ICA FEATURE EXTRACTION AND SUPPORT VECTOR MACHINE IMAGE CLASSIFICATION

By

JEFF FORTUNA, B.Eng, M.Eng

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Doctor of Philosophy

McMaster University

April 2005

Your file Votre référence
ISBN: 0-494-04234-6
Our file Notre référence
ISBN: 0-494-04234-6

ICA Feature Extraction and Support Vector Machine Image Classification

DOCTOR OF PHILOSOPHY (2005) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: ICA Feature Extraction and Support Vector Machine Image Classification

AUTHOR: Jeff Fortuna
B.Eng, M.Eng

SUPERVISOR: David Capson

NUMBER OF PAGES: xii, 155

Abstract

This thesis presents a detailed examination of the use of Independent Component Analysis (ICA) for feature extraction and a support vector machine (SVM) for applications of image recognition. The performance of ICA as a feature extractor is compared against the benchmark of Principal Component Analysis (PCA). Given the intrinsic relationship between PCA and ICA, the theoretical implications of this relationship in the context of feature extraction is investigated in detail. The thesis outlines specific theoretical issues which motivate the need for a feature selection scheme with ICA when used with Euclidean distance classification. Experimental verification of the behavior of ICA with Euclidean distance classifiers is provided by pose and position measurement experiments under conditions of lighting variance and occlusion. It is shown that (provided that the features are selected in an appropriate way), ICA derived features are more discriminating than PCA. ICA's utility in object recognition under varying illumination is exemplified with databases of specular objects and faces. A new application for ICA is illustrated by using ICA derived filters for face recognition with the a multi-class support vector machine (SVM) classifier. The ICA filters function in a similar way to Laplacian of Gaussian (LoG) filters by providing a degree of lighting invariant recognition. However, they are tuned to the specific spatio-frequency and orientation characteristics of the face dataset. The application shows that the performance of the classifier is sensitive to the tuning of the filters. As such, the use of filters derived from the data by ICA provides comparable performance to LoG filters without the need for tuning.

Conceived as a method to further improve the classification of PCA and ICA derived features, a novel algorithm for improving support vector machine performance by the modification of such features derived from an image database is presented. Specifically, the modification is performed iteratively by adjusting the position of the support vectors in the linear feature space which are hypothesized to be outliers. Convergence is shown to occur when there were very few support vectors to modify. A new basis for the database is then computed from linear regression on the modified features. In this way, the SVM is used to both classify the dataset and derive a set of features which result in compact classes that provide maximum margin. This provides a simple and effective way of unifying the process of feature extraction and classification. The performance of the compact class SVM is demonstrated with a series of Gaussian mixture, object and face databases. It is shown that the compact classes which result from the use of the algorithm provide a significant improvement in the generalization ability of the SVM, by dramatically increasing the margin and decreasing the number of support vectors. For the case of image classification, the technique is particularly effective (in some cases resulting in the maximum achievable margin) illustrating that image datasets can be well described by compact classes.

Acknowledgements

I would like to thank Dr. David Capson, my supervisor, for his support and guidance with this research. I also thank my fellow researchers in the lab and everyone else who had patience with me while I completed this work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Definition

The general problem of pattern recognition is the task of classifying objects into different categories (classes). Objects in this context can be any measurements which need to be categorized, such as images, voltage waveforms from a sensor or data from financial reports. These objects are examples of patterns which are to be recognized. Each pattern must have an instantiation (which may or may not be unique) in the domain in which the classification is to take place. When a computer is used to perform pattern recognition, each pattern is typically represented as a number or set of numbers. This set is described as a feature vector, of the form:

$$\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$$

This feature vector could be defined as, for example, all of the measurements made on the object, such as the intensity values of every pixel in an image, or all of the voltage samples from an audio signal. The length of a feature vector n so defined provides a measure of the dimensionality of the raw pattern data. Each scalar value (x_1,x_2 etc.) is referred to as a feature. The feature space of the raw data often has a very large dimension, so it is usually prudent to re-represent the original measurements

in more compact form (a shorter feature vector). The process of selecting features from the raw measurements is one of feature selection or feature extraction. It is, in general, a problem of dimensionality reduction. The ultimate goal of feature selection/extraction is to find the minimum number of features required to capture the essential structure in the raw data. This minimum number of features is termed the intrinsic dimensionality of the data. This dimensionality reduction is accomplished by applying a transformation (linear or non-linear) to the the input data. In this thesis, the transformation process is referred to as feature extraction, except in a few specific cases where features are selected from the transformed data. These cases will be clear from the context.

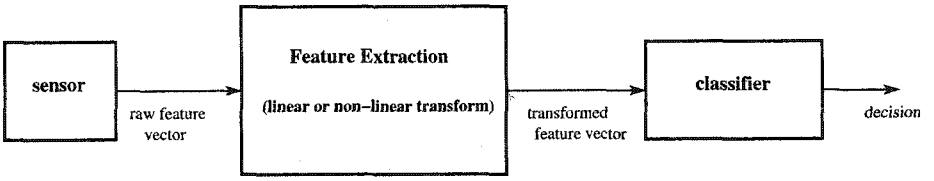


Figure 1.1: General Pattern Recognition Problem

This thesis is concerned with the recognition of objects which are instantiated as images gathered from an image sensor. Unfortunately, for this problem, the intrinsic dimensionality is impossible to know a-priori. As such, linear statistical techniques (such as Principal Component Analysis (PCA), Independent Component Analysis (ICA)), or non linear techniques (such as kernel methods) can be used to reduce the dimensionality of the data. The underlying hypothesis is that the transformed feature vectors have a dimensionality approaching the intrinsic dimensionality of the data. These techniques have the advantage that they use the statistical measures from the data to find features which are (hopefully) intrinsic to the object's important characteristics without requiring prior knowledge. Thus, fully automated general recognition systems can be constructed to recognize a wide variety of objects. A significant disadvantage, however, is that this feature extraction process is sub-optimal

with respect to any specific recognition problem, since it only uses statistical information about the input data and there is often much more information available. Since the goal of pattern recognition is ultimately to classify unknown objects into classes, any information about how the objects are grouped would allow for the extraction of features which aid in the grouping process. Commonly, information about the a-priori classification of some input data examples is available. It would then be useful to use this information when designing a feature extractor. This implies that in general, feature extraction and classification are coupled - the design or performance of a feature extractor has a significant impact on the design or performance of a classifier. To make the preceding ideas more concrete, a simple example will be described in the context of recognizing visual patterns.

