small object differences (that is, when comparing two objects of the same type, like two human faces or hands). However, for large object differences, it is still not clear how robust the current similarity measures (like the *distance within and from feature space* of Moghaddam and Pentland [93]) are.

There are several aspects of our work which, we believe, need further investigation:

1. It is important to make the shape learning process *incremental.* Our current method (Chapter 4) aligns (and computes a shape distance between) *every pair* of shapes in the training set. As a consequence, for large training sets, the number of alignments that have to be performed becomes prohibitive. An incremental learning strategy attempts to "grow" shape clusters online; a new shape example is assigned to one of the current clusters whose prototype is "closest" (based on the alignment error) if the distance between the two is small enough (see Section 4.6.2). However, if

the new shape example cannot properly fit any of the current clusters, it will become the starting seed of a new cluster.

2. Refining the left ventricle localization procedure (Section 3.2) and designing a new segmentation paradigm for tracing the ventricle walls since it appears that the results produced by the current deformable template method are not sufficiently accurate. The approach we envision is shown in Fig. 7.1: the profiles corresponding to each detected box in Fig. 7.1(a) are warped onto their corresponding average feature profiles and the location of two salient points (the intersection between the profile and the medial axis of the ventricle wall) on each profile is estimated (Fig. 7.1(c)). Then, by applying a voting procedure, the center and the medial axis of the ventricle wall can be estimated quite reliably (Fig. 7.1(d)). Finally, a radial tracking procedure can be employed to produce the exact segmentation of the ventricle wall.

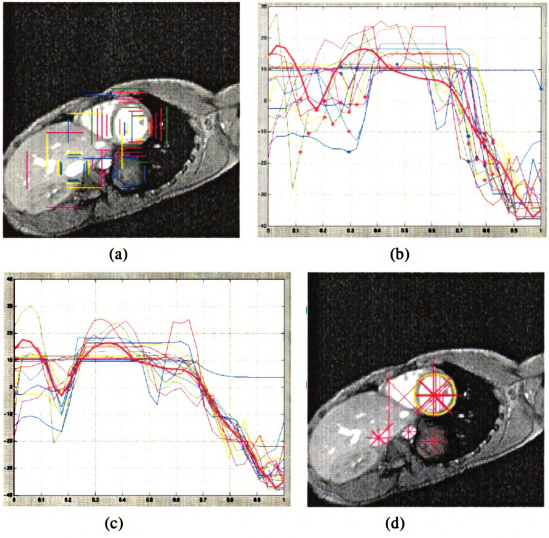


Figure 7.1: Detailed localization of the left ventricle. a) Multiple detections produced by the maximum discrimination method (Section 3.2). b) Horizontal feature profiles corresponding to each detected box in (a). An average horizontal profile is shown in red. c) The profiles in (b) aligned to the average profile using dynamic time warping. d) The feature profiles corresponding to the detected boxes in (a) *after warping to their corresponding average profiles.* A voting procedure is applied to estimate the center and the medial axis of the ventricle wall (shown in yellow).

Bibliography

Bibliography

[1] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines: a Stochastic Approach to Combinatorial Optimization and Neural Computing.* Wiley, Chichester, 1989.

[2] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 12(9):855–867, 1990.

[3] Y. Amit, D. Geman, and B. Jedynak. Efficient focusing and face detection. In *Face Recognition: From Theory to Applications*, H. Wechsler *et al.* (*eds.*), 1997. NATO ASI Series F, Springer Verlag, Berlin.

[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] S. Arya, D. M. Mount, N. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate neareast neighbor searching in fixed dimension. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 573–582, 1994.

[6] J. Beis and D. Lowe. Indexing without invariants in 3D object recognition. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 21(10):1000–1015, 1999.

[7] S. Ben-Yacoob. *Fast object detection using MLP and FFT.* IDIAP Research Report 97-11, IDIAP, Switzerland, 1997.

[8] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 14(2):239–256, 1992.

[9] D. Beymer and T. Poggio. Image Representations for Visual Learning. *Science*, 272:1905–1909, 1996.

[10] B. Bhanu and T. Jones. Image understanding research for automatic target recognition. *IEEE AES Systems Magazine*, pages 15–22, 1993.

[11] F. L. Bookstein. Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–244, 1997.

[12] F. L. Bookstein. Shape and the information in medical images: A decade of the morphometric synthesis. *Computer Vision and Image Understanding*, 66:97–118, 1997.

