Recognition of Visual Object Classes

Thesis by

Michael C. Burl

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1997

(Submitted November 11, 1996)

Acknowledgements

As I stand on the verge of completing the requirements for the Doctor of Philosophy degree, it is natural to look back on my life and acknowledge some of the people who have helped me reach this point.

First and foremost, I would like to thank my wife Maureen. From leaving her family behind in New England to learning to sleep with the lights on so I could do my late night problem sets to running volcano experiments at JPL, she has been by my side all the way and has been extremely supportive.

Next I would like to thank my family. My parents and grandparents have done so much for me it is impossible to write anything adequate. The same goes for my great uncle George who lived with us while I was growing up and was always there for me. My uncles, Jeff and Ron Burl, and my cousin, Mike Elliott, helped me get ahead in math and science and provided the scholastic example I tried to emulate. Despite frequent fights with my brother when we were younger, we have managed to become close friends. I think my going away to college (out of state) probably helped that process, but now that I have been in school for so long we are both hoping to see each other more.

I would also like to thank some of my early teachers, especially the ones who allowed and even encouraged me to learn at my own pace. Sally Coohon was my math and science teacher for fifth and sixth grade. After some initial testing, she gave me an eighth grade math book and said, "Do the problems and grade them yourself. If you need help with any of the topics, come see me." Away I went. In junior high I unfortunately had to sit through adding two and three digit numbers again despite efforts by my mother to change the system. In high school I was able to get another good math teacher, Jack Nutter, whose philosophy was: as long as you knew the material and did well on the tests you didn't need to turn in any homework. He gave me some more-advanced books to study on my own, which enabled me to start

taking calculus and statistics at a local college by the end of my sophomore year. My guidance counselor, Zay Reynolds, was instrumental in helping me investigate colleges. Among other things, she gave me a book which listed Caltech as the school with the highest freshman math SAT scores. (I had not even heard of Caltech before.) After further investigation, I decided to apply and come to Caltech as an undergraduate — in retrospect, an excellent choice.

My closest friends from back in Ovid, Michigan were Dan Hill, Steve and Scott Gardner, Allen Ward, Jim Staley, and Tim Personious. We had a lot of fun growing up together, although we have largely lost touch now. As an undergraduate at Caltech, John Mann, Steve Roskowski, and I did everything together including cramming into Steve's Fiat convertible to make food runs after preseason football practice. John and I kept in contact after leaving Caltech as we both went to work at MIT Lincoln Laboratory. At LL I made many new friends: Shawn Verbout, Jonathan Marsden, Bill Irving, Michael Sechtin, Greg Owirka, and Jeff Nanis. These guys "kept me sharp" by constantly testing me with puzzles and math problems on my white board, which made the transition back to graduate school easier. It is hard to believe, but Mike and Jeff have passed away. They were both young and bright with a lot of life still to be lived. It seems unfair that they didn't get the chance.

I would also like to thank Les Novak, my supervisor at LL, who really got me interested in target detection and classification. Les' influence on my interests and my method of approaching problems still remains and is apparent throughout this thesis. Les, as well as my group leader, Gerald Morse, and Professor Berthold Horn of MIT, were kind enough to write the graduate school recommendation letters that got me back into Caltech.

After the first year of taking classes, I joined the research group of Pietro Perona who at that time was a new professor at Caltech. Pietro turned out to be a good advisor with a laid back style that suited me well. I enjoyed the research group, especially the *asados*. Special acknowledgements go to the people with whom I worked most closely: Thomas Leung, Markus Weber, Mario Munich, Luis Goncalves, Jean-Yves Bouguet, Chris Kolb, and Joe Weber. Markus, thanks especially for putting

together the live demo for the thesis defense; I know at least one committee member left the room quite excited after seeing the demo. Thanks also go to Kevin Stiemke who spent a summer in our lab putting together a JAVA applet demonstrating the face localization work. In addition, I would like to thank Irene Loera and Lavonne Martin for their assistance, as well as Bob Freeman for keeping the computers up and running (and finding extra disk space when I needed it most). Also, thanks to Pili Munich for generating the cursive handwriting data used in the Chapter 8 experiments.

Next I would like to thank the members of my candidacy and thesis defense committees. The candidacy committee consisted of Pietro, Demetri Psaltis, Marvin Simon, Michelle Effros, and Joel Franklin. The thesis committee consisted of Pietro and Demetri again, along with Yaser Abu-Mostafa, Padhraic Smyth, and Usama Fayyad. Their questions and comments have made for a better thesis.

After leaving Caltech, I will be joining the Machine Learning Systems Group at JPL. I already have a long history with this group dating from the summer of 1992 when I started working on the volcano problem with Usama and Padhraic. At various times, I have benefited from discussions with Steve Chien, Joe Roden, and Alex Gray. I would also like to thank Paul Stolorz, Shakti Walia, and Rich Doyle for their efforts to get me an excellent offer. In addition, I would like to thank geologists Jayne Aubele and Larry Crumpler of Brown University for their assistance in labeling and analyzing the Magellan volcano data.

There are numerous others who have made my life at Caltech more fun. Kristina Fayyad has been a good friend and was especially supportive during a difficult time. We thank Len and Shelley Mueller, Jim Ostrowski, and Liz Price for encouraging us to apply for a position as Resident Associates. Thanks also to Kim West for actually hiring us. It has been a lot of fun working with the students and other R.A.'s. I thank all the Fleming students for not causing any disasters this term to sidetrack my thesis. In return, I hope that we have made their lives here more fun.

I have made many good friends at Caltech through sports, especially volleyball. Although there is not enough space to list everyone, Jose Navarro (and parents), Paul and Stacey Barriga, Jennifer Harris, Aaron and Melinda Kiely, Andy Berkin, Andreas

Masuhr, Dave and Lourdes Szumlas, Richard Chin, Steve Lundy, Ingrid Wysong, and Dan Bridges deserve to be singled out. I would also like to thank the women on this year's undergraduate volleyball team for their efforts to make the season fun. I almost forgot to mention my volleyball friends from New England: Joe Landry, Mary McCabe, Amy Lewis, Salah Chnioui, Sergio Cafarelli, and Elizabeth Coliaguri.

Thanks also to Sharon Lacey and "The Kid" (Kimberly) for helping out after Orion was born this summer, and thanks to Maureen's parents, Pat and Don Adams, for taking care of him during the fall so I could work on my thesis and assist Maureen coaching volleyball.

Finally, I would like to dedicate this thesis to my new son, Orion. At five months of age, he is already better at visual recognition than my computer algorithms, which I have worked on for five years. Maybe someday he too will wonder about the mathematics underlying visual recognition, and maybe someday he will be able to solve the problems that I have only started.

Sponsors:

I would like to thank the people and organizations that have helped fund my education. For my undergraduate support, I would like to thank the Michigan Mathematics Prize Competition as well as the Fred and Florence Dengler Scholarship Fund. I would also like to thank Federal Mogul Corporation for sponsoring me through a National Merit Scholarship. In addition, Caltech provided a number of grants and loans that made my undergraduate education possible.

My graduate work, i.e., the work reported in this thesis, was supported in part by the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation Engineering Research Center Program and by the California Trade and Commerce Agency, Office of Strategic Technology. Support for the face localization work was provided by INTEL Corporation. The work on automatically cataloging the volcanoes of Venus was carried out in collaboration with the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Financial support was provided by JPL DDF grant

#61584.

Abstract

Humans can look at a scene or a photograph and easily recognize objects. Outside my window I can see cars, people walking a dog on a brick pathway, trees, buildings, etc. This perception is so effortless that it belies the difficulty of the task. Visual perception begins with light that is reflected from the scene into the eye. The light impinges upon the retina and is transduced by a two-dimensional array of photoreceptors into noisy electrical signals. The brain must then accomplish the difficult task of transforming from this low-level representation to a higher-level understanding of the scene in terms of regions, surfaces, textures, and objects.

For computer vision the problem is the same, but the hardware is different. A camera approximates the function of the eye and retina; that is, the camera produces a two-dimensional array of numbers (pixel values) representing the intensity of light reflected from the scene. The fundamental question addressed in this thesis is the following: *what mathematical processing should be applied to the pixel values in order for a computer to recognize objects?* The methods we propose are not intended as a model of human brain function, although they may provide some insight. We are simply trying to solve the same visual recognition problems as the brain without concern for whether (or how) our algorithms could be realized in neuronal "hardware."

We have developed a new framework for recognizing visual object classes in which the class members consist of characteristic parts in a deformable spatial configuration. Human faces are an object class of this type, since faces consist of eyes, nose, and mouth arranged in a configuration that varies depending on expression and pose and also from one person to another. A second object class is cursive handwriting, which consists of loops, cusps, crossings, etc. arranged in a deformable pattern. In our approach, the allowed object deformations are represented through shape statistics, which are learned from examples. Instances of an object in an image are detected by finding the appropriate features in the correct spatial configuration. Our algorithm is

robust with respect to partial occlusion, detector false alarms, and missed features.

Potential applications include intelligent tools for finding objects in image databases, human-machine interfaces, user authentication, intelligent data gathering and compression, signature verification, and keyword spotting. Experimental results will be presented for two problems: (1) locating quasi-frontal views of human faces in cluttered scenes and with occlusions and (2) spotting keywords in on-line cursive handwriting data.

Contents

List of Figures

Chapter 1 Introduction

1.1 The Problem of Visual Object Recognition

Humans can look at a scene and easily identify the objects that are present. Outside my window I can see and recognize cars, people walking a dog on a brick pathway, trees, buildings, etc. How is this possible?

In ancient times (circa 6th century BC) it was incorrectly believed that vision worked via "feeler rays," which were sent out from the eye to probe an object. This idea, attributed to Pythagoras, persisted until at least 1000 AD [Enc] and by some accounts even until the 1600's [UCI]. We now know that in the correct interpretation of vision, the eye is a passive sensor; the sensation of sight occurs when light is reflected *from* an object *into* the eye. This idea was reportedly put forward by the Greek philosopher Epicurus (circa 300 BC) [Enc], but it was not widely accepted until the feeler-ray theory was finally discredited by Christopher Scheiner, who in 1625 demonstrated that an optical image is formed on the inside rear wall of the eyeball [UCI].

At the rear of the eyeball is a two-dimensional array of photoreceptors — the retina. Light striking the retina is transduced into (noisy) electrical signals. Here the truly difficult part of vision begins: the low-level representation of the world provided by the retina must be processed by the brain into a higher level understanding in terms of regions, surfaces, textures, and objects.

For computer vision the problem is the same, but the hardware is different. A camera approximates the function of eye and retina, producing a two-dimensional array of numbers (pixel values), which represent the intensity of light reflected from the scene. For scenes that vary in time, the camera provides a time-sampled sequence of two-dimensional snapshots. The fundamental question of object recognition is the following: *what mathematical processing should be applied to the pixel values (or*

sequence of values) in order for the computer to recognize an object?

We know from our practical experience that visual cues such as motion, stereo (vision with two eyes), and color can be quite helpful for object recognition. But, we also know that, in most cases, the same objects can be recognized from black and white, still-frame photographs. This fact indicates that reliable computer object recognition should be possible even from static, monocular, monochrome images.

There are a number of difficulties, however. The object must first be segmented from the background. Since the object pose, camera position, and lighting conditions all affect the appearance of the object (pixel values), algorithms must be invariant or at least insensitive to these effects. Algorithms should also work when an object is partially occluded by other objects.

Object recognition is both about recognizing specific objects ("That is my dog Spot.") and about recognizing classes of objects ("That is a dog."). We will focus on the latter problem, even though there is not a precise definition for what constitutes a class. In some cases, for example with human faces, the objects in a class are visually similar; we will refer to this as a *visual object class.* In other cases, for example with chairs, two objects in a class may not look at all alike — the only similarities are in function; we will refer to this as a *functional object class.* Since recognition of functional object classes requires higher-level cognitive reasoning (beyond the scope of the thesis), we will restrict our attention to computer recognition of visual object classes.

If humans are so adept at recognizing objects, why do we need to duplicate this behavior with computers? One reason is simply scientific curiosity. Man has had tremendous success understanding the world around him through sciences such as physics, chemistry, and astronomy. However, little is known about the brain and the visual system. By trying to develop computer vision systems, we may hope to gain insight into the computational principles that govern human vision.

Another reason is to make computers more interactive and autonomous. Judging from the wide range of biological organisms that depend on vision (crustaceans, fish, cephalopods, birds, mammals, insects, etc.), it is evident that evolution has found

sight to be an essential modality for interacting with the world. Robots and computer systems will also need this capability to function in unconstrained environments, especially those that are unsafe or inconvenient for humans. Here we are talking about environments such as hazardous industrial areas, military automatic target recognition (ATR) platforms, underwater, on spacecraft, inside pipes, and even inside the human body. Already people are working on vision systems for autonomous vehicle navigation and obstacle avoidance on national highways [DC89, DM92].

New forms of human-computer interfaces will be needed as people seek more natural ways to interact with their computers. There is currently a trend toward miniaturization; however, the goal of developing a credit-card sized computer is limited by the size of the display and keyboard. As cameras become progressively smaller, it may be possible to eliminate the keyboard and use gesture or handwriting-based input. Both of these ideas, however, depend upon the availability of fast and reliable visual recognition algorithms. Together with speech understanding, vision would make it possible for humans to interact with computers as they do with other humans.

Finally, we note that humans are, well ... "only human." Although they are very good at recognizing objects, humans need to take breaks, eat, and sleep. They also become quite bored with repetitive tasks and become susceptible to mistakes. Many sensors today generate enormous volumes of data and/or data rates. Military pilots are so overloaded with information that they cannot effectively look for targets and still fly their aircraft. As another example, the recently completed NASA/JPL Magellan mission to Venus returned more data than all previous planetary missions combined, a staggering $30,000$ $1K \times 1K$ pixel images from the first pass alone. Planetary scientists are simply overwhelmed with data. The manual methods of data analysis that they have used in the past on hard-copy photographs are no longer feasible. Computers are helping, but there is still no way for the scientists to find the geological features of interest (e.g., volcanoes, impact craters, etc.) without manually looking through all the data, a process which they estimate would take ten man-years for the first-pass Magellan data. Similar problems abound in industry where there are numerous quality control and inspection tasks that rely on the human visual system to detect errors and

anomalies.

Closer to home, the Internet and World Wide Web provide access to a massive wealth of information including pictures, video clips, and other forms of multimedia. Unfortunately, it is very difficult to locate any multimedia items that you might want such as pictures of horses or a particular scene from a movie. Text-based search engines simply do not provide an adequate interface for finding pictures; new methods and new ways of specifying queries are needed. The QBIC (Query By Image Content) system [FSN+95] being developed at the IBM Almaden Research Center is a step in this direction, although currently the system makes use only of lower-level image properties such as color, motion, and texture rather than recognizing higher-level objects. The Photobook system [PPS96] developed at MIT also provides a set of tools to interact with image databases. Additional approaches are presented in the recent recent PAMI special issue [PP96] on digital libraries.

Visual recognition of objects by computer is a technology that promises to revolutionize industry. However, there are a number of difficult technical challenges that must be overcome. Foremost among these is the problem of invariance. Consider imaging a simple rigid object such as a coffee mug; the low-level pixel representation captured by the camera will be highly dependent upon the lighting conditions, the pose of the mug, the relative position and orientation of the camera, etc. How can we conclude that a test image such as the one shown at the top of Figure 1.1 is indeed a picture of a coffee mug? If we were just concerned about recognizing this particular mug, we might try to subtract the test image from a reference image of the same mug and compute the RMS (root mean square) error. In this case, the test image was taken under brighter lighting conditions and with the mug in a slightly different pose. The RMS error between the two images is 24.17 units per pixel. On the other hand, the RMS error between the test image and the image of a computer mouse is only 23.16 units per pixel. Thus, in terms of RMS error the test image is closer to the computer mouse image.

For recognizing object classes, the problem is even harder because of the variability between the underlying physical objects. For example, Figure 1.2 shows a variety of

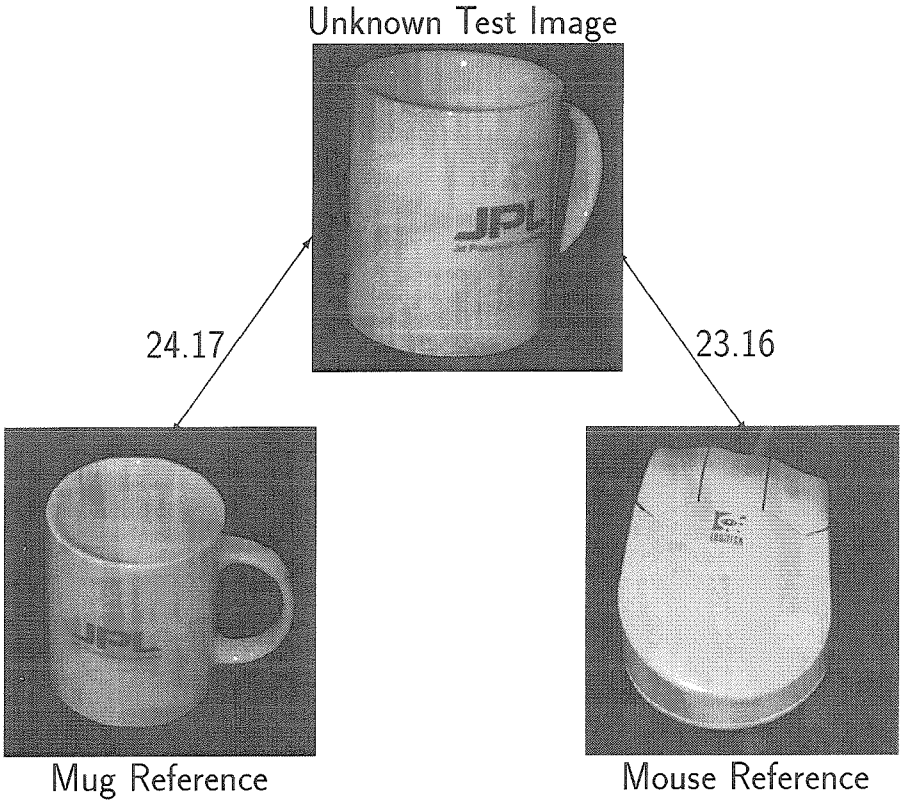


Figure 1.1: The pixel-space representation of an object is highly dependent upon the lighting conditions, object pose, and the relative position and orientation of the camera. The test image is closer to the mouse image in RMS error than to the mug reference image.

Figure 1.2: Examples from the object class "coffee mug."

coffee mugs. To reliably recognize an object class such as this, an algorithm must be (largely) invariant to the differences between individual instances. The same considerations apply to recognizing flexible or deformable objects such as a pair of scissors or a human face.

1.2 Overview of our Approach

Many early pattern recognition algorithms were based on techniques developed for radar and communication systems. Radar systems transmit a known RF (radio frequency) signal. If the signal strikes an object, some portion of the signal may be reflected back to the radar receiver[1] (depending on the geometry and roughness of the object's surface). The receiver does not know when to expect the returned signal or even whether the signal will return (e.g., there may be no objects present). Thus,

[1]Interestingly enough, radar *is* like the "feeler ray" model put forward by the early Greeks.

the receiver must constantly "listen" and try to determine if the transmitted signal has returned. It was shown by North in 1943 [Nor43] that the optimal detector for this problem takes the form of a *matched filter*. The receiver should perform a cross-correlation between the noisy received signal and a copy of the transmitted signal. If the correlation exceeds a predetermined threshold, the receiver declares that a target has been found.

The correlation method, which is also known as *template matching*, is attractive because it provides a mathematically optimal solution for the problem. No detector can perform better for finding a known signal in additive white noise. Unfortunately, in most pattern recognition and computer vision applications, the assumption that "the signal is known exactly" does not hold, so considerable effort has been devoted to extending the template matching method to handle target signal variability. (The white noise assumption may also be inappropriate, but we do not focus on that problem here.)

One generalization of template matching consists of approaches based on principal components or "eigenfeatures" [SK87, TP91, BP93, MN95, BFP+94]. In standard template matching, a single template is used to represent an entire object class, but with the principal components approach, a set of basis templates is used and the object class is represented by linear combinations of the basis templates. This approach provides a better model for how members of the class vary, thereby allowing computer algorithms to ignore the noncritical differences such as those due to lighting, pose, and individual variability within the class. The principal components approach will be discussed in more detail in Chapters 3 and 4; in particular, we will show that the method works well for finding localized patterns such as the small volcanoes in the Magellan SAR imagery of Venus.

The volcano problem, however, is special in some ways. Since the Magellan imaging was done with synthetic aperture radar (SAR), the illumination source and the receiver are co-located; hence, there is no need to worry about illumination invariance. In addition, the underlying physical objects are (to first order) rotationally symmetric, so there is no need to worry about rotational invariance. Approximately 80% of the

volcanoes have resolvable summit pits which appear near the center as either a bright spot or a backwards "C". The pits allow the volcanoes to be reliably centered, reducing the amount of translation invariance required. The volcanoes do vary considerably in scale. But, near the center there is usually a visible summit pit and a transition from bright shading on the side sloped toward the radar and dark shading on the side sloped away from the radar. Good performance can be obtained by simply focusing on this central area and ignoring the outer edges of the volcano. Thus, the volcano problem is especially suitable for principal components analysis since (1) there is a limited amount of variability within the class and (2) the defining information is well-localized.

A good way to interpret the principal components approach is to think of an image as a point in a high-dimensional space. Thus, an $N \times N$ image would correspond to an $(N^2 \times 1)$-dimensional vector. The set of all possible images for an object class forms a manifold in this high-dimensional space. Locally the manifold can be well-approximated by a tangent hyperplane; principal components, however, attempts to *globally* approximate the manifold with one hyperplane. If the amount of variability within the object class is not too large, this may be a reasonable approach, as we have found in the volcano study.

For more difficult problems, such as recognizing objects that consist of characteristic parts arranged in a deformable spatial configuration, more powerful methods are needed. For these types of problems, we have developed an approach in which the allowed object deformations are represented through shape statistics learned from examples. The word "shape" is used here in the sense of Kendall [Ken84, Ken89], Bookstein [Boo84, Boo86], and others [DM91]. That is, "shape" refers to properties of a set of labeled points that are invariant with respect to some group action. In our case, the labeled points are the locations of object parts on the image plane, and the group action is translation, rotation (in the image plane), and scaling. Instances of an object in an image are detected by finding the appropriate object parts or features in the correct spatial configuration. The nature of the features may depend on the application. For example, with human faces the useful features might be areas with

distinctive brightness patterns (eyes), texture (hair), color (lips), motion, or symmetry. For handwriting, the features could be quite different: locations of pen lifts and drops, cusps, humps, crossings, etc. Features may come at different scales of resolution: a low-resolution version of the whole face is as much a feature as a high resolution version of an eye corner [Bur].

Local feature detectors are used to identify candidate locations for object parts. As noted, the detectors may be general or specific to a particular application. For object parts that are defined by distinctive brightness patterns, the principal components paradigm may be particularly good; on the other hand, for finding cusps in handwriting, other types of detectors might be more appropriate. Our focus is not on the feature detectors themselves, but instead on what one should do once a set of detectors has been specified.

The basic problem with local feature detectors is that they are not perfectly reliable. Two types of errors can occur: (1) the detector may fail to respond at a true feature location (missed feature) and/or (2) the detector may respond at erroneous locations (false alarms). Hence, the locations identified by a particular detector are treated only as candidates for the actual object part. These candidates are then grouped into hypotheses, which are scored based on the spatial arrangement of the features. Differences due to translation, rotation, and scaling are eliminated by transforming the hypotheses into an appropriate shape space.

The number of hypotheses that can be formed if M candidate locations are generated for each of N object parts is M^N. Thus, brute force inspection of all hypotheses is not typically practical. This problem is avoided by using conditional search. Given the (hypothesized) positions of two object parts, the position and uncertainty of any other part location can be predicted. For example, given the position of two eyes, we can predict where the nose should be found. Only nose candidates falling inside the appropriate search region are used to form hypotheses.

Finally, hypotheses are transformed into shape space and scored based on the joint distribution of the shape variables. It is important to use joint probabilities since we know that object parts do not usually vary independently. For example, if one mouth

corner is found higher than normal, it is more probable that the other mouth corner is also higher. This particular variation is much more likely than one mouth corner high and the other corner low. Using a joint distribution permits the proper type of deformability, while penalizing abnormal variations. This is the key to obtaining invariance with respect to different instances from the same object class.

1.3 Related Work

There is a large body of literature on the problem of object recognition. Of necessity, we have restricted the discussion of other work to only the most closely related approaches. Huttenlocher and Ullman [HU87] introduced the method of object recognition by alignment, which essentially involves establishing correspondences between features in an image and projected model points. Our shape method can be viewed as a generalized form of alignment in which the matching is done with an allowance for deformations of the model object. Although it is not presented in the language of shape statistics, the Bayesian hashing approach of Rigoutsos and Hummel [RH95] can be viewed as a way to compute an approximate shape probability for a configuration of features. Since feature positions in shape space are assumed to be independent, however, their method does not always work well in practice. (Also, there are practical implementation problems associated with hashing.) A different view of our method is that it is a probabilistic version of invariants [MZ92]. The shape of a configuration is an invariant with respect to Euclidean transformations of the image plane. Similarly, affine-shape is an invariant with respect to affine transformations of the image plane. For a given model (object class), the invariants will take on different values, which we model using probability distributions. More detailed references are given by subject area in the following subsections.

1.3.1 Principal Components Analysis

The technique of principal components analysis (PCA) was originally developed in the 1930's by the statistician Hotelling [Hot33]. Sirovich and Kirby [SK87] used principal

components analysis to provide a low-dimensional characterization of human faces. Turk and Pentland [TP91] appear to be the first to have used PCA for visual recognition problems, although there is an earlier neural-network approach by Fleming and Cottrell [FC90] that is similar to PCA. O'Toole and colleagues [OADD93] studied the use of higher order principal components for determining whether a person is familiar or unfamiliar to the system. The tangent distance approach developed by Simard [SYD93] for handwritten digit recognition is similar to PCA in that object classes are modeled with local hyperplane approximations. Burl *et al* [BFP+94] applied principal components analysis to the problem of detecting natural objects (volcanoes) in remote sensing imagery. Murase and Nayar [MN95] have worked on using principal components to do object recognition under varying illumination conditions and pose. Moghaddam and Pentland [MP95, MP96] and Burl [Bur95] independently derived PCA approaches that combine in-space distance with out-of-space distance to classify unknown examples. There are now numerous applications of PCA for recognition of localized objects. Bichsel and Pentland [BP94] have noted that the PCA approach may work well locally but not for large rotations or scale changes.

1.3.2 Volcanoes

Wiles and Forshaw [WF93] proposed a system to automatically locate volcanoes in the Magellan [PFJ+91] imagery of Venus using normalized cross-correlation. To evaluate the effectiveness of their method they generated simulated radar images and compared algorithm performance on this artificial dataset with that of human experts. Our approach improves performance significantly by using a multi-stage approach in which cross-correlation or matched filtering serves merely as a focus of attention mechanism. More sophisticated processing (PCA) is then applied to the regions of interest identified by the FOA. Gains of 15 percentage points in detection rate are achieved by using the multistage approach. We have also developed better methods for handling the lack of ground truth, both in training and evaluation [SBFP94, SBF+94, BFPS94, SBFP95, SBF+95]. Recent results by Asker [AM97] have shown slight improvements

in performance by preclustering the training data into subclasses and using a sequential application of multiple classifiers.

1.3.3 Shape

The statistical theory of shape was developed by Kendall [Ken84, Ken89, LK93], Bookstein [Boo84, Boo86], and others [DM91, MD89, LK93]. The key result we use in our work is due to Dryden and Mardia [DM91] who derived the exact shape space density induced by a general multi-dimensional Gaussian density in figure space. The use of shape statistics in computer vision applications has also been explored by Wilson [Wil95] and Bookstein [Boo95].Using a slightly different (more flexible) definition of shape, Grenander [Gre93] has looked at models for recognizing human hands and more recently [CMVG96] has developed a method for registering anatomical atlases of the human brain. Cootes, Lanitis, and Taylor [CT96] have proposed using eigenapproximations to model shape densities. He and Kundu [HK91] have used hidden Markov models (HMMs) to model the perturbations in the shape of boundary contours.

1.3.4 Faces

The problem of face recognition has received considerable attention in the literature [Kan77, Yui91, TP91, BP93, OADD93, LQP93, VA+94, KS+94, CWS95]; however, in most of these studies, the faces were either embedded in a benign background or were assumed to have been pre-segmented. For any of these recognition algorithms to work in real-world applications, a system is needed that can reliably locate faces in cluttered scenes and with occlusions.

Recent studies have begun to address the problem of face localization. Bichsel [Bic91] provided one of the first attempts to combine face localization with recognition. Burel and Carel [BC94] proposed a method using multi-resolution analysis and learning from examples (multi-layer perceptron) to search for faces in an image. Yang and Huang [YH94] have described a hierarchical knowledge-based method for

locating faces. Graf [G+95] has combined facial feature detectors with a simple model of the arrangement of the features to perform face localization. Rowley, Baluja, and Kanade [RBK95] have developed a neural network approach that appears to work well provided the faces are unoccluded. Also, several of the flexible/deformable approaches discussed in the next subsection have been developed in the context of face localization and/or recognition. Our algorithm improves upon these other systems in two primary respects: (1) we are able to explicitly handle occlusions, and (2) we are able to exploit the statistical structure of face images in a principled way.

1.3.5 Flexible/Deformable Models

Lades, von der Malsburg, and colleagues [L+93, WvdM93] have developed a recognition method that uses Gabor filters to characterize the local areas of an image. These areas (nodes) are then linked together via a deformable mesh. Given an incoming image, the standard mesh is overlaid and adjusted to obtain the best match between the node descriptors and the image, subject to a penalty on the amount of deformation. The advantage of our approach is that instead of using an ad hoc energy function to penalize deformations, we encode the allowed deformations with a probability density that is learned from actual data.

The deformable template work by Yuille [Yui91] is similar in that local parts are linked together in an ad hoc fashion based on analogies with physical systems (in this case springs). The difference between our work and the triangulated graph matching approach used by Amit [AGW95, AK93b] to align X-ray images of hands falls along similar lines.

Lanitis, Cootes, et al [LTCA95, CT95, C+94] have developed a system that uses a shape description based on eigenmodes. Although this approach can be viewed as an approximation of the probability density over feature positions, it is not clear that their snake-based features will work in cluttered scenes or with occlusion. Recently, Cootes has reported used an approach that closely follows our work to do initial localization before using snakes.

The work by Pope and Lowe [PL95, PL94] is similar in flavor to ours. They also use probability distributions to describe an object's appearance with a model that is separated into local appearance and spatial arrangement. Our approach is much stronger and more rigorous in the handling of probabilistic spatial arrangements. We are able to model the joint variation in feature (part) positions, whereas they assume independence. The fact that they are estimating a viewpoint transformation from the data implies that the transformed positions of the features computed by this transformation cannot possibly be independent. Further, for modeling variations of object classes such as faces, it is quite obvious that the part positions are jointly distributed. If one corner of the mouth is considerably lower than expected, then with high probability the other corner of the mouth will also be lower than expected.

Despite this weakness in their approach, Pope and Lowe have made an attempt do two things that do not currently appear in our framework. First, for each object part they associate a vector of attributes such as the scale and orientation at which the part was detected. These local attributes are then factored into their overall scoring function. Second, they attempt to learn the spatial models from training data in which the ground truth positions of object parts are not labeled. A closely related problem involving learning shape models from video sequences is explored in [BH94].

Shams [Sha95] represents objects such as tanks and jeeps using an elastic graph in which the nodes correspond to object features. The method, which is loosely based on the human visual system, uses neuronal dynamics and annealing to find matches.

1.3.6 Handwriting

The Hidden Markov Model (HMM) in various forms has been widely used for recognizing degraded machine printed text [AK93a, KC94, BK94], signature verification [LB96, YWP95], recognition of cursive characters and handwritten digits, keyword spotting [KA94], and general cursive handwriting recognition [BB+94, CKS95, CK94].

Our shape-based approach to recognizing keyword fragments offers several advantages and disadvantages with respect to HMMs. First, the shape method is applicable

to both on-line and off-line handwriting, yet can be adapted to exploit temporal information if it is available. HMMs seem to be best suited for on-line recognition problems in which the temporal information is available, although they have been used for some off-line applications. The shape approach does not require segmentation of the writing into letters or words. This is also true in some, but not all [CKS95, CK94] of the HMM approaches. In the shape approach, the position of a particular feature can depend on the positions of a number of other local features, while in HMMs only first or second order dependencies are typically assumed.

A disadvantage of the shape method is that to learn the appropriate spatial statistics, we need a number of training examples with ground truth. HMMs, however, can be trained from a relatively small number of examples that have not been specially labeled. Also, HMMs provide a model for the entire writing trajectory, while the shape method provides only a model for the keypoint positions.

1.4 Outline

We begin in Chapter 2 by reviewing some basic results from decision theory. Applying these results to the problem of detecting a known signal in white noise leads to the idea of matched filtering or template matching. The problem with this approach, however, is that for recognizing visual object classes, there is inherent variability between different instances from the same class beyond simple measurement noise. To produce better recognition algorithms, we must work harder at modeling the variability in an object class.

In Chapter 3 we examine methods based on modeling an object class as a linear combination of basis functions. We show that a particularly good choice for the basis functions is the set generated by principal components analysis since these encode the directions of maximum variance in the object class.

Chapter 4 is an applications chapter in which we use matched filtering and principal components analysis for an important practical problem: locating small volcanoes on Venus in a large database of synthetic aperture radar (SAR) imagery. This chapter

may be skipped by the reader interested primarily in theory.

Chapter 5 provides the transition from the first half of the thesis to the second. Much of the material in the first half is now well known (see the discussion in the next section), but it is included in the presentation because it provides essential background for the second half. In this chapter, we introduce an object class in which instances from the class consist of characteristic parts in a deformable spatial configuration. For example, human faces consist of eyes, nose, and mouth arranged in a configuration that varies within one individual due to expression and pose, as well as from individual to individual. Using a toy example, we show that the methods of matched filtering and principal components break down on this type of recognition problem. A key contribution of the thesis is a new model that combines local detectors for the parts of an object with a (principled) probabilistic model for the spatial configuration of the parts. Since the recognition process should be largely invariant to translation, rotation, and scaling of objects from the class, spatial configurations are represented using *shape variables* (as pioneered by Kendall [Ken84, Ken89] and Bookstein [Boo84, Boo86]) and probability distributions over shape.

The representation of spatial configurations using shape variables is explored more fully in Chapter 6. Dryden and Mardia have derived a probability density for the shape variables induced by a multivariate Gaussian figure space density. The Dryden-Mardia density (and possibly mixtures of this density) will prove convenient for modeling the joint distribution over shape for a number of important deformable object classes such as human faces and handwriting.

In Chapter 7 we derive a maximum a posteriori procedure for finding the most object-like configuration of points from a pool of candidates generated by local detectors. Since local detectors are typically unreliable, we provide a framework in which objects can be correctly detected and localized despite false alarms and missing features. Prior knowledge about the reliability of the local detectors can be incorporated in the scoring function. We also discuss a search algorithm that exploits the shape information to generate only the most reasonable object hypotheses.

Chapter 8 is an applications chapter in which the shape-based methods of the

previous two chapters are used to locate human faces in cluttered scenes. Correct localization performance is demonstrated on several face sequences embedded in a cluttered background and with occlusions of parts of the face. To demonstrate the robustness of the method across problem domains, additional results are presented for a keyword spotting problem in on-line handwriting data.

In Chapter 6 through Chapter 8, we conducted our analysis from the assumption that the object parts are initially detected using local detectors and then the detector candidates are grouped together and evaluated to find the most object-like configuration. This approach, based on shape alone, works well provided the object parts can be detected reliably and the object shape is distinctive enough to separate it from the random configurations of points formed from detector false alarms. In Chapter 9, we reconsider the toy problem of Chapter 5 from first principles and show that by combining shape and the degree of match of the parts we can achieve better performance than by shape alone.

Chapter 10 summarizes the main points developed in the thesis. We also discuss limitations of the methods and directions for future work.

1.5 Contribution

The primary contribution of this thesis is a new theoretical framework for recognizing certain classes of objects based on features and their mutual positions (Chapters 6–9). The treatment of mutual positions using probability distributions over shape improves upon a number of previous ad hoc methods such as energy, springs, angles, and distances. The system has been designed (and demonstrated) to work in realistic scenes with cluttered backgrounds and with partial occlusion of objects. Prior knowledge about the performance of the local feature detectors can be incorporated into the framework in a rigorous way. The system has been demonstrated on two very different problem domains: face localization and cursive handwriting. Because of its robustness and versatility, we believe this framework may offer a unified approach to a wide range of visual recognition problems.

A secondary contribution is the volcano detection study of Chapter 4. Although the volcano system is made up entirely of known components (matched filter, principal components analysis, and Gaussian classifier), there are certain aspects of the problem that are novel. This system provided the first effective demonstration of the principal components approach on geological objects in remote sensing data. The volcano study also highlighted the benefits and limitations of a learning-based approach to recognition. On homogeneous data sets, the system was able to achieve performance near, but slightly below, that of human experts, while on a more diverse set of images, the performance degraded considerably from that of humans. Finally, the absence of ground truth for Venus raised some important issues regarding training and evaluation in the face of uncertainty. These issues are not emphasized in the thesis, but they are discussed at length in [SBF+95, SBFP95, BFPS94] and the cited references.

Following is a chapter-by-chapter breakdown of which results and ideas in the thesis are new and which were previously known:

Chapter 2: All of the material in this chapter was previously known, except possibly for the combination of matched filtering and CFAR detection, which was derived in Section 2.6 as a way to handle unknown DC and contrast.

Chapter 3: All of the material in this chapter was previously known except for Sections 3.4 and 3.4.1. The interpretation of the optimal classifier for two classes described by separate SVD bases was derived independently by Burl [Bur95] and Moghaddam and Pentland [MP95, MP96]. The basic result is that an unknown point should be classified based on a "distance" that consists of two terms: one term measures how well a given SVD basis represents a point (i.e., how far is the point from the hyperplane spanned by the SVD basis vectors) and the other term measures how well the projections of the point onto the SVD hyperplane agree with projections of other examples from the class. This result synthesizes the reconstruction error metric used by Turk and Pentland [TP91] with approaches that were based on classification in projection-space.

Chapter 4: As noted earlier, all of the components comprising the JARtool vol-

cano detection system were previously known. However, the application of PCA techniques to recognizing objects in remote sensing imagery is new. The experimental sensitivity studies characterizing detection performance as a function of the number of PCA components are also new.

Chapter 5: This chapter primarily consists of definitions and experiments using classical methods on a particular deformable object class T_ρ. No new theory is derived.

Chapter 6: The idea that the "shape" of a configuration is the information that remains after the effects of translation, rotation, and scaling have been factored out is due to Kendall and Bookstein. The results presented in Sections 6.1 through 6.3 were previously known. In particular, the shape density theorem was originally derived by Dryden and Mardia [DM91]. The properties presented in Section 6.4, however, appear to be new. Although some of these are straightforward (inherited almost directly from the assumed Gaussian figure space density), they appear to have never been explicitly stated in the shape statistics literature. The properties are very important, however, in the context of recognition since features may be missing due to occlusion or detector failure. The properties provide a mechanism to compute the shape density with respect to different baseline pairs and over subsets of shape variables.

The Shape Mixture Theorem (Section 6.4.3) and derivation are new. This theorem is quite important, since in practice it eliminates the need to assume the figure space points follow a multivariate Gaussian density. In our experimental work, we have not yet found it necessary to use shape mixtures, but we expect to eventually encounter problems of this type. The discussion of non-Gaussian figure space models in Section 6.4.4 is also new.

Chapter 7: In this chapter, we form object hypotheses from candidate parts identified by the local detectors. The maximum a posteriori (MAP) rule for selecting the best hypothesis depends on the entire set of observed candidates. We show

that (with appropriate assumptions) the MAP criteria can be reduced to a simple hypothesis scoring function that depends only on the points in the hypothesis. The idea of using a scoring function based on the configuration of points in the hypothesis is not new; in fact, it is quite intuitive. What is new is the derivation showing that the scoring function is actually equivalent to using a MAP rule on the entire set of observations (again under the stated assumptions).

Chapter 8: This chapter contains new experimental results, but no new theory.

Chapter 9: The material in this chapter is entirely new. We have derived the optimal (Bayes) detector for the deformable object class T_ρ defined in Chapter 5. We have also derived an approximately optimal detector for the case when the perturbations of the object parts are not independent. This almost-optimal detector combines two terms: (1) the degree of match between the ideal part and the image and (2) a measure of the overall configuration of the parts.

1.6 Summary

The problem of visual recognition occurs in many domains ranging from medical and biological imaging to surveillance, astronomy, product inspection, human-machine interfaces, database search tools, etc. Our long-term goal is to develop an object recognition system that will work robustly across a variety of problem domains. We envision that such a system will be trained from examples and hints provided by the user since this will permit portability across domains without the need for explicit reprogramming; for a new problem, the user simply provides a new set of examples and hints.

One of the fundamental challenges in building a system of this type is that the recognition algorithms must be sensitive to differences between object classes, yet insensitive to differences within the same object class. To accomplish this task, it is essential to develop good models for the variations in appearance of instances from within the same class. One type of model that works well for localized patterns is

based on principal components analysis and linear combinations of basis functions. We demonstrate this approach for the problem of detecting small volcanoes in the Magellan imagery of Venus. For more difficult problems, however, such as recognizing objects that consist of a set of characteristic parts in a deformable spatial configuration, more powerful methods are needed. For these object classes, we have developed a recognition algorithm that uses local "part" detectors and a probabilistic model for the shape of the allowed object deformations. The algorithm is successfully demonstrated for locating faces in cluttered scenes and with occlusion, as well as for spotting keywords in on-line handwriting data. The combination of local features and probabilistic shape models constitutes the major thrust of the thesis.

Chapter 2 Matched Filtering

2.1 Introduction

In this chapter, we begin by reviewing some elementary results from decision theory. To make the discussion more concrete consider the following hypothetical example. Suppose we are building a machine such as the one shown in Figure 2.1 to separate apples and oranges. The machine works as follows: the user pours a mixture of apples and oranges into the hopper, then one item of fruit at a time travels down the conveyor belt. Along the belt various sensors measure features such as the weight and diameter of each fruit. At the end of the belt, the machine must direct each fruit into either the apple bin or the orange bin. How should the machine make its decision? What if the user will be very angry if some stray apples sneak into the orange bin, but will not care much if a few oranges get into the apple bin? What if the user initially puts a larger percentage of oranges in the hopper than apples? How will these things affect the machine's decision?

Statistical decision theory provides the answers to these questions. If we simply want to minimize the average number of errors (apples put in the orange bin and vice versa), the optimal solution is to use the maximum a posteriori (MAP) rule. Given the feature measurements $\mathbf{x}$, a fruit should be placed into the bin ω (ω_1 = apple, ω_2 = orange) for which $p(\omega|\mathbf{x})$ is maximum. We provide a proof of this result along with generalizations for unequal prior probabilities, multiple classes, and unequal error costs.

These results are then applied to the problem of detecting a known signal in additive white Gaussian noise (ω_1 = signal present, ω_2 = noise only). For this problem, the MAP rule reduces to a matched filter or template matching procedure. A copy of the target signal, "the matched filter," is correlated against the input. If the result exceeds a threshold, a detection is declared. The problem of discriminating between

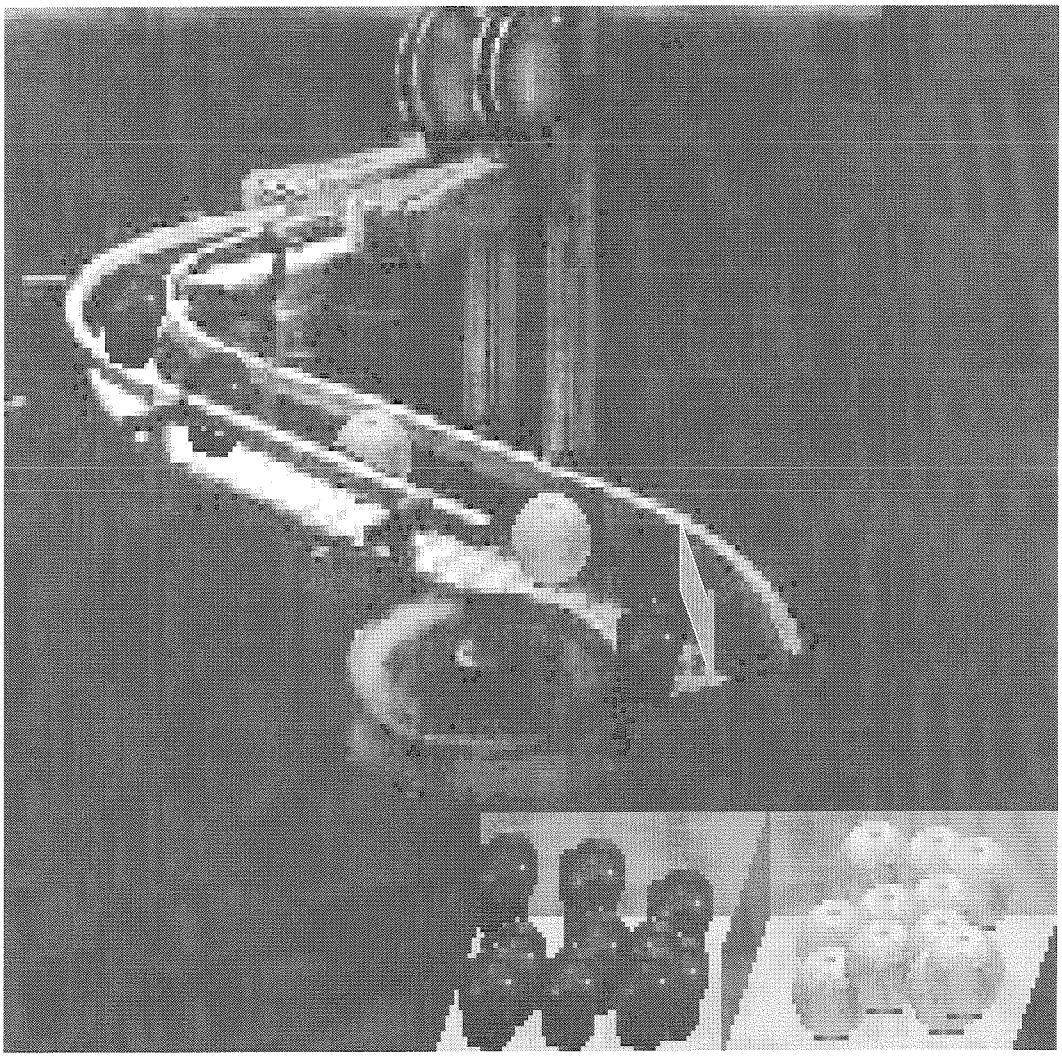


Figure 2.1: Fruit Separation Machine. Based on measurements of each piece of fruit, e.g, diameter, weight, and height, the machine must decide whether a fruit is an apple or an orange. The light gray paddle is shifted to direct each fruit into the proper basket at the end of the conveyor belt.

two known signals in noise is similar. The MAP rule takes the form of a matched filter; however, in this case the filter is matched to the difference between the two signals.

For communication systems the assumptions behind the matched filter (signal known exactly and additive white Gaussian noise) are quite reasonable. For recognizing visual object classes, however, these assumptions break down. Variations in the world, such as illumination conditions and the object pose, cause variability in the appearance. The goal in this thesis is to provide better models for the appearance of instances from an object class. We begin in this chapter by discussing generalizations of the matched filter to cases where the DC level and contrast are unknown and to cases where the target signal is randomly chosen from a discrete set of possibilities.

2.2 The (Bayes) Optimal Decision Rule

This section provides a brief review of some elementary results from decision theory. For more details the interested reader is referred to [DH73, Fuk90]. First, suppose that the universe consists of two exhaustive and mutually exclusive states or classes ω_1 and ω_2. When ω_1 is true, we observe a vector of variables $\mathbf{x}$ generated according to the probability density $p(\mathbf{x}|\omega_1)$. On the other hand, when ω_2 is true, we observe $\mathbf{x}$ generated according to the probability density $p(\mathbf{x}|\omega_2)$. The basic problem of decision theory is the following: *given an observation* $\mathbf{x}$*, determine whether the true state of the world is* ω_1 *or* ω_2. Thus, we seek a decision rule $\mathcal{R}$ that maps $\mathbf{x}$ values to the set $\{\omega_1, \omega_2\}$. Some examples of decision rules are given below:

$$\mathcal{R}_1(\mathbf{x}) = \omega_1 \quad \forall \mathbf{x} \tag{2.1}$$

$$\mathcal{R}_2(\mathbf{x}) = \begin{cases} \omega_1 & \text{if } \mathbf{w}^T\mathbf{x} \geq T \\ \omega_2 & \text{if } \mathbf{w}^T\mathbf{x} < T \end{cases} \tag{2.2}$$

$$\mathcal{R}_3(\mathbf{x}) = \begin{cases} \omega_1 & \text{with probability} = q(\mathbf{x}) \\ \omega_2 & \text{with probability} = 1 - q(\mathbf{x}) \end{cases} \tag{2.3}$$

The first rule decides that ω_1 is true without considering the value of $\mathbf{x}$ (probably not a good rule!). The second rule decides ω_1 if the projection of $\mathbf{x}$ along a vector $\mathbf{w}$ is greater than or equal to a threshold T, and decides ω_2 otherwise. This rule is equivalent to using a hyperplane to separate the space of $\mathbf{x}$'s into two areas or *decision regions*. If $\mathbf{x}$ falls in the first area, the rule chooses ω_1; otherwise, the rule chooses ω_2. The third rule is a bit unusual in that it is not deterministic. Based on the value of $\mathbf{x}$, it guesses ω_1 with probability $q(\mathbf{x})$ and ω_2 with probability $1 - q(\mathbf{x})$.

How can we determine whether a rule is good or not? One method is to use the expected cost. If a decision rule chooses ω_i when the true state is really ω_j, a cost c_{ij} is incurred. Hence, the average cost of using rule $\mathcal{R}$ when presented with observation $\mathbf{x}$ is given by:

$$\text{cost}_{\mathcal{R}}|\mathbf{x} \;=\; \sum_{i,j} c_{ij} \cdot p\left(\mathcal{R}(\mathbf{x}) = \omega_i\right) \cdot p(\omega_j|\mathbf{x}) \tag{2.4}$$

Note that we have allowed for both deterministic rules (such as $\mathcal{R}_1$ and $\mathcal{R}_2$) and probabilistic rules (such as $\mathcal{R}_3$). For deterministic rules, the probability $p\left(\mathcal{R}(\mathbf{x}) = \omega_i\right)$ for a specified $\mathbf{x}$ value will equal 1 for exactly one value of i and 0 otherwise.