Consider a sample database of 100 images of human faces. Within this database, there are 20 different people, with 5 facial images per person. Each person is assigned a label in the database, so we have 20 labels defining 20 classes. The recognition task that has been proposed for the purposes of the example is to assign the correct label to an unknown test person's face where five images (but not necessarily any the same as the test image) of that person exist in the database. The pattern recognition problem can then be defined. Feature extraction involves devising some scheme which produces a set of reference numbers (a vector) for each database face which characterize the face. These can be derived from the geometric arrangement of the pixels in the image, image statistics, or a wide variety or combination of techniques. Once the vectors are derived, classification involves grouping these vectors in such a way that the group of 5 vectors for the 5 database images of one person's face can be separated from the group for another person. Thus a group of 5 feature vectors for the same face make up a class and the classifier must be designed to separate the classes. Once this part of the design is done, classifying an unknown face can proceed by extracting a feature vector in the same way as was done on the database images. The class in the database which is most similar to the unknown feature vector determines the label assigned to the unknown face. The unknown face is then declared to be a member of

that label's class and hence that person.

1.2 Feature Extraction and the Human Visual System

Devising a scheme for extracting representative features of a visual image is a very difficult task. While the human visual system is generally quite adept at finding significant features of visual objects, it appears to employ a complex hierarchical scheme to accomplish this (example schema are proposed by Ullman [1] and Hoffman [2]). In this thesis we are primarily focused on low level feature representations which can be derived from image statistics. Much is known about the lower levels of vision in the visual cortex of humans (see Kandel, Schwartz and Jessell [3] for a detailed neurological study) and statistical learning theory plays an important role in understanding cortical interactions from a neuro-informational perspective.

Statistically based feature extraction (as motivated by neuroscience) would function by employing a network of neurons in the visual cortex which learn a "code" to represent the sensor input from the retinal ganglion cells. For example, a network implementation of PCA could be supplied input from retinal cells and the resulting output code would provide coefficients representing strength of response in the principal directions of the input. This code would reduce redundancy by decorrelating the retinal response. This type of coding scheme is commonly used in image coding for transmission and storage (compression) and is typically Discrete Cosine Transform or Karhunen Loeve Transform based. While this ensures efficiency from a storage point of view, the visual system must process as well as store images, making this a sub-optimal representation. In other words, decorrelated representations are not necessarily ideal when higher order structure is important in the extraction of information from images.

Since the pioneering work of Hubel and Wiesel (see [4] for a summary) over 20 years

from 1958 to the late 1970's, a large amount of attention has been placed on the neural mechanisms of human vision. Recent interest has been given to sparse representations in the primary visual cortex. Early attempts at suggesting cortical representations based on second order statistical redundancy (PCA) [5] have largely been supplanted by the consideration of higher order statistical dependencies. Olshausen and Field [6], [7] suggested that sparse coding (coded values exhibit a highly kurtotic distribution) may be the goal of the lower levels of the visual system. Bell and Sejnowski applied ICA to the problem of sparse coding in vision in [8] and found similar results to Olshausen and Field but through different means. ICA is a statistical technique which can separate signals which are assumed to have non-Gaussian distributions. Olshausen and Field enforced sparsity directly while Bell and Sejnowski utilized the information maximization idea to yield a highly kurtotic representation. The net result of this effort was to show that the use of ICA to derive spatial filters from image sets arrived at spatially localized and oriented Gabor-like (sine function modulated by a Gaussian function) filters which are similar to those found in the low levels of V1, the primary visual cortex (the first point in the visual pathway where the receptive fields are different from those of the retina). The work of Olshausen, Field, Bell and Sejnowski provide motivation in this thesis that perhaps ICA derived filters would offer some advantages as feature extractors for the purpose of pattern recognition.

1.3 Feature Extraction for Machine Vision with ICA

1.3.1 Feature Extraction Model

An image expressed as a random vector $\mathbf{x} \in \mathbb{R}^m$ by concatenating rows or columns together can be expressed as a linear superposition of n basis images in a matrix

$\mathbf{A} \in \mathbb{R}^{m \times n}$. The basis images $\mathbf{A}$ are arranged as n columns of length m:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1.1}$$

where $\mathbf{s} \in \mathbb{R}^n$ are coefficients for each basis image. $\mathbf{A}$ can also be considered as a mixing matrix which mixes n sources $\mathbf{s}$ together to form a mixture $\mathbf{x}$. The standard source separation problem (see Section 1.5.1) seeks a demixing matrix which recovers the original sources $\mathbf{s}$ from the mixture $\mathbf{x}$. The goal for feature extraction is to extract features from data by:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \text{ where } \mathbf{W} = \mathbf{A}^{-1}. \tag{1.2}$$

$\mathbf{W}$ is then a demixing matrix which seeks to recover the coefficients $\mathbf{y}$ for the basis images (mixing matrix) $\mathbf{A}$ which will be used as features to represent the original random image $\mathbf{x}$, as shown in Figure 1.2. In the source separation problem, the elements of $\mathbf{y}$ are estimates of the original sources.