[13] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 10:849–865, 1988.

[14] J. G. Bosch, J. H. C. Reiber, Burken G., J. J. Gerbrands, A. Kostov, van de A. J. Goor, M. Daele, and J. Roelander. Developments towards real time frame-to-frame automatic contour detection from echocardiograms. *Computers in Cardiology*, pages 435–438, 1991.

[15] K. W. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical ROC curves. In *Proceedings of CVPR-'99*, pages 354–359, Fort Collins, CO, 1999.

[16] R. Brunelli and T. Poggio. Face recognition: Features versus Templates. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 15(10):1042–1052, 1993.

[17] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[18] W. Byne, R. Bleier, and L. Houston. Variations in human Corpus Callosum do not predict gender: A study using Magnetic Resonance Imaging. *Behavioral Neuroscience*, 102:222–227, 1988.

[19] A. Califano and R. Mohan. Multidimensional indexing for recognizing visual shapes. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 16(4):373–392, 1994.

[20] C. H. Chiu and D. H. Razi. A nonlinear multiresolution approach to echocardiographic image segmentation. *Computers in Cardiology*, pages 431–434, 1991.

[21] L. Cohen. Note on active contour models and balloons. *Comp. Vision, Graphics, and Image Proc.*, 53(2):211–218, March 1991.

[22] A. Colmenarez and T. Huang. Face detection with information-based maximum discrimination. In *Proceedings of CVPR-'97*, pages 782–787, San Juan, PR, 1997.

[23] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proceedings of CVPR '97*, pages 750–755, Code posted at *http://www.caip.rutgers.edu/ ~comanici/segm_images.html*, San Juan, PR, 1997.

[24] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of European Conference on Computer Vision*, pages 484–498, Freiburg, Germany, 1998.

[25] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. Use of active shape models for locating structures in medical images. *Image & Vision Computing*, 12(6):355–366, 1994.

[26] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. In *Proceedings of British Machine Vision Conference*, pages 110–119. BMVA Press, 1997.

[27] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.

[28] I. J. Cox, S. B. Rao, and Y. Zhong. "Ratio regions" a technique for image segmentation. In *Proceedings of the International Conference on Pattern Recognition*, pages 557–564, Vienna, Austria, 1996.

[29] R. W. Cox. AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research, also at (http://varda.biophysics.mcw.edu/~cox)*, 29:162–173, 1996.

[30] I. Craw, N. Costen, T. Kato, and S. Akamatsu. How should we represent faces for automatic recognition? *IEEE Trans. Pattern Anal. and Machine Intelligence*, 21(8):725–736, 1999.

[31] J. Daugman. Recognizing persons by their iris pattern. In *Biometrics: Personal Identification in Networked Society*, pages 103–121, A. Jain, R. Bolle and S. Pankanti (*eds.*). Kluwer Academic, Boston, 1999.

[32] C. Davatzikos, M. Vaillant, S. M. Resnick, J. L. Prince, S. Letovsky, and R. N. Bryan. A computerized approach for morphological analysis of the corpus callosum. *J. Comp. Assis. Tomogr.*, 20:88–97, 1996.

[33] C. de Lacoste-Utamsing and R. L. Holloway. Sexual dimorphism in the human Corpus Callosum. *Science*, 216:1431–1432, 1982.

[34] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* Wiley, New York, 1998.

[35] R. Duara, A. Kushch, K. Gross-Glenn, W.W. Barker, B. Jallad, S. Pascal, D.A. Loewenstein, J. Sheldon, M. Rabin, B. Levin, and H. Lubs. Neuroanatomic differences between dyslexic and normal readers on magnetic resonance imaging scans. *Archives of Neurology*, 48:410–416, 1991.

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

[37] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley, New York, 2000.

[38] J. Duncan and N. Ayache. Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 22(1):85–106, 2000.

[39] N. Duta and A. Jain. Learning the human face concept in black and white images. In *Proceedings of ICPR '98*, pages 1365–1367, Brisbane, Australia.

[40] N. Duta, A. K. Jain, and M. P. Jolly. Automatic construction of 2D shape models. *To appear in IEEE Trans. Pattern Anal. and Machine Intelligence.*

[41] N. Duta, A. K. Jain, and M. P. Jolly. Learning-based object detection in cardiac MR images. In *Proceedings of ICCV '99*, pages 1210–1216, Corfu, Greece, 1999.