For multiple classes $\omega_1, \ldots, \omega_K$ (again mutually exclusive and exhaustive), Equation 2.4 can be written more conveniently in matrix notation as follows:

$$\text{cost}_{\mathcal{R}}|\mathbf{x} \;=\; \mathbf{q}^T(\mathbf{x})\,\mathbf{C}\,\mathbf{y}(\mathbf{x}) \tag{2.5}$$

where

$$\mathbf{q}(\mathbf{x}) \;=\; \left[p\left(\mathcal{R}(\mathbf{x}) = \omega_1\right), \ldots, p\left(\mathcal{R}(\mathbf{x}) = \omega_K\right)\right]^T \tag{2.6}$$

$$\mathbf{y}(\mathbf{x}) \;=\; \left[p(\omega_1|\mathbf{x}), \ldots, p(\omega_K|\mathbf{x})\right]^T \tag{2.7}$$

and $\mathbf{C}$ is the matrix with (i, j) entry c_{ij}.

As indicated by the notation on the left-hand side of Equations 2.4 and 2.5, this is a conditional cost for a particular $\mathbf{x}$ value. To find the true expected cost, we need to

multiply by $p(\mathbf{x})$ and integrate.

$$\begin{aligned} \text{cost}_{\mathcal{R}} &= \int_{\mathbf{x}} (\text{cost}_{\mathcal{R}}|\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x} \\ &= \int_{\mathbf{x}} \left(\mathbf{q}^T(\mathbf{x})\, \mathbf{C}\, \mathbf{y}(\mathbf{x})\right) p(\mathbf{x})\, d\mathbf{x} \end{aligned} \tag{2.8}$$

However, since $p(\mathbf{x})$ is always non-negative, we can minimize the average cost by minimizing $\text{cost}_{\mathcal{R}}|\mathbf{x}$ pointwise, i.e., for each $\mathbf{x}$ value. Thus, to minimize Equation 2.8 we need to find $\mathbf{q}(\mathbf{x})$ such that the quantity on the right-hand side of Equation 2.5 is minimized, subject to the constraints $\mathbf{q}(\mathbf{x}) \geq 0$ and $\mathbf{1}^T\mathbf{q}(\mathbf{x}) = 1$. The solution is to set $\mathbf{q}$ equal to a vector of all zeros except for the position corresponding to the minimum entry of $\mathbf{Cy}$, which should be set to one. If $\mathbf{Cy}$ does not have a unique minimum element then several different $\mathbf{q}$'s will work (in fact, convex combinations of any $\mathbf{q}$'s that work will also work).

For the two-class case, the minimum cost decision rule reduces to the following form:

$$\mathcal{R}_*(\mathbf{x}) = \begin{cases} \omega_1 & \text{if } \frac{p(\omega_1|\mathbf{x})}{p(\omega_2|\mathbf{x})} > T \\ \omega_2 & \text{otherwise} \end{cases} \tag{2.9}$$

where

$$T = -\frac{c_{12} - c_{22}}{c_{11} - c_{21}} \tag{2.10}$$

By applying Bayes rule

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i) \cdot p(\omega_i)}{p(\mathbf{x})} \tag{2.11}$$

we can rewrite the optimal decision rule in terms of the class-conditional densities:

$$\mathcal{R}_*(\mathbf{x}) = \begin{cases} \omega_1 & \text{if } \frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x}|\omega_2)} > \tilde{T} \\ \omega_2 & \text{otherwise} \end{cases} \tag{2.12}$$

where

$$\tilde{T} = T \cdot \frac{p(\omega_2)}{p(\omega_1)} = -\frac{c_{12} - c_{22}}{c_{11} - c_{21}} \cdot \frac{p(\omega_2)}{p(\omega_1)} \tag{2.13}$$

The ratio of class-conditional densities appearing in Equation 2.12 is generally referred to as the *likelihood ratio* and denoted by $\Lambda(\mathbf{x})$. The optimal decision rule can be viewed as a comparison of $\Lambda(\mathbf{x})$ to the appropriate threshold $\tilde{T}$ or, equivalently, as a comparison of $g(\Lambda(\mathbf{x}))$ to $g(\tilde{T})$, where $g(\cdot)$ is any monotonic function such as $\log(\cdot)$.

Note that the minimum probability of error decision rule is a special case of the minimum expected cost rule with $\mathbf{C} = \mathbf{C}_E = \mathbf{1} \cdot \mathbf{1}^T - \mathbf{I}$. For both the two-class and multi-class problem, $\mathbf{C}_E\mathbf{y} = \mathbf{1} - \mathbf{y}$. The minimum element of $\mathbf{C}_E\mathbf{y}$ will be the one for which the posterior probability $\mathbf{y}$ is maximum. Thus, to obtain the *minimum* probability of error, one should choose the class with *maximum* posterior probability.

2.3 Known Signal vs White Noise

In this section, we apply the results of the previous section to the problem of detecting a known signal in additive, white Gaussian noise (AWGN). In particular, we will show that the optimal decision rule reduces to the classical matched filter.

Let ω_1 correspond to the state "signal present" and ω_2 to "signal absent." When ω_1 is true, we will observe the signal $\mathbf{s}$ plus noise. On the other hand, when ω_2 is true, we will observe only noise. Therefore, the observed n-dimensional vector $\mathbf{x}$ has class-conditional densities given by:

$$p(\mathbf{x}|\omega_1) = \mathcal{N}(\mathbf{x}; \mathbf{s}, \sigma^2\mathbf{I}) \tag{2.14}$$

$$p(\mathbf{x}|\omega_2) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \sigma^2\mathbf{I}) \tag{2.15}$$

where σ^2 is the noise variance (per pixel), $\mathbf{I}$ is the $n \times n$ identity matrix, and $\mathcal{N}$ is the

multivariate Gaussian density:

$$\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \triangleq \frac{1}{(2\pi)^{n/2}\cdot|\boldsymbol{\Sigma}|^{1/2}}\cdot\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{2.16}$$

We can write the posterior probabilities as follows (for $i = 1, 2$):

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{p(\mathbf{x}|\omega_1)p(\omega_1)+p(\mathbf{x}|\omega_2)p(\omega_2)} \tag{2.17}$$

but a simpler approach is to pass directly to the log of the likelihood ratio:

$$\log\Lambda(\mathbf{x}) = \log\mathcal{N}(\mathbf{x};\mathbf{s},\sigma^2\mathbf{I}) - \log\mathcal{N}(\mathbf{x};\mathbf{0},\sigma^2\mathbf{I}) \tag{2.18}$$

Simplifying this expression yields:

$$\begin{aligned}\log\Lambda(\mathbf{x}) &= -\frac{1}{2\sigma^2}\cdot\left[(\mathbf{x}-\mathbf{s})^T(\mathbf{x}-\mathbf{s})-\mathbf{x}^T\mathbf{x}\right] \\ &= \frac{\mathbf{s}^T\mathbf{x}}{\sigma^2}-\frac{\mathbf{s}^T\mathbf{s}}{2\sigma^2}\end{aligned} \tag{2.19}$$

which should be compared to the threshold $\log\tilde{T}$. Equivalently, we can compare $\hat{\mathbf{s}}^T\mathbf{x}$, where $\hat{\mathbf{s}}$ is a unit vector in the direction of $\mathbf{s}$, to the modified threshold

$$T_a = \frac{\sigma^2}{\|\mathbf{s}\|}\log\tilde{T}+\frac{\|\mathbf{s}\|}{2} \tag{2.20}$$

Therefore, the optimal detector for a known signal in AWGN is the matched filter, which is given by:

$$\mathcal{R}_{\text{MF-a}}(\mathbf{x}) = \begin{cases}\omega_1 & \text{if } \hat{\mathbf{s}}^T\mathbf{x} > T_a \\ \omega_2 & \text{otherwise}\end{cases} \tag{2.21}$$

Note that with symmetric costs $c_{ij} = c_{ji}$ and equal prior probabilities $p(\omega_1) = p(\omega_2)$, the term $\log\tilde{T}$ in Equation 2.20 will equal zero, and the threshold T_a reduces to $\frac{\|\mathbf{s}\|}{2}$.

The matched filter has a nice pictorial interpretation as shown in Figure 2.2. Here

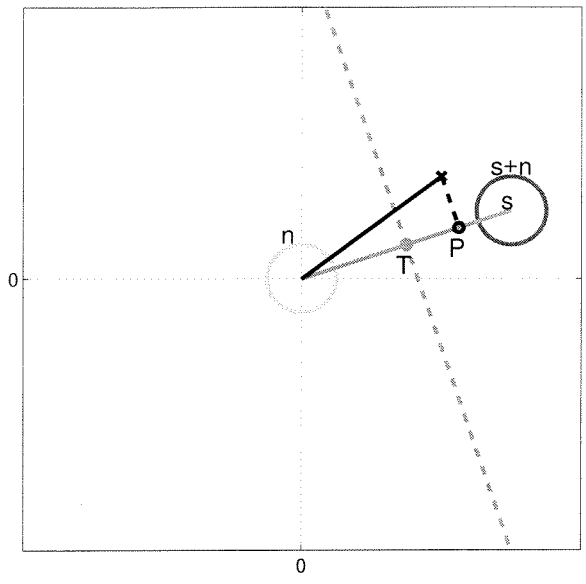


Figure 2.2: Matched Filter - Known Signal vs. Noise

we consider an observation vector consisting of just two components x_1 and x_2. The circles indicate the one standard deviation contours of the noise and signal-plus-noise probability distributions, respectively. The optimal decision rule for equal prior probabilities is to project $\mathbf{x}$ onto $\hat{\mathbf{s}}$ (this projection is indicated by P) and compare to the threshold T. Note that all points $\mathbf{x}$ lying to the right of the almost-vertical, dashed line through T will have projections on $\hat{\mathbf{s}}$ that exceed the threshold; hence, any observations falling in this area will be classified as "signal present."

2.4 Known Signal vs Known Signal

The analysis in the previous section pertained to the problem of discriminating a known signal plus noise from white noise. Another important problem is that of discriminating between one known signal (plus noise) and another known signal (plus noise). In this case our two possible states will be: ω_1 corresponding to signal 1 present and ω_2 corresponding to signal 2 present. The class-conditional densities are

then given by:

$$p(\mathbf{x}|\omega_1) = \mathcal{N}(\mathbf{x}; \mathbf{s}_1, \sigma^2\mathbf{I}) \tag{2.22}$$

$$p(\mathbf{x}|\omega_2) = \mathcal{N}(\mathbf{x}; \mathbf{s}_2, \sigma^2\mathbf{I}) \tag{2.23}$$

Working through the likelihood ratio, we find that the optimal decision rule is

$$\mathcal{R}_{\text{MF-b}}(\mathbf{x}) = \begin{cases} \omega_1 & \text{if } (\mathbf{s}_1 - \mathbf{s}_2)^T\mathbf{x} > T_b \\ \omega_2 & \text{otherwise} \end{cases} \tag{2.24}$$

where T_b is given by

$$T_b = \sigma^2 \log \tilde{T} + \frac{\mathbf{s}_1^T\mathbf{s}_1 - \mathbf{s}_2^T\mathbf{s}_2}{2} \tag{2.25}$$

This rule is illustrated in Figure 2.3. Note that if signal 2 is the zero vector, the solution simplifies to the result of the previous subsection (simply substitute $\mathbf{s}_2 = \mathbf{0}$ into Equations 2.24 and 2.25).

2.5 Theoretical Performance

Since the matched filter is derived from the MAP rule, we know that no detector can perform better, assuming, of course, that the assumptions are valid. In this section we explicitly calculate the theoretical performance of the matched filter.

We will focus on the probability distribution of the statistic $h = \mathbf{w}^T\mathbf{x}$. For the matched filter, $\mathbf{w}$ has a specific form, but for now let $\mathbf{w}$ be any linear filter. Since the conditional density of $\mathbf{x}$ is jointly Gaussian and h is a linear combination of the components of $\mathbf{x}$, we know that the conditional density of h is also Gaussian; specifically,

$$p(h|\omega_1) = \mathcal{N}(h; \mu_1, \sigma_1^2) \tag{2.26}$$

$$p(h|\omega_2) = \mathcal{N}(h; \mu_2, \sigma_2^2) \tag{2.27}$$

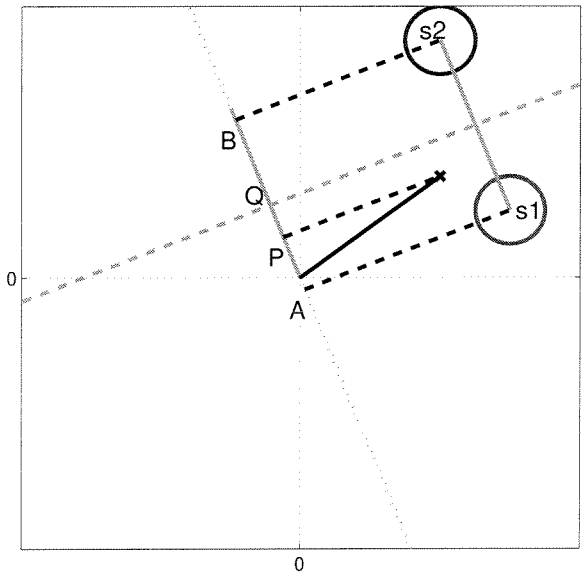


Figure 2.3: Matched Filter - Known Signal vs. Known Signal. Given the observation $\mathbf{x}$, we want to decide whether it corresponds to signal 1 or signal 2. As in Figure 2.2, the circles denote the one sigma probability contours. The optimal decision rule is to project $\mathbf{x}$ onto $\mathbf{s}_2 - \mathbf{s}_1$ and compare to the threshold Q. For equal priors Q is the bisector of the line AB, where A is the projection of $\mathbf{s}_1$ and B is the projection of $\mathbf{s}_2$. The almost-horizontal, dashed line through Q divides the plane into two decision regions; $\mathbf{x}$'s falling in the lower region are assigned to class ω_1 and in the upper region to ω_2.

with

$$\mu_1 = \mathbf{w}^T \mathbf{s}_1 \qquad \mu_2 = \mathbf{w}^T \mathbf{s}_2$$
$$\sigma_1^2 = \sigma^2 \mathbf{w}^T \mathbf{w} \qquad \sigma_2^2 = \sigma^2 \mathbf{w}^T \mathbf{w} \tag{2.28}$$

Let ω_1 be the target signal and ω_2 be a nuisance signal. Consider the decision rule which chooses ω_1 if h is greater than or equal to a threshold T and ω_2 otherwise. There are two types of errors that can occur: (1) misses (true class = ω_1, but the rule says ω_2) and (2) false alarms (true class = ω_2, but the rule says ω_1).

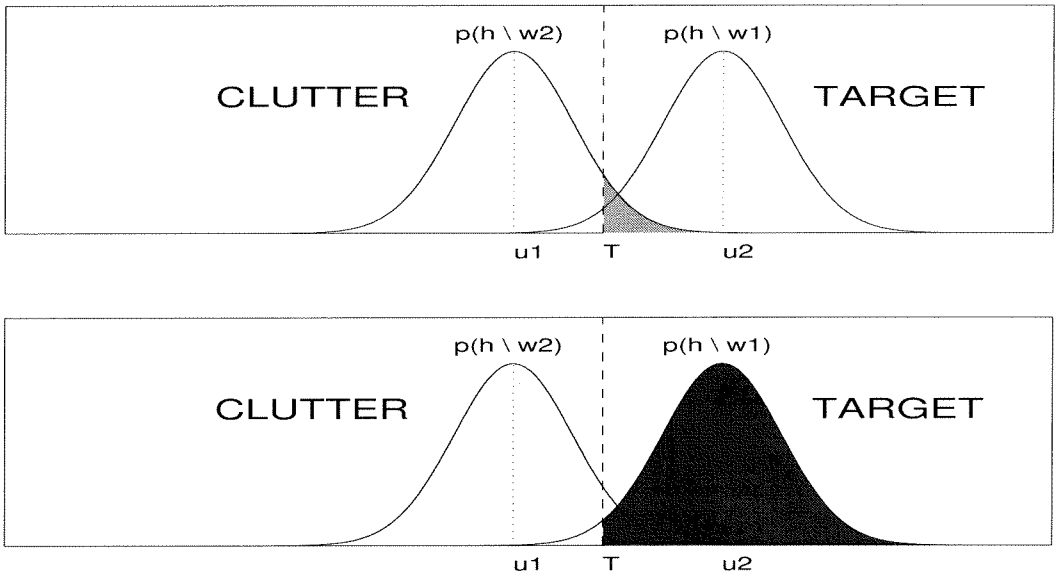


Figure 2.4: (a) The probability of false alarm is the gray shaded area under $p(h|\omega_2)$ to the right of the threshold T. (b) The probability of detection is the black shaded area under $p(h|\omega_1)$ to the right of T.

The performance of a decision rule can be characterized by its probability of detection p_{d} and its probability of false alarm p_{fa}. Of course, these probabilities depend on the threshold T as shown in Figure 2.4.

If the threshold is set very low (aggressively), then we will be sure to detect the target signal, but we will also get an increased number of false alarms. If the threshold is set very high (conservatively), we will get fewer false alarms, but we will also miss

the target signal more often. A curve known as a receiver operating characteristic (ROC curve) [Van68] can be used to show the tradeoff between p_{d} and p_{fa} as a function of T.

$$\begin{aligned} p_{\mathrm{fa}}(T) &= \Pr(h \geq T \mid \text{true class} = \omega_2) \\ &= \int_T^{\infty} p(h|\omega_2)\, dh \qquad (2.29) \\ p_{\mathrm{d}}(T) &= \Pr(h \geq T \mid \text{true class} = \omega_1) \\ &= \int_T^{\infty} p(h|\omega_1)\, dh \qquad (2.30) \end{aligned}$$

Since the class-conditional densities (Equations 2.26 and 2.27) are Gaussian, we can express the probability of detection and false alarm by

$$\begin{aligned} p_{\mathrm{fa}}(T) &= 1 - \Phi(T; \mu_2, \sigma_2) \\ &= 1 - \Phi\left(\frac{T - \mu_2}{\sigma_2}\right) \qquad (2.31) \\ p_{\mathrm{d}}(T) &= 1 - \Phi(T; \mu_1, \sigma_1) \\ &= 1 - \Phi\left(\frac{T - \mu_1}{\sigma_1}\right) \qquad (2.32) \end{aligned}$$

where Φ is the cumulative distribution function of the Gaussian

$$\begin{aligned} \Phi(x; \mu, \sigma) &\triangleq \int_{-\infty}^{x} \mathcal{N}(x; \mu, \sigma)\, dx \qquad (2.33) \\ \Phi(x) &\triangleq \Phi(x; 0, 1) \qquad (2.34) \end{aligned}$$

We could also express the results in terms of the error function erf using the relationship

$$\mathrm{erf}(x) = 2\Phi\left(\frac{x}{\sqrt{2}}\right) - 1 \qquad (2.35)$$

If the threshold T is expressed as follows:

$$T = \mu_2 + K\sigma_2 \tag{2.36}$$

the probabilities of detection and false alarm simplify to:

$$\begin{aligned} p_{\text{fa}}(K) &= 1 - \Phi(K) && (2.37) \\ p_{\text{d}}(K) &= 1 - \Phi\left(\frac{\mu_2 + K\sigma_2 - \mu_1}{\sigma_1}\right) \\ &= 1 - \Phi(K - \frac{\mu_1 - \mu_2}{\sigma_1}) && (2.38) \end{aligned}$$

The quantity $(\mu_1 - \mu_2)/\sigma_1$ is typically called the signal-to-noise ratio (SNR). To maximize the probability of detection for a given false alarm rate, we must maximize the SNR since $\Phi(\cdot)$ is a monotonically increasing function of its argument. Substituting from Equation 2.28, we have the following result which is true for any linear filter $\mathbf{w}$:

$$\begin{aligned} \text{SNR} &= \frac{\mathbf{w}^T(\mathbf{s}_1 - \mathbf{s}_2)}{\sigma \cdot \sqrt{\mathbf{w}^T\mathbf{w}}} \\ &= \frac{\hat{\mathbf{w}}^T(\mathbf{s}_1 - \mathbf{s}_2)}{\sigma} \end{aligned} \tag{2.39}$$

where $\hat{\mathbf{w}}$ is a unit vector in the direction of $\mathbf{w}$. Equation 2.39 is maximized when $\hat{\mathbf{w}}$ is in the direction $(\mathbf{s}_1 - \mathbf{s}_2)$, i.e, when $\mathbf{w}$ is the matched filter. Substituting for $\mathbf{w}$ yields the SNR achieved by the matched filter

$$\begin{aligned} \text{SNR}_{\text{mf}} &= \frac{1}{\sigma}\sqrt{(\mathbf{s}_1 - \mathbf{s}_2)^T(\mathbf{s}_1 - \mathbf{s}_2)} \\ &= \frac{\sqrt{E}}{\sigma} \end{aligned} \tag{2.40}$$

where E is the energy in the difference signal. Thus, the performance for the matched filter is given by

$$p_{\text{d}}(K) = 1 - \Phi\left(K - \sqrt{\frac{E}{\sigma^2}}\right) \tag{2.41}$$

$$p_{\mathrm{fa}}(K) = 1 - \Phi(K) \tag{2.42}$$

If desired, we can express the probability of detection directly in terms of the false alarm probability, i.e.,

$$p_{\mathrm{d}}(p_{\mathrm{fa}}) = 1 - \Phi\left(\Phi^{-1}(1 - p_{\mathrm{fa}}) - \sqrt{\frac{E}{\sigma^2}}\right) \tag{2.43}$$

$$\tag{2.44}$$

The corresponding ROC curves are shown in Figure 2.5 as a function of SNR.

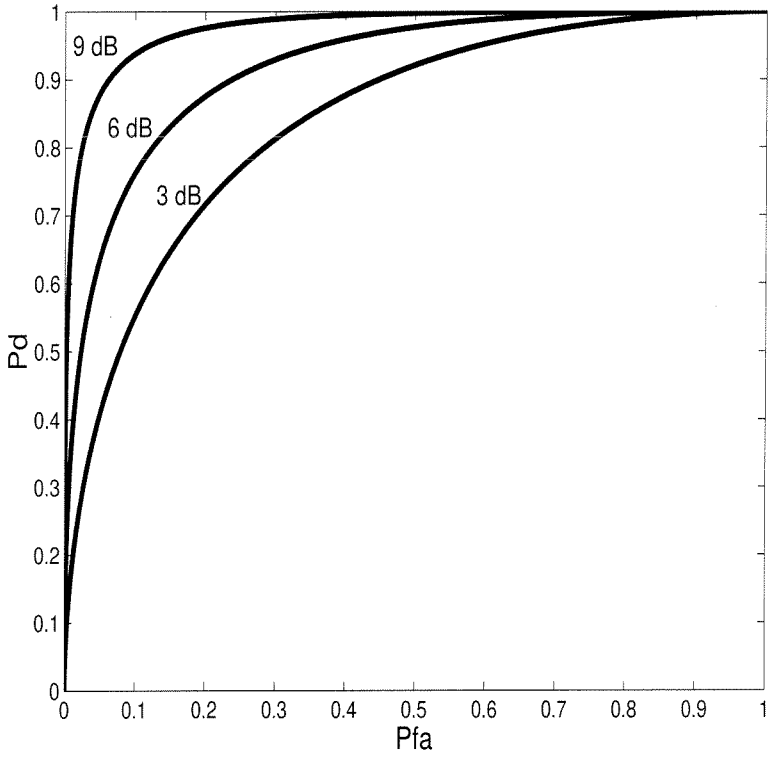


Figure 2.5: Theoretical performance of the matched filter as a function of signal-to-noise ratio.

2.6 Unknown DC and Contrast

In the matched filter derivation, we assumed a target class consisting of one signal that is known exactly. This model is useful in some radar and communications applications; however, for recognizing visual object classes, more complex target models are needed. The first generalization we consider is the problem of detecting a known signal $\mathbf{s}$ in white noise when there is an unknown DC level μ and contrast σ (for both the signal and the noise). Under this model, the observed vector $\mathbf{x}$ can be described as follows:

$$\mathbf{x} = \begin{cases} \mu \cdot \mathbf{1} + \sigma \cdot (\mathbf{s} + \tilde{\mathbf{n}}) & \text{under hypothesis } \omega_1 \\ \mu \cdot \mathbf{1} + \sigma \cdot \tilde{\mathbf{n}} & \text{under hypothesis } \omega_2 \end{cases} \tag{2.45}$$

where $\mathbf{1}$ is a vector of ones and $\tilde{\mathbf{n}}$ is a vector of zero mean, unit variance white Gaussian noise. We refer to this type of model as a signal with *parametric variability* since given the parameter values $\boldsymbol{\theta} = [\mu\ \sigma]^T$, the signal is known exactly. The class-conditional densities of $\mathbf{x}$ given $\boldsymbol{\theta}$ are as follows:

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta}, \omega_1) &= \mathcal{N}\left(\mathbf{x}; \theta_1\mathbf{1} + \theta_2\mathbf{s}, \theta_2^2\mathbf{I}\right) \\ p(\mathbf{x}|\boldsymbol{\theta}, \omega_2) &= \mathcal{N}\left(\mathbf{x}; \theta_1\mathbf{1}, \theta_2^2\mathbf{I}\right) \end{aligned}$$

To eliminate the conditioning on $\boldsymbol{\theta}$, we multiply by $p(\boldsymbol{\theta}|\omega_i)$ and integrate.

$$p(\mathbf{x}|\omega_1) = \int p(\mathbf{x}|\boldsymbol{\theta}, \omega_1) \cdot p(\boldsymbol{\theta}|\omega_1)\, d\boldsymbol{\theta} \tag{2.46}$$

$$p(\mathbf{x}|\omega_2) = \int p(\mathbf{x}|\boldsymbol{\theta}, \omega_2) \cdot p(\boldsymbol{\theta}|\omega_2)\, d\boldsymbol{\theta} \tag{2.47}$$

We will now assume that samples from the background area immediately surrounding the area under test can be used to estimate $\boldsymbol{\theta}$. A fully Bayesian approach [DH73] would treat $\boldsymbol{\theta}$ as a random vector and combine the observed background examples with a hypothesized prior distribution to produce a refined posterior distribution that could be used in place of $p(\boldsymbol{\theta}|\omega_i)$. Instead, we will use a maximum likelihood framework to produce an estimate $\hat{\boldsymbol{\theta}}$. With enough background samples, we can get a very

accurate estimate of the $\boldsymbol{\theta}$ so $p(\boldsymbol{\theta}|\omega_i) \approx \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$. The class-conditional densities then become:

$$\begin{aligned} p(\mathbf{x}|\omega_1) &= \mathcal{N}\left(\mathbf{x}; \hat{\theta}_1 \mathbf{1} + \hat{\theta}_2 \mathbf{s}, \hat{\theta}_2{}^2 \mathbf{I}\right) \\ p(\mathbf{x}|\omega_2) &= \mathcal{N}\left(\mathbf{x}; \hat{\theta}_1 \mathbf{1}, \hat{\theta}_2{}^2 \mathbf{I}\right) \end{aligned}$$

We can either use the results in Equations 2.24 and 2.25 or compute the log of the likelihood ratio directly to find the optimal detector:

$$\mathcal{R}(\mathbf{x}) = \begin{cases} \omega_1 & \text{if } \frac{(\mathbf{x} - \mathbf{1}\hat{\mu})^T \hat{\mathbf{s}}}{\hat{\sigma}} > T_3 \\ \omega_2 & \text{otherwise} \end{cases} \tag{2.48}$$

where

$$T_3 = \log \tilde{T} + \frac{\|\mathbf{s}\|}{2} \tag{2.49}$$

Thus, the optimal detector can be interpreted as follows. Given an area of the image where we want to test for the presence of the signal, we first estimate the DC level and contrast from the area surrounding the test area and use these estimates to "normalize" the test area. A matched filter for the signal is then applied to the normalized test area. If the result exceeds the threshold, we say the signal is present.

There is a second interpretation that is also useful. As shown in Figure 2.6, the matched filter is applied to the entire image. This process is then followed by an adaptive thresholding procedure in which the mean and standard deviation of the response image around the test area are used to establish the appropriate threshold for the matched filter. Asymptotically (in the case of many samples), this procedure is (statistically) the same as doing the normalization on each patch. For a finite number of background samples, the two interpretations will yield slightly different results. Note that the second interpretation is equivalent to using a standard matched filter followed by the classical two-parameter CFAR algorithm [Gol69] commonly used in radar systems to detect point targets.

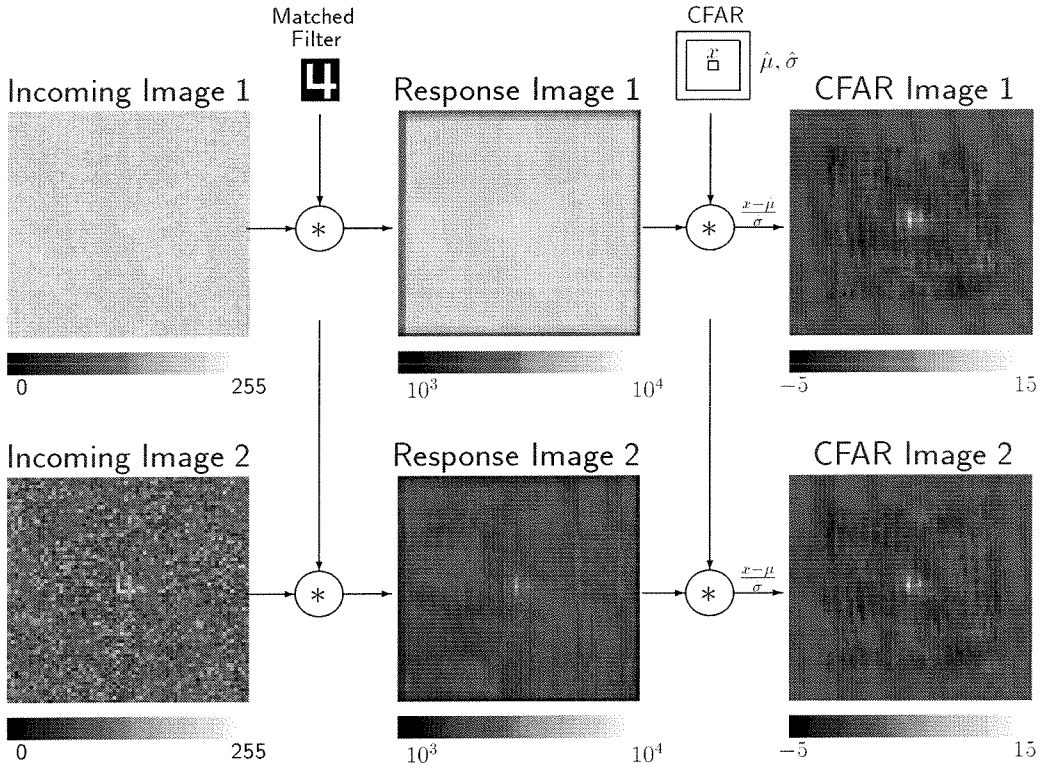


Figure 2.6: When the DC level and contrast are unknown but constant or slowly varying over an image, the optimal detector is a matched filter followed by an ideal two-parameter CFAR. The CFAR essentially normalizes the matched filter response images using statistics it has estimated from nearby areas of the image. To illustrate, image 1 has a DC level = 200 and contrast = 6, while image 2 has DC level = 100 and contrast = 24. Although the matched filter response images are very different in terms of numerical values, the CFAR images are the same.

One concern with the CFAR thresholding algorithm is that if the estimate of σ is anomalously low, the algorithm will detect signals that do not match well because the response value is scaled up by $1/\sigma$. To avoid this problem, a minimum value of σ can be combined with the estimate as follows:

$$\hat{\sigma}_{\text{mod}} = \sqrt{\hat{\sigma}^2 + \sigma^2_{\text{min}}} \tag{2.50}$$

In the human visual system, this loosely corresponds to the fact that the internal signals of the brain are noisy. Weak signals will not be detected because of this additional internal noise.

2.7 Generalization to Subclasses

Thus far, we have considered target classes consisting of a single exemplar signal. An important extension is the case where the signal to be detected is selected probabilistically from one of k subclasses. For this problem, we can show that the likelihood ratio is simply the weighted sum of likelihood ratios from the subclasses.

Suppose that the target signal is selected from subclass k with probability p_k. We want to determine whether there is any target signal present. The likelihood ratio for this problem is given by

$$\begin{aligned} \Lambda(\mathbf{x}) &= \frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x}|\omega_2)} && (2.51) \\ &= \frac{\sum_k p_k \cdot p(\mathbf{x}|k,\omega_1)}{p(\mathbf{x}|\omega_2)} && (2.52) \\ &= \sum_k p_k \Lambda^{(k)}(\mathbf{x}) && (2.53) \end{aligned}$$

where $\Lambda^{(k)}$ is the likelihood ratio for discriminating between subclass k and the background (ω_2).

For the case in which each of the subclasses is a single exemplar known exactly,

the subclass likelihood ratios can be expressed as matched filter operations.

$$\log \Lambda^{(k)}_{\mathrm{MF}}(\mathbf{x}) = \frac{1}{\sigma^2}\left(\mathbf{s}_k^T\mathbf{x} - \frac{\mathbf{s}_k^T\mathbf{s}_k}{2}\right) \tag{2.54}$$

The optimal detector uses a bank of matched filters; each computes its own likelihood ratio, and these are weighted by the p_k's and combined to produce the overall likelihood ratio. This is illustrated in Figure 2.7.

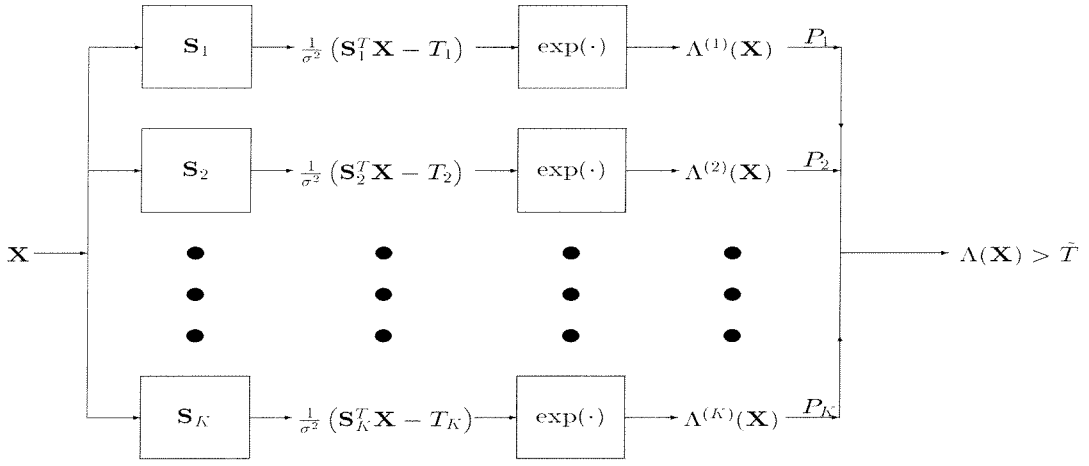


Figure 2.7: Matched Filter Bank. The optimal decision rule when the signal class has subclasses corresponding to the exemplars $\mathbf{s}_1,\ldots,\mathbf{s}_K$ is to use a bank of matched filters. The separate likelihood functions are combined using the probabilities $p_1, \ldots, p_K$, where p_k is the mixture probability for subclass k.

There is an even simpler decision rule that is approximately optimal. The filter $k = \tilde{k}$ which yields the largest value of $\left(\mathbf{s}_k^T\mathbf{x} - \frac{\mathbf{s}_k^T\mathbf{s}_k}{2}\right)$ may dominate the summation since $\Lambda^{(k)}$ is the exponential of this quantity. Hence,

$$\log \Lambda(\mathbf{x}) \approx \log p_{\tilde{k}} + \frac{1}{\sigma^2}\left(\mathbf{s}_{\tilde{k}}^T\mathbf{x} - \frac{\mathbf{s}_{\tilde{k}}^T\mathbf{s}_{\tilde{k}}}{2}\right) \tag{2.55}$$

and the simple decision rule will be:

$$\mathcal{R}(\mathbf{x}) = \begin{cases} \omega_1 & \text{if } \mathbf{s}_{\tilde{k}}^T \mathbf{x} > T_* \\ \omega_2 & \text{otherwise} \end{cases} \tag{2.56}$$

where

$$\tilde{k} = \arg\max \left(\mathbf{s}_{\tilde{k}} \mathbf{x} - T_k \right)$$

and

$$T_* = \sigma^2 \left(\log \tilde{T} - \log p_{\tilde{k}} \right) + \frac{\mathbf{s}_{\tilde{k}}^T \mathbf{s}_{\tilde{k}}}{2} \tag{2.57}$$

In other words, $\mathbf{s}_{\tilde{k}}$ is the best matched-filter from the filter bank. This simplified decision rule is commonly called "winner-take-all." Obviously, if the assumption that there is one dominant k value does not hold (e.g., if two k values co-dominate), then winner-take-all performance will be degraded with respect to the optimal rule.

Figure 2.8 illustrates the relationship between the optimal decision rule of Equation 2.53, the winner-take-all rule of Equation 2.56, and the simple matched filter of Equation 2.21. The dashed lines show the simple matched filter decision boundaries for discriminating between $\mathbf{s}_k$ and noise for $k = 1, 2, 3$. These lines assume that each signal has the same prior probability as the noise. The solid (slightly jagged) line just outside of the dashed lines shows the optimal decision boundary. Approximating the optimal boundary with three *straight* lines would give the winner-take-all decision boundary. Note that the optimal boundary differs from the separate matched filter boundaries for two reasons. First, the subclass k has prior probability $p(\omega_1) \cdot p_{\tilde{k}}$ (rather than $p(\omega_1)$), which pushes the decision boundaries outward slightly. Second, the corners are rounded because points near the corners are about the same distance from two of the subclass exemplars and therefore no single term dominates in Equation 2.53.

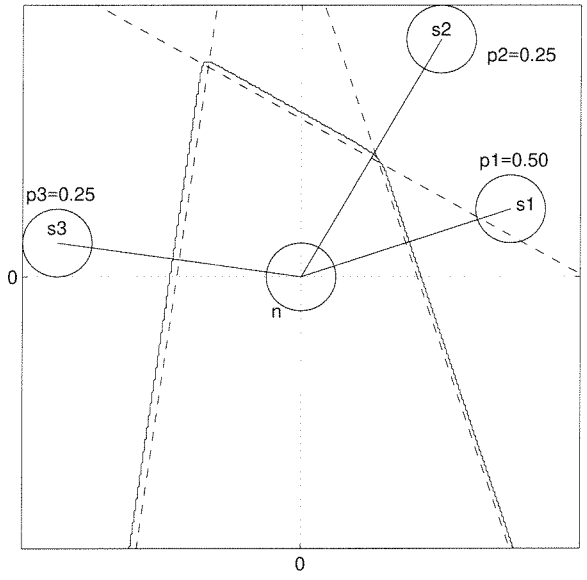


Figure 2.8: Subclass Decision Rules. The dashed lines show the three pairwise matched filter decision boundaries under the assumption that each subclass has the same prior probability as the noise. Using the correct priors moves the decision boundary out to the (slightly jagged) solid lines. Also, notice that the corners of the optimal boundary are rounded because these points are approximately equidistant from two exemplars. The winner-take-all strategy ignores this interaction and just approximates the optimal boundary with three straight lines.

2.8 Summary

In this chapter, we reviewed some of the basic principles of decision theory that provide the foundation for the remainder of the thesis. In particular, the maximum a posteriori rule was shown to be optimal (in terms of minimum probability of error) for choosing between K mutually exclusive and exhaustive classes given observations $\mathbf{x}$ that depend on the true class. For the problem of detecting a known signal in white noise, the optimal decision rule reduces to the classical matched filter. Similarly, for discriminating between two known signals in white noise, the optimal rule is a linear filter matched to the difference signal.

The performance of any detector can be characterized using receiver operating characteristics (ROC curves), which show the probability of detection versus the probability of false alarm. The theoretical performance of the matched filter can be expressed as a function of the signal-to-noise ratio.

Finally, two more complex target models were analyzed. For detecting a known signal in white noise when the DC level and contrast are unknown, we showed that a combination of matched filtering and the classical two-parameter CFAR algorithm is nearly optimal. The CFAR algorithm estimates the background statistics (DC level and contrast) from samples near the area under test. With an infinite number of samples, the variance of the estimates goes to zero and the algorithm is optimal; however, with only a finite number of samples, there is some degradation in performance.

For detecting a target class in which the target signal is probabilistically selected from k known signals, we showed that the optimal detector can be implemented as a bank of matched filters along with some nonlinearities.

In subsequent chapters, we will pursue more complex models of the target class. In particular, we next consider a model in which the target signal consists of a linear combination of basis functions. In later chapters, we generalize to models consisting of characteristic parts in a deformable spatial configuration. The goal throughout is to generate a model with parameters that can easily be estimated, yet which is

complex enough to model the variations that appear in visual object classes. By more accurately modeling the target class we can hope to better decide whether an object in an image is an example from the class or not.

Chapter 3 Linear Combinations of Basis Functions

3.1 Introduction

In this chapter we consider a model for visual object classes in which instances from a class are represented by linear combinations of basis functions. A particular instance from an object class can be viewed as a point in a high-dimensional pixel space; an object class corresponds to the cloud of points generated by all the instances from the class. With the model used in the matched filter derivation, an object class is just a single point in pixel space (i.e., the target signal) perturbed by white noise. The cloud of points is then a hypersphere centered around the target signal. An obvious disadvantage with this model is the target signal typically has inherent variability that is not well-modeled as white noise. Figure 3.1a shows six images of a still subject taken under different lighting conditions. Figure 3.1b shows the error between the average image and each of the six instances. Clearly, the error signals exhibit structured variation (i.e., non-white noise). If signals from the target class vary strongly in only a few directions in pixel space, a sphere model will be too "loose" because it will be the same size in the directions where the target does not vary as in the directions where the target varies most.

The linear combination model represents an object class using a hyperplane perturbed by noise. There are several arguments that can be used to justify such a model. One argument put forward by Simard [SYD93] is based on tangent planes. The object class is assumed to consist of a single exemplar that is perturbed by small rotations, translations, changes of scale, and other deformations. Since these transformations are continuous, the appearance of the perturbed object can be well-approximated by a multidimensional Taylor expansion in a small enough neighborhood. Equivalently,

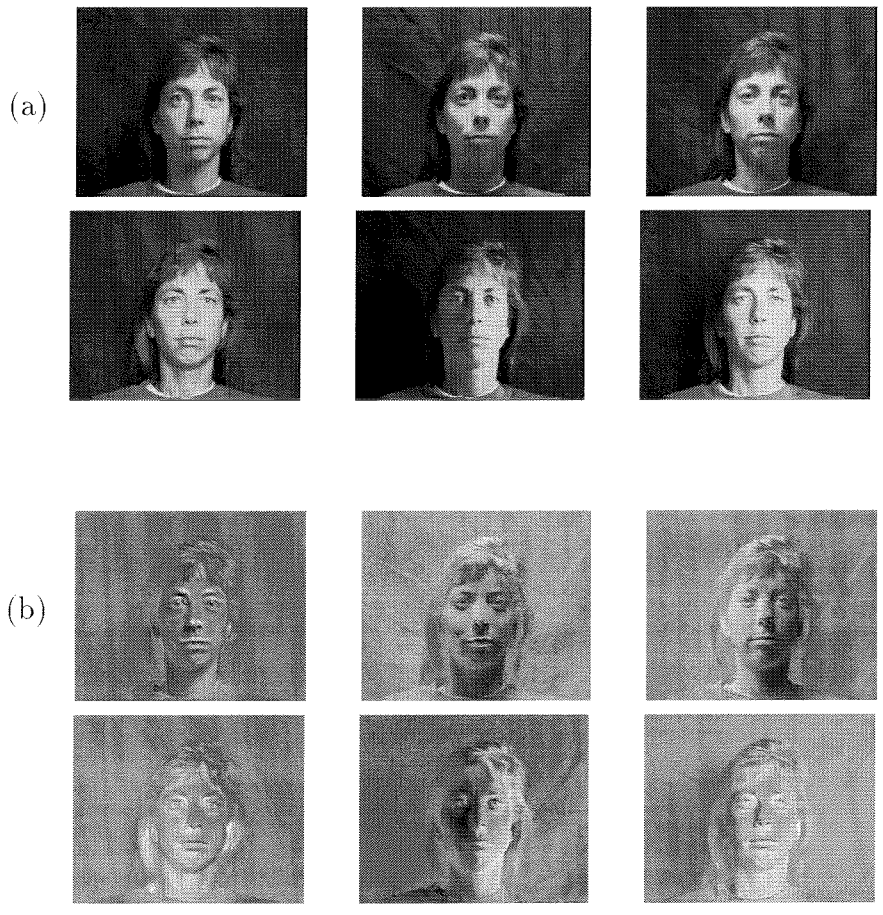


Figure 3.1: (a) Six images of a still subject taken under varying lighting conditions. (b) The errors between the average image and each of the six instances. Clearly, the error images show structured variations. Not all variability is well-modeled by white noise.

we can say that the variations in the target signal lie approximately in the tangent hyperplane.

A second argument is based on the method of principal components analysis (PCA), which was introduced in the statistics literature by Hotelling in the 1930's [Hot33]. PCA provides a way to find the directions of maximum variation in a multidimensional dataset. When applied to examples from a visual object class, one typically finds that most of the variation in the dataset can be explained with a small number of principal components. In essence, the examples lie approximately on a low-dimensional hyperplane. PCA is closely related to the discrete Karhunen-Loéve expansion [Fuk90], which provides a minimum *expected* error expansion for a *random* vector through linear combinations of covariance matrix eigenvectors.

3.2 Preliminaries

In this section we consider target models consisting of a linear combination of a small number of orthonormal basis functions. As in the previous chapter, we will string the pixel values of an object into a long N-dimensional column vector $\mathbf{x}$. Note that there is a loss of spatial neighborhood information with this representation. Algorithms that treat the data in this form will not explicitly know that certain pixels were adjacent in the original image data. Nevertheless, with this caveat an object class ω_i can be modeled as:

$$\mathbf{x}|\boldsymbol{\theta}, \omega_i = \mathbf{m}_i + \mathbf{U}_i\boldsymbol{\theta} + \mathbf{n}_i \tag{3.1}$$

where $\mathbf{m}_i$ is the "nominal exemplar" from the class, $\mathbf{U}_i$ is an $(N \times m)$ orthonormal matrix in which each column is one basis vector, $\boldsymbol{\theta}$ is an $(m \times 1)$ vector of weighting coefficients used to combine the basis functions, and $\mathbf{n}_i$ is white noise. In this section we will assume that for any value of $\boldsymbol{\theta}$, the combination $\mathbf{m}_i + \mathbf{U}_i\boldsymbol{\theta}$ will still be a member of the object class. Hence, the object class is the entire hyperplane passing through the point $\mathbf{m}_i$ and spanned by the columns of $\mathbf{U}_i$. In the next section, we

will consider a refinement in which the weighting coefficients $\boldsymbol{\theta}$ are modeled with a probability distribution.

If the goal is to discriminate between a target object class and a nuisance object class in which both classes are represented by hyperplane models with the same noise variance, it is clear that the optimal decision rule (minimum error) is to compute the perpendicular distance to each hyperplane and assign the test example to the nearest hyperplane. If there is only a model for the target class and no model for the "other" class, then a reasonable decision rule would be to compute the distance from the hyperplane and check whether the distance is less than a threshold. In the context of detecting human faces, this is the "distance from face-space" method originally used by Turk and Pentland [TP91].

The distance from the hyperplane is also called the reconstruction error. As illustrated in Figure 3.2, any point x in pixel space can be expressed as a linear combination of basis functions plus a component orthogonal to the hyperplane. The nearest point p in the hyperplane is the best reconstruction of x using the basis functions, so the distance $d(x, p)$ is called the reconstruction error.

3.3 Murase and Nayar

The model of the previous section assumed $\boldsymbol{\theta}$ could be any $(m \times 1)$ vector of numbers and the resulting linear combination of basis functions would still be a member of the target class. This model is too loose for most practical applications since it assumes the instances from the object class can be *anywhere* on the hyperplane. A better model results if we place restrictions on the values of $\boldsymbol{\theta}$.

Murase and Nayar [MN95] use an object class model of this type. In their approach, training examples are collected as various parameters of the imaging process, such as illumination direction and object pose, are varied continuously. The projection of these training examples on the basis functions generates a trajectory through $\boldsymbol{\theta}$-space. Murase and Nayar represent this trajectory with surface splines. To classify an unknown example, they project the example onto the basis functions and compute

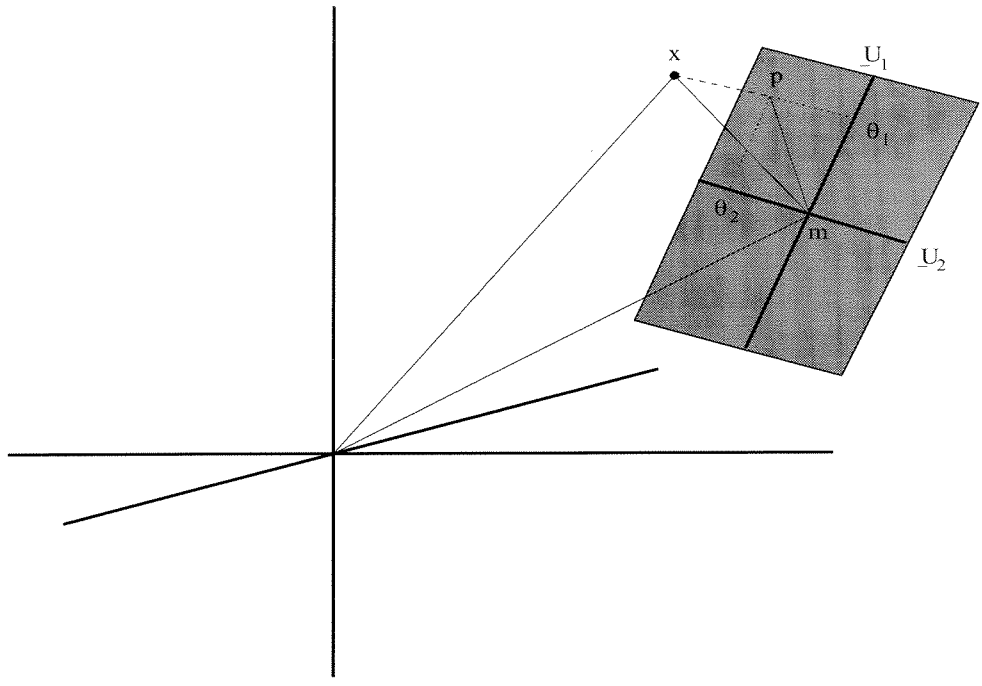


Figure 3.2: Reconstruction Error

the distance from the spline. The trajectory in $\boldsymbol{\theta}$-space corresponds to a trajectory in the original pixel space which is confined to the hyperplane spanned by the basis functions.

3.4 Probabilistic Weighting Coefficients

A more general approach to restricting the values of $\boldsymbol{\theta}$ is to specify a class-conditional probability distribution $p(\boldsymbol{\theta}|\omega_i)$ where ω_i is the class. An illustration of this type of model is shown in Figure 3.3. Here we have used a model for $p(\boldsymbol{\theta}|\omega_i)$ consisting of a mixture of four Gaussians. The ellipses show the equiprobability contours for each of the mixture modes.