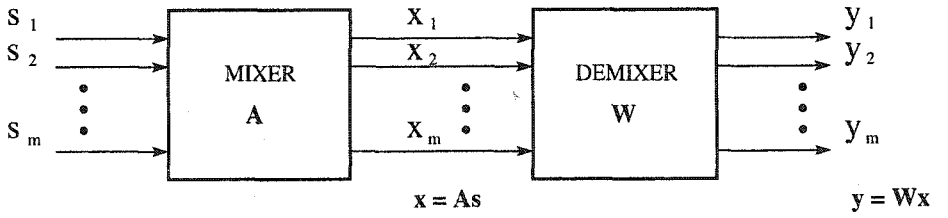


Figure 1.2: Feature Extraction Model

In some signal processing applications, it is possible to work with direct models of the stochastic process which generated $\mathbf{x}$. However, in the case of statistical feature extraction, it will be necessary to work with a finite number of instances of the random vector $\mathbf{x}$, since it is assumed that the stochastic process which generated the random vector is too complex to model directly. A batch mode representation of Equation 1.1 is thus defined:

$$\mathbf{X} = \mathbf{A}\mathbf{S} \tag{1.3}$$

where $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(N)] \in \mathbb{R}^{m \times N}$ and $\mathbf{S} = [\mathbf{s}(1), \mathbf{s}(2), \cdots, \mathbf{s}(N)] \in \mathbb{R}^{n \times N}$

are matrices of N instances of image data and coefficients respectively. In the next chapter, both PCA and ICA will be applied to determine $\mathbf{W}$ for the purpose of feature extraction.

1.3.2 Statistical Features from Higher Order Structure

An example of higher order structure in digital images is the position and intensity of edge features. The distribution of pixel intensities local to these features is highly super-Gaussian (intensities are mostly uniform around the edge and a small number of pixels have a large intensity, which defines the edge). This is a sparse arrangement, often modeled by a Laplacian distribution. For the purposes of this thesis, sparsity will refer to data representations which exhibit a strongly Laplacian distribution. If a sparse coding strategy is employed wherein the resulting code is forced to exhibit a Laplacian distribution, the strategy can discover higher order structure in images, which occur as a result of the oriented lines and edges of images. As a simple example of a coding scheme which enforces sparsity, consider the following optimization [7]:

$$\min_{\mathbf{\Phi}} \left[\min_{\mathbf{a}} \left[\mathbf{x} - \sum_i a_i \mathbf{\Phi}_i \right]^2 + \lambda \sum_i S(a_i) \right] \tag{1.4}$$

where $\mathbf{j}$ is a vector representation of an image (rows or columns appended into a vector), the elements a_i of $\mathbf{a}$ are coefficients of basis images $\mathbf{\Phi_i}$ (as vectors) and λ is a scalar for the penalty function on the coefficients S (the absolute value function is a useful choice for S and is approximated by the differentiable function $S(x) = \log(1 + x^2)$ in [7]). The first term computes the reconstruction error and the second enforces sparsity by penalizing large coefficients. Minimization takes place in two parts. The function is first minimized with respect to the a_i for each image, leaving the $\mathbf{\Phi}$ fixed. Second, over the presentation of many images, the function is minimized with respect to $\mathbf{\Phi}$. A basic premise that is exploited in this thesis is that the significant features in an image are exactly those that are selected from a sparse coding strategy

- namely oriented lines and edges. Higher order forms of redundancy are measured in terms of their higher order statistics. Gaussian distributions have all higher order statistical moments identically zero. So, for the purposes of this work, non-Gaussian is "interesting".

ICA methods use higher order moments to determine the non-Gaussian directions in data. This technique has been used to a limited extent as a method of feature extraction in machine vision, most commonly for the application of face recognition (see Section 1.5.4). Therefore, one important theme of this thesis is to undertake a careful and detailed examination of the use of ICA for feature extraction from both a theoretical and experimental point of view. In the process, it will be shown that there are some subtle but very important considerations that have largely been overlooked in the current literature. Specifically, due to the localized and oriented nature of basis functions that result when ICA is used as a linear generative model for images, some advantages may be expected in applications such as occluded and lighting variant object recognition. This hypothesis is tested in this thesis.

1.4 Classification

1.4.1 Support Vector Machines

Pattern recognition problems typically involve patterns which have some measure of variance across representative elements in a class. For example, images are often taken under a variety of lighting conditions. Additionally, the pose or viewpoint of an object is often a variant in the database. Recognizing objects under these variants represents a challenging problem. Support vector machines (SVM) provide an optimal decision hyper-plane by employing kernel learning, projecting the data into a high dimensional space. This hyper-plane is the separating boundary between objects of one class and objects of another which provides the largest margin (the distances from the closest points in the classes to the boundary). SVMs have been shown to be

very effective classifiers and provide the ability to generalize over imaging variants. The SVM provides a trade-off between a complex decision function giving a specific solution and a simple decision function giving a general solution.

Given that a set of (hopefully) representative features (feature vectors) has been established for a set of data, in the absence of an exact model for the generation of patterns in the features, the problem of pattern recognition becomes one of statistical inference. Historically, statistical inference was classically a problem of parametric inference — the estimation of parameters that define the statistical distributions (density estimation) or functional dependency (regression) underlying the data [9]. Detailed analysis of parametric statistical inference began with the work of Fisher (maximum likelihood) and for general inference with Gilivenko, Cantelli and Kolmogorov's study of the convergence of the empirical distribution to the actual distribution in the 1920's. These approaches are quite different. Parametric statistical inference is based on the belief that the process that generated the data is relatively well known. In this way, the parameters of the underlying generating function can be estimated. General statistical inference assumes that one does not know the underlying process that generated the data, but would like to infer an approximating function from the given examples.

The main shortcomings of the parametric approach were uncovered, beginning in the 1960's. Bellman introduced the notion of "the curse of dimensionality", stating that increasing the number of parameters requires an exponential increase in computation. Real world multi-dimensional problems would quickly become intractable and a small set of parameters would not accurately approximate the underlying process. Tukey also illustrated that many real problems could not be described by the classical statistical distributions. By the end of the 1960's, the notion of empirical risk minimization (ERM) arose — that a decision rule can be described that minimizes the number of training errors (empirical risk). It is this notion that led to the development of support vector machines as a methodology to classify patterns or perform regression by minimizing empirical risk.