[42] N. Duta, A. K. Jain, and K. V. Mardia. *Matching of palmprints.* Submitted, 2000.

[43] N. Duta and M. Sonka. An improved active shape model: handling occlusion and outliers. In *Proceedings of ICIAP '97*, volume 1310, pages 398–405. Springer-Verlag, 1997.

[44] N. Duta and M. Sonka. Segmentation and interpretation of MR brain images: An improved active shape model. *IEEE Trans. Med. Imaging*, 17(6):1049–1062, 1998.

[45] G. J. Edwards, T. F. Cootes, and C. J. Taylor. Face recognition using active appearance models. In *Proceedings of European Conference on Computer Vision*, pages 581–695, Freiburg, Germany, 1998.

[46] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *Int. J. Comp. Vision*, 18:99–119, 1996.

[47] M. Figueiredo, J. Leitao, and A. K. Jain. Unsupervised Contour Representation and Estimation Using B-splines and a Minimum Description Length Criterion. *to appear in IEEE Trans. Image Proc.*, 2000.

[48] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24:381–395, 1981.

[49] R. Fisker, N. Schultz, N. Duta, and J. Carstensen. A general scheme for training and optimization of the Grenander deformable template model. In *Proceedings of CVPR 2000*, Hilton Head, SC, 2000.

[50] P. Fua and C. Brechbuhler. Imposing hard constraints on soft snakes. In *Proc. European Conference on Computer Vision*, volume II, pages 495–506, Cambridge, UK, 1996.

[51] D. Geiger, A. Gupta, L. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 17(3):294–302, 1995.

[52] S. Gold, A. Rangarajan, C. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2D and 3D point matching. *Pattern Recognition*, 31(8):1019–1031, 1998.

[53] S. Gold, A. Rangarajan, and E. Mjolsness. Learning with preknowledge: clustering with point and graph matching distance measures. *Neural Computation*, 8(4):787–804, 1996.

[54] C. Goodall. Procrustes methods in the statistical analysis of shape. *J. Royal Stat. Soc. B*, 53(2):285–339, 1991.

[55] R. M. Gray. *Entropy and Information Theory.* Springer-Verlag, Berlin, 1990.

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

[57] A. Gueziec and N. Ayache. Smoothing and matching of 3-D space curves. *Int. J. Comp. Vision*, 12:79–104, 1994.

[58] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision, Volume I.* Addison-Wesley, Reading, MA, 1992.

[59] A. Hill, A. D. Brett, and C. J. Taylor. Automatic landmark identification using a new method of non-rigid correspondence. In *Proceedings of IPMI '97*, pages 483–488, Poultney, VT, 1997.

[60] A. Hill, C. J. Taylor, and A. D. Brett. A framework for automatic landmark identification using a new method of nonrigid correspondence. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 22(3):241–251, 2000.

[61] B. K. P. Horn. Closed form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4:629–642, 1987.

[62] *http : //www − ai.informatik.uni − dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html.*

[63] *http : //www.cis.hut.fi/research/som_lvq_pak.shtml.*

[64] *http : //www.cs.umd.edu/ ∼ mount/ANN.*

[65] *http : //www.ius.cs.cmu.edu/IUS/usrp0/har/FaceDemo/galleryinline.html.*

[66] A. Hurlbert and T. Poggio. Do computers need attention? *Nature*, 321:651–652, 1986.

[67] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 15(9):850–863, 1993.

[68] G.W. Hynd, J. Hall, E.S. Novey, D. Eliopulos, K. Black, J.J. Gonzales, J.E. Edmonds, C. Riccio, and M. Cohen. Dyslexia and Corpus callosum morphology. *Archives of Neurology*, 52:32–38, 1995.

[69] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 20(6):577–589, 1998.

[70] A. Jain, R. Bolle, and S. Pankanti. Introduction to biometrics. In *Biometrics: Personal Identification in Networked Society*, pages 1–41, A. Jain, R. Bolle and S. Pankanti (*eds.*). Kluwer Academic, Boston, 1999.

[71] A. Jain, R. Bolle, and S. Pankanti. *Biometrics: Personal Identification in Networked Society.* Kluwer Academic, Boston, 1999.