There are now two problems we want to consider: (1) distinguishing between two classes when each class is represented by a linear combination of basis functions with probabilistic weighting coefficients and (2) distinguishing one such class from white noise. We can effectively treat both cases at the same time by focusing on the quantity

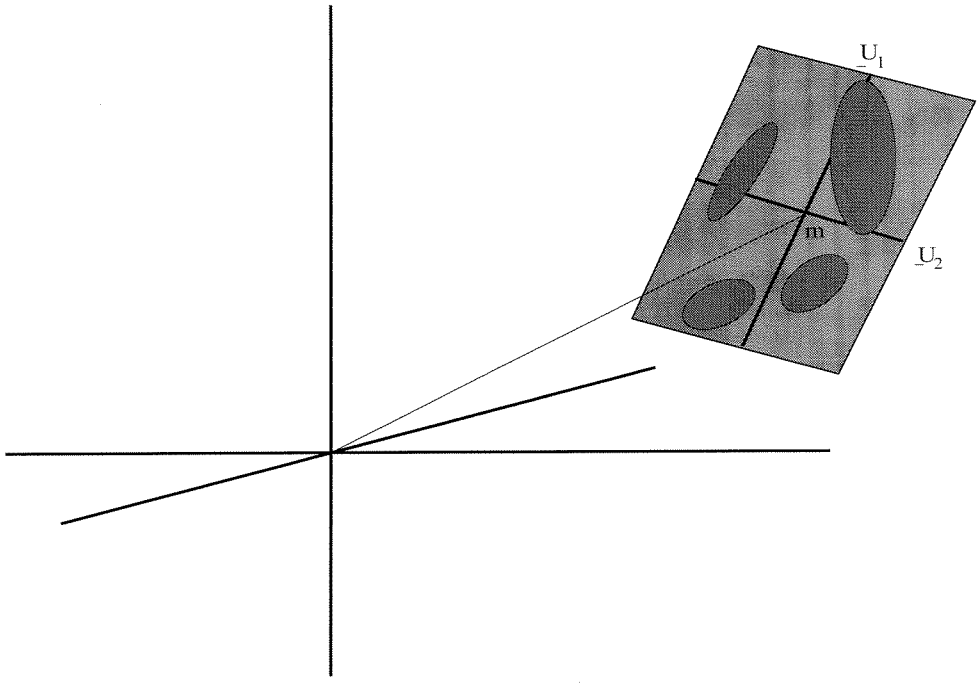


Figure 3.3: Linear combination of basis functions with a probability distribution on the weighting coefficients.

$p(\mathbf{x}|\omega_i)$. Assuming $\boldsymbol{\theta}$ is known, we have the probability density

$$p(\mathbf{x}|\boldsymbol{\theta}, \omega_i) = \mathcal{N}\left(\mathbf{x}; \mathbf{m}_i + \mathbf{U}_i\boldsymbol{\theta}, \sigma_i^2\mathbf{I}\right) \tag{3.2}$$

Essentially, this equation states that if $\boldsymbol{\theta}$ is known then $\mathbf{x}$ is known except for the uncertainty due of the Gaussian observation noise. The class-conditional density of $\mathbf{x}|\omega_i$ (no longer conditioned on $\boldsymbol{\theta}$ can be obtained by multiplying Equation 3.2 by $p(\boldsymbol{\theta}|\omega_i)$ and integrating over $\boldsymbol{\theta}$. Thus,

$$p(\mathbf{x}|\omega_i) \;=\; \int \mathcal{N}\left(\mathbf{x}; \mathbf{m}_i + \mathbf{U}_i\boldsymbol{\theta}, \sigma_i^2\mathbf{I}\right) \cdot p(\boldsymbol{\theta}|\omega_i)\, d\boldsymbol{\theta} \tag{3.3}$$

Following through some straightforward algebra and using the fact that $\mathbf{U}_i^T\mathbf{U}_i = \mathbf{I}$, we obtain

$$p(\mathbf{x}|\omega_i) \;=\; \mathcal{N}\left(\boldsymbol{\Delta}_i; \mathbf{0}; \sigma_i^2\mathbf{I}_{N-m_i \times N-m_i}\right) \cdot \int \mathcal{N}\left(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}_i, \sigma_i^2\mathbf{I}_{m_i \times m_i}\right) \cdot p(\boldsymbol{\theta}|\omega_i)\, d\boldsymbol{\theta} \tag{3.4}$$

where we have introduced the definitions

$$\tilde{\boldsymbol{\theta}}_i \triangleq \mathbf{U}_i^T(\mathbf{x} - \mathbf{m}_i) \tag{3.5}$$

$$\boldsymbol{\Delta}_i \triangleq \mathbf{U}_{i\perp}^T\left[(\mathbf{x} - \mathbf{m}_i) - \mathbf{U}_i\tilde{\boldsymbol{\theta}}_i\right] \tag{3.6}$$

The dimensions of $\tilde{\boldsymbol{\theta}}_i$ and $\boldsymbol{\Delta}_i$ are $(m_i \times 1)$ and $(N{-}m_i \times 1)$, respectively, where m_i is the number of SVD basis vectors (i.e., the dimension of the space spanned by $\mathbf{U}_i$). The symbol $\mathbf{U}_{i\perp}$ designates an $(N \times N{-}m_i)$ matrix with orthonormal columns that are also orthogonal to the columns of $\mathbf{U}_i$. That is, $\mathbf{U}_{i\perp}$ is any orthonormal basis for the subspace orthogonal to the range of $\mathbf{U}_i$. Observe that $\tilde{\boldsymbol{\theta}}_i$ is the projection of $\mathbf{x}$ onto the hyperplane, while $\boldsymbol{\Delta}_i$ is the error between $\mathbf{x}$ and $\tilde{\boldsymbol{\theta}}_i$. We will call $\boldsymbol{\Delta}_i$ the *reconstruction error vector* since $\mathbf{m}_i + \mathbf{U}_i\tilde{\boldsymbol{\theta}}_i$ represents the best possible reconstruction of $\mathbf{x}$ using the $\mathbf{U}_i$ basis functions. In the derivation above we used the fact that

$$\boldsymbol{\Delta}_i^T\boldsymbol{\Delta}_i = (\mathbf{x} - \mathbf{m}_i)^T(\mathbf{x} - \mathbf{m}_i) - \tilde{\boldsymbol{\theta}}_i^T\tilde{\boldsymbol{\theta}}_i \tag{3.7}$$

which is a consequence of the Pythagorean Theorem.

Now we will simplify Equation 3.4 by writing the integral as follows:

$$\int \mathcal{N}\left(\tilde{\boldsymbol{\theta}}_i; \boldsymbol{\theta}, \sigma_i^2\mathbf{I}\right) \cdot p(\boldsymbol{\theta}|\omega_i)\, d\boldsymbol{\theta} \tag{3.8}$$

since $\mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\theta}, \boldsymbol{\Sigma})$. The integral can now be interpreted as the probability density of a random vector $\mathbf{z} = \boldsymbol{\theta} + \mathbf{n}$, evaluated at $\mathbf{z} = \tilde{\boldsymbol{\theta}}_i$, where

$$\mathbf{n} \sim \mathcal{N}\left(\mathbf{0}, \sigma_i^2\mathbf{I}_{m_i \times m_i}\right) \tag{3.9}$$

Therefore, Equation 3.4 reduces to

$$p(\mathbf{x}|\omega_i) = \mathcal{N}\left(\boldsymbol{\Delta}_i; \mathbf{0}; \sigma_i^2\mathbf{I}_{N-m_i \times N-m_i}\right) \cdot q(\tilde{\boldsymbol{\theta}}_i|\omega_i) \tag{3.10}$$

where q is the probability density of $\boldsymbol{\theta}$ blurred by Gaussian noise $\mathbf{n}$. The log likelihood

ratio for discriminating between two signals is given by:

$$\log \Lambda(\mathbf{x}) = (N - m_2) \log \sqrt{2\pi}\sigma_2 - (N - m_1) \log \sqrt{2\pi}\sigma_1 + \frac{\boldsymbol{\Delta}_2^T \boldsymbol{\Delta}_2}{2\sigma_2^2} - \frac{\boldsymbol{\Delta}_1^T \boldsymbol{\Delta}_1}{2\sigma_1^2} + \log \frac{q(\tilde{\boldsymbol{\theta}}_1|\omega_1)}{q(\tilde{\boldsymbol{\theta}}_2|\omega_2)} \tag{3.11}$$

Observe that the solution separates into one term that depends on the reconstruction error and another term that depends upon how well the projections agree with the class-conditional distributions in the hyperplane. This result was obtained independently by Burl [Bur95] and Moghaddam and Pentland [MP95, MP96]. Even if we cannot compute the exact log likelihood ratio (e.g., if the densities are not well-modeled by one of the common parametric forms), we can still make use of Equation 3.11 by combining the reconstruction error term with a classifier in the projection space that produces posterior probability estimates (e.g., mixture densities or kernel-density estimators). Incidentally, the log likelihood ratio for discriminating between zero-mean white noise and a target signal described by a linear combination of basis functions is given by:

$$\log \Lambda(\mathbf{x}) = N \log \sqrt{2\pi}\sigma_2 - (N - m_1) \log \sqrt{2\pi}\sigma_1 + \frac{\mathbf{x}^T \mathbf{x}}{2\sigma_2^2} - \frac{\boldsymbol{\Delta}_1^T \boldsymbol{\Delta}_1}{2\sigma_1^2} + \log q(\tilde{\boldsymbol{\theta}}_1|\omega_1) \tag{3.12}$$

3.4.1 Specialization to a Gaussian Model

To better illustrate the result in Equation 3.11, let us consider a specific form for the densities of $\boldsymbol{\theta}_i$, specifically

$$p(\boldsymbol{\theta}_i|\omega_i) = \mathcal{N}(\boldsymbol{\theta}_i; \boldsymbol{\nu}_i, \boldsymbol{\Sigma}_i) \tag{3.13}$$

Hence, the density of $\tilde{\boldsymbol{\theta}}_i \triangleq \boldsymbol{\theta}_i + \mathbf{n}_i$ is given by:

$$q(\tilde{\boldsymbol{\theta}}_i|\omega_i) = \mathcal{N}(\tilde{\boldsymbol{\theta}}_i; \boldsymbol{\nu}_i, \sigma_i^2 \mathbf{I}_{m_i \times m_i} + \boldsymbol{\Sigma}_i) \tag{3.14}$$

(Since $\tilde{\boldsymbol{\theta}}_i$ is equal to $\boldsymbol{\theta}_i$ plus $\mathbf{n}$, there is an extra variance σ_i^2 attached to the diagonal in the covariance matrix.)

With this model, the log likelihood ratio simplifies to

$$\begin{aligned}\log \Lambda(\mathbf{x}) &= (N-m_2)\log\sigma_2 - (N-m_1)\log\sigma_1 + \frac{\boldsymbol{\Delta}_2^T\boldsymbol{\Delta}_2}{2\sigma_2^2} - \frac{\boldsymbol{\Delta}_1^T\boldsymbol{\Delta}_1}{2\sigma_1^2} + \\ &+\frac{1}{2}\left[d^2(\tilde{\boldsymbol{\theta}}_2;\boldsymbol{\nu}_2,\sigma_2^2\mathbf{I}+\boldsymbol{\Sigma}_2) - d^2(\tilde{\boldsymbol{\theta}}_1;\boldsymbol{\nu}_1,\sigma_1^2\mathbf{I}+\boldsymbol{\Sigma}_1)\right] \qquad (3.15)\end{aligned}$$

where

$$d^2(\boldsymbol{\theta};\boldsymbol{\mu},\boldsymbol{\Psi}) \triangleq (\boldsymbol{\theta}-\boldsymbol{\mu})^T\boldsymbol{\Psi}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}) - \log|\boldsymbol{\Psi}| \qquad (3.16)$$

To summarize, $\mathbf{x}$ is projected onto the two hyperplanes defined by the two bases $\mathbf{U}_1$ and $\mathbf{U}_2$. The two projection coordinate vectors are denoted by $\tilde{\boldsymbol{\theta}}_1$ and $\tilde{\boldsymbol{\theta}}_2$. To determine whether $\mathbf{x}$ should be associated with class ω_1 or ω_2, two questions are important: (1) Is $\mathbf{x}$ well-represented by the basis? (2) Is the projection of $\mathbf{x}$ onto the basis consistent with the projections of other class members? The log likelihood ratio in Equation 3.15 can be interpreted as a nearest distance classifier in which the distance to class i consists of one term involving the reconstruction error and another term involving the Mahalanobis distance in projection space. This result is shown pictorially in Figure 3.4.

3.5 Learning the Basis Functions

We have discussed how to recognize object classes using object models that consist of linear combinations of basis functions. An important practical problem is deciding which basis functions to use, i.e., given a set of training examples, how can we *learn* a good set of basis functions. One idea is to choose a set of basis functions that do the "best" job of representing the examples, where a common definition of "best" is given by the minimum average reconstruction error (mean square error). A different idea is to choose basis functions that maximize the discrimination between the object

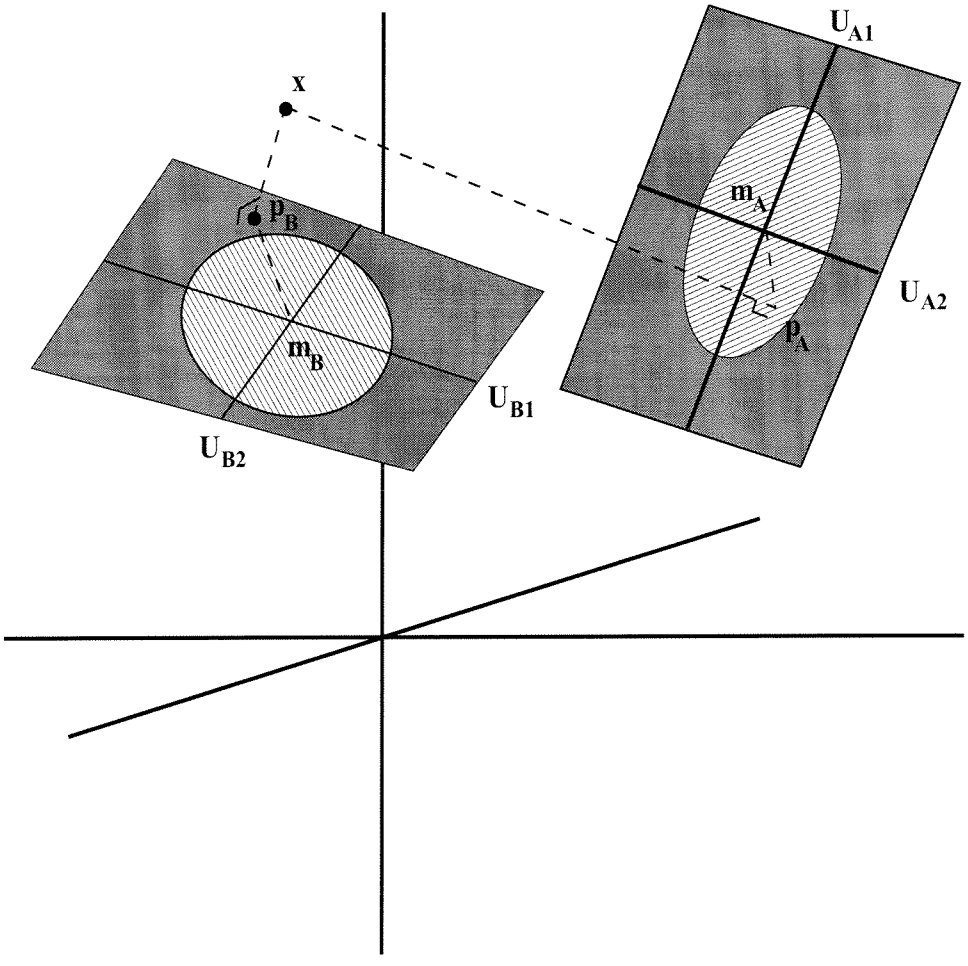


Figure 3.4: Suppose we have two classes A and B. Each class is modeled as a linear combination of basis vectors $\mathbf{U}_A$ and $\mathbf{U}_B$, respectively. To classify an unknown point $\mathbf{x}$, two factors are important: (1) the distance of $\mathbf{x}$ from each hyperplane and (2) the Mahalanobis distance between the projection of $\mathbf{x}$ onto each hyperplane and the corresponding mean. A similar interpretation is possible even when the distributions in projection space are non-Gaussian.

class and another class using a heuristic measure of the class separation such as the Fisher criterion [DH73].

There is an extensive discussion in [Fuk90] on the issue of representation versus discrimination. For recognizing object classes, we argue that the representational approach is best. Humans can generate an approximate drawing or mental image of objects they have seen before. This fact suggests that, at least somewhere in the brain, there is sufficient information to reconstruct the appearance of an object. From economy considerations, it is likely that this same information is used for recognition. Further, an approach based purely on discriminative features would require specialized feature sets to distinguish between *every pair of object classes*, rather than one set of features (model) per class. Also, for many applications the primary concern is to find examples from a specific class. The "other" class is broad and ill-defined, i.e., it consists of everything that is not the object. Discriminative methods, in particular those based on linear discriminant analysis (LDA) and related scatter criteria, cannot deal with this type of catch-all class.

In [Fuk90], Fukunaga argues *against* a representational approach using the following problem as an illustration. People are to be classified as male or female based on a two-dimensional height and weight feature vector. Since height and weight are highly correlated for both males and females, the distributions are approximately as shown in Figure 3.5. Also shown are two one-dimensional distributions obtained by projecting the data. Clearly, the distributions along $\boldsymbol{\phi}_1$ (the principal axis) are highly overlapped, which is taken to mean that representational features are not good for this problem. In fact, if the classes were represented as $\mathbf{m}_i + \alpha\boldsymbol{\phi}_1 + \mathbf{n}$, the methods of the previous section could be applied to yield optimal discrimination performance.

The discriminational approach does have strengths and should be used under certain circumstances. For example, when the problem is to discriminate between similar objects (such as determining whether a person is Asian versus Caucasian), it makes sense to focus on the differences. Similarly, for identifying a particular object within one class (e.g., recognizing your friend Joe), it makes sense to focus on the details that make Joe different from everyone else rather than the fact that Joe has a head

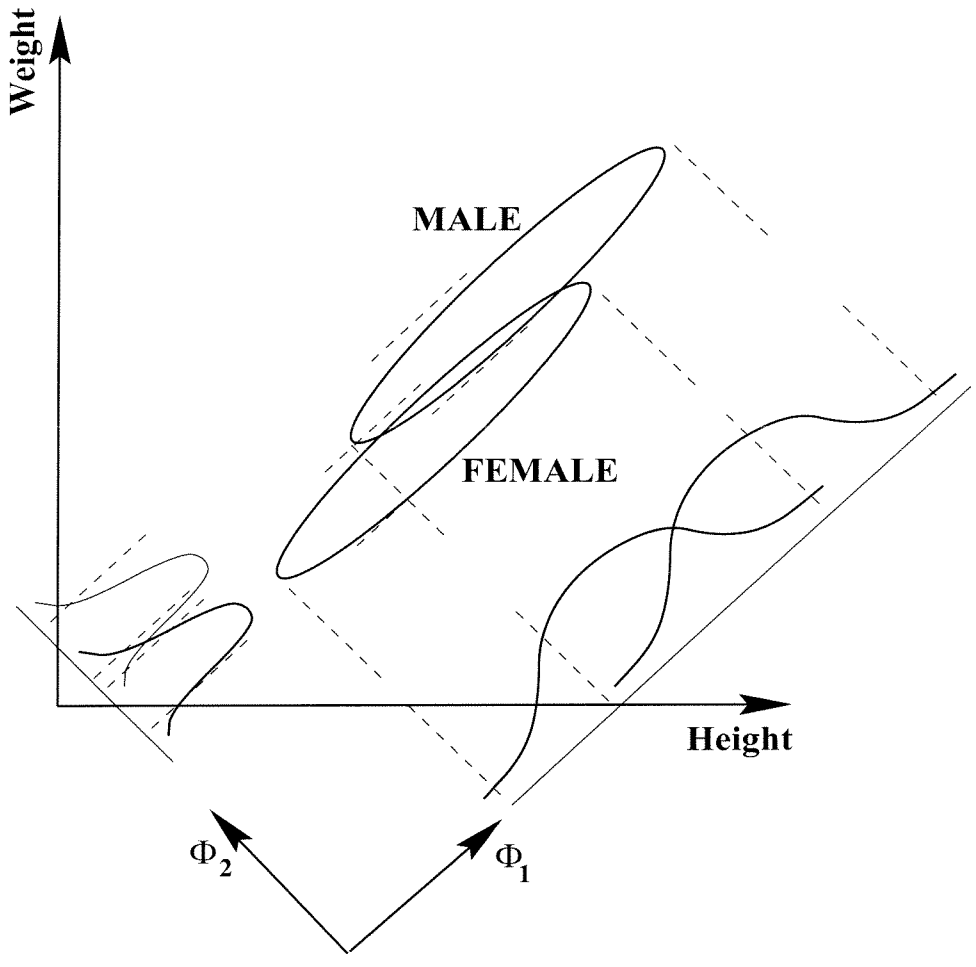


Figure 3.5: Consider the problem of classifying people as male or female based on height and weight feature vectors. Although the projections on ϕ_1 (the principal axis) are highly overlapped, a representational approach as discussed in Section 3.4.1 will yield optimal discrimination. (Adapted from [Fuk90].)

and two eyes, etc.

Since our primary focus is on recognizing *classes* of objects, we will focus solely on basis functions that provide the best representation. We will use the mean square error (MSE) as a measure of the quality of representation; however, we note that there are deficiencies with this metric. For example, consider the problem of recognizing different textures. It is not particularly important to represent the exact microposition of each texture element. Yet, with a MSE metric these details must be encoded, especially if the texture elements have high contrast. In most recognition problems there is a trade-off between the fidelity of representation and whether the extra bits or parameters provide useful information for discrimination. Ideally, we would like a compact description of the appearance of an object class without encoding nonessential details.

Consider a set of M examples designated by column vectors $\mathbf{x_1}$, $\mathbf{x_2}$, ..., $\mathbf{x_M}$. The dimension of each vector is N, where N is the number of pixels in each pattern. We would like to find a set of $m \leq \min(M, N)$ orthonormal basis vectors, $\mathbf{l_1}$, $\mathbf{l_2}$, ..., $\mathbf{l_m}$ that "approximately span" our set of examples. We can make the notion of "approximately spanning" mathematically precise by restating the problem as follows: find a set of m orthonormal basis vectors such that

$$E = \sum_{j=1}^{M} \left\| \mathbf{x_j} - \left(\sum_{i=1}^{m} \alpha_{ij} \mathbf{l_i} \right) \right\|^2 \tag{3.17}$$

is minimized. Notice that we are simply approximating each example as a linear combination of the basis vectors and seeking to minimize the mean squared error. The coefficients α_{ij} are the optimal weights that should be applied to the $\mathbf{l_i}$'s in order to approximate $\mathbf{x_j}$.

Equation 3.17 can be written more compactly using matrix notation. Define

$$\begin{aligned} \mathbf{X} &= [\mathbf{x_1}\ \mathbf{x_2}\ \ldots\ \mathbf{x_M}] && N \times M \\ \mathbf{L} &= [\mathbf{l_1}\ \mathbf{l_2}\ \ldots\ \mathbf{l_m}] && N \times m \\ (\boldsymbol{\alpha})_{ij} &= \alpha_{ij} && m \times M \end{aligned} \tag{3.18}$$

With this notation, the mean square error is given by

$$E = \mathrm{tr}\left[(\mathbf{X} - \mathbf{L}\boldsymbol{\alpha})^{\mathbf{T}}(\mathbf{X} - \mathbf{L}\boldsymbol{\alpha})\right] \tag{3.19}$$

We now seek to minimize E over $\mathbf{L}$ and $\boldsymbol{\alpha}$ subject to the orthonormality constraint $\mathbf{L}^{\mathbf{T}}\mathbf{L} = \mathbf{I}_{m\times m}$. The solution to this problem is generally well known [Pin85]: the optimal basis functions are the m eigenvectors of $\mathbf{X}\mathbf{X}^T$ having the largest eigenvalues. The optimal weighting coefficients are simply the (linear) projections of the examples onto the basis vectors. Note that any orthogonal transformation of the optimal basis provides an equally good basis. For completeness, we have included a derivation of these results in the appendix of this chapter.

The results can also be interpreted in terms of the singular value decomposition (SVD) of $\mathbf{X}$. The SVD of an $N \times M$ matrix $\mathbf{X}$ with $N > M$ produces three matrices $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ sized $N \times M$, $M \times M$, and $M \times M$ respectively. These matrices have the properties that:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}} \tag{3.20}$$

$$\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{I}_{M\times M} \tag{3.21}$$

$$\mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I}_{M\times M} \tag{3.22}$$

$$\mathbf{S} = \mathrm{diag}(s_1, s_2, \ldots, s_M) \tag{3.23}$$

The s_j's are known as the singular values of $\mathbf{X}$ and are arranged in descending order (i.e., $s_1 \geq s_2 \geq \ldots \geq 0$).

Consider now the matrix $\mathbf{X}\mathbf{X}^{\mathbf{T}}$:

$$\mathbf{X}\mathbf{X}^{\mathbf{T}} = \left(\mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}}\right)\cdot\left(\mathbf{V}\mathbf{S}\mathbf{U}^{\mathbf{T}}\right) = \mathbf{U}\mathbf{S}^2\mathbf{U}^{\mathbf{T}} \tag{3.24}$$

Multiplying both sides on the right by $\mathbf{U}$ yields:

$$\mathbf{X}\mathbf{X}^{\mathbf{T}}\mathbf{U} = \mathbf{U}\mathbf{S}^2 \tag{3.25}$$

Thus, the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{XX^T}$. The singular values squared are the corresponding eigenvalues. Since the singular values are arranged in descending order, the m eigenvectors associated with the largest eigenvalues are just $\mathbf{u_1}$, $\mathbf{u_2}$, ..., $\mathbf{u_m}$. These eigenvectors are exactly the set of m orthonormal basis vectors defined by the $\mathbf{l_i}$'s (Equation 3.47 with $\mathbf{Y_1} = \mathbf{I}$). Also note that the matrix $\boldsymbol{\alpha}$ in Equation 3.36 can be associated with the first m rows of $\mathbf{SV^T}$.

3.5.1 Accuracy of Representation

We have shown that the optimal basis of size m for representing a set of examples is obtained by taking the first m columns of $\mathbf{U}$ in the singular value decomposition. In this section, we quantify how accurately these basis functions represent the examples. (Again these results are generally well known [Pin85], but are included for completeness.) Using the SVD notation, the squared error for a single example can be expressed as:

$$e_j^2 = \left\| \mathbf{x_j} - \sum_{i=1}^{m} \mathbf{u_i}\, s_i v_{ji} \right\|^2 \tag{3.26}$$

But according to Equation 3.20, $\mathbf{x_j}$ can be exactly represented as

$$\mathbf{x_j} = \sum_{i=1}^{M} \mathbf{u_i}\, s_i v_{ji} \tag{3.27}$$

Substituting into Equation 3.26, we obtain

$$\begin{aligned} e_j^2 &= \left\| \sum_{i=m+1}^{M} \mathbf{u_i} s_i v_{ji} \right\|^2 \\ &= \left(\sum_{i=m+1}^{M} \mathbf{u_i}^T s_i v_{ji} \right) \cdot \left(\sum_{k=m+1}^{M} \mathbf{u_k} s_k v_{kj} \right) \\ &= \sum_{i=m+1}^{M} s_i^2 v_{ji}^2 \end{aligned} \tag{3.28}$$

(since $\mathbf{u_i}^T\mathbf{u_k} = \delta_{ik}$). The total squared error computed over all the examples is

$$\begin{aligned} E &= \sum_{j=1}^{M} e_j^2 = \sum_{j=1}^{M}\sum_{i=m+1}^{M} s_i^2 v_{ji}^2 \\ &= \sum_{i=m+1}^{M} s_i^2 \sum_{j=1}^{M} v_{ji}^2 = \sum_{i=m+1}^{M} s_i^2 \end{aligned} \tag{3.29}$$

where for the last step, we used the fact that the columns of $\mathbf{V}$ are normalized (Equation 3.22).

Equation 3.29 states that the accuracy of the representation depends on the sum of squares of the lower-order singular values. The number of basis functions m should be chosen based on the decay of the singular values. If the singular values decay very rapidly, then only a few basis vectors are needed to accurately represent the examples. On the other hand, if the singular values decay slowly, many basis vectors would be needed to represent the data with the same error.

The reader is cautioned that *Equations 3.27–3.29 apply only when* $\tilde{\mathbf{x}}_\mathbf{j}$ *is one of the examples used to compute the SVD.* The problem is that the SVD produces $M < N$ basis vectors (the $\mathbf{u_i}$'s). Using all M of these vectors, we can exactly span all the examples without any error (Equation 3.27). However, we cannot expect M basis vectors to span every pattern in an N dimensional space! In order to exactly represent an arbitrary pattern $\mathbf{t}$ in N-dimensional space, the set of basis vectors produced by the SVD must be augmented with $N - M$ additional orthonormal vectors, say $\boldsymbol{\omega_i}$ for $i = M+1, \ldots, N$. In this augmented basis, any pattern $\mathbf{t}$ can be exactly represented as

$$\mathbf{t} = \sum_{i=1}^{M} (\mathbf{u_i^T t})\, \mathbf{u_i} + \sum_{i=M+1}^{N} (\boldsymbol{\omega_i^T t})\, \boldsymbol{\omega_i} \tag{3.30}$$

$$= \sum_{i=1}^{M} \alpha_i \mathbf{u_i} + \sum_{i=M+1}^{N} \beta_i \boldsymbol{\omega_i} \tag{3.31}$$

where the definitions of α_i and β_i are obvious. The error between $\mathbf{t}$ and its recon-

struction using the first m principal components will be

$$||\mathbf{t} - \hat{\mathbf{t}}||^2 = \sum_{i=m+1}^{M} \alpha_i^2 + \sum_{i=M+1}^{N} \beta_i^2 \tag{3.32}$$

If $\mathbf{t}$ happens to be one of the training examples used to compute the SVD, it is easy to show that the β_i's will all be zero. The reconstruction error given in Equation 3.32 then reduces to the expression given previously (Equation 3.28).

3.6 Relationship to Principal Components Analysis

Although we have derived the basis functions from the standpoint of finding the best linear basis of rank m to represent a set of examples, there is another interpretation. If we rewrite the matrix product $\mathbf{XX^T}$ of Equation 3.24 in terms of the columns of $\mathbf{X}$, we find

$$\mathbf{XX^T} = \sum_{j=1}^{M} \mathbf{x_j x_j^T} \tag{3.33}$$

Now suppose the $\mathbf{x_j}$'s are random vectors with zero expected value. Then, the right hand side of Equation 3.33 divided by M corresponds to the sample covariance matrix $\hat{\mathbf{\Sigma}}$ of the data, i.e,

$$\hat{\mathbf{\Sigma}} = \frac{1}{M} \cdot \mathbf{XX^T} \tag{3.34}$$

The eigenvectors of $\mathbf{XX^T}$ are therefore the eigenvectors of the sample covariance matrix, and the singular values normalized by $\frac{1}{\sqrt{M}}$ are the standard deviations of the data along the eigenvector directions. We can think of the sample covariance matrix as a hyper-ellipsoid, with the $\mathbf{u_i}$'s being the major axes. The direction $\mathbf{u_1}$ corresponds to the direction of maximum variance; the direction $\mathbf{u_2}$ corresponds to the direction of maximum variance orthogonal to $\mathbf{u_1}$, and so on. Since the $\mathbf{u_i}$'s provide a compact representation of the primary directions of variance in a set of examples, they are also referred to as the principal components.

3.7 Summary

In this chapter we examined target models consisting of a linear combination of basis functions. We showed that the "best" set of basis functions in terms of approximating the examples with minimum RMS error can be obtained from the singular value decomposition. In particular, if the examples are written as columns of a matrix $\mathbf{X}$, the "best" m basis functions are the first m columns of $\mathbf{U}$ in the singular value decomposition of $\mathbf{X}$. The optimal classifier for discriminating between two object classes that consist of linear combinations of basis functions was derived. A test pattern should be classified based on the "distance" from each class, where the distance consists of two terms. The first term measures how well the test example is represented by the basis functions. The second term measures how well the weighting coefficients agree with other examples from the class.

3.8 Appendix: Derivation of the Optimal Basis

The mean square error that results from representing a set of examples $\mathbf{X}$ by linear combinations of basis vectors $\mathbf{L}$ with weighting coefficients $\boldsymbol{\alpha}$ was given in Equation 3.19 as

$$E = \mathrm{tr}\left[(\mathbf{X} - \mathbf{L}\boldsymbol{\alpha})^{\mathbf{T}} (\mathbf{X} - \mathbf{L}\boldsymbol{\alpha})\right] \tag{3.35}$$

To derive the optimal basis functions, we seek $\mathbf{L}$ and $\boldsymbol{\alpha}$ to minimize E subject to the orthonormality constraint $\mathbf{L}^{\mathbf{T}}\mathbf{L} = \mathbf{I}_{m \times m}$.

Given a tentative solution for $\mathbf{L}$, we can solve for $\boldsymbol{\alpha}$ by differentiating E with respect to $\boldsymbol{\alpha}$ (a matrix) and equating the result to $\underline{\mathbf{0}}$. This process yields:

$$\boldsymbol{\alpha} = \mathbf{L}^{\mathbf{T}}\mathbf{X} \tag{3.36}$$

Equation 3.36 states that the optimal coefficients $\boldsymbol{\alpha}$ are just the projections of the examples along the basis vectors.

Substituting Equation 3.36 into Equation 3.19 reduces the problem to a minimization over $\mathbf{L}$ of the quantity

$$\begin{aligned} E &= \mathrm{tr}\left[\mathbf{X}^{\mathbf{T}}\left(\mathbf{I} - \mathbf{L}\mathbf{L}^{\mathbf{T}}\right)^{\mathbf{T}}\left(\mathbf{I} - \mathbf{L}\mathbf{L}^{\mathbf{T}}\right)\mathbf{X}\right] \\ &= \mathrm{tr}\left[\mathbf{X}^{\mathbf{T}}\left(\mathbf{I} - \mathbf{L}\mathbf{L}^{\mathbf{T}}\right)\mathbf{X}\right] \\ &= \mathrm{tr}\left[\mathbf{X}^{\mathbf{T}}\mathbf{X} - \mathbf{X}^{\mathbf{T}}\left(\mathbf{L}\mathbf{L}^{\mathbf{T}}\right)\mathbf{X}\right] \end{aligned} \tag{3.37}$$

Minimizing E is equivalent to maximizing

$$Q = \mathrm{tr}\left[\mathbf{X}^{\mathbf{T}}\mathbf{L}\mathbf{L}^{\mathbf{T}}\mathbf{X}\right] \tag{3.38}$$

$$= \mathrm{tr}\left[\mathbf{L}^{\mathbf{T}}\mathbf{X}\mathbf{X}^{\mathbf{T}}\mathbf{L}\right] \tag{3.39}$$

where we have used the identity $\mathrm{tr}\,[\mathbf{AB}] = \mathrm{tr}\,[\mathbf{BA}]$ to obtain the second line from the first.

Since $\mathbf{XX^T}$ is symmetric, there exists an orthonormal matrix $\mathbf{P}$ such that

$$\mathbf{XX^T P} = \mathbf{P\Lambda} \tag{3.40}$$

with $\mathbf{\Lambda}$ a diagonal matrix. For convenience, we will rearrange the columns of $\mathbf{P}$ so that $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_N \geq 0$. Thus, the first column of $\mathbf{P}$ is the eigenvector of $\mathbf{XX^T}$ having the largest corresponding eigenvalue.

We can rewrite $\mathbf{XX^T}$ in terms of $\mathbf{P}$ and $\mathbf{\Lambda}$ as

$$\mathbf{XX^T} = \mathbf{P\Lambda P^T} \tag{3.41}$$

Substituting into the expression for Q in Equation 3.39, yields

$$\begin{aligned} Q &= \mathrm{tr}\left[\mathbf{L^T P\Lambda P^T L}\right] \\ &= \mathrm{tr}\left[\mathbf{Y^T \Lambda Y}\right] \\ &= \mathrm{tr}\left[\mathbf{\Lambda Y Y^T}\right] \end{aligned} \tag{3.42}$$

where $\mathbf{Y}$ is defined to be $\mathbf{P^T L}$. We originally wanted to maximize Q over $\mathbf{L}$ subject to the constraint $\mathbf{L^T L} = \mathbf{I}$, but this is equivalent to maximizing Q over $\mathbf{Y}$ subject to the constraint $\mathbf{Y^T Y} = \mathbf{I}$ (since $\mathbf{P}$ is invertible and $\mathbf{Y^T Y} = \mathbf{L^T L}$).

Equation 3.42 can be expressed as follows:

$$\begin{aligned} Q &= \sum_i \lambda_i \sum_k y_{ik}^2 \\ &= \sum_i \lambda_i \gamma_i \end{aligned} \tag{3.43}$$

where γ_i is the sum of the squares of elements of the i^{th} row of $\mathbf{Y}$.

Because of the constraint $\mathbf{Y^T Y} = \mathbf{I}$,

$$\sum_i \sum_k y_{ik}^2 = m = \sum_i \gamma_i \tag{3.44}$$

Also, $0 \leq \sum_k y_{ik}^2 \leq 1$. We can prove this inequality by augmenting $\mathbf{Y}$ with $N - m$

normalized columns that are orthogonal to the first m columns to get a square matrix $\tilde{\mathbf{Y}}$. Then $\tilde{\mathbf{Y}}^{\mathbf{T}}\tilde{\mathbf{Y}} = \mathbf{I}$ implies that $\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^{\mathbf{T}} = \mathbf{I}$, because $\tilde{\mathbf{Y}}$ is a square matrix. Therefore, $\gamma_i = \sum_k y_{ik}^2 \leq 1, \forall i$.

We now have a fixed budget of energy m to place into the γ_i's (or equivalently, into the rows of $\mathbf{Y}$). Our goal is to maximize Q, so we want to avoid spending energy on the lower rows if possible because this energy is amplified by a smaller λ_i. If it were possible, we would put all the energy in the first row because it would be amplified by the biggest λ. Unfortunately, the inequality shown above prevents any row from having more than 1 unit of energy. Thus, the best we can do is to put one full unit of energy in each of the first m rows and none in the remaining rows i.e.,

$$\begin{aligned} \gamma_1 &= \gamma_2 = \cdots = \gamma_m = 1 \\ \gamma_i &= 0, \forall i > m \end{aligned} \tag{3.45}$$

The distribution of energy that achieves the maximum Q can be obtained using any $\mathbf{Y}$ of the form

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y_1} \\ \cdots \\ \mathbf{0} \end{bmatrix} \tag{3.46}$$

where $\mathbf{Y_1}$ is a square $(m \times m)$ matrix such that $\mathbf{Y_1^T Y_1} = \mathbf{I}$. (This constraint on $\mathbf{Y_1}$ is necessary in order to satisfy $\mathbf{Y^T Y} = \mathbf{I}$).

The optimal solution for $\mathbf{L}$ is, therefore, given by:

$$\mathbf{L} = \mathbf{PY} \tag{3.47}$$

$$= \mathbf{P} \begin{bmatrix} \mathbf{Y_1} \\ \cdots \\ \mathbf{0} \end{bmatrix} \tag{3.48}$$

Hence, any orthogonal transformation of the first m columns of $\mathbf{P}$ is an optimal solution

for $\mathbf{L}$. With $\mathbf{Y_1} = \mathbf{I}$, the columns of $\mathbf{L}$ are simply the m eigenvectors of $\mathbf{XX^T}$ corresponding to the largest eigenvalues.

To summarize the derivation, the set of m orthonormal basis vectors that best span a set of examples $\mathbf{X}$ are the m eigenvectors of $\mathbf{XX^T}$ having the largest eigenvalues (or any orthogonal transformation of these eigenvectors). As discussed in Section 3.5, an equivalent solution is provided by the singular value decomposition (SVD) of $\mathbf{X}$.

Chapter 4 Recognition of Small Volcanoes on Venus

4.1 Introduction

This is an applications chapter[1] in which we discuss JARtool[2] (JPL Adaptive Recognition Tool), a prototype system for automatically locating and cataloging geological features in remote sensing databases. For a number of reasons, the problem of finding small volcanoes in the Magellan SAR imagery of Venus was selected as the initial testbed for JARtool algorithm development. The Magellan dataset has provided scientists with the most comprehensive global picture ever of any planetary surface, including even Earth since our planet's surface is largely obscured by water. Magellan was successful in imaging over 98% of the Venusian surface and, in fact, returned more data than all previous planetary missions combined. Planetary geologists are understandably excited about the potential scientific impact of this dataset, but are lacking automated tools to aid in the analysis of the data.

One of the dominant geological processes on Venus is volcanism. Preliminary global surveys of the Magellan data have shown that there are approximately 1400 volcanic features larger than 20km in diameter [H+91]. Based on previous observations from Soviet Venera 15/16, U.S. Pioneer Venus, and ground-based radar, planetary geologists estimate the number of small volcanoes (diameter $<$ 20km) to be $\sim 10^6$ [AS90]. Generating a comprehensive, global catalog that includes the location and size of each volcano is essential in order for the geologists to validate scientific theories about the relationship between volcanoes and local tectonic structure and to

[1] For the reader primarily interested in theory, this chapter may be skipped without loss of continuity.

[2] The JARtool project was a collaboration between the Machine Learning Systems Group at JPL and the Vision Group at Caltech. Principal investigators were M.C. Burl, U.M. Fayyad, P. Perona, and P. Smyth

understand heat flow patterns within the planet. Much of our current understanding of planetary geology is derived from experience on Earth, so analysis of Venus through the Magellan data will provide an invaluable second data point.

4.2 Magellan Imagery

The fundamental objective of the Magellan mission was to provide global mapping of the surface of Venus. Due to the dense cloud cover surrounding Venus, it was necessary to use synthetic aperture radar (SAR) to perform the mapping. A complete description of the Magellan SAR imaging system is given in [PFJ+91], so here we will summarize only the most important characteristics. The spacecraft was inserted in a polar elliptical orbit in August of 1990. Figure 4.1 [MGN] shows an artist's depiction of Magellan. With each orbit the radar imaged a 17–28 km swath on the ground. Over the course of one Venusian day (243 Earth days), most of the planet's surface (84%) was successfully imaged and relayed back to Earth. Subsequent passes boosted the total surface coverage to 98%.

The nominal incidence angle in the Magellan data ranges from 15° – 45° as a function of latitude on the planet. The imagery is available at several resolutions with the highest resolution data product being the F-MIDR's (Full Resolution Mosaicked Image Data Records) which have a resolution of 120m in azimuth and 120m–360m in range. The F-MIDR images are 1024 ×1024 pixels with pixel spacing equal to 75m (slightly oversampled).

A 30 km × 30 km subimage from one of the F-MIDRs is shown in Figure 4.2. This area located near (lat 30°N, lon 332°) contains a number of small volcanoes. Most of these volcanoes have the classic radar signature one would expect based on the topography and illumination direction (illumination is from the lower left); that is, the upward sloping surface of the volcano in near-range (close to the radar) scatters more energy back to the sensor than the surrounding flat plains and therefore appears bright. The downward sloping surface of the volcano in far-range scatters energy away from the sensor and therefore appears dark. Together, these effects cause the volcano

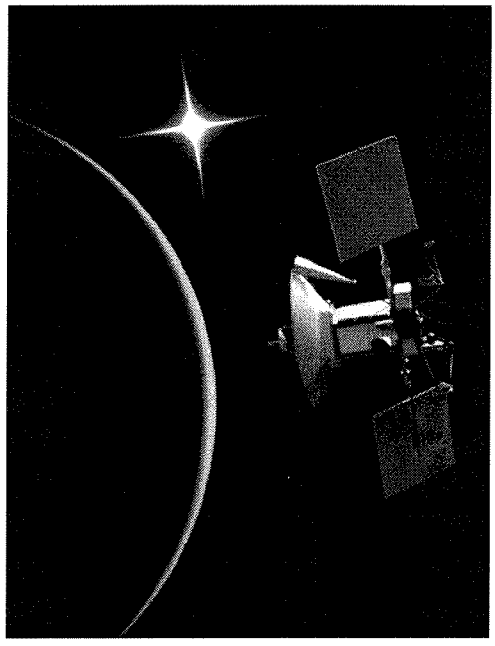


Figure 4.1: Artist's depiction of Magellan spacecraft at Venus.

to appear as a left-to-right *bright-dark pair* within a circular planimetric outline. Near the center of the volcanoes, there is usually a summit pit that appears as a *dark-bright* pair because the radar energy backscatters strongly from the far-range rim. Small pits, however, may appear as only a bright spot or not at all depending upon the pit size relative to the radar resolution.

The topography-induced features described above are the primary visual cues that geologists report using to locate volcanoes. However, there are a number of other more subtle cues. The apparent brightness of an area in a radar image depends not only on the macroscopic topography but also on the surface roughness relative to the radar wavelength. If the flanks of a volcano have different roughness properties than the surrounding plains, the volcano may appear as a bright or dark circular area instead of as a bright-dark pair. Volcanoes may also appear as radial flow patterns, texture differences, or disruptions of graben. (Graben are ridges or grooves in the planet surface, which appear as bright lines in the radar imagery — see Figure 4.2.)

An added difficulty with the Magellan dataset is that there is no absolute ground

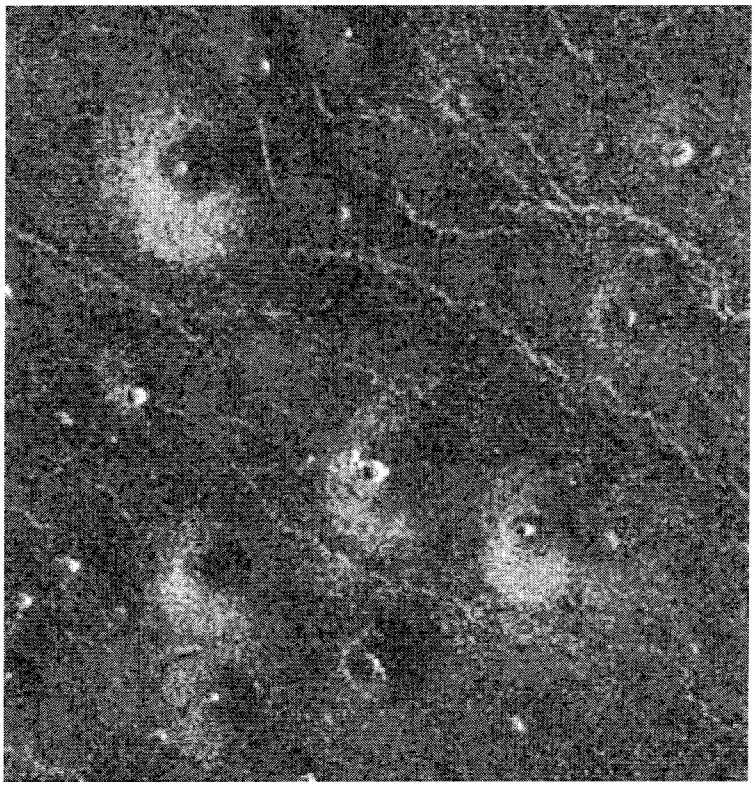


Figure 4.2: Magellan SAR subimage: A 30km $\times$ 30km region containing a number of small volcanoes. Illumination is from the lower left; incidence angle $\approx 40°$.

truth. No one has ever been to the surface of Venus, apart from a Russian robotic lander that melted within minutes, so any ground truth must be derived from the imagery. The Magellan imagery is the best available, but trained scientists still cannot say with 100% certainty whether a particular feature in the imagery is indeed a volcano. There is considerable subjectivity due to the radar resolution, noise level, etc. Thus, only uncertain ground truth is available for generating training examples and evaluating performance.

Part of JARtool is a graphical user interface (GUI) that enables scientists to provide training examples by fitting circles around image features that may correspond to volcanoes. The scientists also provide a label indicating their subjective confidence p that the selected object is indeed a volcano. The confidence labels are quantized into four categories, which were determined based on discussions with the scientists:

Category 1: $p \in [0.95, 1.0]$. Almost certainly a volcano, with all primary visual cues present.

Category 2: $p \in [0.75, 0.95]$. Probably a volcano, but a non-essential visual cue is missing.

Category 3: $p \in [0.5, 0.7]$. Possibly a volcano, but at least two of the primary cues are missing.

Category 4: $p \approx 0.5$. Only a pit is visible; could be a volcano, but more evidence is needed.

Figure 4.3 shows examples of volcanoes in each confidence category.

"Consensus ground truth" is generated by several scientists working together and discussing the merits of each candidate volcano. The consensus data is then used as if it were the actual ground truth. Figure 4.4 shows consensus data for a typical image. Of course, an individual scientist who labels a set of images will not produce exactly the same results as the consensus. This fact is illustrated in Figure 4.5 which shows the confusion matrices for two individual scientists (A and B) relative to the consensus. The (i, j) entry is interpreted as the number of volcanoes labeled i by an

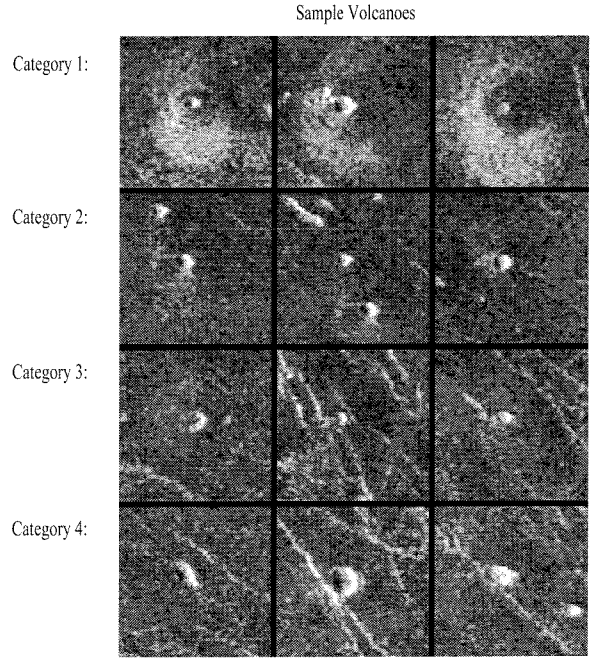


Figure 4.3: Examples of volcanoes from each confidence category.