The modern era of statistical learning theory with ERM began with the work of Vapnik and Chervonenkis in the early 70's. The basic idea is that a measure of the richness or flexibility of the function class describing the data (often called capacity) can be described by a quantity known as the VC (Vapnik–Chernovenkis) dimension [10]. Loosely speaking, the VC dimension, defined on a set of functions used to divide a set of points into classes, is the maximum number of points which can be correctly divided by the function set (said to be 'shattered' by the function set). For a VC dimension of n, there is at least one set of n points which can be shattered and not necessarily all sets of n points. For example for the function set of all lines in the plane, every set of 2 points can be shattered. Most sets of 3 points can also be shattered, but no sets of 4 points can be shattered. Thus the VC dimension is 3. While a detailed discussion of VC dimension is far beyond the scope of this thesis, the important feature of the work of Vapnik and Chervonenkis is that even for data with a high VC dimension, it can be shown that the use of high (or infinite) dimensional learning spaces are advantageous for classification. It is this principle which defines the approach of the support vector machine. An excellent summary of the concepts of VC dimension and SVMs is provided by Burges in [11]. Some further details on statistical learning theory will be provided in Chapter 3.

For the purposes of this thesis, the process of statistical classification will be examined in the context of the support vector machine (SVM) and statistical learning theory. Initially, simple classifiers such as minimum Euclidean distance will be examined, but the shortcomings of this approach, particularly when trying to examine ICA, become apparent immediately. SVM and kernel learning form a general group of classification techniques that rely on very high dimensional representations of data. While a general rule in pattern recognition is to keep the dimensionality of data small, statistical learning theory shows that it is necessary to exceed the VC dimension of the data in order to ensure it is separable. SVM classification has emerged as a preeminent statistical methodology for recognizing objects and as such has been examined in great detail for a wide variety of applications.

1.4.2 ICA with Support Vector Machine Classification

It is relatively common in the literature to apply PCA subspace feature extraction and then use a support vector machine for classification (for example [12] for face recognition). Due to the relatively recent use of ICA for feature extraction, there are fewer examples of ICA and SVMs. One representative example was Doermann, Qi and DeMenthon [13], in which ICA and a SVM were used for face detection. They derived the mixing matrix by minimizing the Kullback Leibler (KL) divergence between the source signal and its estimate. The ICA features were then simply used as input to the SVM. A very different application was illustrated in [14]. In this work the authors attempt to match human subjective ratings for scenic beauty estimation to help manage forest resources for the United States Forest Service. Interestingly, ICA is claimed to outperform PCA in this application, however no claim is given as to why this might be so. Another interesting application was extracting features and classifying forward-looking infrared (FLIR) imagery for target classification [15]. Here, ICA for feature extraction was compared against manually derived feature templates. ICA was used to derive the templates which are formed from the columns of the mixing matrix.

This thesis uses ICA and SVM classification for the application of face recognition. However, the application was not simply a marriage of the two techniques. Specifically, ICA derived filters are applied in place of LoG filters to provide a measure of lighting invariance. The tuning of SVM kernel radial basis function widths is examined in the context of the additional tuning of LoG filters. This new application of ICA eliminates the need for tuning LoG filters with respect to the SVM kernel, since ICA filters are derived from the spatio-frequency characteristics of the data. This experimental work is described in Chapter 5.

1.4.3 Improvements to Feature Extraction for Support Vector Machines

Classifying objects into logical groups fits into a hierarchical model in the human visual system. However, classification and feature extraction are also somewhat unified in the sense that the human visual system creates the objects that we see based on a fairly rigid set of rules which allow us to logically characterize objects. Optical illusions such as the two faces / vase image (see Hoffman [2] for this and other examples) illustrate that simple areas of a homogeneous shade together with its boundary (features) define objects which are constructed by a visual system trying to classify the objects as two equally plausible results. There is strong evidence that the human visual system moves up the hierarchy from low level feature extraction to high level recognition and back down again many times as it tries to reconcile significant features with logically consistent objects as is evidenced by the multiple feedback paths between the hierarchical regions of the human visual cortex (Candle et. al. [3] provides a reference for the current knowledge of the major visual pathways).

In light of this knowledge of the human system in which feature extraction and recognition are integrated, some examination of a technique which would facilitate this would seem to be in order. There has been very little literature in this regard with respect to statistical learning techniques. In this work, a unique proposal is made for an algorithm which integrates statistical feature extraction and SVM classification into a technique which provides a step in the direction of unifying these two parts of a recognition task.

Attempts to link feature extraction and support vector machines are currently almost absent in the literature (see [16] for one of the few examples). One other notable example provides a new description of data by a modification of the SVM which enforces a spherical decision boundary [17]. Data that can be scaled to fit this description can then be classified more robustly [18]. The concept of scaling data to fit the decision boundary was employed in this thesis as the basis for a new

approach that links the feature extraction and the SVM classification process. The proposed algorithm uses the support vectors to modify the principal and independent component data representations. Modification of the bases is used to improve the generalization of the classifier. Imaging variants are considered to increase the intra-class variance. However, the class distributions of image feature sets are strongly non-Gaussian, so the increase in variance corresponds to an increase in the number of support vectors necessary to represent the classes. The proposed algorithm adjusts the positions of the support vectors and recalculates the basis vectors thus providing a measure of invariance to the features. This thesis takes the idea further, showing how a new feature set which has different characteristics of both ICA and PCA (although closer in character to ICA) can be extracted from the SVM optimization process. Chapter 6 provides the details of this process.