[72] A. K. Jain and N. Duta. Deformable matching of hand shapes for user verification. In *Proceedings of ICIP '99*, Kobe, Japan, 1999.

[73] A. K. Jain and P. Flynn. Image segmentation using clustering. In *Advances in Image Understanding*, pages 65–83, K. Bowyer and N. Ahuja (*eds.*). IEEE Computer Society Press, 1996.

[74] A. K. Jain, A. Ross, and S. Pankanti. A prototype hand geometry-based verification system. In *2nd Int'l Conference on Audio- and Video-based Biometric Person Authentication*, Washington D.C., 1999.

[75] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, B. Schlkopf, C. Burges and A. Smola (*eds.*). MIT Press, 1999.

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

[77] T. Kohonen. *Self-Organizing Maps.* Springer Series in Information Sciences, Heidelberg, 1997.

[78] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola. LVQ_PAK: The Learning Vector Quantization Program Package. *Helsinki University of Technology, Report A30*, 1996.

[79] S. Lakshmanan and H. Grimmer. A deformable template approach to detecting straight edges in radar images. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 18(4):438–443, 1996.

[80] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 19(7):743–756, 1997.

[81] J.P. Larsen, T. Høien, and H. Ødegaard. Magnetic resonance imaging of the Corpus callosum in developmental dyslexia. *Cognitive Neuropsychology*, 9:123–134, 1992.

[82] T. K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labelled graph matching. In *Proceedings of ICCV-'95*, Cambridge, MA, 1995.

[83] M. Lew and N. Huijsmans. Information theory and face detection. In *Proceedings of ICPR-'96*, pages 601–610, Vienna, Austria, 1996.

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

[85] S. H. Lin, S. Y. King, and L. J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. Neural Networks*, 8(1):114–131, 1997.

[86] A. Lundervold, N. Duta, T. Taxt, and A. K. Jain. Model-guided segmentation of Corpus Callosum in MR images. In *Proceedings of CVPR '99*, pages 231–237, Fort Collins, CO, 1999.

[87] J. B. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[88] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[89] S. Menet, P. Saint-Marc, and G. Medioni. B-snakes: Implementation and application to stereo. In *DARPA Image Understanding Workshop*, pages 720–726, 1990.

[90] T. Mitchell. Neural networks for face recognition. *http:// www.cs.cmu.edu/afs/ cs.cmu.edu/user/ mitchell/ftp/faces.html.*

[91] T. Mitchell. *Machine Learning.* McGraw Hill, New York, 1997.

[92] B. Moghaddam, C. Nastar, and A. Pentland. A Bayesian similarity measure for direct image matching. In *Proceedings ICPR-'96*, Vienna, Austria, 1996.

[93] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 19(7):696–710, 1997.

[94] S. K. Nayar, H. Murase, and S. Nene. Parametric Appearance Representation. In *Early Visual Learning*, pages 131–160, S. K. Nayar and T. Poggio (*eds.*). Oxford University Press, 1996.

[95] A. Neumann and C. Lorenz. Statistical shape model based segmentation of medical images. *Computerized Medical Imaging and Graphics*, 22:133–143, 1998.

[96] E. Osuna, R Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proceedings of CVPR-'97*, pages 130–136, San Juan, PR, 1997.

[97] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of ICCV '98*, pages 555–562, Bombay, India, 1998.

[98] B. F. Pennington, P. A. Filipek, D. Lefly, J. Churchwell, D. N. Kennedy, J. H. Simon, C. M. Filley, A. Galaburda, M. Alarcon, and J. C. DeFries. Brain morphometry in reading-disabled twins. *Neurology*, 53:723–729, 1999.

[99] J. P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16:295–306, 1998.

[100] P. J. Phillips, H. Moon, P. Raus, and S. A. Rizvi. The FERET evaluation methodology for face-recognition algorithms. In *Proceedings of CVPR-'97*, pages 137–143, San Juan, PR, 1997.

[101] T. Poggio and D. Beymer. Regularization Networks for Visual Learning. In *Early Visual Learning*, pages 43–66, S. K. Nayar and T. Poggio (*eds.*). Oxford University Press, 1996.

[102] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, 1993.

[103] A. L. Ratan, W. E. L. Grimson, and W. M Wells. Object detection and localization by dynamic template warping. In *Proceedings of CVPR '98*, pages 634–640, Santa Barbara, CA, 1998.