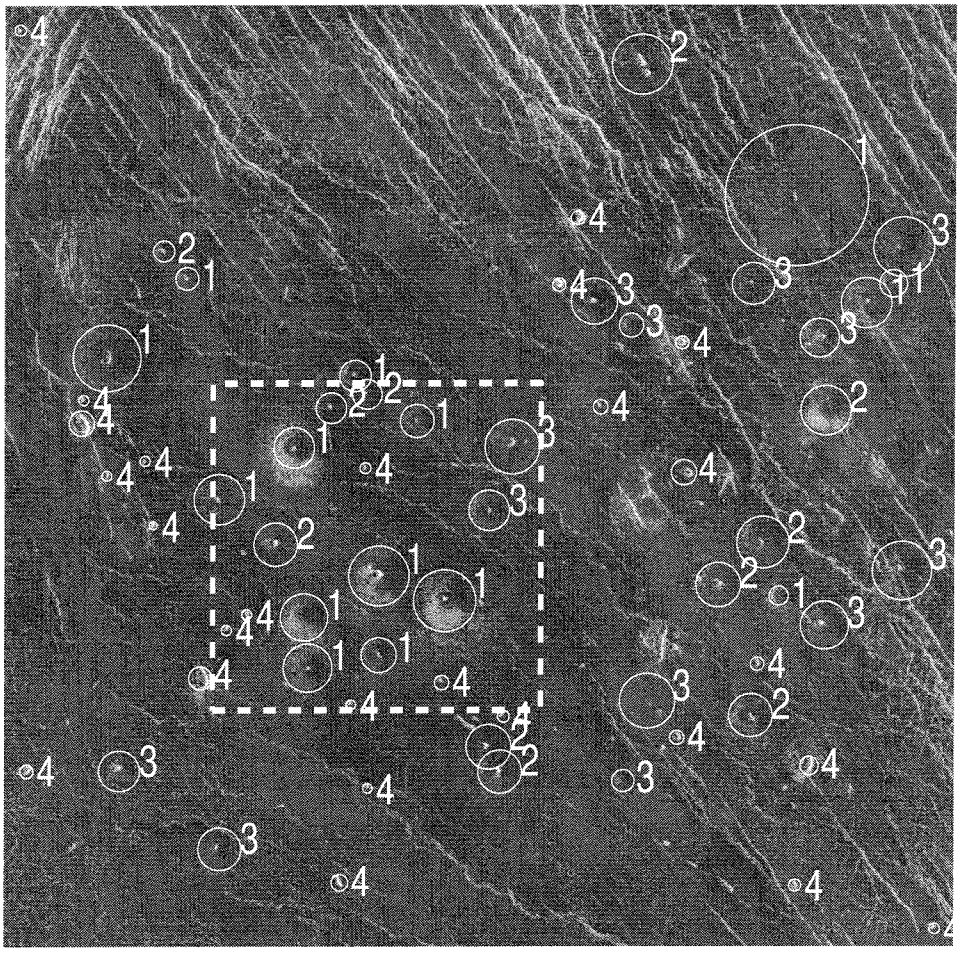


Figure 4.4: Magellan image (75 km $\times$ 75 km) with consensus ground truth showing suspected small volcanoes including size, location, and subjective confidence. The dashed box shows the area depicted in Figure 4.2.

		Consensus				
		1	2	3	4	0
A	1	28	11	6	1	0
	2	5	9	8	9	9
	3	1	2	20	8	31
	4	1	2	5	26	13
	0	0	6	11	4	0

		Consensus				
		1	2	3	4	0
B	1	19	6	6	3	1
	2	9	5	9	4	6
	3	4	13	18	6	37
	4	0	3	3	25	18
	0	3	3	14	10	0

Figure 4.5: The performance of two individual scientists (A and B) compared to 'consensus' ground-truth.

individual that were labeled j in the consensus. The last row of the confusion matrix shows the number of misses (volcanoes not labeled by the individual), while the last column shows the number of false alarms. The $(0, 0)$ entry has no meaning so we have defined it to be 0. Our goal in developing an automatic volcano-detection algorithm is to achieve performance relative to the consensus that is comparable to that of an individual scientist (also judged relative to the consensus). The philosophy here is that if an individual scientist is qualified to perform the analysis, then it is sufficient if our algorithms perform comparably.

4.3 Algorithm Description

The JARtool algorithm consists of three stages: focus of attention (FOA), feature learning/measurement, and classification. In the first stage of processing, the FOA algorithm is used to quickly scan through an image and output a list of candidate volcano locations. Regions not identified by the FOA are eliminated from subsequent processing so it is important that the algorithm not miss too many of the true volcanoes

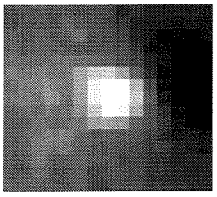
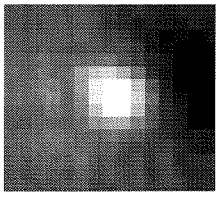
(a)  (b) 

Figure 4.6: (a) Matched filter $\mathbf{s}_1$ constructed by averaging internally normalized volcano examples. (b) Matched filter $\mathbf{s}_2$ constructed by averaging CFAR normalized examples. Both methods produce almost the same filter (aside from different DC value and scale factor). Notice that the matched filter contains many of the characteristics that planetary geologists report using to manually locate volcanoes. In particular, the filter has a bright central spot corresponding to the volcanic summit pit and left-to-right bright-dark shading induced by the volcano topography.

at this stage. The FOA should also be relatively cheap computationally since it must be applied to every pixel in an image. Given these constraints, we have chosen a simple matched filter as the basis for the FOA. The matched filter is synthesized from the example volcanoes in the training image set. Given the assumption that the volcano class consists of a single prototype volcano corrupted by noise, a good estimate for the ideal matched filter is obtained by computing the mean of the training examples.

Due to differences in DC and contrast between the images, however, it is necessary to first normalize the training examples. There are two basic ways this can be done: (1) internal normalization and (2) CFAR normalization. Let $\mathbf{v}_i$ denote a $k \times k$ pixel region around the i-th training volcano. Both methods replace $\mathbf{v}_i$ by the normalized example

$$\tilde{\mathbf{v}}_i = \frac{\mathbf{v}_i - \mu_i \cdot \mathbf{1}}{\sigma_i} \tag{4.1}$$

The matched filter is then calculated by averaging the $\tilde{\mathbf{v}}_i$'s. The difference between the two methods is in the calculation of μ_i and σ_i. With the internal normalization method, μ_i is the mean of the pixels in $\mathbf{v}_i$ and σ_i is the standard deviation. With CFAR normalization, μ_i and σ_i are calculated from pixels in the *background* near $\mathbf{v}_i$. For the volcano problem, the two methods produce similar filters as shown in Figure 4.6.

When the matched filter is applied to an image, the local image patch must also

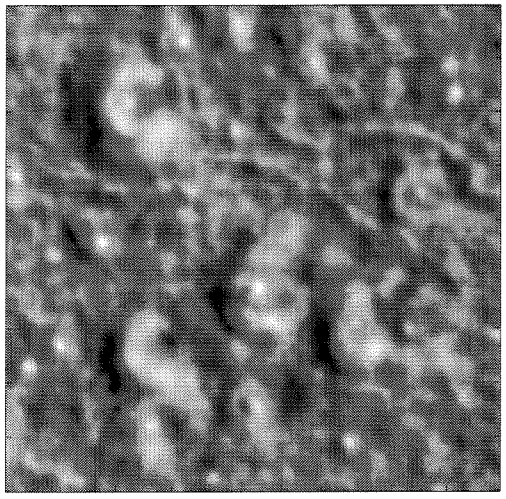
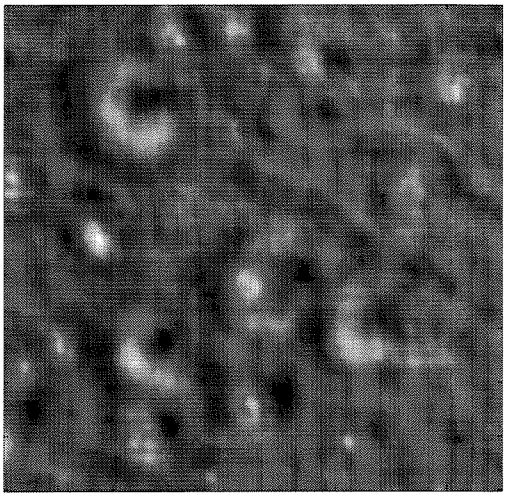
(a)  (b) 

Figure 4.7: (a) Response of matched filter $\mathbf{s}_1$ on the subimage from Figure 4.2. using the internal image normalization method. (b) Corresponding result for $\mathbf{s}_2$ and the CFAR image normalization method. In both images, bright points indicate a strong match — these will be selected as candidate volcano locations.

be normalized, again using either internal or CFAR normalization. A third alternative is to apply the matched filter without image normalization followed by CFAR thresholding on the response image. Asymptotically (i.e., with a large number of samples to the estimate background statistics), this method should produce the same result as performing CFAR normalization on each image patch. Figure 4.7a shows the response of the matched filter $\mathbf{s}_1$ to the subimage from Figure 4.2 using the internal image normalization method. Figure 4.7b shows the corresponding result for $\mathbf{s}_2$ and CFAR normalization. In general the contrast of the response image produced with CFAR normalization appears better. However, sometimes we do not get a large response where we would expect one. The summit pit of the large volcano in the upper left is readily detected with the internal image normalization method, but not with CFAR normalization. The reason is that the CFAR stencil used to estimate the background statistics overlaps with the bright volcano flanks and produces anomalously high estimates of the background DC level.

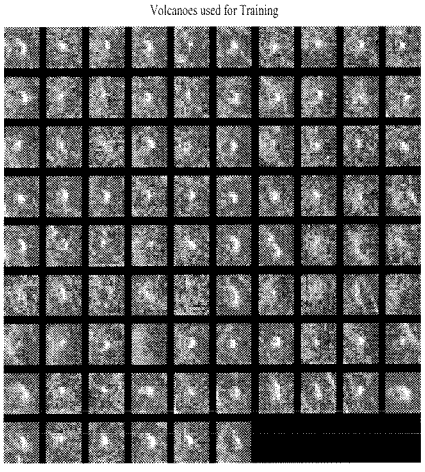
We can try to improve upon the matched filter by using the SVD-based methods discussed in Chapter 3 to better model the variability of the volcano class. The goal

here is to reject any false alarms generated by the FOA while retaining as many of the true volcanoes as possible. The volcano class is modeled as a linear combination of the basis functions produced by principal components analysis with a particular probability distribution on the weighting coefficients. For an unknown test chip, the weighting coefficient for each SVD basis chip is simply the projections of the test chip onto the basis chips. These weighting coefficients serve as feature values which can be used to classify a chip as belonging to the volcano class (ω_1) or not (ω_2). The orthogonal out-of-subspace component (reconstruction error) could also be used in the classification decision as discussed in Chapter 3, but we have not done that in the experiments reported here.

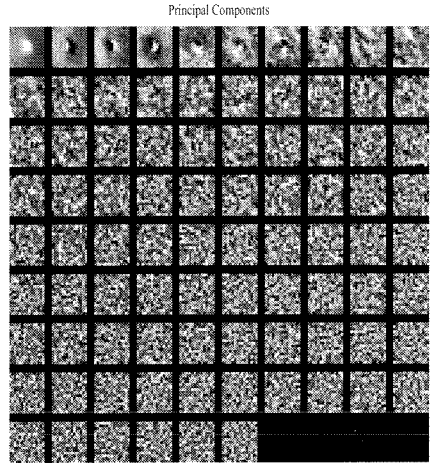
Figure 4.8a shows a set of example volcanoes from one of the cross-validation training sets. Figure 4.8b shows the corresponding set of SVD basis function ordered from left-to-right and then top-to-bottom by singular value. Any of the example volcanoes in the training set can be expressed *exactly* as a linear combination of the SVD basis chips. Further, any of the example volcanoes can be expressed *approximately* using only the first K basis chips. The error of the approximation depends on the decay of the singular values (see Equation 3.29). Notice from Figure 4.8c that the first six to ten singular values dominate. Also, the first six to ten basis functions in Figure 4.8b are the ones that show visual structure. The basis functions corresponding to smaller singular values look very random indicating that they merely encode noise in the training set.

For our experiments we have represented the volcano class using a single SVD basis, but we have not attempted to model the background object class. Since the background class consists of everything that is not a volcano, we believe this class is too complex to be modeled as a linear combination of a small number of basis functions. Thus, we try to classify unknown examples based on how well the features (projection coefficients) agree with those of the volcano class; the reconstruction error was not used in the decision. We have experimented with a number of classifiers to do the mapping from projection space to class identity (ω_1 or ω_2). Among these are quadratic classifiers, nearest neighbor, neural network, decision trees, and kernel

(a)



(b)



(c)

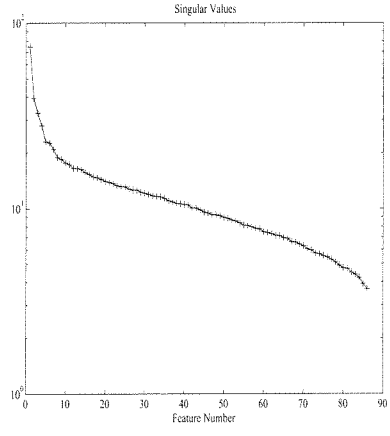


Figure 4.8: (a) Volcano training set. (b) Basis functions (principal components) ordered from left to right by decreasing singular value. Note that the first six to ten basis functions show visual structure, while the others appear to be random noise. (c) Singular values corresponding to the basis functions.

density estimators. All yield similar results so we have used the quadratic classifier as our default algorithm. The quadratic classifier is the optimal classifier if the class-conditional densities in projection space are Gaussian.

4.4 Experimental Performance Results

Preliminary experiments were conducted using a cross-validation paradigm on four images `OLD4` containing 163 small volcanoes and covering a 150km $\times$ 150km area of the planet. All results are scored relative to the scientists' consensus labeling with confidence categories 1-4 treated as true volcanoes. The figure of merit measured is the percentage of true volcanoes detected versus the number of false alarms per square kilometer. Additional tests were conducted using a set of 38 homogeneous images `HOM38` from the same area of the planet, and a set of 36 heterogeneous images `HET36` selected at random over the surface of the planet. Normally with the cross-validation paradigm, $n-1$ images are used for training and the remaining image is used for testing; this process is repeated n times so that each image serves as the test image. For the larger image sets, however, images were placed into subgroups such that each subgroup had approximately the same number of volcanoes. Training was performed using $k-1$ subgroups with testing done on the subgroup left out of training; the process was repeated so that each subgroup served as the test set.

The overall performance of the JARtool system can be summarized using a curve similar to the receiver operating characteristics (ROC curves) that we used in Section 2.5. The percentage of true volcanoes detected provides a reasonable estimate for the probability of detection. However, it is not clear how one should estimate the *probability* of false alarm since there are not a fixed number of "false alarm opportunities." In this situation, a related curve called an FROC (free-response ROC) is typically used [CW90]. The FROC simply shows the trade-off between detection probability and the *number* of false alarms per image or per unit area as the aggressiveness of the algorithm is varied. For convenience, we will use the term ROC for both types of curves since the x-axis label can be used to identify whether a given curve is actually

an FROC.

4.4.1 Performance on OLD4

The performance of the matched filter on the initial set of four test images OLD4 is shown in Figure 4.9. The performance curve, which is implicitly parameterized by the threshold applied to the matched filter output, shows the trade-off between missed volcanoes and false alarms. For aggressive settings of the threshold (low values), the algorithm readily declares things to be volcanoes. Hence, most of the true volcanoes are detected, but a bad side-effect is that many non-volcanoes are mistakenly accepted as volcanoes. Increasing the threshold will decrease the number of false alarms, but more of the true volcanoes will be missed.

Performance curves are shown for both the matched filter using internal normalization and the matched filter using CFAR normalization. On this set of images, the internal normalization method works better yielding approximately half as many false alarms at the same detection level. The performance for two other methods is also shown. The probability weighted filter uses the scientist subjective labels to weight the examples during training. The idea is to bias the matched filter towards objects that are certainly volcanoes while weakening the influence of uncertain volcanoes. The performance, however, is almost identical to the standard (uniformly-weighted) matched filter. The size-binned matched filter experiments will be discussed in Section 4.5.

To assess algorithm performance relative to humans, we evaluated three planetary scientists who are all familiar with the Magellan data and with the appearance of volcanoes in the data. The scientists' performance points relative to the consensus are shown on Figure 4.9 as asterisks. Note that the matched filter performance is significantly worse than the cluster of scientist points. This result is not surprising since we knew in advance that the target class model assumed by the matched filter was too simplistic.

The performance of the end-to-end algorithm, which consists of the matched filter FOA, projection onto SVD basis functions, and classification with a quadratic classi-

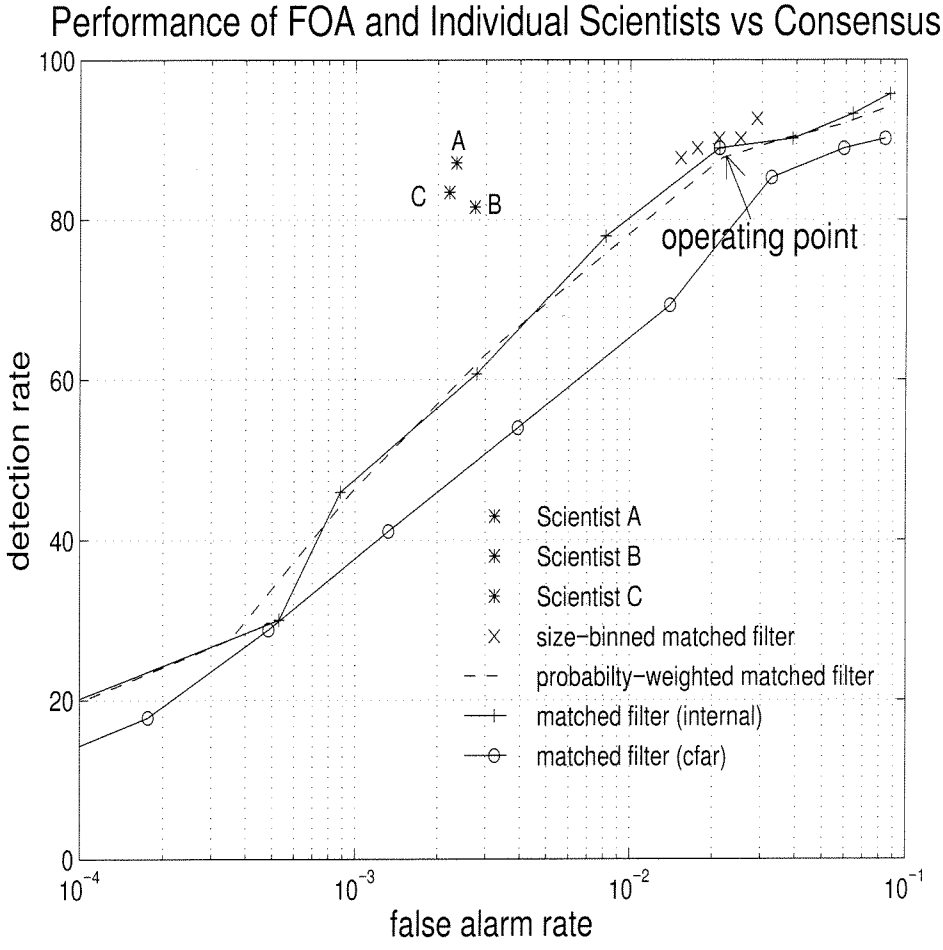


Figure 4.9: Matched filter performance compared to scientists.

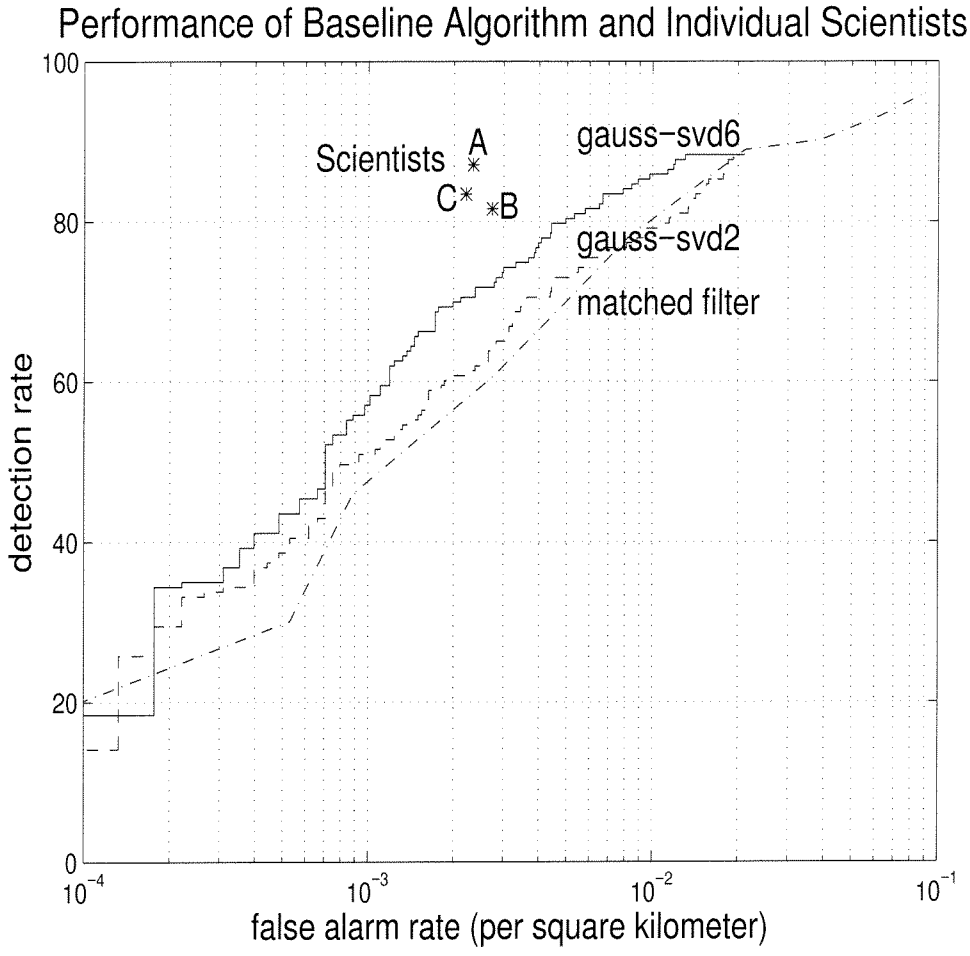


Figure 4.10: Baseline performance on OLD4 for the combined matched filter and SVD approach. Gauss-svd6 shows the performance with 6 principal components, while Gauss-svd2 shows the performance with only two components.

fier, is shown in Figure 4.10 for the OLD4 images. Here we have used a six dimensional feature space (six SVD basis functions). As with the FOA, the classifier has a parameter that can be varied to select aggressiveness. At maximum aggressiveness, every candidate from the FOA is declared to be a volcano; hence, the classifier performance curve is constrained to start from the FOA operating point. The combination of FOA and classification is clearly better than the matched filter alone (which was proposed in [WF93]).

4.4.2 Performance on HOM38

The OLD4 images were initially selected for algorithm development and testing because they contained a high density of small volcanoes. These images, which are located on the planet near $(30°N, 332°E)$, are actually part of a larger 7×8 block of images. Of these, 14 images are blank due to a gap in the Magellan data acquisition process. The remaining 38 images $(56 - 14 - 4 = 38)$ were selected to provide an expanded set of images for training and testing. Since all of the images are from the same area of the planet, the volcanoes are more homogeneous in appearance than one could typically expect from volcanoes selected at random over the surface of the planet. Thus, we will refer to this dataset as HOM38. There are approximately 480 volcanoes in HOM38. The volcanoes were labeled independently by two scientists (A and B) and the author (MCB). As with the OLD4, there is no absolute ground truth for these images so we can only assess performance relative to one of the scientists.

To limit the number of cross-validation runs, we partitioned the HOM38 images into 6 subgroups of 6 images, each containing approximately 80 volcanoes. The two remaining images were not part of any subgroup and were always included in the training set. As discussed above, cross-validation was done on a subgroup basis. That is, we trained on five subgroups (plus the extra two images) and tested on the other subgroup. The process was rotated so that each subgroup served as a test set.

The performance of the end-to-end algorithm on the six cross-validation partitions is shown in Figure 4.11. The labeling of Scientist A is treated as ground truth. The solid circle shows the performance of Scientist B relative to A, while the plus sign shows the performance of MCB relative to A.

The performance of the end-to-end algorithm versus the matched filter alone is shown (only for one cross-validation partition) in Figure 4.12. The use of linear combinations of basis functions to better model the variability in the volcano class has indeed improved performance over the matched filter alone. Notice that the end-to-end performance curve originates from the operating point on the FOA curve. At this point, the classifier is simply saying everything generated by the FOA is a volcano. As

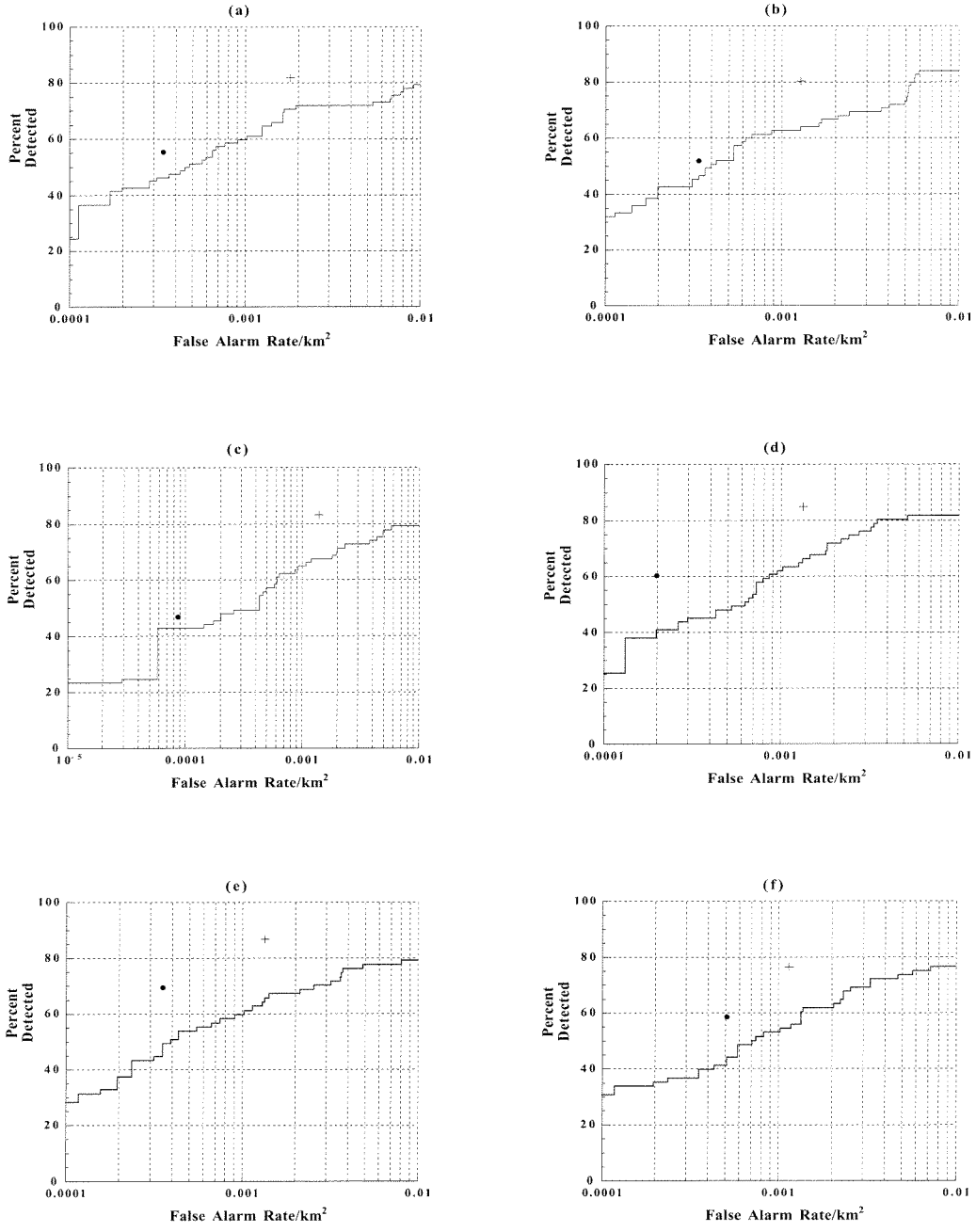


Figure 4.11: Overall performance on HOM38 for each of the 6 cross-validation partitions. In each case, training was done on 32 images and testing on 6. The solid circle shows the performance of Scientist B; the plus shows the performance MCB. Both humans and algorithms are judged relative to the labeling of Scientist A.

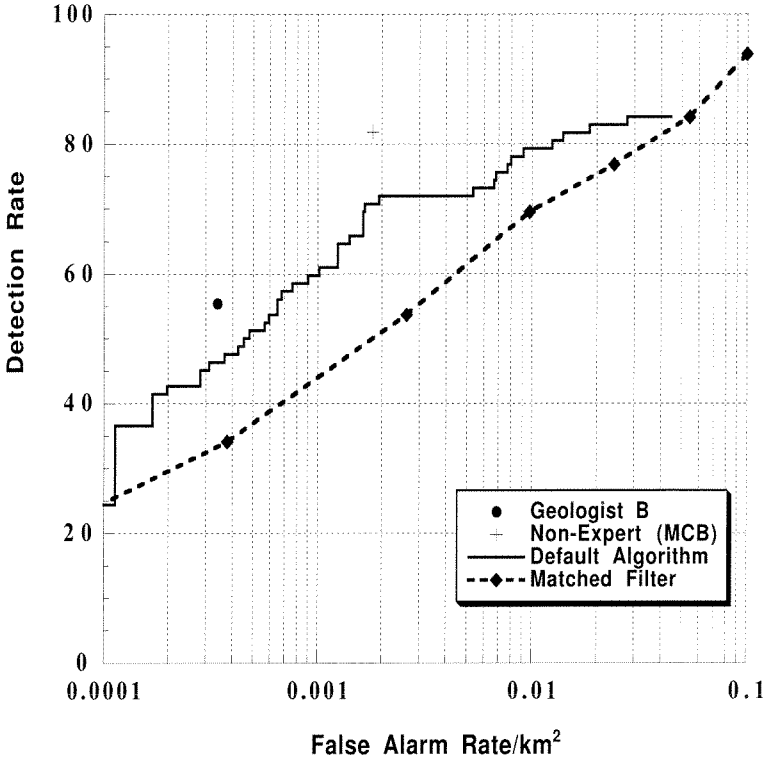


Figure 4.12: Performance of the matched filter alone (dashed line) compared to the combination of matched filter, SVD, and Gaussian classifier (solid line). The solid circle and plus sign show the performance of Scientist B and MCB respectively. These results are for a single cross-validation partition (partition a).

the classifier threshold is varied to make the classifier more selective, the false alarm and detection rate follow the solid curve.

4.4.3 Performance on HET36

The results on OLD4 and HOM38 are somewhat rosier than we can expect in general since these images are from the same area of the planet. To test performance in a less controlled setting, we randomly selected 36 images from scattered locations on the planet. These images are quite heterogeneous with significantly more variability both in appearance of the volcanoes and the background. Thus, this data set is designated HET36. There are approximately 670 volcanoes in these images. The images were placed into 4 subgroups of 9 images; cross-validation training and testing was done on a subgroup basis. The performance of the end-to-end system is shown

in Figure 4.13. Here performance is relative to a consensus labeling done jointly by Scientists A and B. Unfortunately, we do not have individual labeling for the entire set of 36 images. On those images where we do have multiple labelings, the relative performance of the scientists seem consistent with their performance on HOM38. As you will notice, however, by comparing Figure 4.13 and Figure 4.11, the algorithm performance is significantly worse on the heterogeneous images. This is not surprising since the algorithm is modeling the volcano class as linear combinations of a single set of basis functions. Clearly, this type of model is only appropriate for a small range of variability in appearance. It is possible that first clustering the volcanoes and then performing separate SVD's on each cluster might improve the algorithm performance.

4.5 Auxiliary Experiments

4.5.1 Size-Binned Matched Filter

Both the matched filter and SVD basis functions are derived from 30 pixel by 30 pixel regions selected from the center of the training examples. Since many of the volcanoes are larger than this, it is natural to wonder whether better performance could be obtained by trying to account for the size information. To investigate this possibility, we have conducted experiments with a variation of the matched filter, which we call the size-binned matched filter. The training volcanoes are grouped into four clusters based on the scientist-fitted diameters. A separate matched filter is constructed for each size range. The candidate locations identified by each of the matched filters are then merged and consolidated into a single master list of candidates.

For the size-binned algorithm it is difficult to obtain an ROC performance *curve* since each of the filters has its own threshold. In principle we could evaluate performance at many combinations of threshold settings and use the outer envelope of performance as the ROC curve. In practice, this is difficult, so we have just evaluated the performance for several different threshold combinations. The corresponding detection and false alarm points are shown in Figure 4.9 with x's.

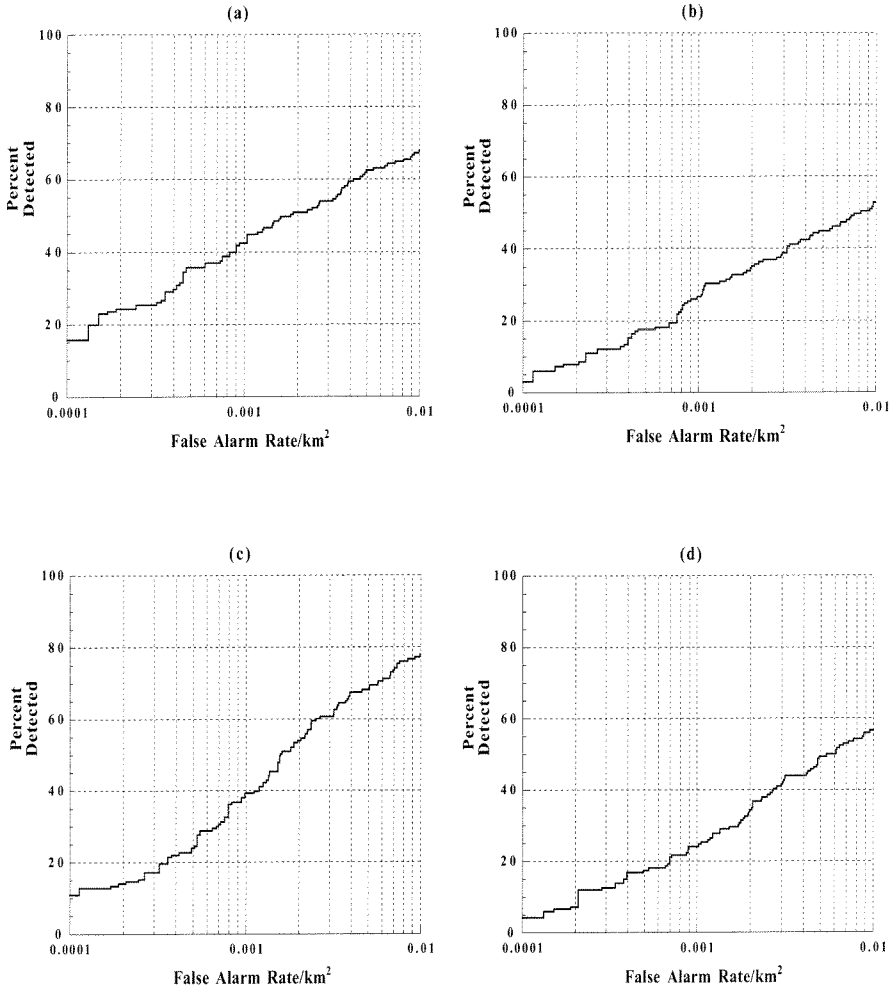


Figure 4.13: Overall performance on HET36 for each of the 4 cross-validation partitions. In each case, training was done on 27 images and testing on 9. Notice that the performance is significantly worse than on HOM38 (Figure 4.11).

Observe that the size-binned matched filter improves only slightly upon the performance of the single-scale matched filter. The size-binned algorithm, however, requires considerably more computation time, is more difficult to synthesize, and has more parameters to adjust than the single-scale version. It appears that the marginal improvement is not worth the increased complexity. Thus, we continue to use the single-scale version.

4.5.2 Sensitivity to Matched Filter Operating Point

A major concern with any algorithm is how sensitive the performance is to the exact settings of the parameters. If a parameter must be accurate to 10 digits in order for the system to work and the parameter is to be estimated from a limited number of noisy training examples, the system will not be of much use in practice. One of the key parameters in the JARtool system is the threshold applied to the matched filter output since this establishes how many true volcanoes and false alarms are passed on to the SVD and classification stages for further processing. Figure 4.14 shows the end-to-end performance of the JARtool algorithm as a function of the matched filter threshold. (These curves were generated from tests on partition (a) of `HOM38` , but are typical of the results on other partitions.) To simplify the display, we show the detection rate at three fixed false alarm rates versus the threshold parameter. Observe that the performance is relatively stable as the threshold varies from 0.35 to 0.45.

4.5.3 Sensitivity to Number of SVD Features

An empirical study was performed to evaluate the sensitivity of the algorithm to the number m of SVD features used. Figure 4.15 shows the measured detection rate versus m at a few selected false alarm rates. Since the detection curves are relatively flat with respect to m, we conclude that the performance is insensitive to the exact number of features, provided at least four are used.

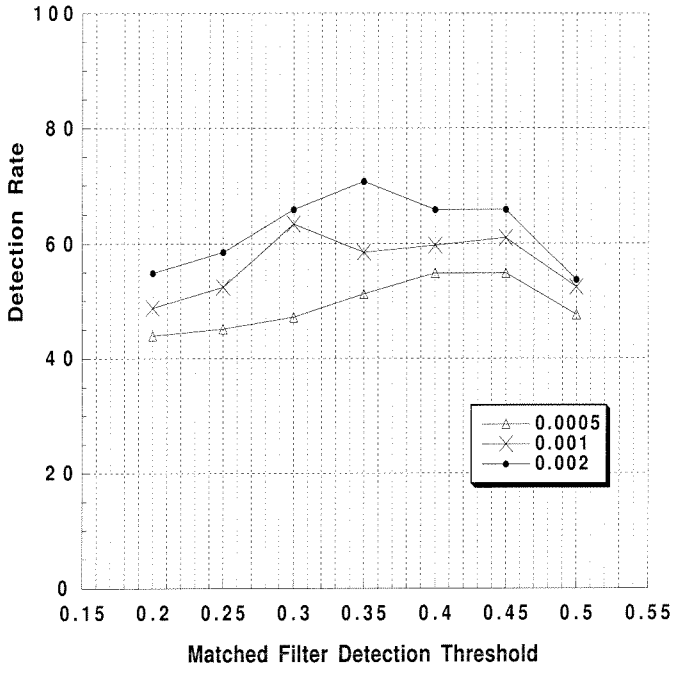


Figure 4.14: Sensitivity of overall performance to the FOA operating point. The three curves correspond to the probability of detection at three different false alarm rates as a function of the threshold applied to the matched filter output. The default threshold used in the experiments was 0.35.

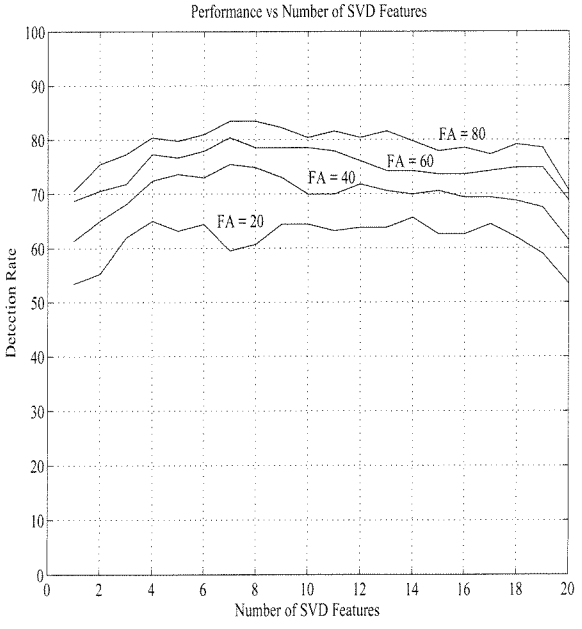


Figure 4.15: Empirical performance versus number of SVD features.

4.6 Summary

JARtool is a prototype system for automatically locating and cataloging geological features in remote sensing databases. The problem of finding small volcanoes in the Magellan imagery of Venus was selected as an initial testbed for algorithm development because of the potential scientific impact. The sheer size of this dataset amplifies the importance of developing automated tools for finding objects of interest in the data; the traditional approach of manually analyzing each collected image is no longer feasible[3].

The prototype JARtool system consists of three stages: focus of attention (FOA), feature learning/measurement, and classification. The FOA is based on image normalization techniques and matched filtering. The purpose of the FOA is to reduce computational costs by providing a quick method to prescreen the data and eliminate uninteresting areas (e.g., barren plains) from further consideration. Candidate areas identified by the matched filter are passed on to later stages of the algorithm, where the goal is to eliminate false alarms while keeping as many of the true volcanoes as possible. To accomplish this goal, it is important to model the variability in appearance of the volcano class. Given a set of training examples, principal components analysis can be used to determine the directions of maximum variance in the data. Volcanoes are approximately modeled as linear combinations of six basis functions (principal components) with a particular probability distribution over the weighting coefficients (features). Based on the projections of a candidate chip onto the basis functions, a chip can be classified as *volcano* or *not-volcano*. We experimented with a number of classifiers including quadratic, nearest neighbors, neural networks, decision trees, and kernel density estimators; all yielded similar performance so we have used the quadratic classifier as the default.

JARtool was evaluated on two homogeneous sets of images from the same area of the planet (`OLD4` and `HOM38`). The performance on these images was good but somewhat below the level of human experts. The principal components approach

[3]The scientists estimate that it would take 10 man-years to catalog all the small volcanoes in the first pass Magellan data

provided a significant improvement over matched filtering, but this should not be surprising since PCA provides a better model for how members of the volcano class vary.

So is PCA the solution for all pattern recognition problems? No. The volcano problem is special in some ways. Since the Magellan imaging was done with synthetic aperture radar (SAR), the source of illumination and the receiver are co-located; hence, there is no need to worry about illumination invariance. In addition, the underlying physical objects are (to first order) rotationally symmetric, so there is no need to worry about rotational invariance. Approximately 80% of the volcanoes have resolvable summit pits which appear near the center as a bright spot or backwards "C". The pits allow the volcanoes to be reliably centered, reducing the amount of translation invariance required. Although the volcanoes do vary considerably in scale, near the center there is usually a visible summit pit and a transition from bright shading on the side sloped toward the radar to dark shading on the side sloped away from the radar. Good performance can be obtained by simply focusing on this central area and ignoring the outer edges of the volcano. Thus, the (homogeneous) volcano problem is especially suitable for principal components analysis since (1) there is a limited amount of variability within the class and (2) the defining information is well-localized.

For the image set `HET36`, however, the performance of PCA is significantly degraded. We believe the increased variability of the volcano class (and the background) accounts for the difference. With heterogeneous images, the volcano class can no longer be adequately represented as a linear combination of a small number of basis functions. Also, for problems in which the defining information is spatially distributed, we anticipate that principal components analysis will not perform adequately. This hypothesis is explored in the next chapter and the remainder of the thesis.

Chapter 5 Deformable Spatial Configurations

5.1 Introduction

In this chapter, we consider object classes in which instances can be modeled as a set of characteristic parts in a deformable spatial configuration. As an example, consider human faces, which consist of two eyes, a nose, and mouth. These parts appear in an arrangement that depends on the individual, his expression, and the viewpoint of the observer.

As shown in Figure 5.1, deformable object classes arise in a number of different ways. An object class may be generated by a single underlying physical object that is deformable. Images from one person with different facial expressions fall in this category. A deformable object class may also be generated by a single physical object that is rigid, but imaged over a range of viewpoints. As shown by the two penguin images in Figure 5.1b, the relative positions of the object parts on the image plane vary causing the penguin to appear to be deformable. Object classes consisting of a number of different physical instances of the same type of object such as different human faces or automobiles can be modeled as deformable configurations of characteristic parts. A final example is handwriting, which consists of different realizations of a single conceptual object. Each time a person writes a word, the same basic strokes and parts are present but the relative positions vary depending on the writer's haste, writing geometry, etc.

Just as we cannot precisely define what constitutes an object class, we cannot define what constitutes an object "part." Generally speaking, a part is any piece of the object that can be reliably located using local information. The part may be defined through a variety of visual cues such as a distinctive brightness or orientation

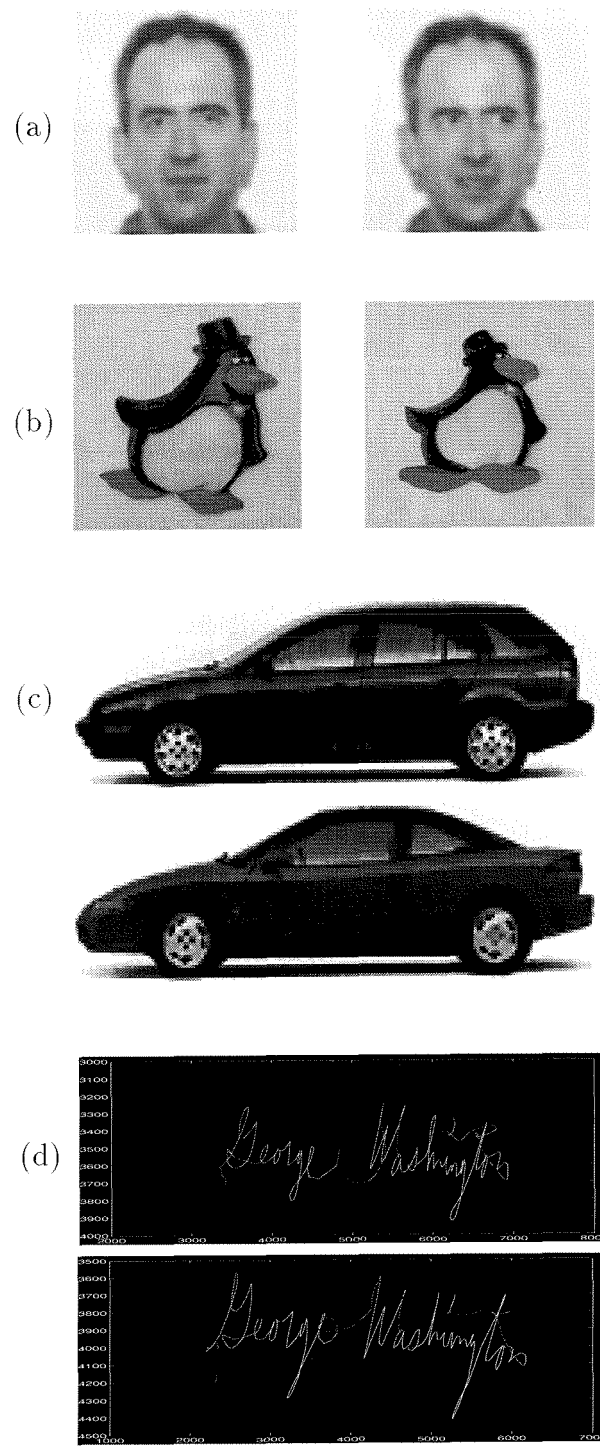


Figure 5.1: Examples of deformable object classes.

pattern, texture, color, motion, or symmetry. Although we have not experimented with audio cues, these could in principle also be "parts." The main requirement is that a part can be detected and localized with some reasonable degree of reliability. Parts may also be defined at multiple scales. A coarse resolution head is as much a "part" as a fine resolution view of an eye corner. An object can also contain a number of parts that are locally the same. For example, in handwriting a word may contain the same letter or strokes in several positions. This is acceptable provided the parts are not so repetitious that they become a texture rather than a pattern.

In the next section we introduce a simple object T_0 consisting of four parts in a specific spatial arrangement. This object can be used to define a deformable object class T_p by allowing each part to be spatially perturbed from its nominal position. We will then show that the local methods discussed earlier (matched filtering and principal components) break down on this problem. In subsequent chapters, we will present a new method for recognizing this type of object class based on a combination of local part detectors and spatial configuration.

5.2 Simplified Model

Consider a 2-D object consisting of N image-based parts P_i, each with a nominal spatial position (x_i^0, y_i^0). The nominal object T_0 is given by

$$T_0(x, y) = \sum_{i=1}^{N} \alpha_i P_i(x - x_i^0,\ y - y_i^0) \tag{5.1}$$

where the α_i's are scalar weighting coefficients that control the signal-to-noise ratio of the respective parts. Although each of the parts could be different from the others and could also have inherent variability, we will consider a simpler case in which each

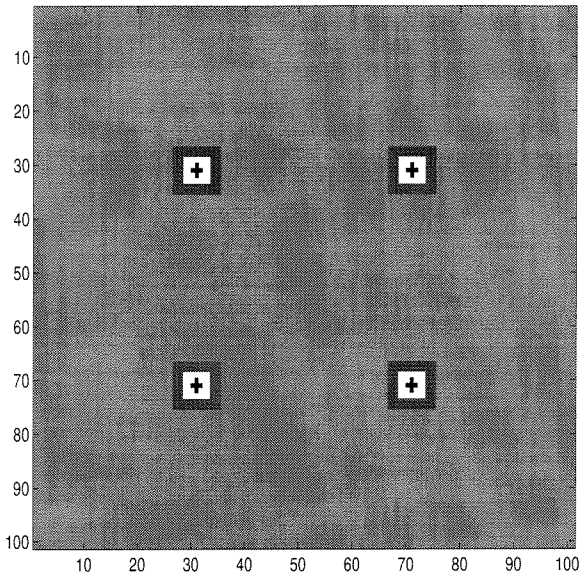


Figure 5.2: The nominal object T_0 consists of four parts arranged at the vertices of a square.

of the P_i's takes the following form:

$$P_i = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 2 & 2 & 2 & 2 & 2 & -1 & -1 \\ -1 & -1 & 2 & 2 & 2 & 2 & 2 & -1 & -1 \\ -1 & -1 & 2 & 2 & 2 & 2 & 2 & -1 & -1 \\ -1 & -1 & 2 & 2 & 2 & 2 & 2 & -1 & -1 \\ -1 & -1 & 2 & 2 & 2 & 2 & 2 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \tag{5.2}$$

We will also restrict our attention to a specific arrangement of parts as shown in Figure 5.2. The part centers are located at the following positions: $(31, 31)$, $(31, 71)$, $(71, 31)$, and $(71, 71)$. The scale factors α_i are the same for each of the parts.

As it stands, the object is not very interesting; however, we will form an object class by allowing the parts to be perturbed from their nominal positions. In particular,

the part positions $\mathbf{X}$ will be modeled with the following probability density:

$$p_{\mathbf{X}}(\mathbf{X}) \;=\; \mathcal{N}_{2N}(\mathbf{X};\, \boldsymbol{\mu}, \rho^2\mathbf{I}) \tag{5.3}$$

where $\boldsymbol{\mu}$ is the vector of nominal positions

$$\boldsymbol{\mu} \;=\; \left[\; x_1^0 \;\; x_2^0 \;\; \ldots \;\; x_N^0 \;\; y_1^0 \;\; y_2^0 \;\; \ldots \;\; y_N^0 \;\right]^T \tag{5.4}$$

Each of the object parts can be independently displaced in x and y from the nominal by a Gaussian perturbation having standard deviation ρ. We will designate the resulting object class as T_ρ.

To generate an object from this class, we first generate a random vector $\mathbf{X}$ according to the density on the right hand side of Equation 5.3. Since this vector determines the part positions, we then place the pattern P_i from Equation 5.2 at each of these positions. Several instances from the class T_3 are shown in Figure 5.3. The plus signs indicate the positions of the parts in the nominal (unperturbed) object.