1.5 Applications

1.5.1 ICA for Feature Extraction and Blind Source Separation

The use of ICA as a feature extraction technique for pattern recognition applications was motivated by the use of ICA to solve the blind source separation problem. This problem, described with Equation (1.1), was first examined in detail in the mid 80's by Herault. Often called the "cocktail party" problem, it was conceptualized as a method for separating individual conversations from a mixture of talkers in a room. In general, problems of this form are considered to be ones of Blind Source (Signal) Separation (BSS). The filtering operation is said to be "blind" since no information is directly available about the original sources nor the mixing matrix. Many techniques have been applied to address this problem. The basic methodology is to gather statistics about the observed mixtures and extract the mixing matrix. The first solutions to this problem, accredited to [19], employed higher order cumulants (beyond second

order). Early use of this methodology was provided by Cardoso [20], [21]. Temporal correlations of data were also exploited to demix signals using only second order statistics [22]. Since then, the blind source separation problem and a linear solution which exploits higher order statistics has been couched in the context of Independent Component Analysis (ICA) [23]. A popular application for ICA for blind source separation in recent years has been in EEG, MEG and ECG analysis. Due to the spatial arrangement of the sensor array and the validity of the instantaneous mixing process of EEG and MEG signals (which have most energy below 1 kHz), ICA is a reasonable technique for the separation of these signals. There is a significant amount of published literature for these applications including [24] and [25]. Schechner, Shamir and Kiryati [26] used ICA in an intuitive way as a blind source separation technique to decorrelate transparent layers of images (for example, looking out of a window, both the outside world and the semi-reflection of the inside objects is seen). The scene is imaged through a polarizing filter at two orientations and ICA is used to separate the components as if they were the result of the superposition of two statistically independent sources. All of the applications for ICA in this thesis are in the context of feature extraction, although the underlying idea is the same as for blind source separation. For feature extraction, the problem is simply re-organized into the form shown in Equation 1.2. The difference, then, is in the interpretation and use of the separated signals.

Comon's fundamental paper [23] formalizes the problem of ICA mathematically. Specifically, it described how equations of the form of Equation 1.1 can be solved assuming statistical independence of the source signals. Once the problem had been clearly defined, algorithms were developed to provide an efficient solution. A number of information theoretic approaches can be taken to derive an algorithm for ICA. Information maximization was proposed by Bell and Sejnowski in [27] to derive a stochastic gradient algorithm. Maximum likelihood estimation (MLE) can also be used for ICA, as it was shown by Cardoso [28] that MLE is identical to information maximum for the ICA problem. Nonlinear PCA [29] provides another solution, with

the relationship to Comon's algorithm shown by Lee, Girolami, Bell and Sejnowski in [30]. Additional improvements in algorithm computation time were made by Hyvarinen in [31] by utilizing a fixed point optimization methodology. An excellent summary of the relationship between the aforementioned approaches along with the use of negentropy and kurtosis is provided in [32]. Chapter 2 highlights the significant contributions of Bell and Sejnowski's algorithm, along with Hyvarinen's FastICA. The methodologies of each of these algorithms has implications for their suitability for feature extraction and pattern recognition. These implications are discussed in Chapter 3 and are pointed out again in Chapters 4, 5, and 6, when they impact the experiments described therein.

1.5.2 Subspace Methods for General Pattern Recognition

Subspace methods of pattern recognition are derived from the concept of statistically extracting features from a data set by assuming a linear model for the generation of the data. Features can then be extracted from a linear transformation of the data matrix. Oja described this technique in detail in [33]. Since the inception of the IEEE journal "Transactions on Pattern Analysis and Machine Intelligence", approximately three hundred papers have analyzed this and other statistical techniques [34]. A few linear feature extractors have achieved preeminence — Principal Component Analysis (PCA), Fisher's Discriminant Analysis (FDA) and Independent Component Analysis (ICA). Another very promising subspace technique involves a non-linear extension to PCA — Kernel PCA (KPCA).

A number of recent articles have attempted to summarize the relationships and effectiveness of visual subspaces for recognition, including Duin, Jain and Mao [34] and Moghaddam [35]. Unfortunately, a disproportionate number of the comparisons are conducted using face recognition as the test application, making it difficult to derive general conclusions about the effectiveness of each method. This thesis will seek to rectify this problem.

1.5.3 Subspace Methods for Position and Pose Measurement

Specific use of PCA for position and pose measurement is generally first credited to Murase and Nayar in [36]. In Murase and Nayar's work, the principal component coefficients of the training images form a manifold wherein unknown images can be matched to training images by their position on (or near) this manifold. Using this same technique by creating a manifold from a training object under varying pose (orientation), Murase and Nayar also estimated the pose of unknown objects. Similarly, the unknown position of a camera can be determined by constructing a manifold with training images from a camera at known positions. Nayar et. al. [37] demonstrated the use of eigenspace methods for determining the position of a camera equipped end-effector relative to a circuit board for a chip insertion task by acquiring a set of training images with the robot arm perturbed around the insertion point. Similarly, determining the position of a mobile robot within a room has also been accomplished previously via PCA. Jogan and Leonardis [38] as well as Winters, Gaspar and Santos-Victor [39] used PCA with omnidirectional cameras to accomplish this task. Martinez et. al. [40] compared the performance of PCA and FDA as well as Gaussian mixture models for the task of autonomously navigating a mobile robot equipped with a camera.

In this thesis, Chapter 5 illustrates the use of ICA for the application of position and pose measurement. ICA has not been examined in any amount of detail for this application previously in the literature. This chapter provides some direct evidence that the characteristics of ICA derived features can reduce measurement errors under conditions of occlusion or lighting variance.

1.5.4 Subspace Methods for Face Recognition

The camera position determination problem described above has similarities with face recognition, however a difference is that rather than matching discrete face classes, camera position is determined over a continuum of possible positions over the defined

movement range. First credit for the application of subspaces to face recognition is given to Turk and Pentland [41]. In this work, the subspace was generated from PCA. A minimum Euclidean distance measure was used for classification. In the ten or so years since this pioneering effort, methods which employ face models have developed alongside subspace techniques and have been successfully applied to face recognition. In this thesis, however, in keeping with the context of ICA, only subspace developments will be detailed.

Of particular interest is the ongoing debate over PCA vs. ICA as a subspace representation for faces [42], [35], [43], [44], [45]. While each one of these papers produces a different result and argues in favor of a different subspace, none of them seek to describe why their chosen technique is better. The reasons are often hidden in differences in implementation technique or testing databases. Curiously, in light of the debate over PCA and ICA subspaces, de Ridder, Messer and Kittler [46] approached PCA and ICA feature selection differently than in the face recognition work. They chose a floating search technique [47] to pick the most effective features. This contrasted with selecting ICA features by combining pre-whitening and dimensionality reduction with PCA. The latter method has the effect of making the ICA basis orthogonal in the whitened space and ensuring that the ICA basis spans the same space as the PCA case. Much more will be mentioned about this in Chapters 2 and 3.