[104] J. Ratches, C. Walters, R. Buser, and B. Guenther. Aided and automatic target recognition based upon sensory inputs from image forming systems. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 19(9):1004–1019, 1997.

[105] R. A. Rauch and J. R. Jinkins. Variability of corpus callosal area measurements from midsagittal MR images: effect of subject placement within the scanner. *American Journal of Neuroradiology*, 17:27–28, 1996.

[106] R. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1989.

[107] F. Robichon. *Personal communication*, 1999.

[108] F. Robichon and M. Habib. Abnormal callosal morphology in male adult dyslexics: relationship to handedness and phonological abilities. *Brain and Language*, 62:127–146, 1998.

[109] R. Ronfard. Region-based strategies for active contour models. *Int. J. Comp. Vision*, 13(2):229–251, October 1994.

[110] H. Rowley. *Neural Network-based Face Detection.* Ph.D. thesis, Carnegie Mellon University, 1999.

[111] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 20(1):23–38, 1998.

[112] J. M. Rumsey, M. Casanova, G.B. Mannheim, N. Patronas, N. De Vaughn, S.D. Hamburger, and T. Aquino. Corpus callosum morphology, as measured with MRI, in dyslexic men. *Society of Biological Psychiatry*, 36:769–775, 1996.

[113] H. Schneiderman. *A Statistical Approach to 3D Object Detection.* Ph.D. thesis, Carnegie Mellon University, 2000.

[114] H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. In *Proceedings of CVPR '98*, pages 45–51, Santa Barbara, CA, 1998.

[115] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proceedings of CVPR 2000*, Hilton Head, SC, 2000.

[116] S. Sclaroff. Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition*, 30(4):1019–1031, 1997.

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

[118] J. Shufelt. Performance evaluation and analysis of monocular building extraction from aerial imagery. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 21(4):311–326, 1999.

[119] C. G. Small. *The Statistical Theory of Shape.* Springer-Verlag, Berlin, 1996.

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

[121] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 20(1):39–52, 1998.

[122] J. Ton and A. K. Jain. Registering Landsat images by point matching. *IEEE Trans. Geosci. Remote Sensing*, 27(5):642–651, 1989.

[123] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[124] S. Ullman. Aligning pictorial descriptions: An approach to object recognition. *Cognition*, 32(3):193–254, 1989.

[125] A. Vailaya, Y. Zhong, and A. K. Jain. A hierarchical system for efficient image retrieval. In *Proceedings of the International Conference on Pattern Recognition*, pages 356–360, Vienna, Austria, 1996.

[126] R. Vaillant, C. Monrocq, and Y. Le Cun. An original approach for the localization of objects in images. *IEE Proceedings Vision, Image and Signal Processing*, 141(4):245–250, 1994.

[127] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, Heidelberg, 1995.

[128] K. von Plessen *et al.* Size and shape of the Corpus Callosum in children with dyslexia - an MRI study. *In preparation.*

[129] J. Weng, N. Ahuja, and T. S. Huang. Learning, recognition and segmentation using the Cresceptron. *Int. J. Comp. Vision*, 25:105–139, 1997.

[130] J. Weng, A. Singh, and M. Y. Chiu. Learning-based ventricle detection from cardiac MR and CT images. *IEEE Trans. Med. Imaging*, 16(4):378–391, 1997.

[131] S. Witelson. Hand and sex differences in the isthmus and genu of the human Corpus Callosum. *Brain*, 112:799–835, 1989.

[132] H. Wu, Q. Chen, and M. Yachida. Face detection from color images using a fuzzy pattern matching method. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 21(6):557–564, 1999.

[133] C. Xu and J. L. Prince. Snakes, Shapes, and Gradient Vector Flow. *IEEE Trans. Image Proc.*, pages 359–369, March 1998.

[134] K. C. Yow and R. Cipolla. Feature-based human face detection. *Image and Vision Computing*, 15(9):713–735, 1997.

[135] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *Int. J. Comp. Vision*, 8(2):133–144, 1992.

[136] Y. Zhong and A. K. Jain. Object localization using color, texture and shape. In *Proc. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 279–294, Venice, 1997. Springer-Verlag.

[137] R. Zunkel. Hand geometry based verification. In *Biometrics: Personal Identification in Networked Society*, pages 87–101, A. Jain, R. Bolle and S. Pankanti (*eds.*). Kluwer Academic, Boston, 1999.