5.3 Breakdown of Local Methods

We now examine how well the methods discussed earlier for recognizing localized patterns (matched filtering and principal components) work for the problem of detecting instances from the object class T_ρ. The position of the nominal object T_0 in the image will always be the same, so there is no concern for translation, rotation, or scale invariance.

5.3.1 Matched Filtering

From Chapter 2, we know that the optimal detector for the nominal object T_0 in white noise is the matched filter. However, for detecting *the object class* T_ρ, the matched filter will not be optimal. Intuitively, we expect that as the deformability of objects in the class increases (by increasing ρ), the performance of the matched filter will quickly

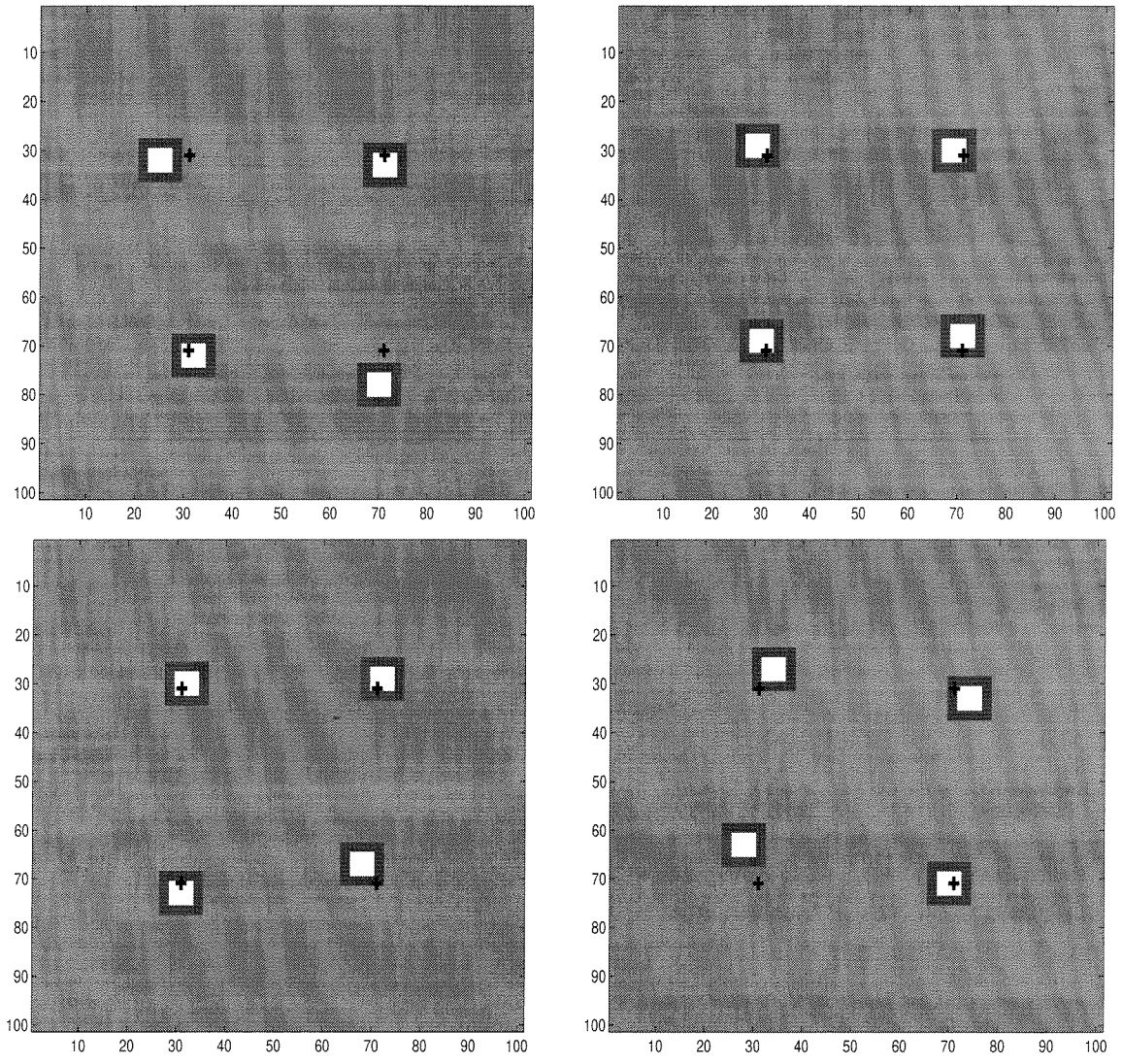


Figure 5.3: Four instances from the deformable object class T_3. The class was generated by perturbing the part positions of the nominal object T_0 shown in Figure 5.2.

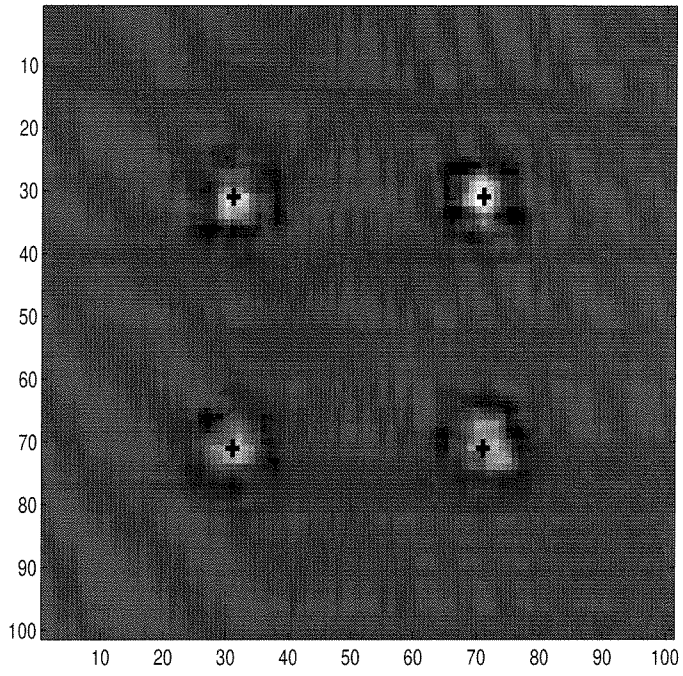


Figure 5.4: Matched filter $\mathbf{m}_2$ computed by averaging training examples from T_3.

degrade.

There are two basic ways we might construct a matched filter for this problem. The first method simply uses the nominal unperturbed object T_0 as the filter. The second method uses an average filter constructed from a large number of training instances from the object class. This method is based on the assumption that the training instances are a single signal with white noise added. Thus, averaging should produce a filter that is closer to the true signal. For this object class, however, the training instances are not well-modeled as a single signal plus white noise. Hence, averaging produces a matched filter that is a blurred version of the nominal object (caused by averaging displaced versions with the nominal). Figure 5.4 shows this filter, as computed from 200 training examples from T_3.

We empirically evaluated the performance of the two matched filters $\mathbf{m}_1$ and $\mathbf{m}_2$ using the following paradigm. One set of 200 noise examples was used to test the resilience of the filters to false alarms. A second set of 200 signal-plus-noise examples was generated to evaluate the detection performance. The signal-plus-noise examples

were produced by generating a set of "clean" examples from T_ρ. These examples were then (amplitude) scaled by the appropriate α and combined with noise examples to provide signal-plus-noise examples at the desired SNR.

$$\alpha = 10^{\left(\mathrm{SNR}_{\mathrm{desired}} - \mathrm{SNR}_0\right)/20.0} \tag{5.5}$$

where the SNR values are in decibel (dB) units and SNR_0 is given by

$$\mathrm{SNR}_0 \;=\; 10 \cdot \log_{10}\left(\frac{NE_0}{\sigma^2}\right)$$

The background noise variance per pixel is denoted by σ^2. The quantity E_0 is the energy in a single object part. (Note that we have implicitly assumed that parts do not overlap.) Figure 5.5 shows the nominal object embedded in noise for several different SNR values.

Figure 5.6 shows the ROC performance of the average matched filter $\mathbf{m}_2$ as a function of the spatial perturbation, for $\rho = 0, 1, \ldots, 5$. We also evaluated the nominal matched filter $\mathbf{m}_1$, but the performance was significantly worse than for the average filter. In both cases the SNR was set to 18dB and the matched filter was applied only at the center of the image (i.e., without convolution). The degree of degradation with ρ depends on the decorrelation length of the component parts. Thus, an object composed of different parts may have greater or lesser degradation than shown in the figure.

It is interesting to note that we can write an expression for the theoretical ROC performance of the (non-sliding) nominal filter as a function of ρ. If part i is displaced by $\Delta_i = (\delta x_i, \delta y_i)$, then the output from the matched filter will have expected value given by:

$$E(\boldsymbol{\Delta}) = \sum_{i=1}^{N} R_i(\delta x_i, \delta y_i) \tag{5.6}$$

where $R_i(x, y)$ is the autocorrelation function for part i and $\boldsymbol{\Delta}$ is the vector of dis-

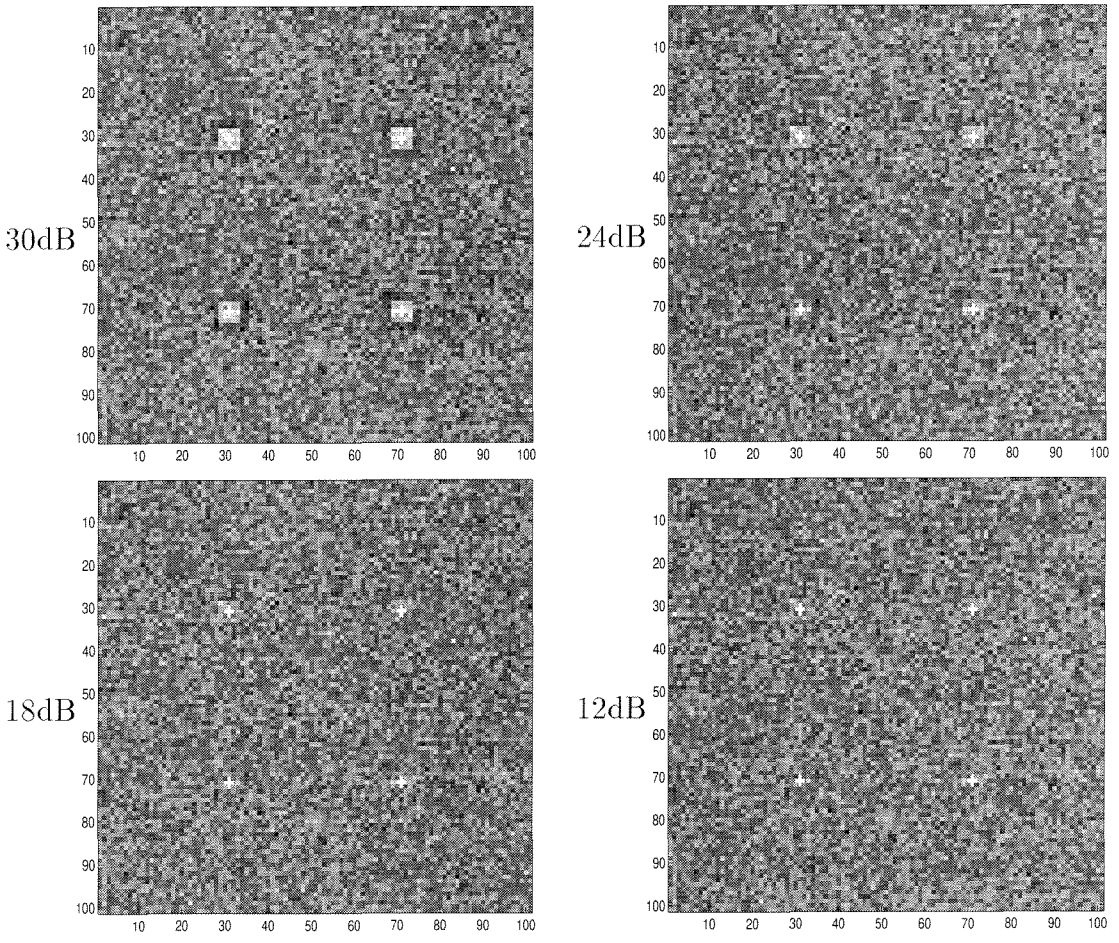


Figure 5.5: Nominal object embedded in white noise at several SNR settings.

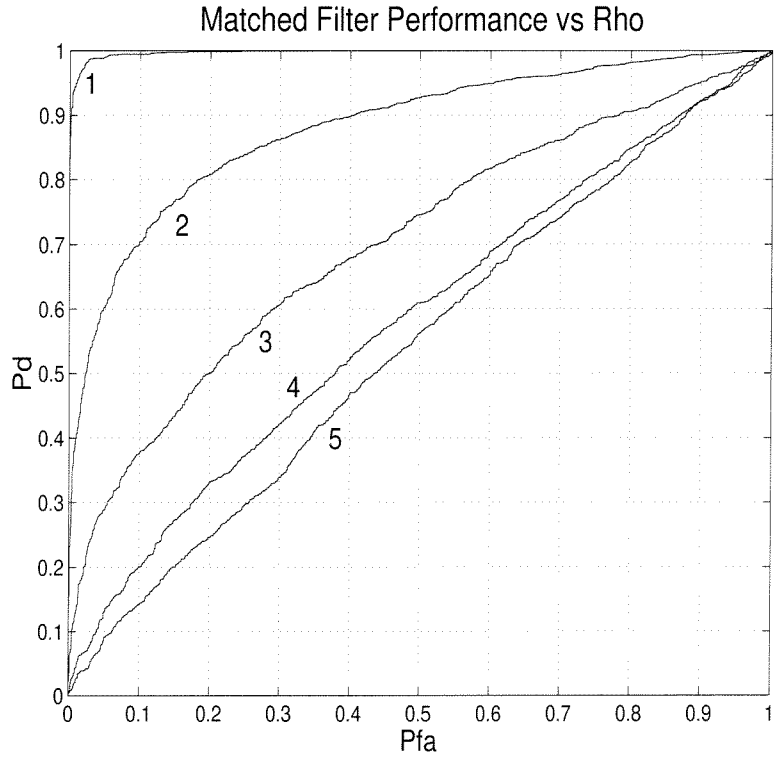


Figure 5.6: Empirical performance of the average matched filter as a function of the spatial perturbation at SNR = 18dB. The performance on the unperturbed object was perfect for the sample size tested (2000 samples from each class).

placements

$$\Delta = [\delta x_1, \delta y_1, \ldots, \delta x_N, \delta y_N]^T \tag{5.7}$$

The false alarm performance of the matched filter will remain unchanged. However, the detection performance will be degraded due to the loss in effective signal energy. The expected ROC performance will be:

$$p_{fa} = 1 - \Phi(K) \tag{5.8}$$

$$p_d = 1 - \int_{\mathbf{\Delta}} \Phi\left(K - \frac{E(\mathbf{\Delta})}{\sigma^2}\right) \cdot p(\mathbf{\Delta})\, d\mathbf{\Delta} \tag{5.9}$$

where $p(\mathbf{\Delta})$ is the probability distribution over the vector displacement. If the auto-correlation functions R_i fall off slowly relative to the fall off of $p(\mathbf{\Delta})$, then we see that the performance will not be degraded significantly.

5.3.2 Principal Components

For T_ρ, the matched filter performance degrades quickly with ρ. This result is not surprising since the matched filter attempts to represent the entire object class with a single exemplar. The principal components approach improves upon the matched filter by modeling the variability in the object class with a linear set of basis functions.

The principal components approach has a number of parameters that must be estimated from training data. We have used the same approach that was used in Chapter 4. for locating volcanoes. That is, basis functions were estimated from only the positive training examples. Both positive and negative training examples were then projected onto the basis functions. The class-conditional densities in projection space were modeled with unimodal Gaussian densities.

The basis functions were estimated from 200 "clean" object examples from class T_3. Based on the singular value decay shown in Figure 5.7a and the appearance of the basis functions in Figure 5.7b, we estimated that $L = 20$ basis functions would provide a reasonable approximation to the set of training examples. In Figure 5.7b,

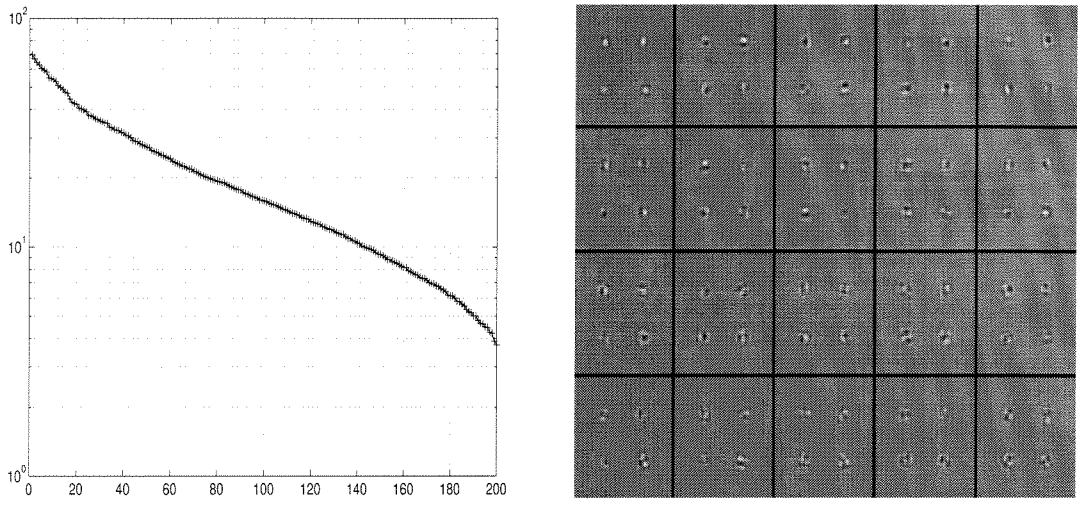


Figure 5.7: (a) Singular value decay. (b) The first 20 basis functions ordered from left to right by singular value. The top left chip is the direction of maximum variation in the training set.

the basis functions are ordered from left to right and top to bottom by singular value. Thus, the top left chip is the direction of maximum variance in the training set.

The basis functions in Figure 5.7b span a 20-dimensional subspace in the space of all 101×101 pixel patterns. This subspace is the best 20-dimensional approximation (in the RMS sense) to the set of training examples. We will refer to this as "SVD space". Positive and negative training examples were projected into SVD space and modeled with Gaussian densities. Unknown test examples were then classified by projecting into SVD space and estimating the posterior probabilities.

Empirical performance of the PCA approach is shown in Figure 5.8 for object class T_3. For comparison, the performance of the matched filter and the optimal detector for this problem (which will be discussed further in Chapter 9) is also shown. The PCA performance is considerably better than the matched filter, but is degraded significantly with respect to the optimal detector. As more degrees of freedom are added to the object class, e.g., translation, rotation, scaling, and variability in the part appearances, the ability of a small set of linear basis functions to encode the variability will break down completely.

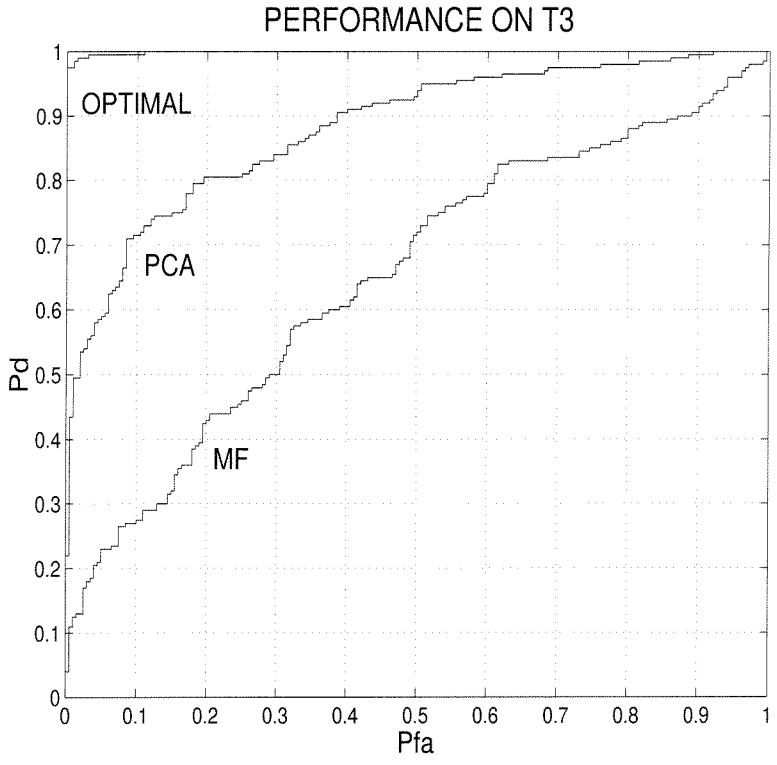


Figure 5.8: The performance of the principal components approach on the deformable object class with $\rho = 3$. For comparison, performance of the optimal detector and the matched filter are also shown. Although principal components provides significant improvement over the matched filter, the performance is significantly degraded from the optimal.

5.4 Summary

Object classes that consist of characteristic parts arranged in a deformable spatial configuration arise in a variety of ways. There may be a single underlying physical object that is deformable as in the case of an individual person making facial expressions. A single underlying physical object that is rigid but viewed from different directions will also appear on the image plane to be deformable. Collections of similar objects of the same type (different human faces or automobiles) can also be treated as deformed instances of a single object.

In the first half of the thesis, two methods were discussed for recognizing localized patterns: matched filtering and principal components analysis. These methods, however, break down for this new class of objects. Since the object parts do not always appear in the same relative positions, the matched filter will not line-up properly with a particular instance of the object. Experimentally, we showed that this leads to degraded performance. The singular value decomposition is somewhat better suited for this type of problem, but is still limited by the ability of linear combinations of basis functions to encode all the inherent variability in an object class. Specifically, variability in the appearance of the individual parts and their global arrangement, as well as variability due to translation, rotation, and scaling, must be encoded using linear combinations of basis functions.

An idea that we will pursue in the next three chapters is to use local methods such as matched filtering or principal components to detect individual parts of an object. As we have seen in Chapter 4, these techniques are not perfect even for localized patterns: some true parts may be missed (this could also happen because of occlusion) and a number of false alarms will occur. Thus, the local detector outputs serve only as candidate locations for the object parts. We will explore methods that group these candidate part locations into object hypotheses. The hypotheses are then evaluated based on the spatial arrangement of the parts. Invariance to translation, rotation, and scale is obtained by representing configurations with "shape" variables. Variability in the spatial configuration of the parts can then be modeled using probability densities

over shape.

Chapter 6 Shape Statistics

6.1 Introduction

In our approach to recognizing deformable object classes, the allowed deformations are represented through shape statistics, which are learned from examples. The word "shape" is used here in the sense of Kendall [Ken84, Ken89] and Bookstein [Boo84, Boo86]. That is, "shape" refers to properties of a set of labeled points that are invariant with respect to some group of transformations. In our case, the labeled points are the locations of object parts on the image plane, and the transformations are translation, rotation *in the image plane*, and scaling. Instances of an object in an image are detected by finding the appropriate object parts or features in the "correct" spatial configuration, where "correct" is intended in a probabilistic sense. We have also begun to investigate a more general set of transformations known as affine transformations [BWLP96, Leu95].

In this chapter, we briefly cover some of the key results from the statistical theory of shape. The main result we use is due to Dryden and Mardia [DM91] who derived the shape space density induced by a general Gaussian figure space density. New results that we have derived are presented in Section 6.4 including a theorem on shape space mixture densities. In the next chapter, we discuss how shape densities can be used to evaluate the correctness of a spatial configuration of points (and of partial configurations). Combined with a hypothesis generation procedure, this yields a recognition strategy for visual object classes. Applications to face localization and cursive handwriting are presented in Chapter 8.

6.2 Definition of Shape

A spatial arrangement of N labeled points in a plane can be described by a $2N$-dimensional vector containing the x and y coordinates of each point. This representation, however, is not convenient to use for recognition since the important information is obscured by differences due to translation, rotation, and scaling (TRS). What we really want is a representation in which the *shape* of a configuration is separated from the effects of TRS.

Let the (raw) figure space representation be given by the vector of $2N$ image plane coordinates:

$$\mathbf{X} = [x_1, \ldots, x_N, y_1, \ldots, y_N]^T \tag{6.1}$$

TRS can be eliminated by mapping two points to fixed reference positions; the positions of the other points will then represent the shape. This process is illustrated in Figure 6.1. First translation is eliminated by mapping point 1 to the origin. Then rotation and scaling are eliminated by mapping point 2 to $(1, 0)$. The $(2N-4)$-dimensional vector

$$\mathbf{U} = [u_3, \ldots, u_N, v_3, \ldots, v_N]^T \tag{6.2}$$

represents the shape of the configuration. Essentially, one dimension is dropped for factoring out scale, one for rotation, and two for translation.

The transformation of the first figure point (x_1, y_1) to the origin can be accomplished by premultiplying $\mathbf{X}$ by the $2N \times 2N$ matrix $\mathbf{L}^T$ defined by:

$$\mathbf{L}^T = \begin{bmatrix} \mathbf{I} - \underline{\mathbf{1}}\mathbf{e}_1^T & \underline{\mathbf{0}} \\ \underline{\mathbf{0}} & \mathbf{I} - \underline{\mathbf{1}}\mathbf{e}_1^T \end{bmatrix} \tag{6.3}$$

where $\underline{\mathbf{1}}$ is an $N \times 1$ vector of all ones and $\mathbf{e}_1$ is the $N \times 1$ vector $[1, 0, \ldots, 0]^T$. $\mathbf{I}$ and $\underline{\mathbf{0}}$ are the $N \times N$ identity and zero matrices, respectively. Following this transformation,

we have

$$\mathbf{X}^* = \mathbf{L}^T\mathbf{X} = [0, x_2^*, \ldots, x_N^*, 0, y_2^*, \ldots, y_N^*]^T = \mathbf{L}^T\mathbf{X} \tag{6.4}$$

where $x_i^* = x_i - x_1$ and $y_i^* = y_i - y_1$. Omitting the fixed (zero) values from $\mathbf{X}^*$ yields a reduced $(2N\text{–}2)$-dimensional vector $\mathbf{Y}$,

$$\mathbf{Y} = [x_2^*, \ldots, x_N^*, y_2^*, \ldots, y_N^*]^T \tag{6.5}$$

The transformation from $\mathbf{X}^*$ to $\mathbf{Y}$ can be written as a linear transformation: $\mathbf{Y} = \mathbf{H}^T\mathbf{X}^*$, where $\mathbf{H}$ is an $(N\text{–}2 \times N)$ matrix. Thus, the transformation from $\mathbf{X}$ to $\mathbf{Y}$ is also a linear transformation:

$$\begin{aligned} \mathbf{Y} &= \mathbf{H}^T\mathbf{L}^T\mathbf{X} \\ &= \mathbf{L}_R^T\mathbf{X} \end{aligned} \tag{6.6}$$

where $\mathbf{L}_R^T \triangleq \mathbf{H}^T\mathbf{L}^T$.

Elimination of the effects of scaling and rotation can be achieved by now mapping the points such that $(x_2^*, y_2^*) \rightarrow (1, 0)$. This process yields the shape vector $\mathbf{U}$ (Equation 6.2), where (for $i = 3, \ldots, N$)

$$\begin{aligned} u_i &= (x_i^* x_2^* + y_i^* y_2^*) \,/\, \left(x_2^{*2} + y_2^{*2}\right) \\ v_i &= (y_i^* x_2^* - x_i^* y_2^*) \,/\, \left(x_2^{*2} + y_2^{*2}\right) \end{aligned} \tag{6.7}$$

Note that it is also possible to eliminate TRS by mapping the centroid of the configuration to the origin and then using moments of inertia to eliminate rotation and scaling. The problem with this method, however, is that it will not work if some of the points are missing. Since the local detectors that identify parts of the object may miss some true parts, e.g., due to occlusion, we prefer to do the normalization based on two reference points instead of the centroid and second moments. With our method, if one or both of the reference points happen to be missed, a different

reference basis can be used without difficulty.

6.3 Dryden-Mardia Shape Density

Now suppose that the original configuration $\mathbf{X}$ can be modeled as a random vector from some $2N$-dimensional probability distribution. What density is induced upon the shape vector $\mathbf{U}$? Dryden and Mardia have solved this problem in closed-form [DM91] for the case when $\mathbf{X}$ follows a general $2N$-dimensional Gaussian distribution:

$$\mathbf{X} \sim \mathcal{N}_{2N}(\boldsymbol{\nu}, \boldsymbol{\Omega}) \tag{6.8}$$

Since $\mathbf{Y}$ is a linear combination of the values in $\mathbf{X}$, it also follows a multivariate Gaussian distribution, in particular,

$$\mathbf{Y} \sim \mathcal{N}_{2N-2}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{6.9}$$

where $\boldsymbol{\mu} = \mathbf{L}_R^T \boldsymbol{\nu}$ and $\boldsymbol{\Sigma} = \mathbf{L}_R^T \boldsymbol{\Omega} \mathbf{L}_R$. The shape vector $\mathbf{U}$ is related to $\mathbf{Y}$ by a nonlinear transformation so we cannot expect $\mathbf{U}$ to follow a Gaussian distribution. The actual joint probability density function (pdf) of $\mathbf{U}$ is given by the following theorem due to Dryden and Mardia:

Theorem 1 (Dryden-Mardia Shape Density [DM91]) *Under the multivariate Gaussian model for the figure-space coordinates (Equation 6.8), the joint probability density function of the shape vector* $\mathbf{U}$ *is:*

$$p_{\mathbf{U}}(\mathbf{U}) = \frac{q \cdot \exp(-g/2)}{(2\pi)^{N-2}} \cdot \sqrt{\frac{|\boldsymbol{\Psi}|}{|\boldsymbol{\Sigma}|}} \cdot (N-2)! (2\sigma_2^2)^{N-2} \tag{6.10}$$

where

$$q = \sum_{i=0}^{N-2} \frac{\sigma_1^{2i}}{\sigma_2^{2i}} \mathcal{L}_i^{\left(-\frac{1}{2}\right)}\{-r_1^2\} \mathcal{L}_{N-2-i}^{\left(-\frac{1}{2}\right)}\{-r_2^2\} \tag{6.11}$$

$$g = \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} - \boldsymbol{\xi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\xi} \tag{6.12}$$

$$\boldsymbol{\Psi}^{-1} = [\mathbf{u} \vdots \mathbf{v}]^T \boldsymbol{\Sigma}^{-1} [\mathbf{u} \vdots \mathbf{v}] \tag{6.13}$$

$$\boldsymbol{\xi} = \boldsymbol{\Psi} [\mathbf{u} \vdots \mathbf{v}]^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \tag{6.14}$$

$$\mathbf{u} = [1, u_3, \ldots, u_N, 0, v_3, \ldots, v_N]^T \tag{6.15}$$

$$\mathbf{v} = [0, -v_3, \ldots, -v_N, 1, u_3, \ldots, u_N]^T \tag{6.16}$$

$$r_k^2 = \left(\boldsymbol{\phi}_k^T \boldsymbol{\xi}\right)^2 / \left(2\sigma_k^2\right) \qquad \textit{for } k = 1, 2 \tag{6.17}$$

and $\sigma_1^2 \geq \sigma_2^2$ *are the eigenvalues of* $\boldsymbol{\Psi}$ *with corresponding eigenvectors* $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2$. *The function* $\mathcal{L}_i^{(a)}(x)$ *is the generalized Laguerre polynomial of degree* i*:*

$$\mathcal{L}_i^{(a)}(x) = \sum_{k=0}^{i} (1+a)_i (-x)^k / \left\{(1+a)_k k! (i-k)!\right\} \tag{6.18}$$

where $(1+a)_0 \triangleq 1$ *and* $(1+a)_k = (a+k) \cdot (1+a)_{k-1}$.

Although the Dryden-Mardia density appears complicated, the derivation is relatively straightforward. The basic idea is to introduce a $(2N\text{–}2) \times 1$ vector $\mathbf{W}$ that contains the $(2N\text{–}4)$ shape variables $\mathbf{U}$ *and* x_2^* and y_2^*. The transformation from $\mathbf{Y}$ to $\mathbf{W}$ is one-to-one so the density of $\mathbf{W}$ can easily be related to the density of $\mathbf{Y}$. It is then possible to integrate out the dependence on (x_2^*, y_2^*) from the density of $\mathbf{W}$ leaving the density of $\mathbf{U}$.

The following argument regarding the shape density may initially appear plausible, but in fact it is false. Translation, rotation, and scaling of a configuration of points can be expressed as:

$$\mathbf{U} = \mathbf{M}\mathbf{X} + \mathbf{b} \tag{6.19}$$

where the entries of $\mathbf{M}$ and $\mathbf{b}$ depend on the transformation parameters τ_x, τ_y, θ, and σ (true). Since $\mathbf{X}$ is jointly Gaussian and $\mathbf{U}$ is a linear transformation of $\mathbf{X}$, $\mathbf{U}$ should also be jointly Gaussian (wrong!). This would be true if the transformation parameters were fixed numbers, but here they are actually realizations of random variables. Thus, $\mathbf{U}$ is only *conditionally* Gaussian *given* the transformation parameters. To get

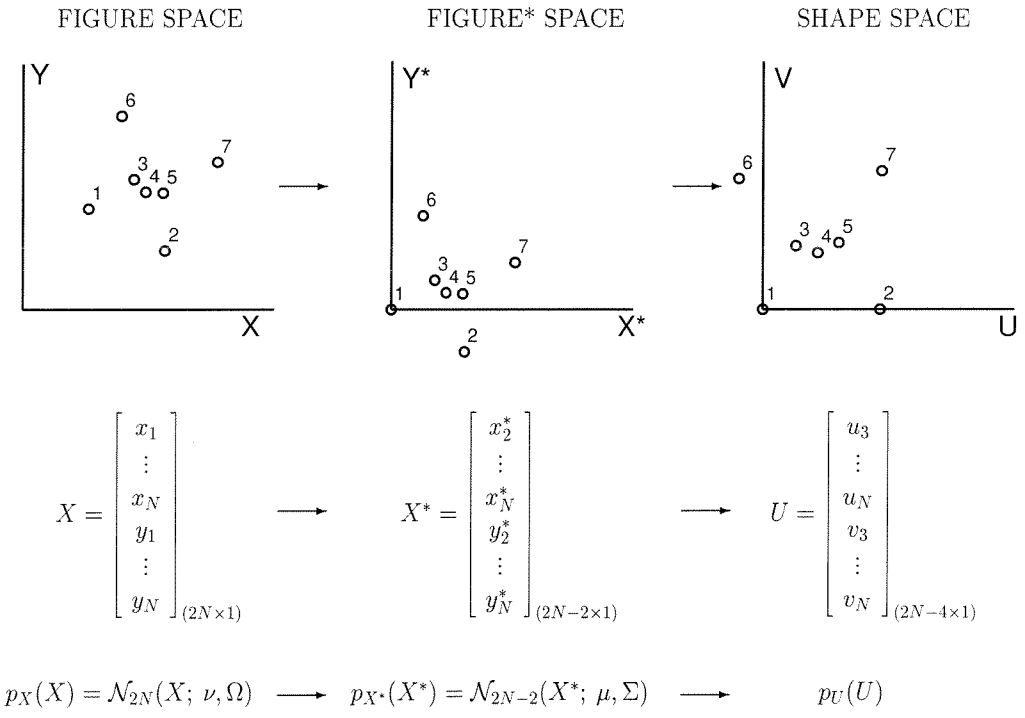


Figure 6.1: A $2N$-dimensional jointly Gaussian density in figure space induces a $(2N-2)$-dimensional Gaussian density in figure$*$ space and a new density $p_{\mathbf{U}}(\mathbf{U})$ in shape space.

the unconditional density, one would have to multiply the conditional density by the density of the transformation parameters and integrate.

As an illustration of the Dryden-Mardia result, we have generated a number of Gaussian random triangles in figure space with the mean triangle and covariance as shown in Figure 6.2a. (Each vertex varies independently of the others; the dashed ellipses show the equiprobability contours for the marginal density of each vertex.) Figure 6.2b shows several random triangles generated according to this distribution. The transformation to shape space maps vertex 1 of each triangle to the origin and vertex 2 to the point $(1, 0)$. The coordinates (u_3, v_3) of the third vertex after this transformation will represent the shape of the triangle. Figure 6.2c shows the shape variables for $5,000$ random triangles. Figure 6.2d shows a surface plot of the theoretically predicted shape space density (from Theorem 1). Figure 6.2e shows the theoretical equiprobability contours in shape space. Observe that the density is clearly non-Gaussian. Finally, Figure 6.2f shows the equiprobability contours of the empirical shape space density estimated from 50,000 random triangles. This result agrees closely with the theoretical density.

6.4 Properties

Since the Dryden-Mardia density is derived from a Gaussian, it has several properties that make it especially convenient to work with. One of these is that it is easy to determine the density of shape variables computed with respect to a different baseline pair. Also, it is easy to compute joint densities over subsets of shape variables.

6.4.1 Different Baseline Pair

It is straightforward to compute the Dryden-Mardia density with respect to a different baseline pair. In Theorem 1 we have written down the density of shape variables computed with respect to point 1 and point 2. That is, point 1 and point 2 were mapped to $(0, 0)$ and $(1, 0)$, respectively, and the coordinates of the remaining points

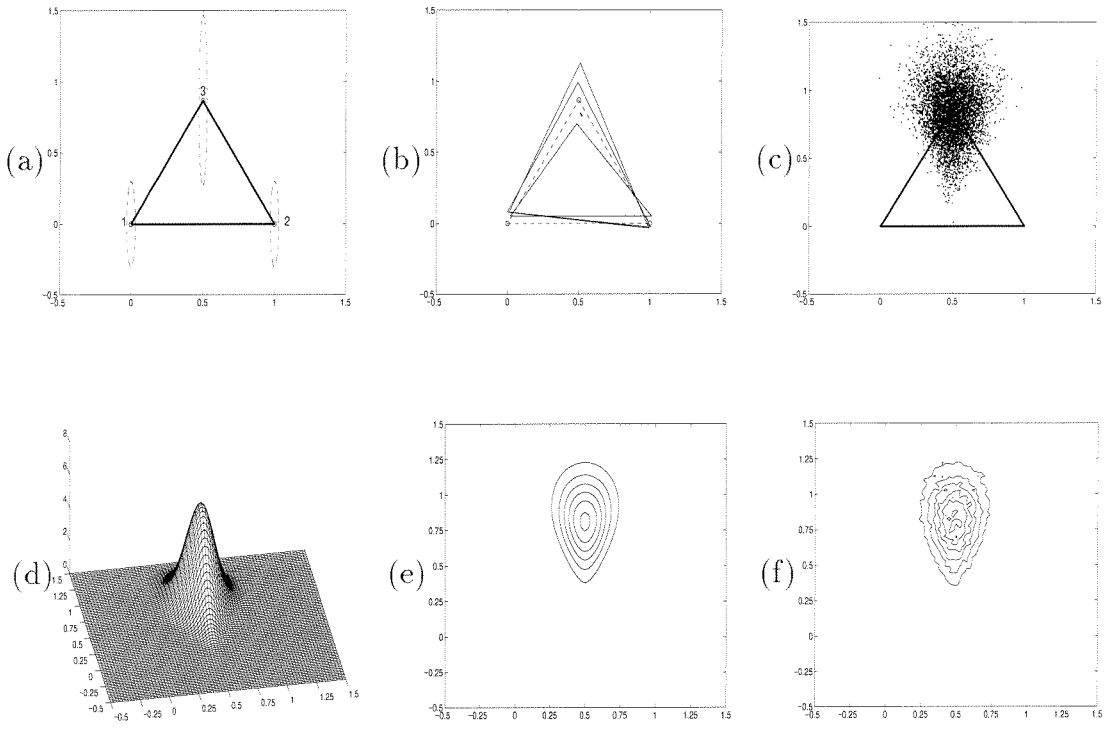


Figure 6.2: (a) Mean triangle in figure space. Dashed lines show the marginal covariance structure. (b) Random triangles in figure space. (c) 5,000 random triangles mapped to shape space. (For clarity, the edges have been omitted.) Notice that the empirical shape-space density is non-Gaussian. (d) The theoretical Dryden-Mardia shape-space density for this problem. (e) Equiprobability contours of the theoretical density. (f) Equiprobability contours of the empirical density as estimated from 50,000 samples.

constituted the shape variables appearing in $\mathbf{U}$. What happens if a different baseline pair, say point 3 and point 4, is used?

The basic idea is to permute the labels of the figure space features. This can be accomplished by multiplying the original figure space vector $\mathbf{X}$ by a $2N \times 2N$ permutation matrix $\mathbf{P}$. For six points, the new figure space vector $\mathbf{X}$ will look as follows:

$$\mathbf{X}_{\text{new}} = [x_3, x_4, x_1, x_2, x_5, x_6, y_3, y_4, y_1, y_2, y_5, y_6]^T \tag{6.20}$$

The mean vector ν and covariance matrix Ω of the original figure space variables must also be permuted appropriately, i.e.,

$$\boldsymbol{\nu}_{\mathbf{P}} = \mathbf{P}\boldsymbol{\nu} \tag{6.21}$$

$$\boldsymbol{\Omega}_{\mathbf{P}} = \mathbf{P}\boldsymbol{\Omega}\mathbf{P}^T \tag{6.22}$$

The density of the new shape vector $\mathbf{U}_{\text{new}}$ is given by Theorem 1 with the parameters $(\boldsymbol{\nu}, \boldsymbol{\Omega})$ replaced by $(\boldsymbol{\nu}_{\mathbf{P}}, \boldsymbol{\Omega}_{\mathbf{P}})$.

6.4.2 Density over Subsets of Shape Variables

Another useful property of the Dryden-Mardia density is that the joint density over subsets of shape variables can easily be computed. Suppose that we have six figure space points. After transformation to shape space, there will be four points whose coordinates constitute the shape variables: $u_3, u_4, u_5, u_6, v_3, v_4, v_5, v_6$. The density over these variables is given by Theorem 1, but what if we only want the joint density over the coordinates of three of the points? What is the density $p(u_3, u_4, u_5, v_3, v_4, v_5)$? As we will see later, it is important to be able to compute such partial densities because the feature detectors used to locate parts of an object are not perfectly reliable. The detectors may miss some of the true object parts; however, there is still a need to test whether the partial configuration of detected parts is consistent with a given object class.

The approach used to compute partial densities is similar to the approach used to compute the density with respect to a different baseline pair. We define a new figure space vector:

$$\mathbf{X}_{\text{new}} = [x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5]^T \tag{6.23}$$

which is related to the original figure space vector by a linear transformation:

$$\mathbf{X}_{\text{new}} = \mathbf{R}\mathbf{X} \tag{6.24}$$

where $\mathbf{R}$ is a 10×12 matrix of zeros and ones (notice the omission of x_6 and y_6 from $\mathbf{X}_{\text{new}}$). The density of $\mathbf{X}_{\text{new}}$ will be jointly Gaussian with parameters given by:

$$\boldsymbol{\nu}_{\mathbf{R}} = \mathbf{R}\boldsymbol{\nu} \tag{6.25}$$

$$\boldsymbol{\Omega}_{\mathbf{R}} = \mathbf{R}\boldsymbol{\Omega}\mathbf{R}^T \tag{6.26}$$

Transforming $\mathbf{X}_{\text{new}}$ to shape space generates the shape variables $u_3, u_4, u_5, v_3, v_4, v_5$. The density over these variables is now given by Theorem 1 with the parameters $(\boldsymbol{\nu}_{\mathbf{R}}, \boldsymbol{\Omega}_{\mathbf{R}})$.

6.4.3 Mixture Models

The shape density derived by Dryden and Mardia is predicated on using a single multivariate Gaussian for the figure space distribution. This Gaussian model is useful when the figure space configurations consist of perturbations around some "average figure." For more complicated figure space distributions, however, the single Gaussian model may be inadequate. In principle this limitation may be overcome by using Gaussian mixture models. We have derived the following result which shows that the induced shape space density is a mixture of the shape densities resulting from the individual Gaussian modes. For the experiments reported later, we have not found it necessary to use mixture densities; however, we believe mixture densities may be

useful for other datasets.

Theorem 2 (Mixture of Shape Densities) *Under a multivariate Gaussian* mixture model *for the figure-space coordinates,*

$$p_{\mathbf{X}}(\mathbf{X}) = \sum_{j=1}^{J} \alpha_j \mathcal{N}_{2N}(\mathbf{X};\ \boldsymbol{\nu}_j, \boldsymbol{\Omega}_j) \tag{6.27}$$

the joint probability density function of the shape vector $\mathbf{U}$ *is a mixture of shape densities corresponding to the modes of the Gaussian mixture density. That is,*

$$p_{\mathbf{U}}(\mathbf{U}) = \sum_{j=1}^{J} \alpha_j p_{\mathbf{U}}(\mathbf{U};\ \boldsymbol{\nu}_j, \boldsymbol{\Omega}_j) \tag{6.28}$$

Proof of this result is straightforward. Conditioned on the mode j of the Gaussian mixture, the shape space density is a Dryden-Mardia density with parameters $\boldsymbol{\nu}_j$ and $\boldsymbol{\Omega}_j$. The unconditional shape density is then given by multiplying by the mode probabilities α_j and summing over j.

6.4.4 Non-Gaussian Figure Space Densities

Although the Dryden-Mardia density is derived from a Gaussian figure-space density, many other figure-space densities lead to the same shape-space density. To illustrate this point, consider the transformation from figure-space to shape-space as a change of variables from $\mathbf{X}$ to $\mathbf{U}$, τ_x, τ_y, θ, and σ, where $\boldsymbol{\tau}$ is translation, θ orientation, and σ scale. The joint density over the new variables can be written as follows:

$$p(\mathbf{U}, \tau_x, \tau_y, \theta, \sigma) = p(\tau_x, \tau_y, \theta, \sigma|\mathbf{U})p(\mathbf{U}) \tag{6.29}$$

To generate a random sample from this density, we first generate a random shape vector $\mathbf{U}$ according to the Dryden-Mardia density. We then generate transformation parameters τ_x, τ_y, θ, and σ according to the density $p(\tau_x, \tau_y, \theta, \sigma|\mathbf{U})$. Applying the transformation parameters to $\mathbf{U}$, generates a figure-space example. The set of all figure-space examples generated in this way will be Gaussian distributed *only if*

$p(\tau_x, \tau_y, \theta, \sigma|\mathbf{U})$ *has a special form, which we designate by* $p_{\mathcal{N}}$. If $\mathbf{U}$ is instead acted on by transformation parameters selected from a different density $q(\tau_x, \tau_y, \theta, \sigma|\mathbf{U})$, there is no guarantee that the resulting figure-space density will be Gaussian. Consider for example a bimodal density q that transforms shape examples to two different scales in figure-space. The resulting figure-space density will not be Gaussian. Conversely, we can (sometimes) start from a non-Gaussian figure space and transform to a shape-space that is well-modeled by a Dryden-Mardia density. Thus, a number of figure-space densities lead to the same shape space density.

As a side-note, we remark that the transformation parameter density $p_{\mathcal{N}}(\tau_x, \tau_y, \theta, \sigma|\mathbf{U})$ induced by a Gaussian figure-space density *may indeed depend on* $\mathbf{U}$. To illustrate this point, consider the following example:

$$\nu = \begin{bmatrix} 0.0 & 1.0 & 0.5 & 0.0 & 0.0 & \sqrt{3}/2 \end{bmatrix}^T \tag{6.30}$$

$$\Omega = 10^{-3} \cdot \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 3.3 & 3.3 & 0.0 & 3.8 & 3.8 \\ 0.0 & 3.3 & 3.3 & 0.0 & 3.8 & 3.8 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 3.8 & 3.8 & 0.0 & 7.8 & 7.8 \\ 0.0 & 3.8 & 3.8 & 0.0 & 7.8 & 7.8 \end{bmatrix} \tag{6.31}$$

This model corresponds to a random triangle, in which the mean triangle is equilateral. Since all entries in the first and fourth rows and first and fourth columns of $\mathbf{\Omega}$ are zero, vertex 1 does not vary in position; it is always at its mean position – the origin. The second and third vertices, however, vary strongly in the direction $[1/2, \sqrt{3}/2]$ and weakly in the orthogonal direction. Further, vertex 2 and vertex 3 are exactly correlated so that if vertex 2 is displaced by a certain amount, vertex 3 is displaced by the same amount. The marginal covariance structure is shown in Figure 6.3a. The dashed lines going between the two ellipses are intended to remind the reader that vertex 2 and vertex 3 are exactly correlated, since it is not possible to show the complete covariance structure on a diagram such as this. We next generated random triangles

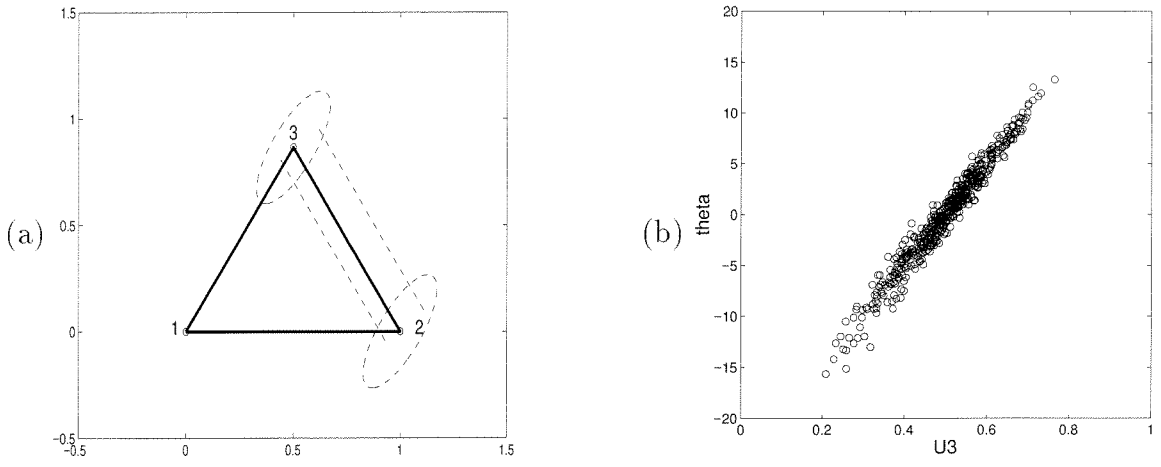


Figure 6.3: (a) Random triangle example in which vertex 2 and vertex 3 are exactly correlated (as indicated by the dashed lines linking the two ellipses). (b) The shape variable u_3 and the parameter θ are strongly correlated.

according to this density. The figure-space variables (coordinates of the vertices) were then transformed into shape variables u_3 and v_3 plus transformation parameters τ_x, τ_y, θ, and σ. Figure 6.3b shows a scatter plot of θ versus u_3, which clearly indicates that these variables are strongly correlated. The density $p_{\mathcal{N}}(\tau_x, \tau_y, \theta, \sigma|\mathbf{U})$, therefore, does depend on $\mathbf{U}$.