The non-linear method of KPCA has very recently been employed for the application of face recognition [48] [49]. Some suggestion has been made that KPCA may outperform both PCA and ICA for this application. A unique use of KPCA has been investigated in this thesis, where it has been compared with PCA and ICA for the purpose of position and pose measurement in Chapter 5.

1.5.5 Lighting and Occlusion Invariant Recognition

Illumination variation in object recognition has historically been a challenging problem. In general, any difference in illumination between the images used as training examples (for designing class decision boundaries or constructing a linear transformation for feature extraction), and the test images to be classified must be accounted for. This illumination variation can be accommodated by either extracting somewhat illumination invariant features or generalizing the classifier to allow for the variation. Direct methods of handling varying illumination have been utilized in subspace methods by constructing an illumination dimension in the subspace. This methodology was employed in [36], where a robot was used to vary the direction of illumination on an object from five different light source positions. It was subsequently shown by Nayar and Murase [50] that under the constraints of a linear reflectance model, bounds could be established on the dimensionality of the illumination space. The linear constraint limits the scope of the characterization to Lambertian reflectances. Specular reflections are non-linear functions of observer viewpoint and a variety of heuristics have been applied to detect and deal with specularities (see [51] for some examples). In general, however, this remains an unsolved problem. Most recent work focus on the linear models and attempt to characterize lighting variation by examining the totality of the lighting space using illumination cones [52] or similar methods. For objects such as faces, which are roughly Lambertian reflectors, this technique can accurately characterize the lighting space in a subspace of relatively low dimension.

A more general approach to illumination variation from a pattern recognition point of view is to abandon the notion of a lighting model and seek illumination invariant features. Typically, the broad assumption made is that the existence of an edge or a sharp intensity gradient is an essential feature of an object and is somewhat robust to illumination change. While shadows and specularities can be indistinguishable from an object's features, the technique of creating features from strong intensity gradients has met with success. A great many machine vision algorithms utilize

object edges for tasks of classification, segmentation, measurement etc. and these function well under modest illumination variation. A common approach to providing illumination invariant features is to employ a LoG (Laplacian of Gaussian) filter to the data during training was used by Bartlett [53] when constructing a basis for a subspace representation. A two dimensional LoG filter is defined by a symmetrical (non-oriented) Laplacian of Gaussian function of a particular scale. An extension to this definition is to derive a scalable, oriented filter. A typical filter applied to early vision tasks is the steerable Gabor wavelet as described by Freeman and Adelson [54]. This filter has the advantage of allowing for maximal response in particular directions and can be scaled to different spatial resolutions. For successful application as a pre-filter for lighting invariant face recognition, LoG and Gabor filters must be tuned to an appropriate spatial resolution. Gabor filters additionally require that the primary orientations in the images be determined a-priori (or multiple filters can be applied). By way of a connection to the previous discussion of human vision, Hoyer and Hyvarinen used sparse coding and a multi-layer network to learn contours from natural images [55] which provide efficient coding of "shape" features. A recent use of filters to provide lighting invariance is described by Wildenauer, Leonardis and Bishof [56] wherein features are extracted which are similar to eigenfeatures derived from PCA but are invariant under filtering. The resulting features are filtered with multiple oriented Gabor wavelets and the best ones are selected by a robust hypothesize and test paradigm based on the Minimum Description Length (MDL) principle.

With respect to object recognition under occlusion, great difficulties arise in trying to categorize or model the effects of occlusion. By definition, an occlusion is any portion of an object blocked from view by another object. The occluding object can in general be of any form. Some amount of robustness to occlusion can be expected when a sufficient number of features exist in the non-occluded object and these features form a holistic representation wherein a few missing data elements does not hinder overall recognition. Eigenfeatures from PCA have been shown to be relatively robust in this context, due to the global nature of these features. Krumm

[57] used these for pose measurement of partially occluded objects. This technique has been extended by Ohba and Ikeuchi [58] to include multiple image windows whereby eigenfeatures in non-occluded windows are unaffected by occlusion . Another recent approach with eigenfeatures was proposed by Leonardis and Bishof [59] where subsets of image points are selected by the robust technique mentioned above.

The supposition that ICA derived features are more effective than a PCA feature space at providing lighting and occlusion invariance to image recognition is a central theme of this thesis. The human visual system has been shown to employ center-surround receptive fields (often modeled by LoG function) for the purpose of lighting invariant vision (see Chapter 5 for details). In all of the experiments conducted in this thesis, lighting conditions are included as a variant in the images. As such, the experiments seek to show that ICA features are effective at providing a degree of lighting invariance and are thus functioning in the same way as the low-level feature extractors of the human visual system.

1.5.6 Extensions To ICA

The basic idea of independent components has been extended in a variety of interesting ways. One possible generalization of ICA is the notion of multidimensional independent components presented by Cardoso [60]. An important generalization is made in Cardoso's work - ICA is not considered as a demixing process and is "matrix-free". In this new model of ICA, a set of orthogonal projection matrices onto each component subspace needs to be determined. In classical ICA, the components are often considered to be geometrically orthogonal. In this approach, the components are not necessarily geometrically orthogonal but are statistically independent (more than statistically orthogonal). By relaxing the assumption of geometric orthogonality, the original data can be grouped into components that are actually independent and those that turn out not to be independent (or are Gaussian). Another way of looking at this is described by Hyvarinen, Hoyer and Inki as Topographic Independent

Components [61]. They use the residual dependence structure that occurs when the components are not independent to create a topographic representation. This idea has been employed in the application of multi-view face detection and recognition [62], [63], where basis components were ordered in a two dimensional map with axes of viewing angle and illumination change. Another application for multidimensional ICA could be to represent signals of different types (for example images and sound). Salem and Erten explored the idea of sensor fusion in [64], although without the use of multidimensional ICA.