6.5 Parameter Estimation

The Dryden-Mardia shape density can be parameterized by ν and Ω, or, equivalently, by μ and Σ. To use the shape density in practice, these parameters must be estimated from training examples. Ideally, the estimation would be based on *shape space examples* $\mathbf{U}_1$, $\mathbf{U}_2$, ..., $\mathbf{U}_K$, but we have not found a reliable procedure for doing the estimation in this way. Counting degrees of freedom, there are $2(N-1)$ parameters in μ and $(2N-1)(2N-2)/2$ parameters in Σ. Two of the parameters in μ serve only to fix the scale and orientation in figure space; thus, they do not correspond to degrees of freedom in shape space. If all the parameters in Σ represent independent degrees of freedom, then the shape density has $N_p = (2N-4) + (2N^2 - 3N + 1) = 2N^2 - N - 3$ degrees of freedom [DM91].

Dryden and Mardia [DM91] suggest using a numerical maximum likelihood procedure to estimate the independent parameters from shape examples, where the likelihood is given by

$$\log \Lambda = \sum_{k=1}^{K} \log\ p_{\mathbf{U}}\left(\mathbf{U}_k | \mu, \Sigma\right) \tag{6.32}$$

Since Σ is a covariance matrix, only the elements in the upper triangle (including the diagonal) are free. However, a straightforward parameterization using Σ_{ij} with $j \geq i$ makes it difficult to enforce the positive definiteness constraint during optimization. To get around this problem, we used a decomposition of Σ known as the $\mathbf{LDL^T}$ factorization [GL89]. For Σ having full rank, the matrix $\mathbf{L}$ is lower triangular (same size as Σ) with ones on the diagonal and $\mathbf{D}$ is non-negative and diagonal. In this parameterization, $\mathbf{L}$ has $(2N-2)(2N-3)/2$ parameters and $\mathbf{D}$ has $2N-2$ parameters. The positive definiteness constraints can now be easily enforced by insisting that the elements in $\mathbf{D}$ remain positive.

Using a numerical gradient descent procedure, we attempted to maximize the log likelihood over the free parameters in μ and the parameters in $\mathbf{L}$ and $\mathbf{D}$ subject to the constraint that the d_i's remain positive. The typical behavior we observed with this approach was that the covariance matrix estimates became increasingly ill-conditioned. Since the inverse of Σ is needed to calculate $p_{\mathbf{U}}(\mathbf{U})$, the procedure eventually failed. We conjecture that the estimation fails because not all the parameters in Σ are truly free — the estimation problem may be ill-posed. In their original paper [DM91], Dryden and Mardia mention having difficulty with the estimation for general covariance structures. In their work they have typically used diagonal or other specialized covariance structures for which the estimation works fine.

In a private communication [Dry95], Dr. Dryden confirmed that they have experienced similar problems with general covariance estimation. An excerpt from this communication follows:

⋮

The subject of estimating shape covariance matrices is still very much in

progress. One cannot estimate all the shape parameters. An equivalence class of covariance matrices leads to the same shape distribution. Also singular covariance matrices can lead to non-singular shape distributions. Therefore there are great problems in estimating the parameters of the covariance matrix. We found that careful specification of covariance matrices can result in reasonable estimators (e.g. certain diagonal matrices) - see the inference section of [DM91]. However, such models can be criticized as being unrealistic.

As for the possible bug — we also found that for many data sets and covariance matrices the numerical routine estimates tended towards singular matrices. So you may not have a bug.

⋮

[Ian Dryden]

Given the apparently intrinsic nature of the problem, it does not appear that more sophisticated approaches to maximum likelihood estimation such as the EM algorithm [DLR77] will provide a breakthrough.

An alternative to performing the estimation based on shape examples is to use *figure-space examples* directly to estimate the parameters of a Gaussian or Gaussian mixture density. (For a single Gaussian, estimation of the parameters is straightforward, while for mixture models, the EM algorithm must be used [DLR77, RH84].) The problem with the single Gaussian approach is that the figure-space examples may not be well-modeled by a Gaussian even if the Dryden-Mardia density is appropriate in shape space. For definiteness consider the problem of detecting human faces. If the training faces are collected with various rotations and scalings, there is basically no hope that the locations of feature points in the image plane will follow a jointly Gaussian distribution. However, by collecting the training images in a special way, the Gaussian assumption becomes more reasonable. Specifically, the camera is held a fixed distance from the subject and the subject's head is upright. Translation is eliminated by mapping one facial feature (say the left eye) to the origin. Under

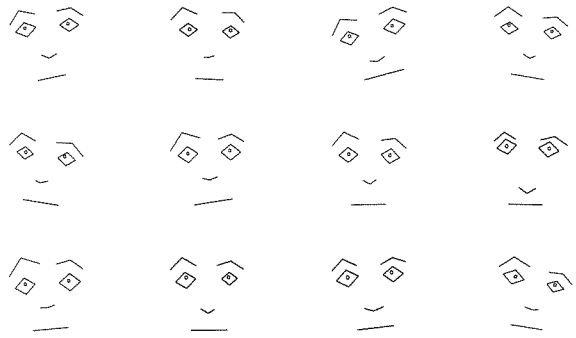


Figure 6.4: These cartoon faces were generated from a multivariate Gaussian distribution determined from training data. The faces show a reasonable amount of variability without deformity indicating that the Gaussian model may be reasonable.

these conditions, the variability in the positions of the other facial feature positions is primarily due to inherent variability between different individuals, and this variability may be reasonably approximated with a multivariate Gaussian.

As a sanity check, we can estimate the shape-density parameters $\hat{\boldsymbol{\nu}}$ and $\hat{\boldsymbol{\Omega}}$ from training examples and then generate random Gaussian vectors according to the estimated distribution: $\mathbf{X} \sim \mathcal{N}\left(\hat{\boldsymbol{\nu}}, \hat{\boldsymbol{\Omega}}\right)$. The vectors $\mathbf{X}$ can be plotted in figure-space where we can check whether the variability looks reasonable. For human faces, we hand-clicked the locations of fifteen facial features on a training set of 180 faces. After eliminating translation by mapping the left eye to the origin, we estimated the mean $\hat{\boldsymbol{\nu}}$ (a 30×1 vector) and the covariance matrix $\hat{\boldsymbol{\Omega}}$ (30×30) from the data. Figure 6.4 shows twelve cartoon faces that were based on the feature positions generated by $\mathcal{N}_{30}\left(\hat{\boldsymbol{\nu}}, \hat{\boldsymbol{\Omega}}\right)$. We have looked at several hundred of these cartoons and found that they exhibit considerable variability without any gross deformities. We believe this indicates that the Gaussian model is not unreasonable. Table 6.1 shows the estimated parameters for the mean and covariance of a reduced set of five facial features: the left eye (LE), right eye (RE), nose/lip junction (NL), left nostril (LN), and right nostril (RN), where left and right are defined with respect to the image. Note that these parameters were estimated from hand-clicked feature positions; they do not include the additional variability that would occur in practice due to the feature detector localization errors.

$$\hat{\nu}_R = \begin{bmatrix} LE_x = 0.0 \\ RE_x = 28.8 \\ NL_x = 14.8 \\ LN_x = 9.7 \\ RN_x = 20.1 \\ LE_y = 0.0 \\ RE_y = -0.1 \\ NL_y = 19.6 \\ LN_y = 17.9 \\ RN_y = 17.8 \end{bmatrix}$$

$$\hat{\Omega}_R = \begin{bmatrix} & LE_x & RE_x & NL_x & LN_x & RN_x & LE_y & RE_y & NL_y & LN_y & RN_y \\ LE_x & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ RE_x & 0.0 & 2.8 & 1.5 & 1.5 & 1.5 & 0.0 & -0.1 & -0.7 & -0.6 & -0.9 \\ NL_x & 0.0 & 1.5 & 3.7 & 3.5 & 3.3 & 0.0 & -3.0 & -2.5 & -1.8 & -2.9 \\ LN_x & 0.0 & 1.5 & 3.5 & 3.7 & 3.1 & 0.0 & -3.1 & -2.7 & -1.9 & -3.1 \\ RN_x & 0.0 & 1.5 & 3.3 & 3.1 & 3.3 & 0.0 & -2.9 & -2.1 & -1.6 & -2.6 \\ LE_y & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ RE_y & 0.0 & -0.1 & -3.0 & -3.1 & -2.9 & 0.0 & 6.2 & 3.4 & 2.3 & 4.4 \\ NL_y & 0.0 & -0.7 & -2.5 & -2.7 & -2.1 & 0.0 & 3.4 & 5.3 & 4.1 & 5.4 \\ LN_y & 0.0 & -0.6 & -1.8 & -1.9 & -1.6 & 0.0 & 2.3 & 4.1 & 3.7 & 4.4 \\ RN_y & 0.0 & -0.9 & -2.9 & -3.1 & -2.6 & 0.0 & 4.4 & 5.4 & 4.4 & 6.2 \end{bmatrix}$$

Table 6.1: The estimated mean and covariance matrix for the positions of five facial features. The rows and columns of zeros in Ω occur because the parameters were estimated from translation-normalized data.

6.6 Summary

In our approach to modeling deformable object classes, the allowed object deformations are represented through shape statistics. We use the term "shape" in the sense of Kendall and Bookstein to mean the properties of a configuration of labeled points that are invariant to transformations such as translation, rotation *in the image plane*, and scaling. Given a spatial configuration of N labeled points in a plane, the shape may be defined by mapping two points to fixed reference positions. The $2N - 4$ coordinates of the remaining points constitute the shape of the configuration. We have also worked out the theory for *affine invariant shape* although this work is presented elsewhere [BWLP96, Leu95].

By using probability densities over shape variables, we hope to encode which object deformations are more likely. A particularly useful density over shape was derived by Dryden and Mardia, who showed that for a Gaussian density in figure space, the induced density in shape space has a special form (the Dryden-Mardia density). With the Dryden-Mardia density we can easily determine the density of shape variables computed with respect to any baseline pair of features. In addition, we can specify partial densities over subsets of shape variables. Finally, if the figure-space density is not well-modeled by a single Gaussian, we can use a mixture of Gaussians, which induces a mixture of Dryden-Mardia densities in shape space. One may wonder whether a Gaussian mixture can be used directly to model the shape space density. This is certainly possible, but then we lose the nice properties of being able to switch between different choices of basis pairs. Also, if we are forcing a Gaussian mixture model where it is not appropriate, then we may need many modes in the model to get a reasonable approximation.

One difficulty with the Dryden-Mardia density, however, is that there is currently no systematic way to estimate the shape density parameters from shape examples. We have tried maximum likelihood approaches but these do not work (the covariance estimates become singular). An alternative is to estimate the parameters from figure space examples, but this method is not as satisfying since the training examples must

be collected in a controlled way. It does provide a workable solution, however.

Chapter 7 Hypothesis Selection

7.1 Introduction

In this chapter we will focus on using shape statistics to provide robust recognition performance. As mentioned in the previous chapters, instances of an object class in an image are detected by finding the appropriate object parts in the "correct" spatial configuration. The part detectors may come in several varieties depending on the particular application. For locating facial features, the principal components analysis approach or other image based methods may work best. Regions of color, texture, or motion can also serve as features. The only requirement is that the features provide localization information. Indeed, even non-visual features, such as a coarse location estimated from audio cues, can be incorporated in our framework. Features may also come at a variety of resolutions. A coarse resolution "head" is as much a feature, as a fine resolution eye corner.

A fundamental fact about feature detectors is that they are not perfectly reliable. There are two basic types of errors: missed features and false alarms. Thus, locations identified by a particular detector are treated only as candidates for the actual object part. In this chapter, we discuss how candidate part locations are grouped into object hypotheses and then scored.

7.2 Problem Formulation

We will suppose that there are N types of object parts with a detector for each type. The nature of the parts may depend on the particular application, but in general the detectors will not be perfectly reliable. Two types of errors can occur: (1) the detector may fail to respond at a true feature location (missed feature) and/or (2) the detector may respond at erroneous locations (false alarms). Hence, the locations identified by

a particular detector should be treated only as candidates for the actual object part. For a robust system, we must be able to rank hypotheses with missing features as well as complete hypotheses. Also, in some recognition problems such as handwriting the same features occur repetitively; only the context of surrounding features can disambiguate whether a cusp is a part of one letter or another. The part candidates can be organized into a data structure $\mathbf{W}$ having N rows (one for each part type) and an uneven number of columns:

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_{11} & \mathbf{w}_{12} & \dots & \mathbf{w}_{1M_1} \\ \mathbf{w}_{21} & \mathbf{w}_{22} & \dots & \dots & \mathbf{w}_{2M_2} \\ \vdots & & & & \\ \mathbf{w}_{N1} & \dots & \mathbf{w}_{NM_N} \end{bmatrix} \tag{7.1}$$

The number of candidates identified for part n is designated by M_n. Note that $\mathbf{w}_{nm}$ contains the (x, y) coordinates of the m-th candidate for part n.

From $\mathbf{W}$, we can formulate hypotheses about which subset of the candidate locations actually constitute an object. A hypothesis is just an N-component vector

$$H = [h_1, h_2, \dots, h_N]^T \tag{7.2}$$

where $h_n \in \{0, 1, \dots, M_n\}$ specifies that element $\mathbf{w}_{nh_n}$ is hypothesized to be the true location of part n. The option $h_n = 0$ allows for the possibility that true part n may not be among the candidates (due to detector failure or occlusion). The problem can now be stated as: *Given the candidates* $\mathbf{W}$, *determine the best object hypothesis* H. Note that we have cast the problem in a hypothesis selection framework; there is another paradigm based on averaging over hypotheses [MR94], which we have not considered.

7.3 Hypothesis Selection

The optimal method for selecting the best hypothesis from a set of mutually exclusive and exhaustive alternatives is well known; as shown in Chapter 2 it is the maximum a posteriori or MAP rule:

$$\text{Select } H_* \text{ if } \quad \Pr(H_*|\mathbf{W}) \geq \Pr(H_k|\mathbf{W}) \quad \forall k.$$

The best hypothesis can also be determined from pairwise ratios of the posterior probabilities. Define

$$R_{jk} \triangleq \frac{\Pr(H_j|\mathbf{W})}{\Pr(H_k|\mathbf{W})} = \frac{P(\mathbf{W}|H_j)}{P(\mathbf{W}|H_k)} \cdot \frac{\Pr(H_j)}{\Pr(H_k)} \tag{7.3}$$

where the second form follows from Bayes' rule. In terms of R, the MAP rule is given by:

$$\text{Select } H_* \text{ if } \quad R_{*k} \geq 1 \quad \forall k.$$

Notice that we could define a "goodness function"

$$G(H) \triangleq P(\mathbf{W}|H) \cdot \Pr(H) \tag{7.4}$$

and use G to rate hypotheses. The problem, however, is that G depends on *all* the points in $\mathbf{W}$, including those that are not hypothesized to be part of the object. To avoid this problem, we will now make two assumptions about the candidate points:

Assumption 1: False alarm locations are distributed independently of the true part locations.

Assumption 2: False alarm locations are distributed independently of each other.

Clearly, neither of these assumptions is strictly true. For example, the presence of an object such as a face in an image alters the false alarm distribution in the area where the object is. Likewise, structure in the clutter background leads to false alarm clumping, which violates Assumption 2. Nevertheless, these assumptions are valuable

because they permit simplification of the goodness function $G(H)$ to a form that depends only on the points listed in H, i.e., the points hypothesized to be part of the object.

Let $\mathbf{W}(H)$ represent the points listed in H, and let $\mathbf{W}(\bar{H})$ represent the points in $\mathbf{W}$ not listed in H. (These points are hypothesized to be false alarms.) Using the first assumption, we can write

$$P(\mathbf{W}|H) \;=\; P_H\left(\mathbf{W}(H)\right) \cdot Q_{\bar{H}}(\mathbf{W}(\bar{H})) \tag{7.5}$$

where P_H denotes the joint probability density for the locations of all object features present in the hypothesis H. For example, if $H = [2, 5, 0, 7, 1]$, then P_H would be the joint density for the locations of object features 1, 2, 4, and 5. Similarly, $Q_{\bar{H}}$ is the (joint) probability density for all the other candidate locations (hypothesized to be false alarms). By Assumption 2, Q factors into a product of terms, which can be expressed using the shorthand notation

$$Q_{\bar{H}}(\mathbf{W}(\bar{H})) \;=\; \prod q\left(\mathbf{W}(\bar{H})\right) \tag{7.6}$$

where the product goes over all points $\bar{H}$ that are not listed as part of the object hypothesis. Substituting Equations 7.5 and 7.6 into Equation 7.3 and canceling terms leads to the following:

$$R_{jk} \;=\; \frac{G_0(H_j)}{G_0(H_k)} \tag{7.7}$$

where

$$G_0(H) \;\triangleq\; \Pr(H) \cdot \frac{P_H\left(\mathbf{W}(H)\right)}{\prod q\left(\mathbf{W}(H)\right)} \tag{7.8}$$

Observe that $G_0(H)$ depends only on the points of $\mathbf{W}$ that are listed in H. Contrast this result with the original goodness function defined in Equation 7.4, where the goodness depended on *all* points in $\mathbf{W}$. To find the best hypothesis now, we must

simply find H_* that maximizes G_0.

7.4 Evaluation of the Goodness Function

The two terms in Equation 7.8 will be referred to as the prior probability and the likelihood ratio. These two terms must be evaluated in order to compute G_0 for a given hypothesis.

7.4.1 Prior Probability

The prior probability of a hypothesis depends on the prior probability $p(\omega_1)$ that the image contains an object and on the performance of the feature detectors. Suppose initially that we have only a single feature. Let the number of candidate locations identified for this feature equal M. Hence, a hypothesis H is simply an element $\in \{0, 1, \ldots, M\}$, with H_0 meaning none of the candidates is the true feature and $H_{j\neq 0}$ meaning that candidate j is the true feature. A convenient way to view the problem is that the true feature is put into the pool of candidates with probability $\tilde{\gamma} = \gamma p(\omega_1)$ and then a random number K of false alarms are added. Here, γ is the probability of detection i.e., the probability the true object feature will be detected given that the object is present in the image, while *gamma* is the joint probability that an object is present *and* detected. Given a set containing M candidates, the prior probabilities are:

$$\Pr(H_0|M) = \frac{(1-\tilde{\gamma}) \cdot P_K(M)}{\tilde{\gamma} \cdot P_K(M-1) + (1-\tilde{\gamma}) \cdot P_K(M)} \tag{7.9}$$

$$\Pr(H_{j\neq 0}|M) = \frac{\tilde{\gamma} \cdot P_K(M-1)/M}{\tilde{\gamma} \cdot P_K(M-1) + (1-\tilde{\gamma}) \cdot P_K(M)} \tag{7.10}$$

where $P_K(k)$ is the probability that the number of false alarms added equals k.

For the case of multiple features, this analysis can be easily generalized. We suppose that the number of false alarms K_n added to candidate pool n is independent from pool to pool; however, the detection probabilities are governed by a joint distribution

$\Gamma(\mathbf{b})$, where $\mathbf{b}$ is a string of N bits. For example, $\Gamma(\mathbf{11000})$ specifies the probability that features 1 and 2 of the object are detected, but features 3–5 are not (again conditioned on the object being present). The function Γ can be estimated from detector performance on training data. Ignoring the detector false alarm statistics (i.e., assuming $P_K(M-1) \approx P_K(M)$), we obtain the approximate result:

$$\Pr(H_0|\mathbf{M}) \approx [1-(1-\Gamma(\underline{\mathbf{0}}))\,p(\omega_1)] \;/\; \prod \mathbf{M}(\mathbf{b}) \tag{7.11}$$

$$\Pr(H_{\neq 0}|\mathbf{M}) \approx p(\omega_1)\cdot\Gamma(\mathbf{b}) \;/\; \prod \mathbf{M}(\mathbf{b}) \tag{7.12}$$

where $\mathbf{b}$ is determined from H by $b_n = 1$ if $h_n \neq 0$ and $b_n = 0$ if $h_n = 0$. $\mathbf{M}$ is a vector containing the number of candidates for each type of feature. The product $\mathbf{M}(\mathbf{b})$ means the product of M_n over all n such that $b_n \neq 0$. For $\mathbf{b} \equiv \mathbf{0}$, we define the product to equal 1.

In practice we have found that there is a problem with using the prior probability given by Equation 7.12. We have set up the probabilities so that the "universe" consisted of all the hypotheses associated with *one image*. That is, the sum over all hypotheses for an image adds up to one. Thus, scores for hypotheses from one image should not really be compared with scores from another image. A configuration of points in one image may receive a very different score from the exact same configuration of points in another image especially if one image has many more background false alarms than the other. A second consequence is that the algorithm favors hypotheses in which object parts having many candidates tend to be declared missing.

An alternative is to replace $\Pr(H|\mathbf{M})$ with the function $F(\cdot)$ defined below. This change corresponds to weighting the score given to a hypothesis by the probability (estimated over many positive examples) of observing the same pattern of present and missing features.

$$\begin{aligned} F(H_0) &\triangleq [1-(1-\Gamma(\underline{\mathbf{0}}))\,p(\omega_1)] \\ F(H_{\neq 0}) &\triangleq p(\omega_1)\cdot\Gamma(\mathbf{b}) \end{aligned} \tag{7.13}$$

Using $F(\cdot)$ has the advantage that (1) missing features are penalized in accordance with the detector statistics and (2) the same configurations of points will receive the same scores in different images, i.e., scores can be compared across images. Empirically, this change eliminated most of the instances where a hypothesis with missing features is scored better than a true hypothesis that has no missing features. For handwriting, where the features are repetitive and each page of writing has the same feature (e.g., a cusp) many times, the replacement of $\Pr(H|\mathbf{M})$ by $F(H)$ was essential.

7.4.2 Likelihood Ratio

The other part of the goodness function consists of the likelihood ratio:

$$L\left(\mathbf{W}(H)\right) = \frac{P_H\left(\mathbf{W}(H)\right)}{\prod q(\mathbf{W}(H))} \tag{7.14}$$

The numerator consists of the conditional probability density given object over the points listed in the hypothesis H. Similarly, the denominator consists of the conditional probability density given background over the points listed in the hypothesis H. Instead of expressing the densities in terms of $\mathbf{W}(H)$, which is a vector containing the x and y-coordinates of the points listed in H, it will be convenient to make a change of variables:

$$\mathbf{W}(H) \rightarrow \tau_x,\ \tau_y,\ \theta,\ \sigma,\ \mathbf{U} \tag{7.15}$$

where τ_x, τ_y, θ, and σ are the translation, rotation, and scale of the configuration and $\mathbf{U}$ is the *shape*, i.e., the information remaining after translation, rotation, and scale (TRS) are factored out. The key observation we need is that the transformation from $\mathbf{W}(H)$ to the new variables is one-to-one so the densities are the same except for a Jacobian factor that depends only on the transformation. This factor cancels out in the ratio; therefore,

$$L\left(\mathbf{W}(H)\right) \;=\; \frac{P(\mathbf{U})}{Q(\mathbf{U})} \cdot \frac{P(\tau_x, \tau_y, \theta, \sigma|\mathbf{U})}{Q(\tau_x, \tau_y, \theta, \sigma|\mathbf{U})} \tag{7.16}$$

Here, $P(\cdot)$ represents the probability density of the argument conditioned on being from the object distribution and $Q(\cdot)$ the probability density conditioned on the background distribution. Although the false alarm locations were assumed to be independent, the normalization transformation introduces correlations among the shape variables; hence, $Q(\mathbf{U})$ is a full joint density.

The first term on the right-hand side of Equation 7.16 is the likelihood of observing shape variables $\mathbf{U}$ given the object distribution versus given the background distribution. This term measures how consistent the spatial configuration of parts is with the object class.

The second term encompasses any information contained in the TRS parameters. When the figure space density is Gaussian, we showed in Chapter 6 that the density of the TRS parameters could be dependent on the shape; thus, the TRS densities are expressed conditioned on $\mathbf{U}$. In practice, however, this dependency can be ignored since the TRS parameters are largely governed by external factors such as the position of the camera relative to the object, the pose of the object, etc. The TRS likelihood term has the following interpretations:

- Prior Knowledge: If we know side information about the TRS parameters, then the TRS likelihood term can be used to incorporate this information. For example, suppose that in the face localization problem, we know the head is upright to within 20°. The observed θ then contains useful information about whether the hypothesized object could be a face or not. The TRS term exploits this type of knowledge in a principled way.

- Invariance: Suppose instead that the algorithm is intended to be completely invariant to TRS. That is, we want to recognize upside-down faces as well as right-side up. In this case, the TRS likelihood term should be discarded. As a caution, unless the class-conditional densities of the TRS parameters (conditioned on object and conditioned on background) are identical, there will be some loss of useful information (and performance) by doing this.

7.5 The Background Distribution

Recall that the likelihood ratio (Equation 7.14) contains the term $P(\mathbf{U})/Q(\mathbf{U})$, where P is the shape density given object and Q is the shape density given background. $P(\cdot)$ can be estimated from training data as described in Section 6.5, but what about $Q(\cdot)$? We believe a reasonable approximation is that the false alarms are Gaussian distributed over the image plane with a greater concentration near the center of the image. That is, the figure space distribution conditioned on background is assumed to be $\mathcal{N}(\underline{\mathbf{0}}, \sigma^2\mathbf{I})$. Because we eventually transform to shape variables, the parameter σ is irrelevant and may be set to one. The false alarm locations in figure space are assumed to be independent, but this is not true in shape space because the TRS normalization introduces correlations. Thus, the density $Q(\mathcal{S})$ is given by Theorem 1 with $\boldsymbol{\nu} = \underline{\mathbf{0}}$ and $\boldsymbol{\Omega} = \mathbf{I}$.

One might wonder whether the background distribution can be ignored or replaced by a constant in the likelihood ratio. This would be equivalent to making a decision based solely on $P(\mathbf{U})$, the density over shape given the object distribution. There are two problems with this idea. First, the value of $P(\mathbf{U})$ depends on the pair of features that is used as the baseline. Second, if features are missing, $P(\mathbf{U})$ will be a probability density in a lower dimensional space and there is no reason to expect these values to be comparable to those from the full density. The likelihood ratio corrects both problems by normalizing $P(\mathbf{U})$ with respect to the background density. The likelihood ratio is invariant to the choice of baseline features since the information in a shape vector $\mathbf{U}_{1,2}$ computed with respect to the baseline pair $(\mathbf{p}_1, \mathbf{p}_2)$ is *exactly the same* as the information in a shape vector $\mathbf{U}_{3,4}$ computed with respect to the baseline pair $(\mathbf{p}_3, \mathbf{p}_4)$. (There is a one-to-one, invertible mapping between the two shape spaces.) Even though the white Gaussian density may not provide a particularly model for the background, it must be included to produce the proper normalization.

7.6 Conditional Search

We have now specified a complete procedure for computing the "goodness" score of a hypothesis. In principle, we could check every possible hypothesis in order to determine the best one; this brute-force approach, however, is computationally expensive. We would like a more efficient way to search through the hypotheses.

Given the positions of two features, the possible positions of all other features are highly constrained. This idea is used as follows. First, we consider partial hypotheses of the form $H = [h_1, h_2, \ldots]^T$ with $h_1 \neq 0, h_2 \neq 0$. Assuming the points $\mathbf{w}_{1h_1}$ and $\mathbf{w}_{2h_2}$ correspond to the true object features, we can use the shape density to estimate where the other features should be and how much uncertainty exists about their location. This allows us to define search regions in the image plane. Hypotheses are only formed which couple the two reference points h_1 and h_2 with candidate points falling inside the appropriate search regions. We then loop over all values of h_1 and h_2 and repeat the procedure. Since our algorithm must be robust to missing features, we also consider hypotheses of the form $[h_1, 0, h_3, \ldots]^T$, $[0, h_2, h_3, \ldots]^T$, and $[0, 0, h_3, h_4, \ldots]$. We have developed a compact recursive procedure to generate all the viable hypotheses in this way.

The resulting procedure is not a search algorithm in the conventional sense, but instead simply a method for enumerating all "plausible" hypotheses. The word "search" refers to the process of looking within the image plane uncertainty regions for suitable candidate points to link together into hypotheses. All plausible hypotheses are then evaluated and compared. We acknowledge that more efficient techniques may exist for finding the best B hypotheses or for finding all hypotheses above a certain minimum score.

The search regions can be determined either from the theoretical shape density or empirically. Figure 7.1 shows the uncertainty for facial features as computed from a training database containing 180 face images (18 people $\times$ 10 instances each). Figure 7.1a shows the definitions of the features on one face from the training database. In Figure 7.1b, the left-eye and right-eye of each training face was mapped to a fixed

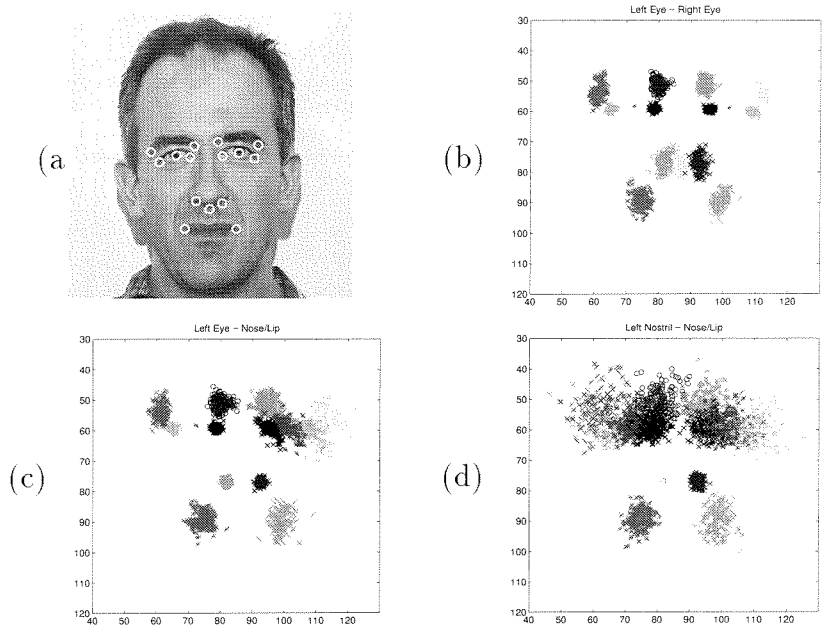


Figure 7.1: (a) Definitions of fifteen facial features on a typical face. (b) Superposition of the features from 180 training faces after being mapped into shape space with the eyes as reference. The clouds show the positional uncertainty for the other features. (c) Uncertainty using the left-eye and nose-lip as reference. (d) Uncertainty using the left-nostril and nose-lip as reference.

reference position. The clouds of points show the superposition of the other facial features. Figure 7.1c shows the uncertainty using the left-eye and nose/lip junction as the baseline pair, while Figure 7.1d shows the uncertainty if the left-nostril and nose-lip are used as reference.

The computational savings of the conditional search procedure versus brute-force are shown in Figure 7.2. These results were obtained during face localization experiments using a face model based on five features: the left-eye, right-eye, nose-lip, left-nostril, and right-nostril. The conditional search procedure holds the number of hypotheses to be evaluated at a manageable level and yields comparable accuracy.

7.7 Summary

We have introduced an object recognition approach based on finding the appropriate object parts in the "correct" spatial configuration. Local feature detectors are used

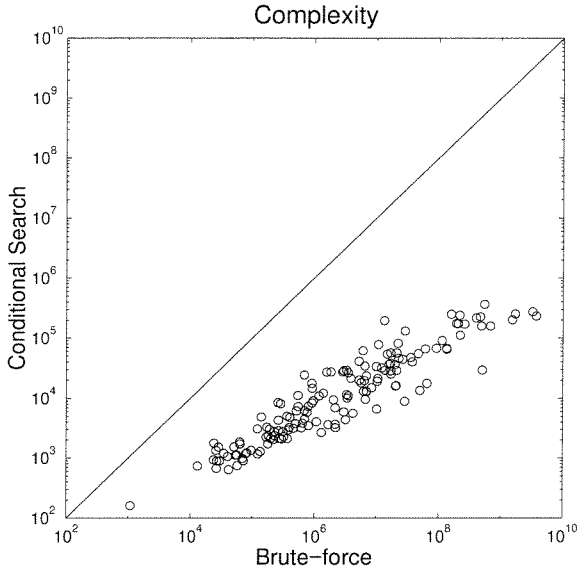


Figure 7.2: The number of hypotheses formed with the conditional search method versus brute-force.

to find object parts; however, these detectors are unreliable and typically result in missed features and false alarms. Thus, the detector outputs are treated as candidate locations for the corresponding parts. To reliably locate an object, the candidate locations are grouped into object hypotheses and scored based on the configuration of the parts (represented using shape variables). Partial configurations, which occur when true parts are occluded or missed by the detectors, can be handled with this method. A priori knowledge about the position, orientation, or scale of an object can also be incorporated.

Since brute-force evaluation of all possible object hypotheses is not typically feasible, a conditional search procedure that utilizes what is known about the object shape is used to form plausible hypotheses. Given the locations of two object parts, the location and uncertainty of all the other parts can be predicted. For example, given the position of two eyes, we can predict where the nose should be found. Only nose candidates falling inside the appropriate search region are used to form hypotheses. The conditional search procedure provides a significant reduction in computation over the brute-force approach without sacrificing accuracy. More efficient procedures for finding the best B hypotheses or for finding all hypotheses above a certain minimum

score may exist, but we have not pursued this avenue of research.

Chapter 8 Shape Experiments

This is an applications chapter in which we present experimental results for the shape-based recognition algorithm described in the last two chapters. To demonstrate the robustness of the shape-based approach, we have included results on two very different problem domains: (1) locating human faces in cluttered scenes and with occlusion and (2) finding keywords in on-line handwriting data collected from a graphics tablet.

8.1 Face Localization

8.1.1 Introduction

Algorithms to reliably detect and track human faces in cluttered backgrounds are essential for the development of advanced human-machine interfaces. Once the face (or faces) can be successfully localized, a number of applications become possible, including user-authentication for credit card and ATM transactions, video teleconferencing, gaze tracking, lip reading, and passive monitoring (for example, watching people's reactions at a movie). Previous work on face localization is described in Section 1.3.4.

8.1.2 Datasets

We have used three different datasets in our work on face localization; a brief summary of each is given in Table 8.1. The first dataset, known as the Burl-Leung database, contains ten images for each of eighteen different subjects in a "studio" setting. Each subject was imaged against a plain white background with controlled lighting conditions and pose. The camera was positioned approximately 2 meters from the subjects. For each subject we obtained ten different instances by asking the subject to slightly vary head pose, gaze direction, and facial expression.

Dataset	# Images	Description
BL Database	180	10 instances $\times$ 18 subjects in a "studio" setting
LAB Sequence	150	1 subject moving freely in a cluttered environment
MM Sequence	400	1 subject with periodic partial occlusion of the face

Table 8.1: Datasets used for face experiments.

The second dataset, known as the LAB sequence, was collected under more realistic conditions. The subject was seated in a chair approximately two meters from the camera and allowed to move freely, make facial expressions, etc. The background was cluttered and additional people moved around behind the subject so the background was not constant. Individual frames (150 total) were grabbed at a rate of approximately 1 frame per second (1 Hz) and analyzed off-line. Although this dataset is a time-sequence, all frames were analyzed independently; the temporal correlation between successive frames was *not used* in any of the experiments.

The third dataset, known as the MM sequence, was generated primarily to demonstrate the robustness of the algorithm to occlusion. The subject was again seated two meters from the camera, but part of the subject's face was periodically blocked with his hand or another object (a white bicycle helmet). This sequence contains 400 frames and was sampled at approximately 5Hz. Again, all frames in the sequence were analyzed independently.

8.1.3 Experiments

We have used two different feature sets $\mathcal{F}_1$ and $\mathcal{F}_2$ in our experiments. The specific features used in each set are listed in Table 8.2. These features were hand-selected based on intuition and the fact that they have distinctive local brightness patterns.

We have also experimented with two different sets of feature detectors. The first set of detectors $\mathcal{D}_1$ is based on multi-orientation, multi-scale filtering with elongated even and odd kernels that are sensitive to lines and edges. The response of the filters at a particular (x, y) location can be viewed as a characterization of the brightness in the local neighborhood. To detect a feature such as an eye, we compare the vector

Name	Symbol	$\mathcal{F}_1$	$\mathcal{F}_2$
Left Eye	LE	X	X
Right Eye	RE	X	X
Nose/Lip Junction	NL	X	X
Left Nostril	LN	X	
Right Nostril	RN	X	
Left Mouth Corner	LM		X
Right Mouth Corner	RM		X

Table 8.2: Features used in various face localization experiments.

of filter responses at each (x, y) location to a template response vector obtained from the eye in a training image. Local maxima of the match score that exceed an absolute threshold serve as eye candidates. A complete description of this method is given in [LBP95].

The second set of detectors $\mathcal{D}_2$ is based on local orientation structure. The image is transformed to an orientation map, which shows the dominant orientation structure at each location in the image. The detectors then look for areas where the local orientation structure matches that of a prototype object part. This method generally shows less sensitivity to illumination changes than the filtering method, which is based directly on the image gray-level values. A more complete description of the method is given in [BWLP96]

For the initial experiments, we tested on the LAB sequence using features $\mathcal{F}_1$ and detectors $\mathcal{D}_1$. The parameters of the shape distribution were estimated from hand-clicked feature locations from the Burl-Leung database. A small scaled identity matrix was added to the estimated covariance to allow for the fact that the detectors introduce additional localization error into the position of the features. In testing, the local detectors were applied to each image. Hypotheses were formed from the detector outputs via the conditional search method outlined earlier. The hypotheses were scored and ranked from best to worst with the highest-ranked hypothesis declared to be a face. If the highest-ranked hypothesis does not represent a correct localization of the face, it is recorded as an error.

On the LAB sequence we obtained a correct localization performance of 87%.

That is, in 87% of the images the highest scored hypothesis corresponded to a correct localization of the face. The LAB sequence contains one particularly challenging section of 15–20 images where the head is rotated significantly in depth. Since both the detectors and shape statistics were trained from quasi-frontal faces, we did not expect the algorithm to work on this section. Scored only over the quasi-frontal face images, the performance is 94%. Figure 8.1 shows the best hypothesis produced by the algorithm on some typical images. Note that the algorithm is still able to locate the face when features are missing.

We also tested the algorithm on the training database, which is easier in the sense that the background is benign, but more challenging in terms of the variety of individuals represented. Figure 8.2 shows that the algorithm still works. However, over the whole database, the performance is not as good (63% correct localization). Possible explanations are that we are not using enough features or that the basic feature detectors are not robust enough to work well across such a variety of individuals.

In a follow-on set of experiments, we used features $\mathcal{F}_2$ and detectors $\mathcal{D}_2$ to detect faces in the MM sequence and the LAB sequence. For these experiments, we decided to estimate the shape statistics directly from the detector outputs to insure that the correct detector localization error statistics would be included in the overall shape density. One problem, however, is that the $\mathcal{D}_2$ detectors are person-specific, so we could not train on the Burl-Leung database. Hence, training of the shape parameters was done on the LAB sequence.

The detector outputs were associated with the true (hand-clicked) features using a distance threshold. The shape statistics were then estimated from the detected locations. To increase the number of training samples, we mirrored the LAB sequence images, which effectively doubles the number of examples. The algorithm was tested both on the LAB sequence (training data) and the MM sequence. In both cases, the first image in the sequence was used to define the prototype orientation map for each of the features. In the remaining images, features were detected by looking for regions where the local orientation structure matched the prototype. The performance on each sequence is above 90%. Figure 8.3 shows the best hypothesis on selected images from

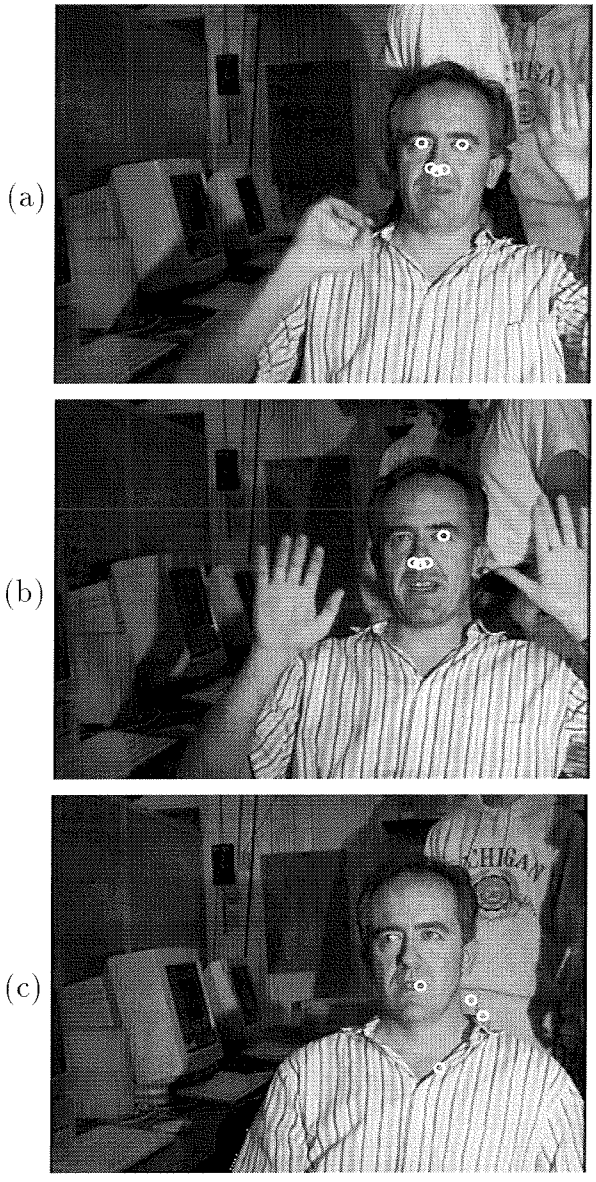


Figure 8.1: Performance on selected images from the LAB sequence. The best hypothesis is shown in each case: (a) correct, (b) correct, despite detector failure for the left eye, (c) incorrect, an error is caused by four false alarms that happen to occur in a face-like arrangement.

Figure 8.2: Performance on a variety of individuals from the training database.

the two sequences. Notice that the algorithm works well despite the strong occlusions present in the MM sequence.

8.1.4 Strawman

As a standard for comparison, we evaluated a simple strawman face localization algorithm based on normalized correlation. The first face in the LAB sequence was selected as a template to be applied to the remainder of the sequence. The template and images were down-sized by a factor of four to reduce computation. The highest correlation value from each image was selected as the face location. Performance on the LAB sequence yielded only 57% correct localization.

We repeated the strawman procedure on the MM sequence using the first face from the sequence as the template. Performance on this sequence was 64%. Surprisingly, the normalized correlation algorithm was not as sensitive to occlusions of the face by the hand as expected (perhaps because of the similarity in skin tone). Occlusions with the white bicycle helmet, however, caused serious problems. Lighting variations due to self-shadowing also caused the correlation algorithm to fail.

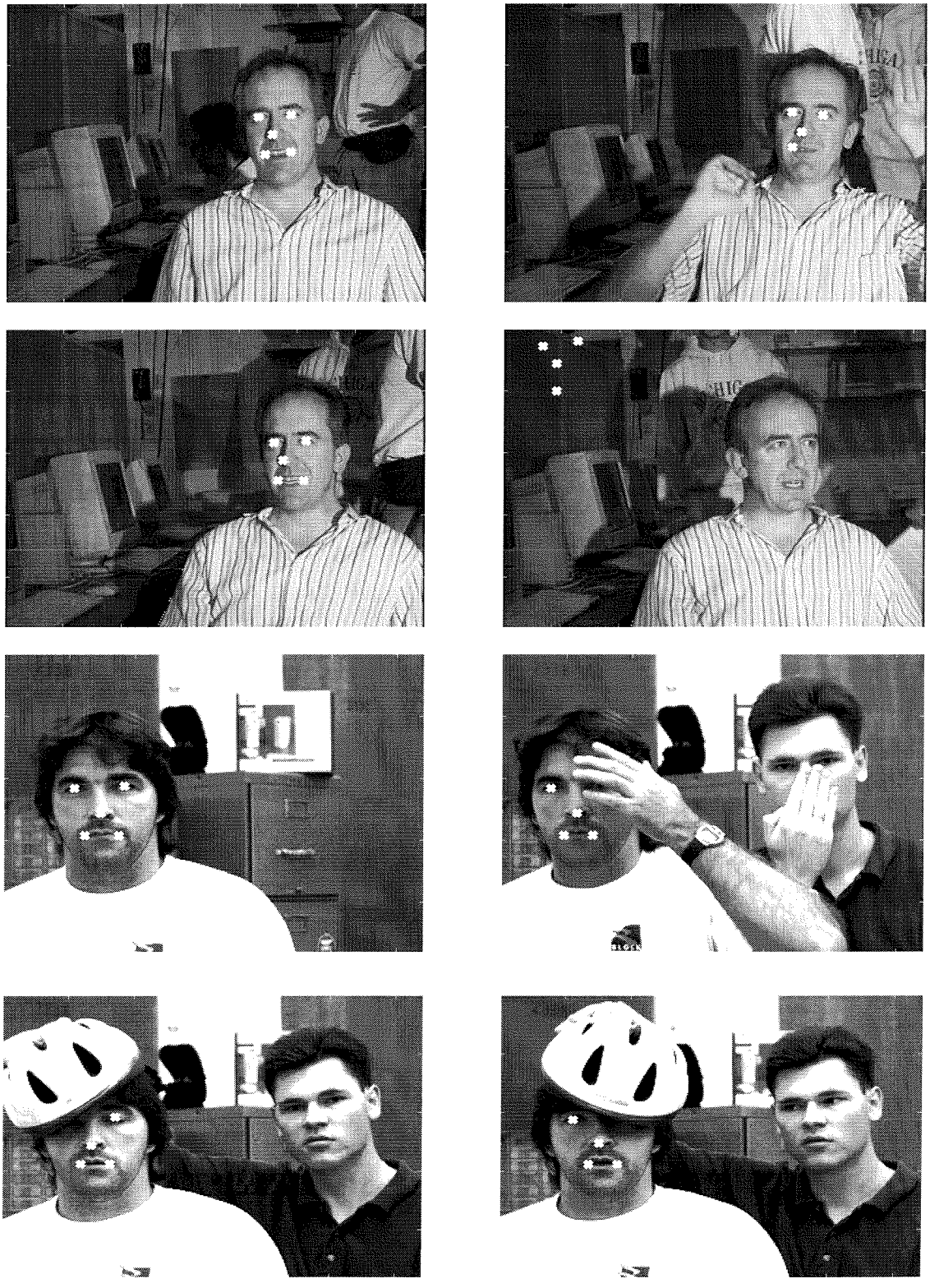


Figure 8.3: Performance on selected images from the LAB and MM sequences. Only the best hypothesis is shown in each case. The second figure in the right column is an error caused by four false alarms that happen to occur in a face-like arrangement.

8.2 Handwriting

8.2.1 Introduction

For many applications, handwriting appears to offer a more natural human-computer interface than the traditional keyboard. Entering data by keyboard is especially frustrating for novice users who possess limited typing skills. Advanced users also encounter difficulty when entering mathematical equations, tables, sketches, and other visually formatted material. Even entering "standard" text is a major hurdle for users working in character-based languages such as Japanese.

As with human faces, handwriting can be modeled as a deformable configuration of parts. The parts we have used include pen lifts/drops, humps, cusps, and crossings. In this section we will discuss using shape models to find keywords in cursive handwriting data. A brief summary of other approaches to handwriting is given in Section 1.3.6. The leading alternative is the Hidden Markov Model (HMM) []. Our shape method has several advantages and disadvantages with respect to HMMs. One advantage is that the shape-based method is applicable to both on-line and off-line handwriting, yet can be adapted to exploit temporal information if it is available. Another advantage of the shape approach is that the position of a particular feature can depend on the positions of a number of other local features, while in HMMs only first or second order dependence is typically assumed.

A disadvantage of the shape method is that to learn the appropriate spatial statistics, we need a number of training examples with ground truth. HMMs, however, can be trained from a relatively small number of examples that have not been specially labeled. Also, HMMs provide a model for the entire writing trajectory, while our method provides only a model for the keypoint positions.

We have developed some simple feature detectors to locate keypoints in signature data collected from a graphics tablet. In our experiments we have used the WACOM tablet, which provides x, y and p (pressure) samples versus time. The detected keypoints are grouped into hypotheses and evaluated based on their spatial configuration.

8.2.2 Feature Detection

The feature detectors for this problem are, of course, domain specific and quite different from the detectors used in the face localization experiments. The detectors we have developed for handwriting are described below.

Lifts and Drops

Pen lifts and drops are relatively easy to detect using the pressure coordinate provided by the tablet. Let p_n be the discrete-time sequence of pressure samples. By applying a simple threshold to this sequence, we can reliably binarize the samples into a sequence b_n of zeros and ones where zero represents pen up (not in contact with the tablet) and one represents pen down. A pen drop is detected at time n if $b_{n-1} = 0$ and $b_n = 1$. Similarly, a pen lift is detected if $b_n = 1$ and $b_{n+1} = 0$.

Humps and Cusps

Figure 8.4 shows a portion of a digitized curve. At each point, the angle θ_n of the segment from (x_n, y_n) to (x_{n+1}, y_{n+1}) is measured with respect to horizontal. Since the quantity of interest for locating humps and cusps is the *change* in θ, we form the sequence

$$\delta\theta_n \;=\; \theta_n - \theta_{n-1} \tag{8.1}$$

Noise is suppressed by smoothing $\delta\theta$ with a narrow Gaussian kernel. Keypoints are then detected as local extrema over a three pixel window on the smoothed sequence. The state of the pen (up or down) is also checked. To qualify as a hump or cusp feature, the pen must be down and local maxima must exceed an absolute threshold of 30°, while local minima must be less than −30°. Features are labeled as "right-turn" or "left-turn" based on the sign of $\delta\theta$ (smoothed) at the keypoint. For very sharp (hairpin) turns, the change in angle may be close to ±180°. Such points should probably

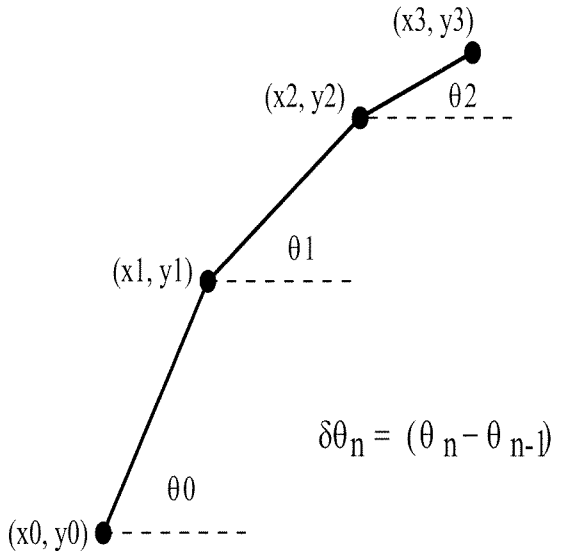


Figure 8.4: Portion of a digitized curve. At each sample point, the change in direction is calculated.

be given a distinct label, but we have not done so in our experiments. Figure 8.5 illustrates the process of detecting humps and cusps on a cursive letter G.