It is possible to eliminate the difference between the mixing matrix and the independent components. This leads to another model called spatiotemporal ICA [65]. Spatiotemporal here refers to performing ICA in the temporal domain (assuming the use of time signals) and in the spatial domain corresponding to the spatial mixing matrix. In this model, both the independent components and the mixing matrix are assumed to be generated by independent random variables. In a similar vein, the concept of priors on the mixing matrix has been applied. Specifically, a sparse prior on the mixing matrix amounts to the same idea as spatiotemporal ICA, since a sparse prior is imposed as a method of maximizing super-Gaussianity. The sparse distribution of natural images may be good candidates for this sparse representation, however direct application of sparse priors on the mixing matrix for image processing have not yet materialized.

A couple of other extensions to ICA have not found widespread application in pattern recognition and image processing. If the number of basis vectors exceeds the dimensionality of the space, we have the situation of an over-complete basis. This could occur in the case of feature extraction from images. Some solutions to this estimation problem have been proposed in [7] and [66]. Aside from simply extracting features, to the author's knowledge, no image based application that explicitly requires the over-complete basis has been examined. Another natural extension to ICA had previously been applied to PCA — nonlinear ICA. There have been a number of algorithms proposed for this idea [67], [68], [69] and likely have not found application

due to the ill-posed nature of the problem.

Almost all of the basic experiments and ideas that have been investigated in this thesis can be extended to include the aforementioned generalizations of ICA. Of particular interest is the use of spatiotemporal ICA and multidimensional ICA for the case of multi-dimensional image databases. As an example, multidimensional ICA could be applied for a position measurement application with multiple degrees of freedom of motion. Some of these extensions and their possible application will be examined in the future research directions described in Chapter 7.

1.6 Summary of Contributions From This Thesis

The are two central themes of this thesis. The first is an examination of the behavior of ICA derived features for the purposes of image recognition under conditions of lighting variation, pose variation and occlusion. The second is providing a method of improving feature extraction by combining feature extraction and support vector machine classification. The contributions of this work are listed below.

- Much of the current literature which uses ICA for feature extraction provides a comparison with PCA. It is shown theoretically that PCA and ICA derived bases are related by an orthogonal transformation and therefore ICA cannot offer a benefit over PCA for the purposes of providing a basis for subspace based recognition, provided that two conditions are met. The first is that an ℓ_2 distance metric is used for classification. The second is that PCA is used for dimensionality reduction prior to the calculation of the independent components. Both of these conditions are often found to hold in the previous literature on the use of ICA for image recognition.

- ICA algorithms provide an approximation to the theoretical ideal of statistically independent features or basis images. As a result, it is shown that these

approximations can provide a small difference in performance as a feature extractor with respect to PCA. These differences are shown to be not statistically significant.

- It is shown that there is still a distinct advantage to using ICA. The individual basis images that are extracted may indeed be more effective at representing important directions in the data. ICA is used with a floating search technique for subspace object recognition. The search was used to select the k best features which maximize the inter-object Euclidean distance in feature space. Thus, the criterion function to be optimized in the feature search is the sum of the distances between the feature vectors of all of the training objects. By selecting features in this way, the condition that PCA is exclusively used for dimensionality reduction no longer holds.

- When measuring the unknown position of an object or a camera from subspace techniques, this work draws the conclusion that PCA is well suited for this application when imaging conditions do not change from training to measurement. The highly correlated nature of the training subspace provides the affinity for PCA. However, PCA's performance is poor when lighting variation or occlusion occur in the images used to determine the unknown position. Other subspaces (kernel PCA and Fisher's linear discriminant) also perform poorly.

- ICA is shown to offer an advantage for position measurement in the presence of lighting variation and occlusion when dimensionality reduction is performed with a search technique to find the best features. Significant improvement in performance (as much as 50% reduction in error) can occur from the use of ICA in the case of occlusion.

- A similar advantage to using ICA is found in general object recognition. When lighting conditions are radically changed from training to recognition, such as is the case when specular objects are to be recognized, ICA provides higher

recognition rates than PCA as the dimensionality of the subspace is reduced. Specular objects cannot be described in a low dimensional subspace in a similar way as Lambertian surfaces, since the reflectance function is non-linear.

- It has been previously shown that some degree of lighting invariant image recognition can be obtained through the use of LoG filters as a pre-filtering step on images. This thesis recognizes the similarity between the bandpass nature of LoG filters and those derived from ICA to arrive at a novel application for ICA as a feature extractor. ICA is used to derive small oriented and bandpass filters from a database of images which are used as pre-filters when performing image recognition in subspace. Specifically, when a support vector machine is used as a classifier, the ICA filtering method avoids a difficult tuning problem with the SVM kernel widths. The ICA derived filters exhibit very similar margin, number of support vector and recognition results to the LoG filters without the a-priori knowledge required to find an appropriate Laplacian of Gaussian width.

- A new algorithm, called a Compact Support Vector Representation (CSVR) is developed in this thesis to improve the performance of subspace techniques with support vector machines. A basis is derived directly from the support vectors by considering these feature vectors as outliers. These feature vectors are subsequently moved and a new basis is derived from the modified features. In this way, the classes are made much more compact and it is shown that these compact classes are a particularly effective arrangement for representing image databases, including those with lighting and pose variation. This algorithm also serves to unify the classification and feature extraction stage of pattern recognition by utilizing information from the class boundaries to guide the computation of a basis.

- The CSVR algorithm is shown to provide better generalization over variations in lighting and object pose than PCA or ICA derived bases for the application

of face or general object recognition. In most cases, the number of support vectors is only about 10% of the raw data or PCA and ICA representations.

- The CSVR algorithm converges rapidly to a large margin (often 500% of the raw data or PCA and ICA margin) in about a hundred iterations when it is started with a basis representing an identity mapping. Convergence speeds up to less than 10 iterations when PCA or ICA bases are used as a starting point.