Crossings

The final feature we have used consists of places where writing crosses over itself as in a cursive ell. Since the digitized handwriting can be broken up into a number of line segments, we simply need to check whether the current line segment intersects with any previously written segments, which is a standard problem in computer graphics [FvD84]. To avoid spurious detections due to pauses and other anomalies, only segments that are within a specific time window in the past are checked.

A line segment with endpoints (x_0, y_0) and (x_1, y_1) can be expressed as follows:

$$\mathbf{s}_1 = (1-\alpha)[x_0\, y_0]^T + \alpha[x_1\, y_1]^T, \quad \text{where } \alpha \in [0,1] \tag{8.2}$$

To check whether this segment intersects with a second segment

$$\mathbf{s}_2 = (1-\beta)[u_0\, v_0]^T + \beta[u_1\, v_1]^T, \quad \text{where } \beta \in [0,1] \tag{8.3}$$

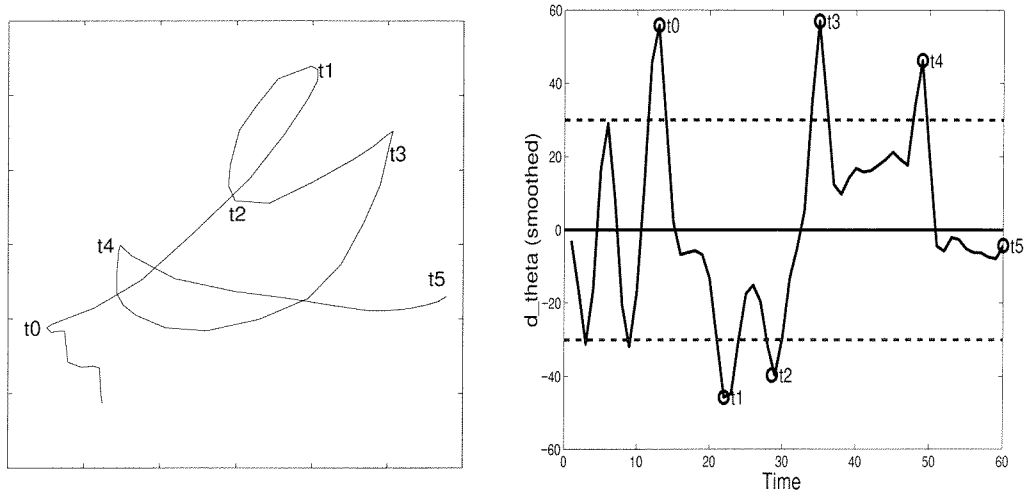


Figure 8.5: (a) Cursive letter G with important time instants marked. (b) Detection of humps and cusps from the smoothed $\delta\theta_n$ sequence.

we first apply some quick "sanity checks." In particular, we check whether

$$\begin{aligned} \max(x_0, x_1) &\geq \min(u_0, u_1) \\ \max(y_0, y_1) &\geq \min(v_0, v_1) \\ \min(x_0, x_1) &\leq \max(u_0, u_1) \\ \min(y_0, y_1) &\leq \max(v_0, v_1) \end{aligned} \tag{8.4}$$

All of these conditions must hold true or the segments cannot possibly intersect because the rectangles bounding each segment will not even intersect.

Assuming the segments pass the sanity checks, we look for solutions α and β for the equations:

$$(1-\alpha)x_0 + \alpha x_1 = (1-\beta)u_0 + \beta u_1 \tag{8.5}$$

$$(1-\alpha)y_0 + \alpha y_1 = (1-\beta)v_0 + \beta v_1 \tag{8.6}$$

This pair of equations can be rewritten in matrix form as follows:

$$\begin{bmatrix} x_1 - x_0 & u_0 - u_1 \\ y_1 - y_0 & v_0 - v_1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} u_0 - x_0 \\ v_0 - y_0 \end{bmatrix} \tag{8.7}$$

Provided the determinant of the matrix is nonzero, a solution for α and β exists in closed form. However, the segments intersect if and only if the solution is such that $\alpha \in [0,1]$ and $\beta \in [0,1]$. Note: If the determinant equals zero, it means the two segments are parallel and they can either be distinct (no solutions) or overlapping (infinite number of solutions).

Crossings are different from the other feature types in that they do not occur at precisely one of the time samples. To associate a time and position with a crossing feature, we must interpolate between values. If segment $\mathbf{s}_1$ is the later segment, which crosses with the earlier segment $\mathbf{s}_2$, the position and time $[x_*, y_*, t_*]^T$ assigned to the crossing feature is given by:

$$\begin{bmatrix} x_* \\ y_* \\ t_* \end{bmatrix} = (1 - \alpha_*) \begin{bmatrix} x_0 \\ y_0 \\ t_0 \end{bmatrix}_* + \alpha_* \begin{bmatrix} x_1 \\ y_1 \\ t_1 \end{bmatrix}_* \tag{8.8}$$

where α_* is the solution from Equation 8.7. Note that the time is taken from the later segment.

Sample Performance

Figure 8.6 shows the performance of the feature detectors on a piece of handwriting collected with the WACOM tablet. The individual segments of the handwriting are shaded according to the pen pressure: light gray lines indicate little or no pen pressure, while dark lines indicate normal to heavy pen pressure. The detections are labeled according to feature type: (1) pen lift, (2) left turn, (3) right turn, (4) pen drop, and (5) crossing.

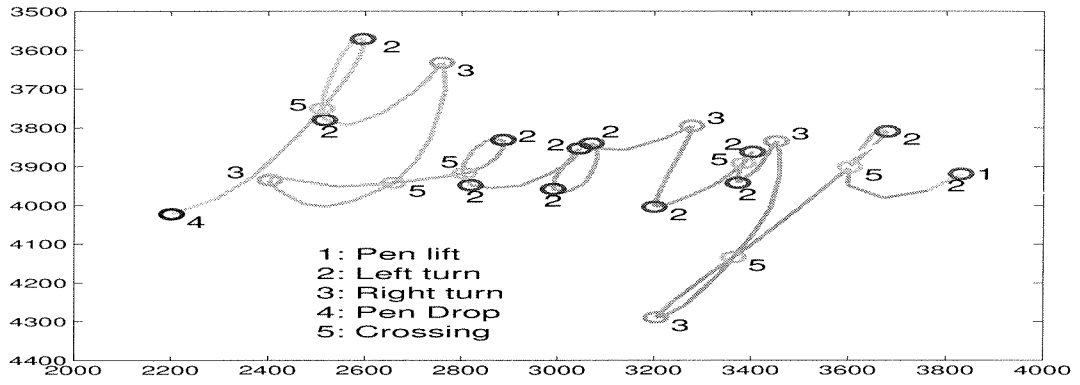


Figure 8.6: Feature detector performance on a sample of handwriting data.

8.2.3 Shape Models

As shown in Figure 8.6, the five basic features occur in many places in a sample of handwriting. To find a particular word or word fragment, we must rely on the spatial configuration of the features. However, since the precise positions of the features vary for different realizations of the same word, joint probability densities are used to encode the allowed deformations.

Figure 8.7a shows a cursive letter G. The eleven numbered locations were manually identified as potential object parts, i.e., places likely to be found by the feature detectors. Figure 8.7b shows the superposition in shape space of 100 realizations of the letter G. The clouds represent the uncertainty in the positions of parts 3–11. For reference the solid line marks the entire shape-space trajectory for one of the samples. From this figure we can only see the *marginal* shape-space density for each part, *not* the joint density.

The joint density over the shape variables can be well-modeled using a Dryden-Mardia density [DM91, BLP95, BLP96], which we denote by $p_{\mathbf{U}}(\mathbf{U};\ \boldsymbol{\mu}, \boldsymbol{\Sigma})$. The parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are estimated from the *detected* positions of the features. Parts that are not detected reliably are omitted from the model as are parts that do not have a ground truth location in every training example (e.g, part 7 and part 9 of the G only exist in 20% of the training examples). Eliminating bad parts leaves a G model

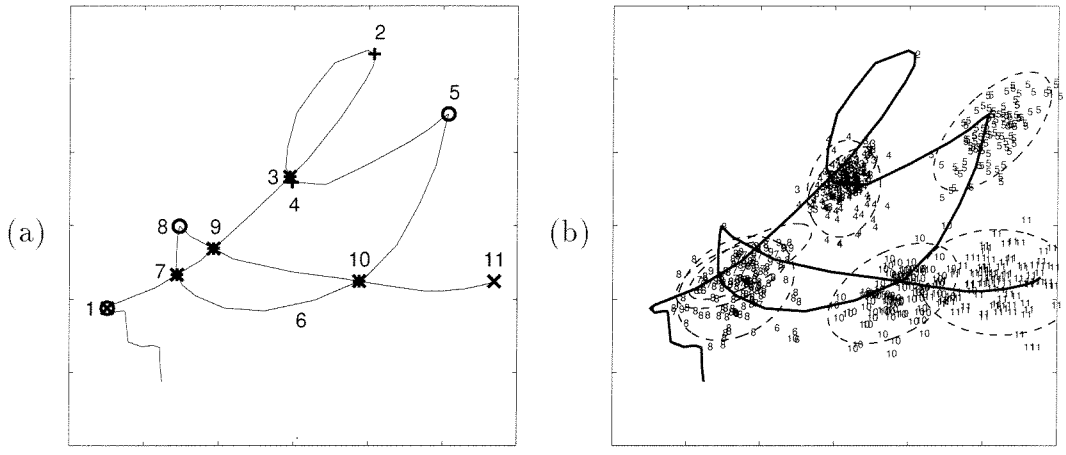


Figure 8.7: (a) A cursive letter G with definitions of hand-selected object parts. (b) Uncertainty regions in shape space (features 1 and 2 used as reference).

consisting of just six parts: 1, 2, 3, 5, 8 and 10.

There are two shape models associated with any object. One is the model of the ground truth positions of the parts, while the second is the model of the *detected* positions of the parts. Since recognition is performed based on the detected positions of parts rather than their ground truth locations, we want to estimate a model of the second type. This is accomplished by applying the feature detectors to each of the 100 training examples. During the training phase, the detected positions are associated with the nearest ground truth part. Thus, we form a table in which each row corresponds to one sample and the columns are the x and y coordinates of the detected positions for each part. Some parts, however, are not detected in every training example. We could treat these as missing data and use the EM algorithm [OW72, BL75, DLR77, RH84] to impute their values, but instead we will simply replace the missing values by the corresponding ground truth coordinates. The final estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are obtained by shifting a given part to the origin in each training example and computing the sample mean and covariance matrix for the other part positions *in figure space.*

To find an object such as the letter G, hypotheses are generated from the set of

feature locations. Each hypothesis is scored using the same criteria as for faces:

$$G_0(H) = F(H) \cdot \frac{p_{\mathbf{U}}(\mathbf{U}; \boldsymbol{\mu}, \boldsymbol{\Sigma})}{p_{\mathbf{U}}(\mathbf{U}; \mathbf{0}, \mathbf{I})} \tag{8.9}$$

where $\mathbf{U}$ is the shape of the configuration under hypothesis H and $F(H)$ is a factor to penalize a hypothesis that is missing features.

8.2.4 Time

Up to this point we have defined a model that describes only the spatial layout of handwriting. We have not said anything about the temporal structure (except in the context of feature detection). To some extent, this is a selling point since the shape method can be used both for on-line and off-line data. However, if the time information is available, we can expect to do better by exploiting it.

In principle, handwriting could be modeled using a joint probability density over both shape and time, but we have taken a simpler approach using time only to guide the selection of the reference features and to prune hypotheses that do not have the correct *time ordering*.

For selecting the baseline pair (i.e., the two reference parts used to map a hypothesis into shape space), the time separation between the two parts is examined to see if it is comparable to the time separation in the training data. Similar checks are also performed on the orientation and length of the baseline, since this allows the user to set limits on the amount of invariance allowed. If the test data is known to be in the usual orientation, there is no need to consider upside down hypotheses.

The other place in the processing where we have used the time information is during the conditional search procedure for generating hypotheses. Here we add the requirement that the features must occur in the correct time order. Although this seems like a simple enough constraint, it is extremely effective in pruning hypotheses because the feature detector labels are so coarse. That is, a detector label 2 stands for a right turn, but this type of feature is a candidate for many object parts. Enforcing the proper time order reduces the number of different combinations.

The time information could also be used *metrically* during the conditional search, although we have not currently implemented this constraint in our algorithms. Given a baseline pair, the location in shape space of the other parts is constrained (as shown in Figure 8.7b). Similarly, the location in time of the other parts is also constrained. As the distance from the baseline pair increases, the spatial search regions tend to become larger and may overlap with writing from the previous or next line. By also considering the time, the number of candidates falling inside the search region is limited to those in the correct time frame. Figure 8.8 shows the time uncertainty for parts of the letter G in relative units, i.e., relative to the time between part 1 and 2. In other words, the time associated with part p in relative units is given by

$$\tilde{t}_p = \frac{t_p - t_1}{t_2 - t_1} \tag{8.10}$$

The figure shows a Gaussian approximation to the probability distribution of $\tilde{t}_p$, for $p = 3, \ldots, 11$. The average time from the start of the G to part 5 is approximately twice as long as the average time from the start to part 2. If parts 1 and 2 were used as the baseline, then part 5 should be found in the time range $[1.5\tilde{t}_2, 2.8\tilde{t}_2]$ and in the ellipse in shape space shown in Figure 8.7b. Only part 5 candidates falling in both the time range and shape uncertainty region would be coupled with this baseline pair in the hypothesis formation process.

8.2.5 Modifications to the Hypothesis Generation Procedure

Handwriting is, by nature, highly repetitive; hence, a single page of writing leads to a large number of detected features. The conditional search procedure described in Section 7.6 can be used to generate hypotheses from the detected features, but even then the number of hypotheses is enormous. To reduce the computational load and memory requirements, we have made several modifications to the hypothesis generation procedure.

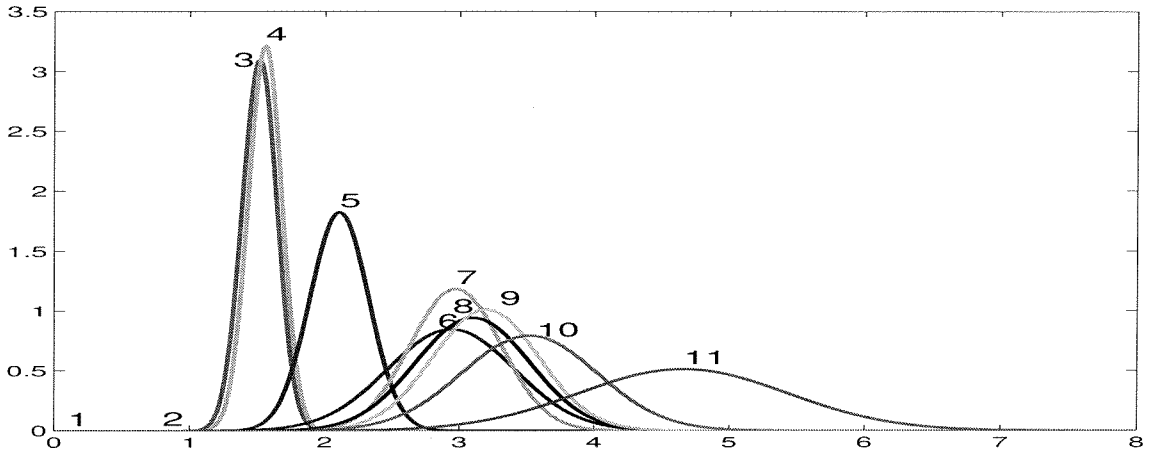


Figure 8.8: Uncertainty in the time location of the parts. The time between parts 1 and 2 is used as the unit of measurement in each sample.

Windowing

Rather than generate hypotheses over an entire page of handwriting, we used windowing to work on smaller portions of a page at one time. Windowing can be implemented either spatially or temporally, but we have used temporal windows in our experiments. The window size was chosen to be three times the maximum time separation between parts of the object (as determined from training data). Successive windows were overlapped by two-thirds to insure that no objects could "fall through the cracks."

Allowed Baseline Pairs

The standard hypothesis generation procedure allows any pair of parts to serve as the baseline that is used to transform to shape space. Since the area of the search regions grows approximately quadratically with the distance from the baseline, using small baselines leads to large uncertainty regions, thereby increasing the number of false alarms falling inside each region. To overcome this problem, we have simply introduced a table which defines whether a given pair of part types is allowed to serve as a baseline. For the detection probabilities with which we are working, only about four baseline pairs are needed to insure a high probability that at least one baseline pair on the object will be considered. The table of allowed baseline pairs is currently

defined manually.

A consequence of considering only selected pairs of baselines is that the recursive hypothesis generation algorithm described in Section 7.6 will no longer work. Instead, we simply loop over pairs of points. If two points are of the right types to serve as a baseline pair, we use the search regions to identify plausible candidates for the other parts. All combinations of plausible part candidates are grouped into hypotheses and evaluated. This process is repeated over all pairs of points.

Degree of Invariance

Another change, which we mentioned briefly in the previous section, is that the user is allowed to specify the amount of invariance desired. For example, if the orientation of the handwriting is known to be upright, there is no point in wasting resources doing a fully rotation-invariant search. Similarly, if the scale is known to be within a particular range, this information can be used to restrict the hypothesis formation process. The desired degree of invariance over the time difference between pairs of parts can also be specified.

During training, the scale (σ), orientation (θ), and time-difference (δt) between each pair of object parts is measured. The minimum and maximum values over the training set between each pair of part types is recorded in a table. For orientation, special care must be taken in computing the minimum and maximum values because of the 2π periodicity. For each pair of part types i and j, the table will contain:

$$[\sigma_{\min}(i,j), \sigma_{\max}(i,j), \theta_{\min}(i,j), \theta_{\max}(i,j), \delta t_{\min}(i,j), \delta t_{\max}(i,j)] \tag{8.11}$$

During testing, each candidate baseline is checked to see whether the scale, orientation, and time difference fall within the allowed range observed during training. The user can specify "invariance factors" to increase the allowed range. In the context of faces, suppose the distance between the two eyes (as observed on training examples) ranges from 30 to 60 pixels. The user can specify two invariance factors, say 0.5 and 1.5. The algorithm will not consider eye candidates that are closer than $0.5 \cdot 30$ pixels

or farther than $1.5 \cdot 60$ pixels as a legitimate baseline pair. If the invariance factors are denoted by σ_{lo}, σ_{hi}, θ_{lo}, θ_{hi}, δt_{lo}, and δt_{hi}, then a candidate baseline with scale (σ), orientation (θ), and time-difference (δt) will not be accepted unless *all* of the following conditions hold:

$$\sigma \in [\sigma_{\min}(i,j) \cdot \sigma_{\text{lo}}, \sigma_{\max}(i,j) \cdot \sigma_{\text{hi}}]$$
$$\theta \in [\theta_{\min}(i,j) + \theta_{\text{lo}}, \theta_{\max}(i,j) + \theta_{\text{hi}}]$$
$$\delta t \in [\delta t_{\min}(i,j) \cdot \delta t_{\text{lo}}, \delta t_{\max}(i,j) \cdot \delta t_{\text{hi}}]$$

The condition on the time difference is actually more complicated than what we have written above, because the time difference between some parts may be negative. The algorithm sets the interval to something reasonable based on whether $\delta t_{\min}$ and $\delta t_{\max}$ are both positive, both negative or of opposite sign. We emphasize that the interval tests are only applied to pairs of parts that are under consideration as a baseline pair. The tests are not applied to any other parts.

Time-ordering

In the previous section, we mentioned that hypotheses are rejected if the parts do not satisfy the correct time ordering. This constraint is quite effective in pruning the number of hypotheses.

Missing Features

For our handwriting experiments we have considered models consisting of up to 10 parts. If each part is allowed to be present or missing, then a full hypothesis can lead to slightly less than 2^{10} sub-hypotheses. Clearly this is unacceptable and wasteful since hypotheses that are missing too many features will not be ranked highly anyway. To reduce the number of hypotheses, we have placed a limit on the number of missing features. (In the experiments the limit was set to 3 or 4 depending upon the number of parts in the model).

Dominant Hypotheses

A new problem is created by abandoning the recursive hypothesis generation procedure: the same hypothesis (and all its missing feature sub-hypotheses) may be generated several times by different baselines. Since the overall algorithm only keeps a record of the best 1000 hypotheses what can happen is that the list of best hypotheses will be filled up by variations of a few of the best hypotheses.

We have introduced a "dominance test" to eliminate hypotheses that are simply variations of better hypotheses. Currently this test is applied after hypotheses from a given window have been scored, but before they are merged into the top-1000 list. The test is applied again to the merged list since the same hypothesis may be selected from two adjacent windows. The basic idea of dominance is that if two hypotheses share a common feature point, the hypothesis with the higher score dominates the hypothesis with lower score. All hypotheses that are dominated by another hypothesis will not appear in the final ranking. It would be very useful if we could determine that a hypothesis is inferior before doing the full evaluation of the scoring function. For example, it may be possible to determine that a particular full hypothesis will dominate all its sub-hypotheses that are missing features. We have not yet pursued this direction, however.

8.2.6 Experimental Results

We have conducted two experiments to show that the shape-based methods used for faces are also viable for recognizing handwriting. These experiments are intended as a demonstration of the generality of shape methods rather than a specification for a complete handwriting recognition system. The basic scenario is that a user has recorded some handwritten notes on a portable graphics tablet and now would like to look back through the notes and find places where certain *keywords* appear.

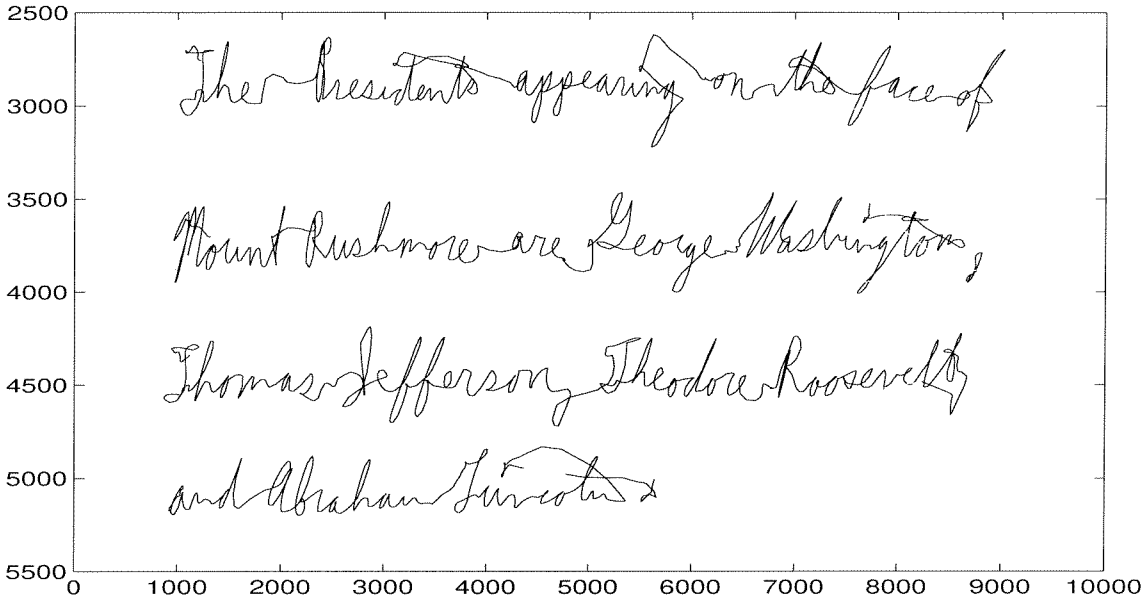


Figure 8.9: Handwritten notes about Mount Rushmore recorded with a pen system.

Rushmore Passage

As a specific example, suppose that on a tour of the United States, a user of the pen system has recorded some notes about Mount Rushmore as shown in Figure 8.9 (actual text below).

```
The Presidents appearing on the face of
Mount Rushmore are George Washington,
Thomas Jefferson, Theodore Roosevelt,
and Abraham Lincoln.
```

After returning home, the user would like to find this passage in his notes. Ideally, he should simply write the word *George* and have the computer retrieve the passage. Since shape models are statistical, only one example of the keyword does not provide sufficient information to determine how the word varies when written a number of times. In practice, there are three potential solutions for this problem. First, the keyword as written by the user could be assumed to be the average example. Variation around the average could be modeled as isotropic, white noise on the feature positions.

Initially, a small noise variance could be used, but then increased if no matches were found. By providing feedback the user could iteratively refine the model (assuming the keyword appears multiple times in the passage). If the algorithm returns a number of matches, the user could verify the correct ones, and from those build a refined model combining the initial white noise model with the new information in a Bayesian fashion.

A second potential solution is for the user to prestore a bank of training data including on the order of 100 handwritten instances of each upper and lower case letter and in some cases bigrams (2-letter combinations). The computer system would first recognize letters in the query word and then synthesize a shape model for the entire word from the stored letter models.

A final solution, which is not practical, is for the user to write the keyword a number of times. Obviously, no one would want to do this, but it does permit a shape model to be generated. Due to the lower overhead, this is the method we have used in our experiments.

In the first experiment, we looked for the keyword *George* in the Mount Rushmore passage. In principle an entire keyword can be mapped to shape space and modeled with a single Dryden-Mardia density or mixture of these densities. However, a better approach is to break the query word into smaller fragments, each with 5–10 features. A separate DM density can then be estimated for each of these fragments. Splitting the word into fragments is a good idea for several reasons: (1) with the allowance for missing features, the number of hypotheses generated from an entire word is too large, (2) the assumption of a Dryden-Mardia density may be more appropriate when applied locally rather than to an entire word, (3) the full joint density over the entire word may not be reliably estimated from a limited amount of training data; however, by splitting the word into pieces we are essentially zeroing out a number of covariance parameters, reducing the number of parameters to be estimated, (4) splitting leads to a hierarchical model in which a word consists of fragments having a particular spatial arrangement, and each of these fragments in turn consists of 5–10 points having their own spatial arrangement, and (5) the detailed arrangement of parts at the end of

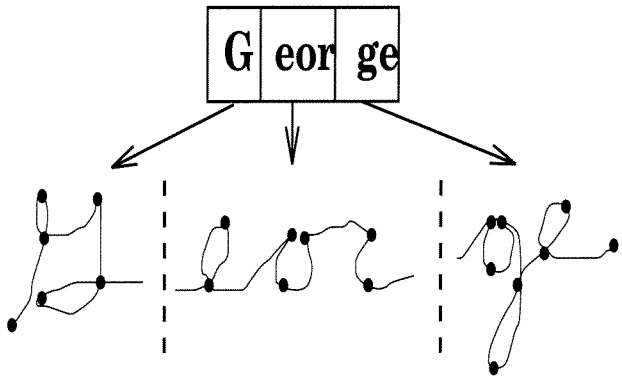


Figure 8.10: Hierarchical Model. The word *George* is divided into three pieces. Each of the pieces is further subdivided into 6–8 parts indicated with large dots. The overall word detector first seeks the individual pieces and then looks for the three pieces in the proper spatial arrangement.

the word does not depend much on the detailed arrangement at the beginning of the word, except through the global scale, position, and orientation parameters, but this dependency can be most easily compensated for at the top level of the hierarchy (through the arrangement of the word fragments).

As shown in Figure 8.10, we manually split the word *George* into three fragments: *G*, *eor*, and *ge*. Figure 8.11 shows the fragment spotting results on the Mount Rushmore passage. Separate shape models were used to detect each fragment. The best hypothesis for the *G* is at the true location (we have just plotted a thick line through the features of the best hypothesis according to their time ordering). The best hypothesis for *eor* occurs on the *eod* in *Theodore*. This mistake looks quite reasonable especially since one of the features on the true *eor* (at the end of the *r*) was missed by the feature detectors. The second best *eor* hypothesis is a variation of the best hypothesis[1]. The third best occurs at the true *eor*, but the last feature, which does not really exist, is misplaced at the bottom part of the circle in the *g* following the *r*. The fourth and fifth best hypotheses are again variations on the *eod* in *Theodore*. The sixth best hypothesis is the correct *eor* in *George* with the last feature declared missing. The best *ge* hypothesis occurs at the correct location.

[1]We had not yet implemented all the modifications to the search procedure at the time this experiment was conducted, and in particular we had not implemented the dominance test.

#	e_{11}	e_{12}	o_{14}	o_{15}	o_{16}	r_{18}	r_{19}	score
1	144	94	265	147	267	120	149	7.5852
2	144	94	265	147	267	121	149	7.5852
3	91	52	163	94	165	71	96	7.4075
4	144	94	265	147	267	120	151	6.8578
5	144	94	265	147	267	121	151	6.8578
*6	91	52	163	94	165	71	0	6.6669
7	144	94	265	0	267	120	149	6.4804
8	144	94	265	0	267	121	149	6.4804
9	91	52	163	94	165	73	96	6.4123
10	91	52	163	94	165	72	96	6.3896

Table 8.3: List of the top ten hypotheses for the location of *eor*. Note that zeros indicate a missing part. The best hypothesis actually occurs on the *eod* of *Theodore*. Many of the other top ten hypotheses are simply variations on this best one. The third best, however, is the true location of *eor* in *George*. The last part, which should be at the bottom of the r, is incorrect. In the sixth hypothesis, the last part is correctly declared to be missing. By combining these hypotheses for *eor* with the hypotheses for G and *ge*, we can identify the best location of the entire word *George*.

Using the results for the separate word pieces, it is apparent that the best hypothesis for the complete word *George* is on the true word. Although the *eod* in Theodore is the best candidate for *eor*, there is no supporting evidence around this fragment to indicate the word is *George*. The same applies to *eor*-hypotheses 2, 4, and 5. The third *eor*-hypothesis has supporting evidence because it is preceded by what appears to be a G and is followed by what appears to be a *ge*. Depending upon the implementation of the top level hierarchy, there may be a minor problem with this hypothesis in that the completion of the *eor* (i.e., the time associated with the last feature) is *after* the start of the *ge*. Hypothesis 6 is the correct *eor*-hypothesis and has the supporting evidence of the G and *ge* as well as the proper time-ordering between the fragments.

Declaration of Independence

The Rushmore passage contains only on the order of 20 words and 115 characters. To test our algorithm under more stringent conditions, we asked a user to transcribe the Declaration of Independence, which contains approximately $1,300$ words and $7,500$

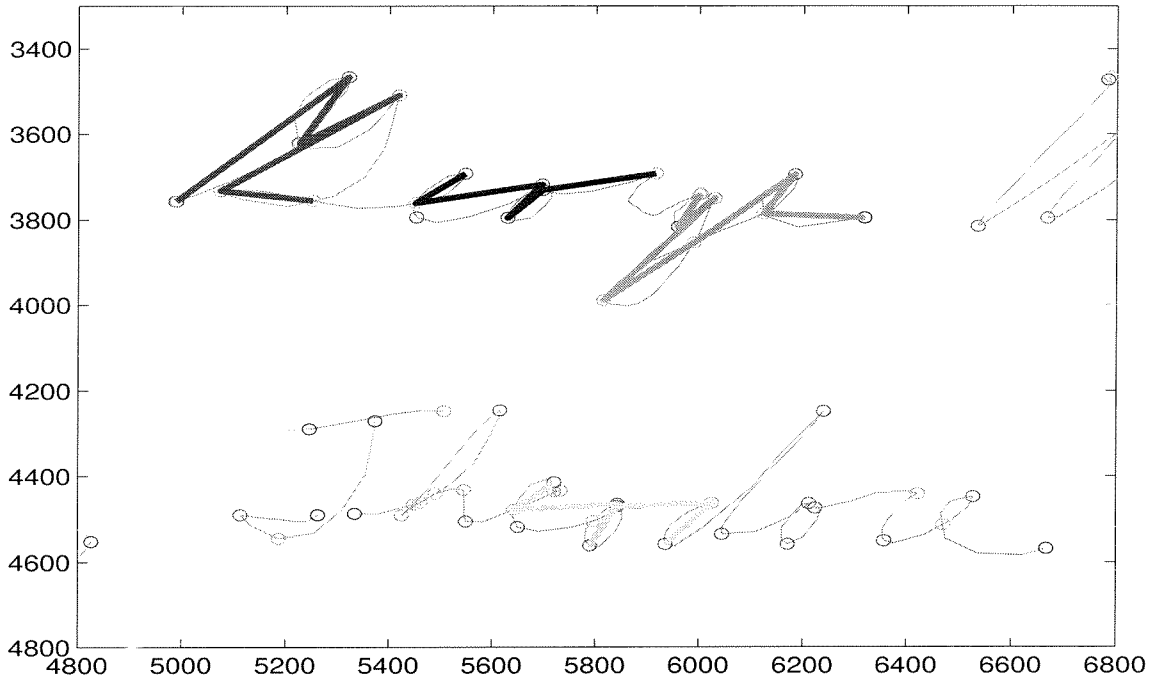


Figure 8.11: Rushmore Passage Results. *Red*: best hypothesis for *G*. *Yellow*: best hypothesis for *eor*. *Green*: best hypothesis for *ge*. *Blue*: sixth best hypothesis for *eor*. The best hypothesis for the word is formed from the red, blue, and green fragments. The yellow fragment, although it is the best *eor* hypothesis, does not have any supporting evidence to suggest the word *George*. Note: colors are only visible in the on-line version of the thesis.

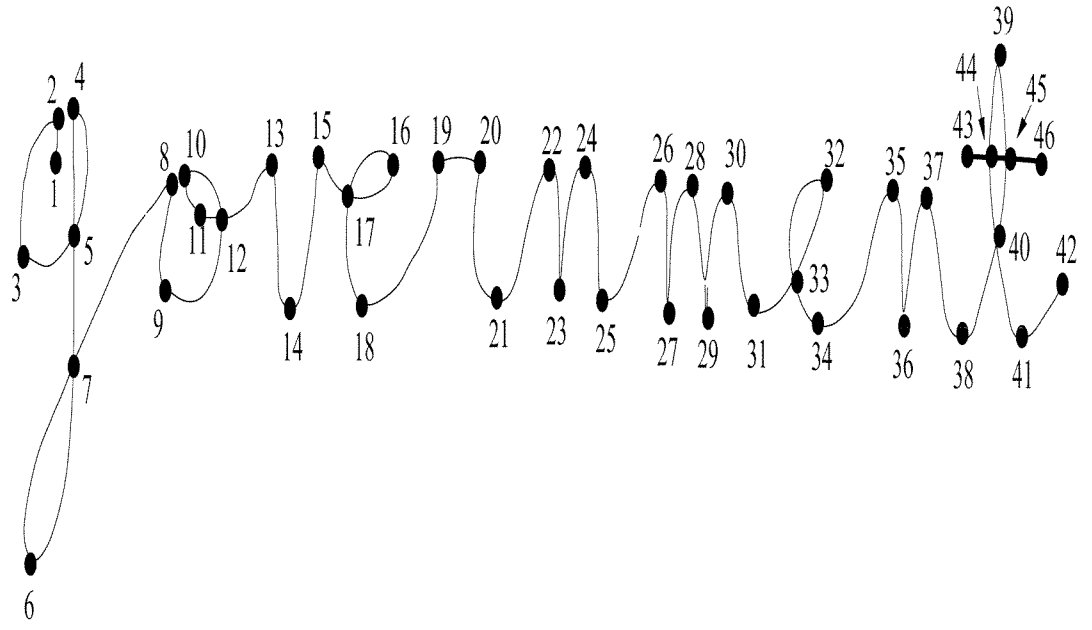


Figure 8.12: Manually identified parts on the word *government.*

characters. The word *government* was selected as a keyword since it appears 11 times in the document. Figure 8.12 shows the parts that were manually identified as potentially useful.

Detection performance for the lowercase letter g was evaluated. Our model consisted of six parts: 1,3,4,5,6, and 7. Part 2 was omitted because it could not be reliably detected. Hypotheses were generated from the detected feature positions and scored based on the spatial configuration. Figure 8.13 shows sample results on a well-known passage of the document containing 13 g's at a threshold of 1.5 on the log score. Twelve of the thirteen g's were detected as well as five false positives. The dark boxes show the detected g's, while the light boxes show the false positives. The dashed dark box shows the g that was missed. Some of the false positives occurred on the letter y which does in fact look like a g with an open top. Errors also occurred on the combination ap since the finish of the a and start of the p together look like an open g.

Figure 8.14 shows the detection performance for the letter g over the entire document using ROC (receiver operating characteristic) curves. The ROC curve shows

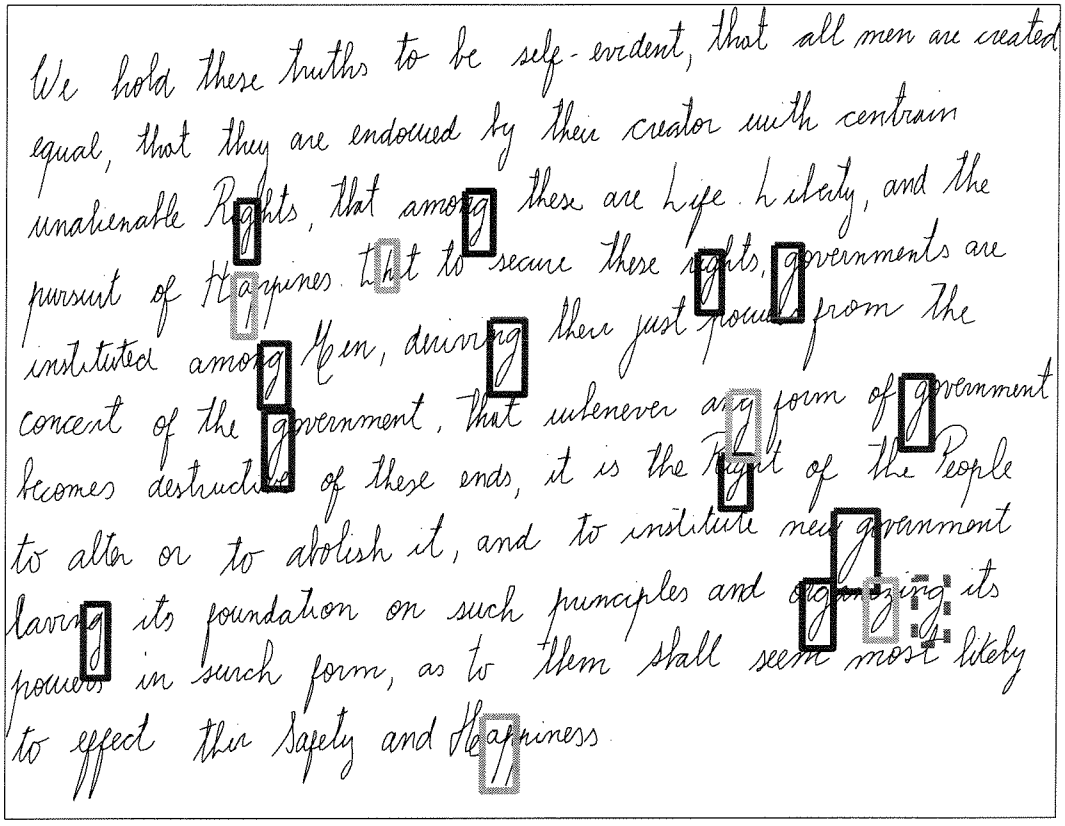


Figure 8.13: Detection of the letter g on a section of the Declaration of Independence. The dark boxes show correct hits, the light boxes show false positives, and the dashed boxes show misses.

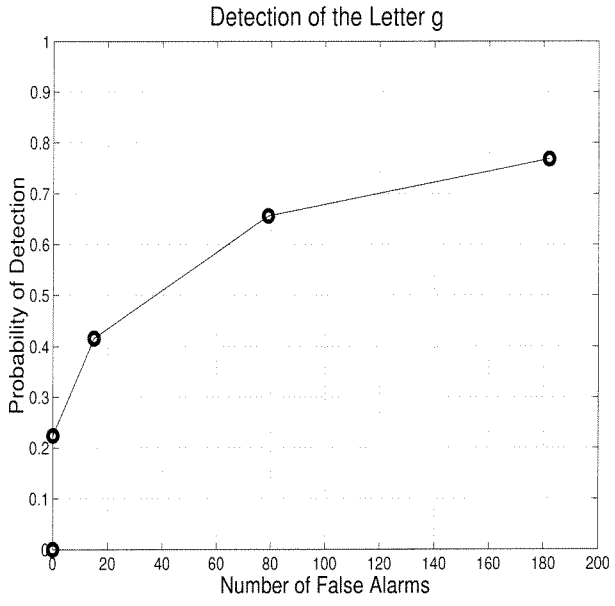


Figure 8.14: ROC performance over the Declaration of Independence for the letter g.

the tradeoff between the percentage of true g's detected (out of $\approx$ 125 opportunities) and the number of false positives.

The detection experiment was repeated for the letter t. Our model for the t consisted of nine parts: 38–46. Figure 8.15 shows the performance of the algorithm at a threshold of 4.0 on the same section of the Declaration of Independence shown in Figure 8.13. The performance over the entire document, which contains slightly less than 600 instances of the letter t, is shown in Figure 8.16. At a threshold of 4.0, the estimated probability of detection was 67% and the number of false alarms over all images was 100.

It turns out that from the two letters g and t, we can reliably detect the word *government* just by looking for a leading g and a terminal t in the same word. Over the entire document, this method will find all 11 instances of the word *government* with only four false positives: *object*, *Assent*, *appealed*, and *Great*. A more rigorous test on the length of the word or on the characters in the middle of the word would probably reject these false positives.

We have also attempted to detect the letter m, but these experiments were not as

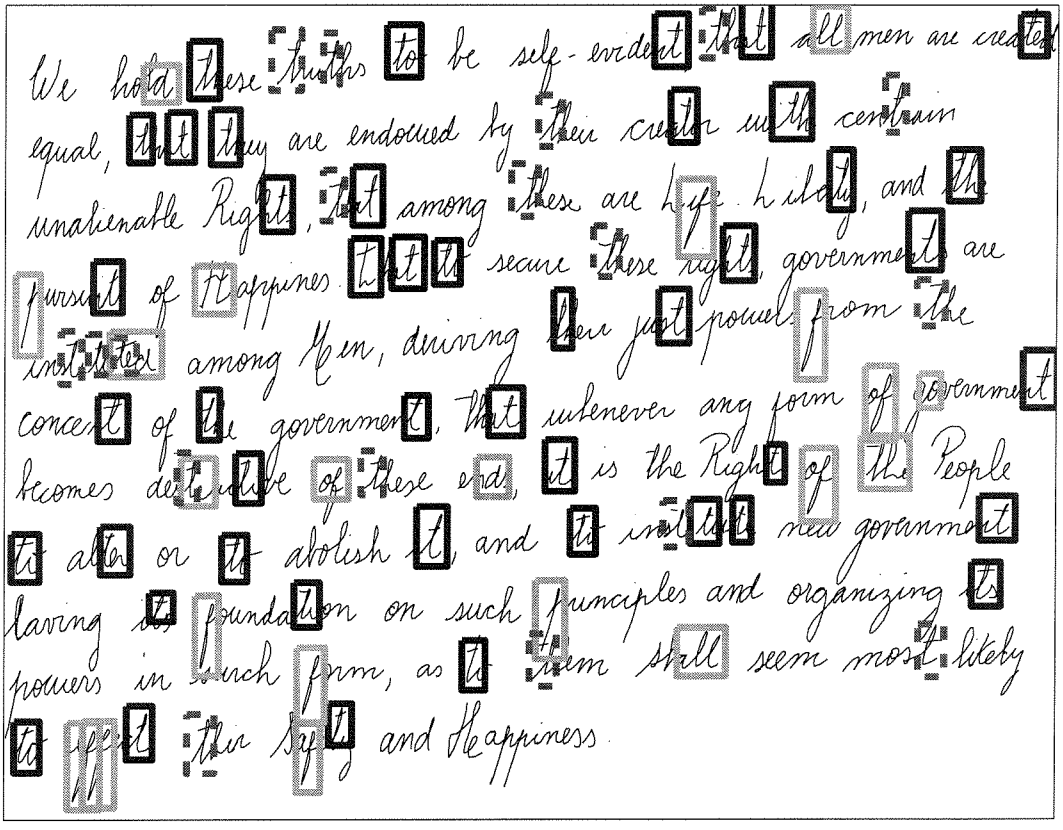


Figure 8.15: Detection of the letter *t* on a section of the Declaration of Independence. The dark boxes show correct hits, the light boxes show false positives, and the dashed boxes show misses.

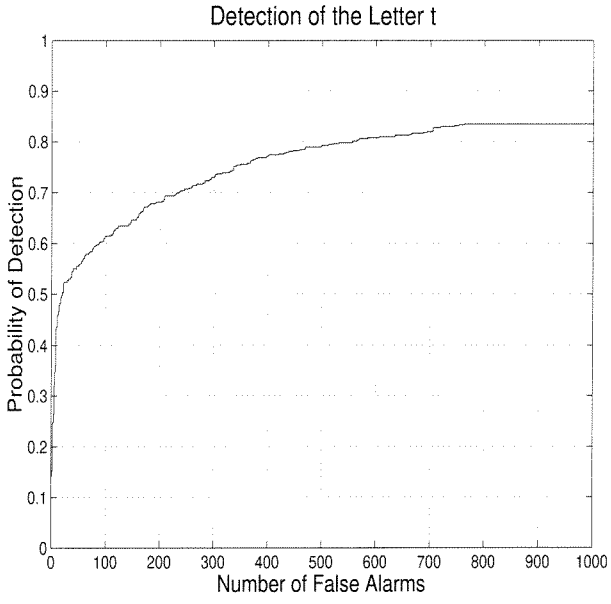


Figure 8.16: ROC performance over the Declaration of Independence for the letter *t*.

successful. There were two basic problems. First, our model specifies only the arrangement of keypoints, not the pen trajectory between them. For the *m* the positions of the keypoints alone are not descriptive enough so many false alarms occur. Second, the test handwriting shows significantly more vertical compression and slant than the training data. As a result, the uncertainty regions frequently do not include the correct object parts. Even when the hypothesis formation procedure works properly, the slanted hypotheses are not scored well because this type of variation doesn't appear in the training data. It isn't clear whether simply giving the algorithm a wider variety of training data would eliminate the problem. Most likely it will be necessary to move to affine-invariant shape descriptions, which can handle both vertical compression and slant.

8.3 Summary

In this chapter we have demonstrated a recognition approach based on probabilistic shape descriptions. The strategy uses local detectors to identify candidate part loca-

tions and then evaluates part groupings based on the overall shape of the configuration. For locating quasi-frontal views of faces in cluttered backgrounds and with occlusion, the algorithm achieves over 90% accuracy. We also demonstrated the generality of the method by automatically spotting keyword fragments in on-line handwriting data.

Chapter 9 Beyond Shape

The shape-based recognition algorithm presented in the previous three chapters was derived as a method to find the most object-like configuration of points from a set of candidate locations. However, there is no guarantee that first hard-detecting the features and then looking for the proper configuration is the best thing to do. In this chapter we reconsider from first principles the problem of detecting deformable object classes.

9.1 Object Class T_ρ Revisited

The object class T_ρ, which was introduced in Chapter 5, will serve as the focal point for our discussion. Recall that T_ρ was defined by perturbing the part positions of a nominal pattern T_0 by independent Gaussian errors having standard deviation ρ. For detecting instances from T_ρ in white noise, we showed that matched filtering and principal components analysis provided degraded performance relative to the optimal detector (derived in the next section).

A natural question to ask is: how well does the shape-based recognition algorithm work on T_ρ? To evaluate the performance, we used matched filtering to hard-detect candidate locations of the object parts. Candidate locations were grouped into hypotheses and evaluated using the shape likelihood ratio. The score from the best hypothesis in each image was used as our figure of merit. The resulting ROC curve is shown in Figure 9.1.

The performance of the shape algorithm appears to be slightly worse than the principal components method, especially at the lower end of the false alarm range. *However, the shape algorithm is looking for the pattern at different translations, rotations, and scales*[1], *while the other three methods are all looking at a single position,*

[1]In this chapter, we use the terms "rotation" and "scaling" in a specialized sense. By rotation we

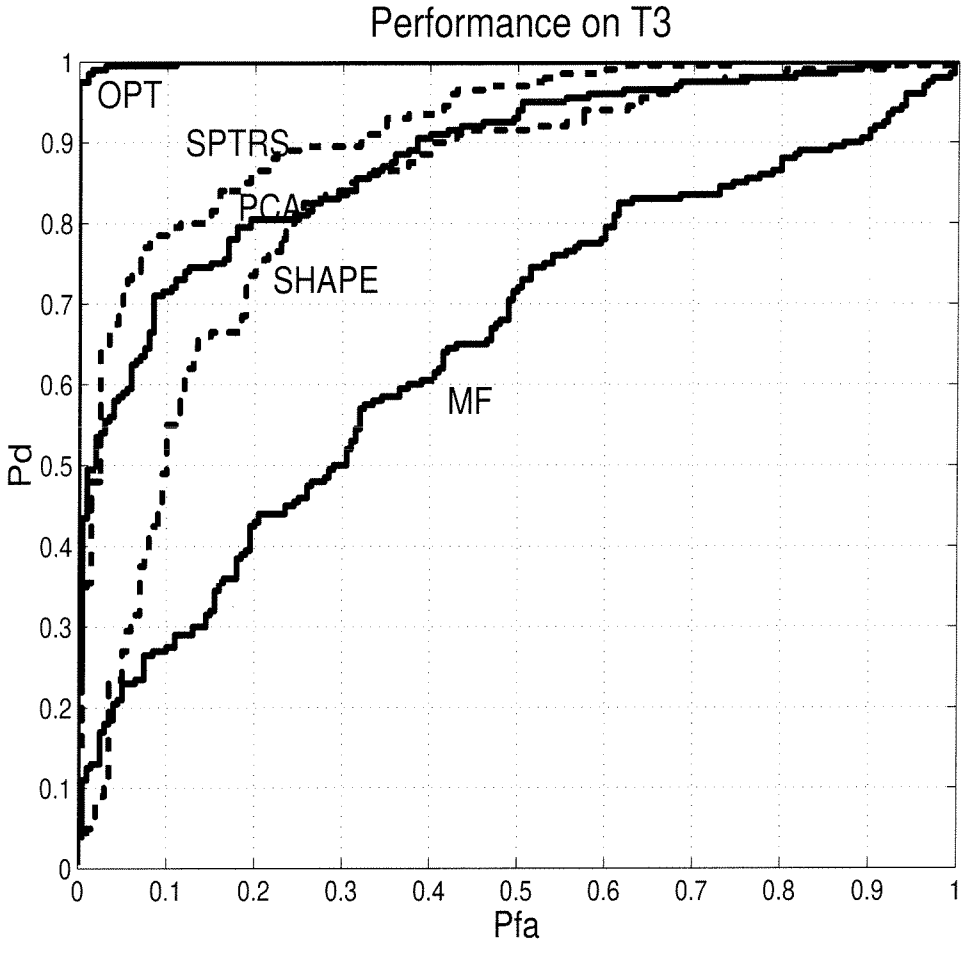


Figure 9.1: The performance of the shape method on the deformable object class T_3. For comparison, the performance of the optimal detector, principal components analysis, and the matched filter are also shown. The curve labeled SPTRS shows the performance of the shape method with constraints on the allowed translation, rotation, and scaling.

orientation, and scale. Clearly, the shape method must pay a penalty to provide TRS-invariant recognition. Consider that for TRS-invariant recognition, accidental arrangements of four detector false alarms in a square pattern at any position, orientation, and scale will be counted as an object false alarm. To verify this point, we did a second experiment using the shape method but also applying hard limits on the permitted amount of translation, rotation, and scaling. Based on training data, we used translation limits of ± 10 pixels, scaling limits of ± 15 pixels and rotation limits of $\pm 20°$. The score from best hypothesis satisfying these TRS constraints was selected as the figure of merit for each image. The resulting ROC curve is shown in Figure 9.1 with the label SPTRS (**S**hape **P**lus **T**ranslation, **R**otation, and **S**caling). Note that the SPTRS curve is still well below the optimal curve.

We believe that the optimal detector performance is probably not achievable by TRS-invariant algorithms. However, we have found that the following simple TRS-invariant algorithm yields good performance. The algorithm finds the maximum response over the image from the part detector (matched filter) and uses this value as the figure of merit. Performance of the maximum part response detector is shown in Figure 9.2. The *optimal* performance for TRS-invariant algorithms must lie somewhere between this curve and the (non-invariant) optimal detector curve. The maximum part response detector does not exploit the full signal energy since it is only looking for one part out of four causing a net 6 dB loss in SNR. From the initial 18 dB SNR, we are left with 12 dB, which is still good enough to provide reasonable performance.

This example clearly shows that hard-detecting the object parts is suboptimal. The shape of this object is not particularly distinctive, so it is easy to find detector false alarms that accidentally appear in the same arrangement. In this case, shape alone does not provide good performance. In the next section, we will derive from first principles the optimal non-invariant detector OPT for this problem. It will turn out that OPT relies on both the spatial configuration *and* the matched filter response

---

mean that the overall square pattern (configuration of the parts) can rotate, but the individual parts maintain their nominal orientation. Similarly, by scale we mean the square pattern can be larger or smaller, but the individual parts retain their nominal scale. We have made this simplification to insure that the matched filter is the optimal part detector (to eliminate the question of whether performance is degraded due to a suboptimal part detection scheme or other factors).

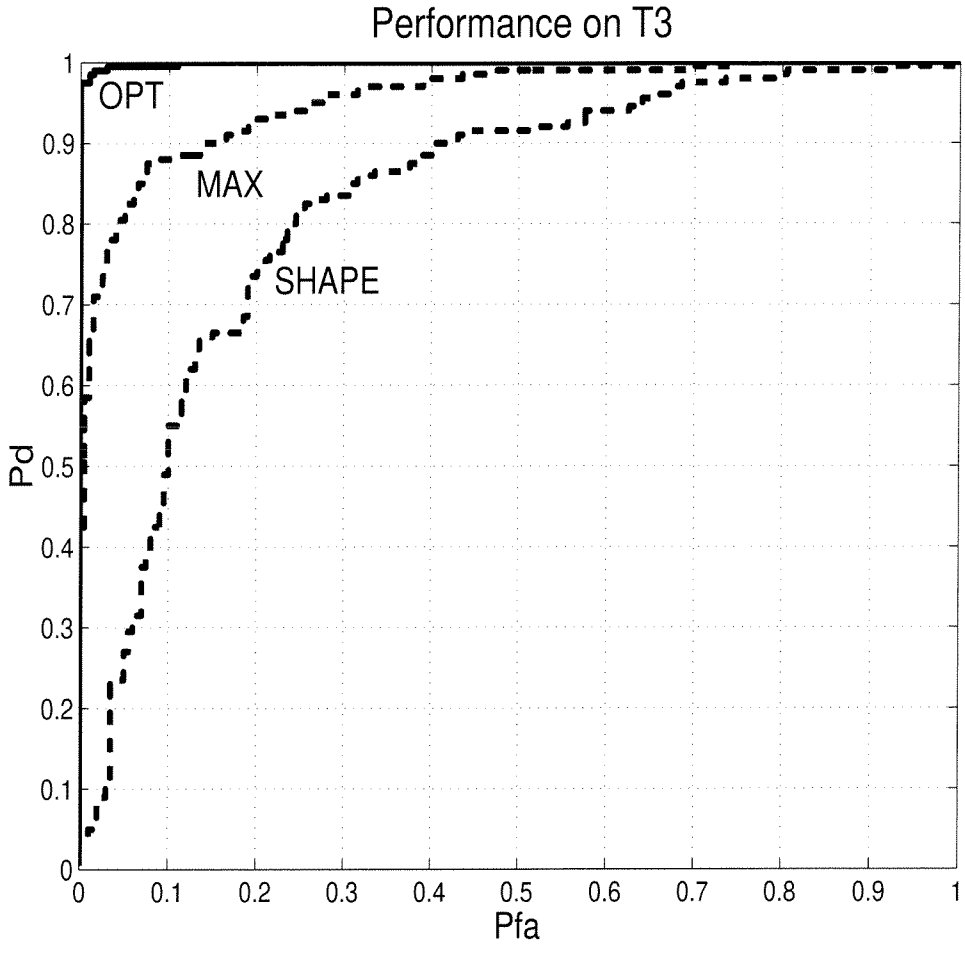


Figure 9.2: The performance obtained using the maximum response over the image from the part detector (MAX) is better than the shape method (SHAPE). Since the maximum response is TRS-invariant, we know that the *optimal* TRS-invariant performance must lie somewhere between this curve and the optimal (non-invariant) detector curve (OPT).

values to make the best possible decision.

9.2 Derivation of the Optimal Detector

The basic problem can be stated as follows: given an image $\mathcal{I}$ determine whether the image contains an instance from T_ρ (hypothesis ω_1) or whether the image is noise-only (hypothesis ω_2). In Chapter 7 we proposed a two-step solution to this problem: (1) apply feature detectors to the image in order to identify candidate locations for each of the object parts and (2) given the candidate locations, find the set of candidates with the most object-like spatial configuration. However, there is nothing to say that first hard-detecting candidate object parts is the right thing to do. In this section, we will directly derive the optimal detector starting from the pixel image $\mathcal{I}$.

The optimal decision statistic is given by the likelihood ratio

$$\Lambda = \frac{p(\mathcal{I}|\omega_1)}{p(\mathcal{I}|\omega_2)} \tag{9.1}$$

We can rewrite the numerator by conditioning on the spatial positions $\mathbf{X}$ of the object parts ($\mathbf{X}$ is as in Equation 6.1). Hence,

$$\Lambda = \frac{\sum_{\mathbf{X}} p(\mathcal{I}|\mathbf{X}, \omega_1) \cdot p(\mathbf{X}|\omega_1)}{p(\mathcal{I}|\omega_2)} \tag{9.2}$$

where the summation goes over all possible configurations of the object parts. Substituting for the class-conditional densities, we have

$$\Lambda = \frac{\sum_{\mathbf{X}} \mathcal{N}(\mathcal{I};\, \boldsymbol{\mu}_{\mathbf{X}}, \sigma^2\mathbf{I}) \cdot p(\mathbf{X})}{\mathcal{N}(\mathcal{I};\, \underline{\mathbf{0}}, \sigma^2\mathbf{I})} \tag{9.3}$$

where $\boldsymbol{\mu}_{\mathbf{X}}$ is the object with parts positioned at $\mathbf{X}$. Expanding the Gaussian densities and combining terms yields:

$$\Lambda = \sum_{\mathbf{X}} \exp\left(\frac{\boldsymbol{\mu}_{\mathbf{X}}^T \mathcal{I}}{\sigma^2} - \frac{\boldsymbol{\mu}_{\mathbf{X}}^T \boldsymbol{\mu}_{\mathbf{X}}}{2\sigma^2}\right) \cdot p(\mathbf{X})$$

$$
\begin{aligned}
&= \sum_{\mathbf{X}} \exp\left(-\frac{\boldsymbol{\mu}_{\mathbf{X}}^T \boldsymbol{\mu}_{\mathbf{X}}}{2\sigma^2}\right) \cdot \exp\left(\frac{\boldsymbol{\mu}_{\mathbf{X}}^T \mathcal{I}}{\sigma^2}\right) \cdot p(\mathbf{X}) \\
&= \sum_{\mathbf{X}} \exp(-\frac{E}{2\sigma^2}) \exp\left(\frac{\boldsymbol{\mu}_{\mathbf{X}}^T \mathcal{I}}{\sigma^2}\right) \cdot p(\mathbf{X}) \\
&= c \cdot \sum_{\mathbf{X}} \exp\left(\frac{\boldsymbol{\mu}_{\mathbf{X}}^T \mathcal{I}}{\sigma^2}\right) \cdot p(\mathbf{X}) \qquad (9.4)
\end{aligned}
$$

The third line follows from the second since $\boldsymbol{\mu}_{\mathbf{X}}^T \boldsymbol{\mu}_{\mathbf{X}}$ is just the energy E in the target signal, which is constant independent of $\mathbf{X}$ provided the parts do not overlap. Note that we could have written the final formula directly using the results in Section 2.7. For each $\mathbf{X}$ value, the object signal is known exactly, so the entire object class consists of a number of subclasses (one for each $\mathbf{X}$ value). Further, each subclass is represented by the signal $\boldsymbol{\mu}_{\mathbf{X}}$ which occurs with probability $p(\mathbf{X})$.

The quantity $\boldsymbol{\mu}_{\mathbf{X}}^T \mathcal{I}$ can be rewritten in terms of matched filter response images. Let A_i be the response image obtained by correlating part i with the image $\mathcal{I}$ and define $\tilde{A}_i = A_i/\sigma^2$. Then,

$$
\boldsymbol{\mu}_{\mathbf{X}}^T \mathcal{I} = \sum_{i=1}^{N} \tilde{A}_i(x_i, y_i) \qquad (9.5)
$$

Substituting into the likelihood equation, we have

$$
\begin{aligned}
\Lambda &= c \cdot \sum_{\mathbf{X}} \exp\left(\sum_{i=1}^{N} \tilde{A}_i(x_i, y_i)\right) \cdot p(\mathbf{X}) \\
&= c \cdot \sum_{\mathbf{X}} \left[\prod_{i=1}^{N} \exp\left(\tilde{A}_i(x_i, y_i)\right)\right] \cdot p(\mathbf{X}) \qquad (9.6)
\end{aligned}
$$

The constant c does not affect the form of the decision rule, so we will omit it from our subsequent equations.

9.2.1 Independent Part Positions

If the part positions are independent, $p(\mathbf{X})$ can also be expressed as a product

$$
p(\mathbf{X}) = \prod_{i=1}^{N} p_i(x_i, y_i) \qquad (9.7)
$$

Thus, we have

$$\begin{aligned}
\Lambda &= \sum_{\mathbf{X}} \left[\prod_{i=1}^{N} \exp\left(\tilde{A}_i(x_i, y_i)\right) p_i(x_i, y_i) \right] \\
&= \sum_{\mathbf{X}} \left[\prod_{i=1}^{N} \exp\left(\tilde{A}_i(x_i, y_i) + \log p_i(x_i, y_i)\right) \right] \\
&= \prod_{i=1}^{N} \left[\sum_{(x_i, y_i)} \exp\left(\tilde{A}_i(x_i, y_i) + \log p_i(x_i, y_i)\right) \right] \qquad (9.8)
\end{aligned}$$

For each object part, we compute the correlation response image normalized by σ^2. To this image, we add the log probability that the part will occur at a given spatial position, take the exponential, and sum over the whole image. This process is repeated for each object part. Finally, the product of scores over all the object parts yields the likelihood ratio.

This detector is optimal for T_ρ since we assumed the part positions were perturbed by independent Gaussian errors. Note, however, that the detector is not invariant to translation, rotation, and scaling since the term $p_i(x_i, y_i)$ includes information about the absolute coordinates of the parts.

9.2.2 Jointly Distributed Part Positions

If the part positions are *not independent*, we must introduce an approximation since summing over all *combinations* of part positions as in Equation 9.4 is infeasible. The basic idea (similar to the winner-take-all approximation made when we discussed subclasses in Section 2.7) is to assume that the summation is dominated by one term corresponding to a specific combination $\mathbf{X_0}$ of the part positions. With this assumption, we have

$$\begin{aligned}
\Lambda_0 &= \exp\left(\frac{\boldsymbol{\mu}_{\mathbf{X_0}}^T \mathcal{I}}{\sigma^2}\right) \cdot p(\mathbf{X_0}) \\
&= \exp\left(\sum_{i=1}^{N} \tilde{A}_i(x_{0i}, y_{0i})\right) \cdot p(\mathbf{X_0}) \\
\log \Lambda_{\mathcal{S}} &= \left(\sum_{i=1}^{N} \tilde{A}_i(x_{0i}, y_{0i})\right) + \log p(\mathbf{X_0}) \qquad (9.9)
\end{aligned}$$

The strategy now is to find a set of part positions such that the matched filter responses are high and the overall configuration of the parts is consistent with $p(\mathbf{X}|\omega_1)$. Again, the resulting detector is not invariant to translation, rotation, and scaling.

9.3 TRS-invariant Approximation to the Optimal Detector

The approximate log-likelihood ratio given in Equation 9.9 can readily be interpreted as a combination of two terms: the first term, $\sum \tilde{A}_i$, measures how well the hypothesized parts in the image match the actual model parts, while the second term, $p(\mathbf{X_0})$, measures how well the hypothesized spatial arrangement matches the ideal model arrangement. The second term, the configuration match, is specified as a probability density over the absolute coordinates of the parts, which in practice is not useful since (a) there is no way to know or estimate this density and (b) this formulation does not provide TRS-invariance.

We can make use of the machinery developed in Chapters 6 and 7 to write down a TRS-invariant detector that closely follows the form of Equation 9.9. Specifically, we replace the term $p(\mathbf{X_0})$ by the shape likelihood ratio defined in Equation 7.14. Instead of working with figure space variables $\mathbf{X_0}$, we will work with the corresponding shape variables $\mathbf{U_0}$.

$$\log \Lambda_1 \;=\; \sum_{i=1}^{N} \tilde{A}_i(x_{0i}, y_{0i}) \;+\; K \cdot \log \frac{P_U(\mathbf{U_0})}{Q_U(\mathbf{U_0})} \tag{9.10}$$

The shape likelihood ratio, rather than just $p_{\mathbf{U}}(\mathbf{U_0})$, is used in place of $p_{\mathbf{X}}(\mathbf{X_0})$ to provide invariance to the choice of baseline features. The likelihood ratio also assigns lower scores to configurations that have higher probabilities of accidental occurrence. The factor of K provides a weighted trade-off between the part match and shape match terms, since the units of measurement for the two terms will no longer agree. (The proper setting for this value can be estimated from training data).

An object hypothesis is now just a set of N coordinates specifying the (hypothes-

ized) spatial positions of the object parts. Any hypothesis can be assigned a score based on Equation 9.10. It is no longer the case that hypotheses must consist only of points corresponding to the best part matches. The trade-off between having the parts match well and having the shape match well may imply that it is better to accept a slightly worse part match in favor of a better shape match or vice versa.

We do not have a procedure for finding the hypothesis that optimizes $\log \Lambda_1$. One heuristic approach $\mathcal{A}_1$ is to identify candidate part locations as in the previous chapters (e.g., at maxima of the matched filter response) and combine these into hypotheses using the conditional search procedure. However, instead of discarding the response values, these should be summed and combined with the shape likelihood. In this approach, the emphasis is on finding the best part matches and accepting whatever spatial configuration occurs. There is no guarantee that the procedure will find the best hypothesis.

A second approach $\mathcal{A}_2$ is to insist on the best shape match and accept whatever part matches occur. This method is equivalent to using a rigid matched filter for the entire object, but applying it at multiple orientations and scales.

Finally, we outline a third approach $\mathcal{A}_3$ that intuitively seems appealing but has not been tested. Candidate part locations are identified as before in $\mathcal{A}_1$ at local maxima in the part response image. From pairs of candidate parts, the locations of the other parts are estimated to provide an initial hypothesis. (So far, this is equivalent to using a fixed-shape template anchored at the two baseline points). From the initial hypothesis, however, a gradient-style search would be employed to find local maximum of $\log \Lambda_1$. Individual part positions are pulled by two forces. One force tries to maximize the response value while the other force tries to improve the shape of the configuration. This approach is similar to several of the ad hoc methods for dealing with deformable objects discussed in Section 1.3.5.

Approach $\mathcal{A}_1$ was applied to the same data from T_3 used in the previous experiments. The ROC curve for $K = 2.5$ is shown in Figure 9.3 using the label `SLPPR` ("slippery") which is an acronym for **S**hape **L**ikelihood **P**lus **P**art **Response**. As expected, the performance is between that of the maximum response detector and

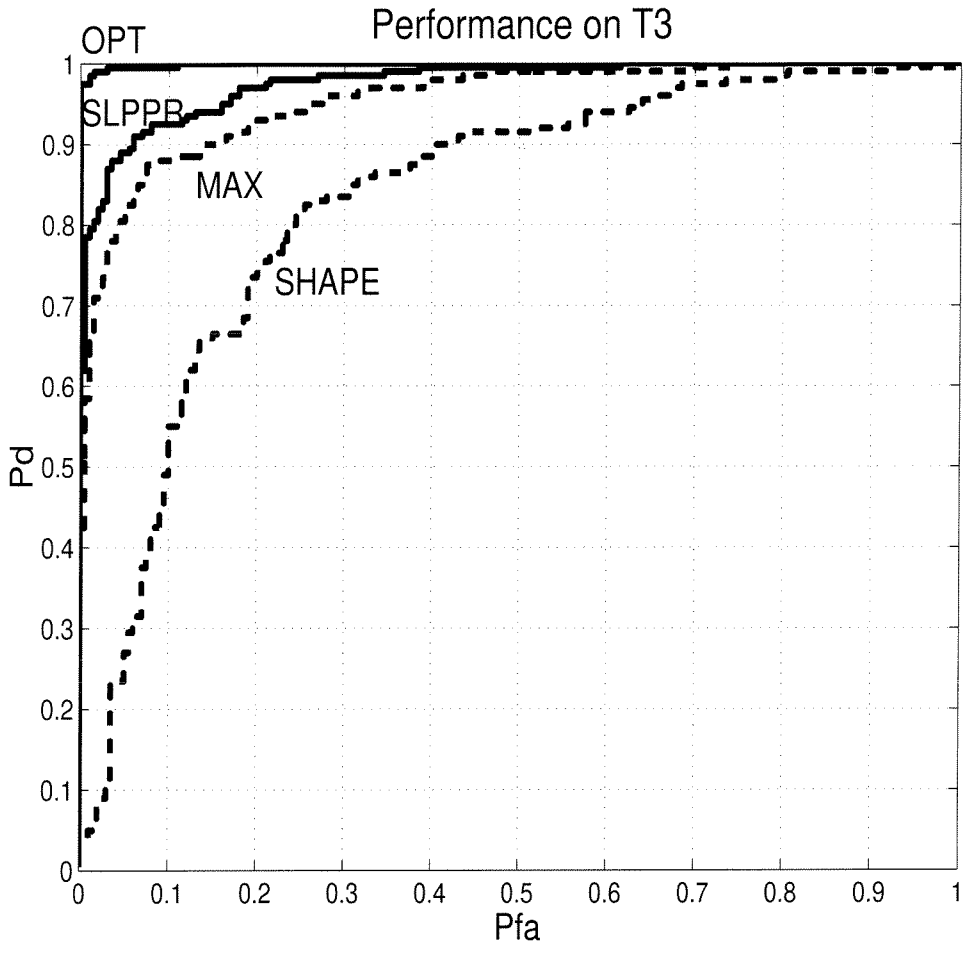


Figure 9.3: Curve SLPPR shows the performance obtained using approach $\mathcal{A}_1$ and $\log \Lambda_1$ on object class T_3. As expected, the performance is between the maximum response detector and the optimal non-invariant detector.

OPT, the optimal non-invariant detector. Values of K from 2.0 to 3.0 yielded nearly identical ROC curves indicating a low sensitivity to the precise setting.

9.4 Summary

In this chapter we have reconsidered from first principles the problem of detecting deformable object classes. The optimal detector for object class T_ρ was derived for the case of independent part positions. When the part positions are jointly distributed the optimal detector is too complicated to evaluate, but it can be approximated

using a winner-take-all simplification. In both cases, the detector is composed of two terms: the first term measures how well the hypothesized parts in the image match the actual model parts, while the second term measures how well the hypothesized spatial arrangement matches the ideal model arrangement.

The configuration match is specified in terms of the absolute positions of the object parts so the optimal detector cannot be used in practice. However, using machinery developed in Chapters 6 and 7, we were able to write an expression that closely follows the form of Equation 9.9, but only exploits the *shape* of the configuration. The resulting criteria combines the part match with shape match and is invariant to translation, rotation, and scaling. We have named the criteria SLPPR for **S**hape **L**ikelihood **P**lus **P**art **R**esponse. The part match is specified in terms of matched filter outputs, but the generalization to parts with inherent variability is straightforward using principal components analysis on each of the parts.

Although we do not have a procedure for finding the hypothesis that maximizes SLPPR, a heuristic approach $\mathcal{A}_1$ worked quite well. In this approach, candidate parts are identified and grouped into hypotheses as in the shape-only method, but, in addition, the response values (part matches) are retained and combined with the shape likelihood. This approach provided excellent ROC performance on T_3.

Chapter 10 Conclusion

10.1 Summary

Visual recognition of object classes is a problem that occurs across many domains including medical and biological imaging, astronomy, planetary geology, military target recognition, product inspection, user authentication, content-based retrieval, etc. One of the fundamental challenges in building an object recognition system is that the *algorithms must be sensitive to differences between different object classes, yet insensitive to the differences within the same object class.* Therefore, it is important to develop good models for the variations in appearance of instances from within the same class.

For object classes consisting of localized patterns that have limited degrees of freedom, principal components analysis (PCA) provides a useful model. With this approach an object class is represented as a linear combination of a small number of basis vectors. The basis vectors can be estimated from training examples using singular value decomposition (SVD). The vectors encode the directions of maximum variance within the training set and hopefully within the object class. A stronger model can be obtained by specifying a probability distribution over the weighting coefficients used to combine the basis functions. Using this technique, we developed a trainable system for locating small volcanoes in homogeneous images of Venus collected by the Magellan spacecraft. Related techniques have been used by various researchers to recognize facial features [MP96], handwritten digits [SYD93], and a number of other localized patterns.

However, not all object classes are well- modeled by principal components analysis. For heterogeneous volcano images, the performance of the PCA algorithm breaks down markedly. This problem could potentially be fixed by pre-clustering the volcanoes into subclasses and then performing PCA on each subclass. However, for object classes

having instances that consist of characteristic parts with inherent variability arranged in a deformable spatial configuration, we demonstrated that principal components analysis is unable to adequately model the variation with a small number of basis functions. Thus, the main thrust of this thesis has been to develop techniques suitable for deformable object classes. Here we use "deformable" to mean that object parts appear in somewhat variable relative positions on the image plane. Thus, a human face which deforms based on different expressions of the owner is a deformable object class. A physically rigid object that is imaged from a range of viewpoints may also be considered to be a deformable object class since the relative positions of the object parts on the image plane vary. Similarly a set of different rigid physical objects from the same class (cars) constitute a deformable object class since the positions of object parts vary from model to model.

The approach we have proposed for recognizing deformable object classes is based on finding the appropriate object parts in the "correct" spatial configuration, where "correct" is used in a probabilistic sense. The allowed object deformations are modeled using shape statistics learned from examples. Here we use the term "shape" in the sense of Kendall and Bookstein to refer to properties of a set of labeled points that are invariant under a group of transformations such as translation, rotation (in the image plane), and scaling. By using a probabilistic shape model, we are able to encode which object variations are most probable.

An important probability density over shape was derived by Dryden and Mardia, who showed that for a Gaussian density in figure space, the induced density in shape space has a particular form known as the Dryden-Mardia density. This density has the advantage that any baseline pair of features can be used to transform to shape space, and joint densities over subsets of shape variables can be determined. Both of these properties are useful, because the local detectors that identify candidate locations for the object parts may fail to detect some of the actual parts. If one of the parts normally used to transform to shape space is missing, the calculations can still be carried through using a different baseline pair. Similarly, if other (non-baseline) parts are missing, the partial configuration can be checked for consistency with the proposed

object class.

To detect instances of an object class in an image, local feature detectors are used to identify candidate locations for parts of the object. The nature of the detectors is not important and may be application-dependent. Since local detectors are not perfectly reliable, the detector outputs are treated as candidate locations for the corresponding object parts. The pools of candidate locations are then combined into hypotheses about the object location. In forming hypotheses, the known spatial structure of the object class is used to limit the number of object hypotheses that must be evaluated. Finally, object hypotheses are scored based on the spatial configuration including a penalty for missing parts.

The technique we have proposed was demonstrated on two challenging recognition problems. For human face localization, we achieved performance above 90% on two sequences of quasi-frontal face images in cluttered scenes and with occlusion. The algorithm worked reliably despite complete occlusion of parts of the face (including each eye) and significantly outperformed a simple normalized cross-correlation strawman. Second, we showed that the shape method could be used to spot keyword fragments in on-line handwriting data. For this problem, we demonstrated that multiple objects could be found in a single image. Also, we showed that thresholding the hypothesis scores provided reliable rejection of objects not in the class.

Finally, we derived the optimal detector for object classes consisting of a deformable configuration of parts. When the part positions are perturbed independently, the optimal detector can be simplified to a form that is easily evaluated. For co-varying parts, however, the optimal detector is too complicated to evaluate. A reasonable approximation can be obtained by applying winner-take-all selection to the set of possible spatial configurations. This pseudo-optimal detector consists of two terms: one term measures the part match while the other term measures the configuration match. The configuration match is specified in terms of a joint probability distribution over the part positions *in absolute image plane coordinates* so it is not directly usable. However, using the machinery developed through the later chapters of the thesis, we were able to write an expression that closely follows the form of the pseudo-optimal detector,

but only exploits the *shape* of the configuration. The resulting detector combines part match with shape match in a criteria we have called SLPPR (**S**hape **L**ikelihood **P**lus **P**art **R**esponse). The SLPPR critera is invariant to translation, rotation, and scale.

10.2 Limitations and Outlook

In this final section, we conclude the thesis with a discussion of limitations of the current work and directions for future work. The material is grouped into subsections by topic to improve the readability.

10.2.1 2-D/Affine/3-D

One obvious limitation is that we have only considered 2-D shape. Our method is invariant to general translations and scalings, but only to rotations *in the image plane.* For rotations in depth, the apparent 2-D shape of an object on the image plane will vary. This variability must currently be treated as deformability in the underlying object. For small rotations (approx $\pm 20°$), the shape densities can absorb the variability, but for handling larger rotations, affine or genuine 3-D shape methods are probably required (pasting together 2-D models from different views might also work).

We have derived the theoretical affine-invariant shape density induced by a general Gaussian figure space density [Leu95, BWLP96], but have yet to test the method experimentally. The affine shape of a point configuration is obtained by mapping three points to fixed reference positions; the positions of the remaining points represent affine shape. Generating hypotheses is more complicated in the affine framework since we must consider triples of points to perform the transformation to shape space. Also, there is an increased chance for false alarms to accidentally fall in an object-like arrangement. Despite these potential drawbacks, affine shape should be able to handle larger rotations in depth, especially for pseudo-planar objects such as human faces. Affine models are essential for robust, shape-based handwriting recognition since the writing slant and vertical compression of cursive writing can vary significantly.

For genuine 3-D shape, there are two versions of the problem. In the first version, a 3-D model probabilistic model is used, and 3-D measurements (e.g., from stereo) are assumed to be available. In the second version, a 3-D probabilistic model is also used, but only 2-D image plane measurements are assumed. In this version, if the 3-D figure space model is Gaussian, then given the viewing direction ψ and a weak perspective camera model, the image plane projections should follow a Dryden-Mardia shape density with parameters $\boldsymbol{\mu}_\psi$ and $\boldsymbol{\Sigma}_\psi$, which are related to the 3-D parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. This suggests that one should form hypotheses, infer the object pose from each hypothesis, and then compute a score based on the DM density $p_U(\mathbf{U}; \boldsymbol{\mu}_{\hat{\psi}}, \boldsymbol{\Sigma}_{\hat{\psi}})$.

10.2.2 Training Requirements and Part Definition

A second limitation in our approach is the need for labeled training examples. The object parts are currently selected manually based on intuition about which features are important and reliable. For example, in the face localization work, we decided somewhat arbitrarily that the eyes, nose, and mouth corners were the object parts. Unfortunately, automatically determining the part definitions appears to be a hard problem. Even given the part definitions, there is a considerable amount of work involved in building the shape model since the parts must be manually labeled in each training image. For image sequences it might be possible to use a bootstrapping method to automate this process. For example, the user might hand-click on the features in the first frame. Templates from this frame could be used to identify the features in the second frame. New templates would be selected from the second frame and applied to the third frame, and so on.

For the keyword spotting problem, we would ideally like the user to write the query word just once. However, a single example will not provide any information about the shape variation of the class. A possible solution might be for the user to prestore samples for each letter (or bigram pair) from the alphabet. After the user writes the query keyword, the computer would break the word into fragments it recognizes and then look for these fragments in the text. An alternative is to start with a simple

model of the variation, for example, the query word plus white noise on each of the part positions. This simple model could be used to identify candidate keywords, which the user would interactively verify or reject. A refined shape model could then be constructed from the verified candidates. Iterating this bootstrap procedure should eventually produce a good model with satisfactory detection/false alarm performance.

10.2.3 Hypothesis Generation

The hypothesis generation procedure based on conditional search worked well for a limited number of object parts. For more parts and more candidates (as in the handwriting experiments), a better method is needed. Currently, the search regions must be quite large to insure that legitimate object parts fall inside the regions, but this allows too many false alarms to be included in the search regions. By carefully choosing which pairs of parts are allowed to serve as the baseline, the search may be improved. The size of the search regions grows with distance from the baseline, so there may be an optimal ordering of the part pairs that minimizes the total search area or a similar criteria. For the handwriting experiments, we manually decided which feature pairs could be used as baseline but there may be an automated procedure that works better. It is interesting to note that only about four pairs are needed to insure a high probability of finding at least one correct baseline pair (on the object).

Another interesting idea for improving the search is to zipper the search regions. When a candidate point is found inside one region, the area where we look for other part candidates should be restricted based on the new information. This process is similar to closing a zipper. As we condition on more points, the uncertainty about the positions of the other points shrinks just as when more close a zipper, each successive tooth brings the others closer to a known position.

A different problem with our hypothesis generation procedure is that the method for allowing for missing parts creates many extra hypothesis. Consider, for example, an object consisting of five parts. If exactly one part falls inside each of the respective search region, the conditional search algorithm will generate $2^5 = 32$ hypotheses since

each part is can be present or missing. For more parts and more candidates, the problem is magnified. With the SLPPR algorithm, we will no longer need to explicitly state whether a part is present or missing. We will simply add up the filter responses at the hypothesized part positions. If a part is missing, it will get a lower score from the filter. A minimum value can be applied to the part match to keep certain bad types of occlusion (e.g., a white bicycle helmet) from ruining an otherwise good hypothesis.

We also acknowledge that there may be efficient algorithms in the general computer science or AI literature for finding the best hypotheses or finding hypotheses above a minimum score given the type of spatial constraints we have. Since hypothesis generation was not a fundamental theme of the thesis, we did not attempt to optimize this part of the algorithm.

For the new SLPPR algorithm, we have used a heuristic algorithm to find good hypotheses, but there is no guarantee it will find the best hypothesis. We outlined a method based on soft part detection and gradient descent that appears promising, but this has not been implemented or tested.

10.2.4 Temporal Structure

A promising area for future work is extending the shape and SLPPR methods to time-varying imagery. In the face localization experiments, we analyzed each frame independently searching over the whole image for faces. In practice, one should exploit the significant time correlation between frames. Having located the faces in several frames, we should be able to apply recursive motion estimation to search for the faces in very localized regions of the next frame. After a number of frames, a global search over the entire image could be initiated to find any new faces that have entered the image. Alternatively, the global search could be triggered by motion cues, e.g., if a new blob enters the image.

In the handwriting experiments, we have made limited use of the temporal information. Specifically, we used the time separation between pairs of features to eliminate bad baselines from consideration. Also, we used *time ordering* to prune the hypo-

thesis generation procedure. Time could also be used to extend the spatial search regions to search *volumes* using the Cartesian product of spatial uncertainty and time uncertainty. For some features such as crossing the letter "t" or dotting an "i", there are additional complications since the time delay depends on the length of the word (assuming the entire word is written and then crosses and dots are added). Along the same lines, it may be worthwhile to consider a joint probability density over shape and time to evaluate hypotheses.

10.2.5 Additional Information

Finally, we remark that there are several types of potentially useful information that are not handled well in the current framework. The points not included in a hypothesis can be as informative as the points included in a hypothesis. For example, in texture-free regions of the face such as the cheeks, we do not expect to see false alarms, but this type of information is not included in our scoring function. (Recall that in Chapter 7 we made an assumption that false alarm locations are independent of the position of any objects, which enabled us to simplify the scoring function. A consequence of that simplification is that any information from the false alarms is discarded.) Perhaps by defining a special object part called "bland region" we could work around this problem without changing the current framework.

A second piece of information is the agreement of local part attributes. For example, parts detected at a particular scale should only be grouped with parts at a similar scale. Similar considerations apply to orientation and polarity. The DC level and contrast of the parts should also be globally consistent. In our framework, these types of constraints can be handled by binning attributes and assigning different labels to different attribute ranges, but there should be a better method. For one approach, see [PL95].

Another limitation is that the current system uses only point configurations. For face localization, where the parts are image-based this does not create a problem. For handwriting, however, where the parts are keypoints, the keypoint configuration is

not descriptive enough to represent all the letters of the alphabet. Ideally we should define shape probabilities over entire pieces of the handwriting contour. Probabilistic contours could also be useful for encoding the boundary of the head. The Lanitis group has done some work along these lines using principal components analysis to encode the directions of maximum variation in a contour [LTCA95].

As defined in our approach, the configuration of a set of points is a *metric* concept. That is, the relative distances between points are meaningful attributes of the configuration. An alternative approach is to consider a configuration from the *relational* viewpoint. For example, the mouth is below the nose which is below the eyes, etc. These types of relational constraints could easily be incorporated in the hypothesis generation process, but do not seem to provide a way of scoring a hypothesis except in a binary fashion (acceptable or not acceptable).

10.3 Rapproachment

Despite the various caveats and limitations, we believe that modeling visual object classes as arrangements of parts in a spatially deformable configuration is a significant step forward. By learning probability distributions over the shape of a configuration, we are able to compactly model which variations in the object class are reasonable. The SLPPR method combining shape and local part responses represents a unification of the local methods developed in the first half of the thesis and the shape methods developed in the second half. Given the strong performance of both the local methods and the shape-only method, we believe this new algorithm has outstanding potential for a number of real-world visual recognition problems.

Bibliography

[AGW95] Yali Amit, Donald Geman, and Ken Wilder. "Recognizing Shapes from Simple Queries about Geometry." source unknown, July 1995.

[AK93a] O.E. Agazzi and S. Kuo. "Pseudo Two-Dimensional Hidden Markov Models for Document Recognition." *AT&T Technical Journal*, pages 60–72, 1993.

[AK93b] Y. Amit and A. Kong. "Graphical Templates for Image Matching." Technical Report 373, Department of Statistics, University of Chicago, August 1993.

[AM97] L. Asker and R. Maclin. "Large Ensembles as a Sequence of Classifiers." submitted to IJCAI, 1997.

[AS90] J.C. Aubele and E.N. Slyuta. "Small Domes on Venus: Characteristics and Origins." *Earth, Moon, and Planets*, 50/51:493–532, 1990.

[BB+94] E.J. Bellegarda, J.R. Bellegarda, et al. "A Fast Statistical Mixture Algorithm for On-Line Handwriting Recognition." *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(12):1227–1233, December 1994.

[BC94] Gilles Burel and Dominique Carel. "Detection and Localization of Faces on Digital Images." *Pattern Recognition Letters*, 15:963–967, 1994.

[BFP+94] M.C. Burl, U.M. Fayyad, P. Perona, P. Smyth, and M.P. Burl. "Automating the Hunt for Volcanoes on Venus." In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, Seattle, June 1994.

[BFPS94] M.C. Burl, U.M. Fayyad, P. Perona, and P. Smyth. "Automated Analysis of Radar Imagery of Venus: Handling Lack of Ground Truth." In *IEEE Intl. Conf. on Image Proc.*, volume III, pages 236–240, 1994.

[BH94] A. Baumberg and D. Hogg. "Learning Flexible Models from Image Sequences." *Lecture Notes in Computer Science*, 800:299–308, 1994.

[Bic91] M. Bichsel. *"Strategies of Robust Object Recognition for the Automatic Identification of Human Faces"*. Ph.D. thesis, ETH Zurich, 1991.

[BK94] C.B. Bose and S. Kuo. "Connected and Degraded Text Recognition Using Hidden Markov Model." *Pattern Recognition*, 27(10):1345–1363, 1994.

[BL75] E.M.L. Beale and R.J.A. Little. "Missing Values in Multivariate Analysis." *J. Royal Statistical Society*, 37:129–145, 1975.

[BLP95] M.C. Burl, T.K. Leung, and P. Perona. "Face Localization via Shape Statistics." In *Int Workshop on Automatic Face and Gesture Recognition*, 1995.

[BLP96] M.C. Burl, T.K. Leung, and P. Perona. "Recognition of Planar Object Classes." In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 1996.

[Boo84] F.L. Bookstein. "A Statistical Method for Biological Shape Comparison." *J. Theor. Biol.*, 107:475–520, 1984.

[Boo86] F.L. Bookstein. "Size and Shape Spaces for Landmark Data in Two Dimensions." *Statistical Science*, 1(2):181–242, 1986.

[Boo95] Fred L. Bookstein. "The Morphometric Synthesis for Landmarks and Edge-Elements in Images." *Terra Nova*, 7(4):393–407, 1995.

[BP93] R. Brunelli and T. Poggio. "Face Recognition: Features versus Templates." *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(10):1042–1052, October 1993.

[BP94] M. Bichsel and A.P. Pentland. "Human Face Recognition and the Face Image Sets Topology." *Computer Vision Graphics and Image Processing*, 59(2):254–261, Mar 1994.

[Bur] P.J. Burt. "A Multiscale Face Recognition System." unpublished.

[Bur95] M.C. Burl. "Detecting a Signal with Parameteric Variability." *Personal memo*, January 1995.

[BWLP96] M.C. Burl, M. Weber, T.K. Leung, and P. Perona. *From Segmentation to Interpretation and Back: Mathematical Methods in Computer Vision*, chapter "Recognition of Visual Object Classes." Springer, 1996.

[C+94] T.F. Cootes et al. "Use of Active Shape Models for Locating Structures in Medical Images." *Image and Vision Computing*, 12(6):355–365, July/Aug 1994.

[CK94] M. Chen and A. Kundu. "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network." *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):481–496, 1994.

[CKS95] M. Chen, A. Kundu, and S. Srihari. "Variable Duration Hidden Markov Model and Morphological Segmentation for Handwritten Word Recognition." *IEEE Trans. on Image Proc.*, 4(12):1675–1688, December 1995.

[CMVG96] G.E. Christensen, M.I. Miller, M.W. Vannier, and U. Grenander. "Individualizing Neuroanatomical Atlases using a Massively-Parallel Computer." *Computer*, 29(1), 1996.

[CT95] T.F. Cootes and C.J. Taylor. "Combining Point Distribution Models with Shape Models Based on Finite Element Analysis." *Image and Vision Computing*, 13(5):403–409, Jun 1995.

[CT96] T.F. Cootes and C.J. Taylor. "Locating Objects of Varying Shape Using Statistical Feature Detectors." In *European Conf. on Computer Vision*, pages 465–474, 1996.

[CW90] D.P. Chakraborty and L.H.L. Winter. "Free-Response Methodology: Alternate Analysis and a New Observer-Performance Experiment." *Radiology*, 174:873–881, 1990.

[CWS95] Rama Chellappa, Charles L. Wilson, and Saad Sirohey. "Human and Machine Recognition of Faces: A Survey." *Proceedings of the IEEE*, 83(5):705–740, May 1995.

[DC89] E. D. Dickmanns and Th. Christians. "Relative 3D-State Estimation for Autonomous Visual Guidance of Road Vehicles." In *Intelligent Autonomous Systems 2 (IAS-2)*, Amsterdam, 11-14 December 1989.

[DH73] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *J. Royal Stat. Soc. B*, 39:1–38, 1977.

[DM91] I.L. Dryden and K.V. Mardia. "General Shape Distributions in a Plane." *Adv. Appl. Prob.*, 23:259–276, 1991.

[DM92] E.D. Dickmanns and B.D. Mysliwetz. "Recursive 3-D Road and Relative Ego-state Recognition." *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:199–213, 1992.

[Dry95] I.L. Dryden. "General Shape Distributions", Oct 1995. *Private communication* (e-mail).

[Enc] Encyclopedia Britannica.

[FC90] M.K. Fleming and G.W. Cottrell. "Categorization of Faces using Unsupervised Feature Extraction." In *Proc. of IJCNN-90*, volume 2, 1990.

[FSN+95] M. Flickner, H Sawhney, W Niblack, et al. "Query by Image and Video Content – The QBIC System." *Computer*, 28(9):23–32, 1995.

[Fuk90] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 2nd edition, 1990.

[FvD84] J.D. Foley and A. van Dam. *Fundamentals of Interactive Computer Graphics.* Addison-Wesley, 1984.

[G+95] Hans Peter Graf et al. "Locating Faces and Facial Parts." In *International Workshop on Automatic Face- and Gesture-Recognition*, pages 41–46, 1995.

[GL89] G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins University Press, 2nd edition, 1989.

[Gol69] Goldstein. "False Alarm Regulation in Weibull and Log-Normal Clutter." *IEEE Trans. on AES*, 1969.

[Gre93] U. Grenander. "On the Shape of Plane Images." *SIAM Journal on Applied Mathematics*, 53(4):1072–1094, 1993.

[H+91] J.W. Head et al. "Venus Volcanic Centers and their Environmental Settings: Recent Data from Magellan." *American Geophysical Union Spring meeting abstracts*, EOS 72:175, 1991.

[HK91] Y. He and A. Kundu. "2-D Shape Classification Using Hidden Markov Model." *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(11):1172–1184, 1991.

[Hot33] H. Hotelling. "Analysis of a Complex of Statistical Variables into Principal Components." *J. Educ. Psych*, 24:417–441,498–520, 1933.

[HU87] D.P. Huttenlocher and S. Ullman. "Object Recognition Using Alignment." In *Int. Conf. on Comp. Vision*, pages 102–111, 1987.

[KA94] S. Kuo and O.E. Agazzi. "Keyword Spotting in Poorly Printed Documents Using Pseudo 2-D Hidden Markov Models." *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):842–848, 1994.

[Kan77] T. Kanade. "Computer Recognition of Human Faces." *Interdisciplinary Systems Research*, 47, 1977.

[KC94] G.E. Kopec and P.A. Chou. "Document Image Decoding Using Markov Source Models." *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(6):602–617, 1994.

[Ken84] D.G. Kendall. "Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces." *Bull. London Math Soc.*, 16:81–121, 1984.

[Ken89] D.G. Kendall. "A Survey of the Statistical Theory of Shape." *Statistical Science*, 4(2):87–120, 1989.

[KS+94] M.S. Kamel, H.C. Shen, et al. "Face Recognition using Perspective Invariant Features." *Pattern Recognition Letters*, 15(9):877–883, 1994.

[L+93] Martin Lades et al. "Distortion Invariant Object Recognition in the Dynamic Link Architecture." *IEEE Transactions on Computers*, 42:300–311, mar 1993.

[LB96] L.L. Lee and T. Berger. "Reliable On-Line Human Signature Verification Systems." *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(6):643–647, 1996.

[LBP95] T.K. Leung, M.C. Burl, and P. Perona. "Finding Faces in Cluttered Scenes." In *Intl. Conf. on Computer Vision*, Cambridge, MA, 1995.

[Leu95] T.K. Leung. "Affine Shape Statistics." Technical report, U.C. Berkeley, Nov 1995.

[LK93] H. Le and D.G. Kendall. "The Riemannian Structure of Euclidean Shape Spaces: A Novel Environment for Statistics." *The Annals of Statistics*, 21(3):1225–1271, 1993.

[LQP93] H-Y.S. Li, Y. Qiao, and D. Psaltis. "Optical Network for Real-time Face Recognition." *Applied Optics*, 32(26):5026–5035, Sept 1993.

[LTCA95] A. Lanitis, C.J. Taylor, T.F. Cootes, and T. Ahmed. "Automatic Interpretation of Human Faces and Hand Gestures Using Flexible Models." In *International Workshop on Automatic Face- and Gesture-Recognition*, pages 90–103, 1995.

[MD89] K.V. Mardia and I.L. Dryden. "Shape Distributions for Landmark Data." *Adv. Appl. Prob.*, 21:742–755, 1989.

[MGN] JPL NASA. *http://nssdc.gsfc.nasa.gov/planetary/mgncraft.gif.* Magellan Homepage.

[MN95] Hiroshi Murase and Shree Nayar. "Visual Learning and Recognition of 3-D Objects from Appearance." *Int J. of Comp. Vis.*, 14:5–24, 1995.

[MP95] B. Moghaddam and A. Pentland. "Maximum Likelihood Detection of Faces and Hands." In *Intl. Wkshp on Automatic Face and Gesture Recognition*, pages 122–128, Zurich, Switzerland, 1995.

[MP96] B. Moghaddam and A. Pentland. *"Probabilistic Visual Learning for Object Representation"*, chapter 5, pages 99–130. Oxford University Press, 1996.

[MR94] D. Madigan and A.E. Raftery. "Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occams Windwo." *Journal of the Amer. Stat. Assoc.*, 89(428):1535–1546, 1994.

[MZ92] J.L. Mundy and A. Zisserman. *"Geometric Invariance in Computer Vision"*. Artificial Intelligence. The MIT Press, 1992.

[Nor43] North. title unavailable. Technical report, RCA, 1943.

[OADD93] A.J. O'Toole, H. Abdi, K.A. Deffenbacher, and D.Valentin. "Low-Dimensional Representation of Faces in Higher Dimensions of the Face Space." *J. Opt. Soc. Am. A*, 10(3), 1993.

[OW72] T. Orchard and M.A. Woodbury. "A Missing Information Principle: Theory and Applications." In *Proc. 6th Berkeley Symp. on Math. Statist and Prob.*, volume 1, pages 697–715, 1972.

[PFJ+91] G.H. Pettengill, P.G. Ford, W.T.K. Johnson, R.K. Raney, and L.A. Soderblom. "Magellan: Radar Performance and Data Products." *Science*, 252:260–265, 1991.

[Pin85] A. Pinkus. *n-Widths in Approximation Theory.* Springer Verlag, 1985.

[PL94] Arthur R. Pope and David G. Lowe. "Modeling Positional Uncertainty in Object Recognition." Technical report, Department of Computer Science, University of British Columbia, 1994. Technical Report # 94-32.

[PL95] Arthur R. Pope and David G. Lowe. "Learning Feature Uncertainty Models for Object Recognition." In *IEEE International Symposium on Computer Vision*, 1995.

[PP96] R.W. Picard and A.P. Pentland. "Introduction to the Special Section on Digital Libraries: Representation and Retrieval." *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):769–853, Aug 1996.

[PPS96] A. Pentland, R.W. Picard, and S. Sclaroff. "Photobook - Content-based Manipulation of Image Databases." *Int. Journal of Computer Vision*, 18(3):233–254, Jun 1996.

[RBK95] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. "Human Face Detection in Visual Scenes." source unknown, July 1995.

[RH84] R.A. Redner and H.F.Walker. "Mixture Densities, Maximum Likelihood, and the EM Algorithm." *SIAM Review*, 26(2):195–239, 1984.

[RH95] I. Rigoutsos and R. Hummel. "A Bayesian Approach to Model-Matching with Geometric Hashing." *Computer Vision and Image Understanding*, 62(1):11–26, 1995.

[SBF+94] P. Smyth, M.C. Burl, U.M. Fayyad, P. Baldi, and P. Perona. "Inferring Ground Truth from Subjective Labelling of Venus Radar Images." In *Neural Info. Processing Systems*, 1994.

[SBF+95] P. Smyth, M.C. Burl, U.M. Fayyad, P. Perona, and P. Baldi. *Advances in Neural Information Processing Systems 7*, chapter "Inferring Ground Truth from Subjectively Labeled Images of Venus." Morgan Kaufman, 1995.

[SBFP94] P. Smyth, M.C. Burl, U.M. Fayyad, and P. Perona. "Knowledge Discovery in Large Image Databases: Dealing with Uncertainties in Ground Truth." In *Proc. of Knowledge Discovery in Databases*, 1994.

[SBFP95] P. Smyth, M.C. Burl, U.M. Fayyad, and P. Perona. *Advances in Knowledge Discovery and Data Mining*, chapter "Knowledge Discovery in Large Image Databases: Dealing with Uncertainties in Ground Truth." AAAI/MIT Press, Menlo Park, CA, 1995.

[Sha95] S. Shams. "Multiple Elastic Modules for Visual Pattern Recognition." *Neural Networks*, 8(9):1439–1456, 1995.

[SK87] L. Sirovich and M. Kirby. "Low Dimensional Procedure for the Characterization of Human Faces." *Journal of Optical Society of America*, 4(3):519–524, 1987.

[SYD93] P. Simard, Y.L.Cun, and J. Denker. "Efficient Pattern Recognition Using a New Transformation Distance." In *Advances in Neural Information Processing Systems 5*, pages 50–58, 1993.

[TP91] M. Turk and A. Pentland. "Eigenfaces for Recognition." *J. of Cognitive Neurosci.*, 3(1), 1991.

[UCI] Univ. Calif. Irvine Vision Group. *http://www.vision.uci.edu/*. UCI Vision Homepage.

[VA+94] D. Valentin, H. Abdi, et al. "Connectionist Models of Face Processing — A Survey." *Pattern Recognition*, 27(9):1209–1230, September 1994.

[Van68] H.L. Van Trees. *Detection, Estimation, and Modulation Theory:Part 1.* John Wiley and Sons, 1968.

[WF93] C.R. Wiles and M.R.B. Forshaw. "Recognition of Volcanoes on Venus using Correlation Methods." *Image and Vision Computing*, 11(4):188–196, 1993.

[Wil95] A. Wilson. *title unavailable.* Ph.D. thesis, Duke, 1995.

[WvdM93] L. Wiskott and C. von der Malsburg. "A Neural System for the Recognition of Partially Occluded Objects in Cluttered Scenes." *Int. J. of Pattern Recognition and Artificial Intelligence*, 7(4):935–948, 1993.

[YH94] G.Z. Yang and T.S. Huang. "Human Face Detection in a Complex Background." *Pattern Recognition*, 27(1):53–63, January 1994.

[Yui91] A.L. Yuille. "Deformable Templates for Face Recognition." *J. of Cognitive Neurosci.*, 3(1):59–70, 1991.

[YWP95] L. Yang, B.K. Widjaja, and R. Prasad. "Application of Hidden Markov Models for Signature Verification." *Pattern Recognition*, 28(2):161–170, 1995.