1.7 Outline of Thesis

In Chapter 2, the theoretical background of subspace pattern recognition is provided. Other subspaces (PCA, kernel PCA and Fisher's Linear Discriminant) used in subsequent experiments are introduced. Chapter 3 provides a discussion of Euclidean distance classification and the need for more sophisticated classifiers such as the support vector machine. Support vector classification is also detailed. Feature extraction for specific use with a SVM is described. Chapter 4 provides examples of using ICA features for position measurement applications under conditions of lighting variance and occlusion. In Chapter 5, the advantages of using an ICA basis and ICA derived pre-filters are illustrated with object and face recognition experiments. The SVM is used for face classification. Chapter 6 shows how the features supplied to a SVM can be modified to provide improved classification results for a Gaussian mixture dataset and object and face databases. Chapter 7 draws conclusions from the results and discussions of the previous chapters and suggests future directions for work with ICA features and SVM classifiers.

Chapter 2

Feature Extraction

Recall the model for feature extraction defined in Chapter 1 as:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \text{ where } \mathbf{W} = \mathbf{A}^{-1} \tag{2.1}$$

This chapter will examine a number of techniques for finding a vector of features $\mathbf{y}$ for a data vector $\mathbf{x}$ under a linear transformation $\mathbf{W}$, namely PCA, ICA, and FLD (Fisher's Linear Discriminant).

2.1 PCA Solution

The general goal of PCA is to find a transformation matrix $\mathbf{W}$ in:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{2.2}$$

as defined in Equation (2.1), that diagonalizes $E\left[\mathbf{y}\mathbf{y}^T\right] = \mathbf{R}_{yy}$. As such:

$$E\left[\mathbf{y}\mathbf{y}^T\right] = E\left[\mathbf{W}\mathbf{x}\mathbf{x}^T\mathbf{W}^T\right] \tag{2.3}$$

With $\mathbf{R}_{xx} = E\left[\mathbf{x}\mathbf{x}^T\right]$:

$$\mathbf{R}_{yy} = \mathbf{W}\mathbf{R}_{xx}\mathbf{W}^T \tag{2.4}$$

From the eigendecomposition $\mathbf{Q}^T\mathbf{R}_{xx}\mathbf{Q} = \mathbf{\Lambda}$ where $\mathbf{Q}$ is matrix of the eigenvectors of $\mathbf{R}_{xx}$ (arranged in columns) and $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues, it is clear that:

$$\mathbf{R}_{yy} = \mathbf{\Lambda} = \mathbf{Q}^T\mathbf{R}_{xx}\mathbf{Q} \tag{2.5}$$

so that $\mathbf{W}_{pca} = \mathbf{Q}^T$.

The batch mode PCA problem is defined by:

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \tag{2.6}$$

with $\mathbf{Y} = [\mathbf{y}(1), \mathbf{y}(2), \cdots, \mathbf{y}(N)] \in \mathbb{R}^{n\times N}$ and $\mathbf{W}$, $\mathbf{X}$ defined as above. PCA can then be performed directly from the singular value decomposition (SVD) of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{N\times N}$ are orthogonal matrices. $\mathbf{\Sigma}$ is a pseudo diagonal matrix with the first n elements on the diagonal containing the first n singular values, ordered from largest to smallest, and the last $m - n$ diagonal elements zero. If we assume noiseless data:

$$\mathbf{X} = [\mathbf{U}_S, \mathbf{U}_N] \begin{bmatrix} \mathbf{\Sigma}_S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{V}_S, \mathbf{V}_N]^T \tag{2.7}$$

where $\mathbf{U}_S \in \mathbb{R}^{m\times n}$ and $\mathbf{\Sigma}_S \in \mathbb{R}^{n\times n}$ is a diagonal matrix of the first n singular values. $\mathbf{U}_S\mathbf{\Sigma}_S\mathbf{V}_S{}^T$ spans the signal subspace of $\mathbf{X}$. From this point on, only the signal space of $\mathbf{X}$ will be considered and the subscript S will be dropped from the $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$ matrices. From the SVD, then,

$$\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}^T\mathbf{X} \tag{2.8}$$

provides a matrix $\mathbf{Y} = \mathbf{\Sigma}\mathbf{V}^T$ with decorrelated columns, so:

$$\mathbf{Y} = \mathbf{U}^T\mathbf{X} \tag{2.9}$$

and $\mathbf{W}_{PCA-Batch} = \mathbf{U}^T$ in Equation 2.6. Note that dimensionality reduction is implicitly modeled since the first d column vectors of $\mathbf{U}$ can be used instead of all n representing a projection into the signal subspace of $\mathbf{X}$. When $\mathbf{Y} = \mathbf{\Sigma}^T\mathbf{U}^T$ with $\mathbf{W} = \mathbf{V}^T$ where $\mathbf{X}$ is replaced with $\mathbf{X}^T$ in the batch mode PCA model:

$$\mathbf{\Sigma}^T\mathbf{U}^T = \mathbf{V}^T\mathbf{X}^T \tag{2.10}$$

The relationship between batch mode PCA with the SVD and general PCA is seen from the estimate of the covariance matrix $\hat{\mathbf{R}}_{xx}$ which is necessary to perform PCA from a finite number N of data samples:

$$\hat{\mathbf{R}}_{xx} = \frac{1}{N}\mathbf{X}\mathbf{X}^T \tag{2.11}$$

since from $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$:

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T \text{ and } \mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T \tag{2.12}$$

It is thus clear that the eigenvectors of the estimated covariance matrix $\mathbf{X}\mathbf{X}^T$ are the left singular vectors $\mathbf{U}$ of $\mathbf{X}$ with eigenvalues of the square of the singular values of $\mathbf{X}$. PCA can then be computed with either the left or right singular vectors, indicating that either $\mathbf{X}\mathbf{X}^T$ or $\mathbf{X}^T\mathbf{X}$ can be used as estimates of $\mathbf{R}_{xx}$. Which one is used depends on the size of $\mathbf{X}$.

2.2 ICA Solution

The general goal of ICA is to find a transformation matrix $\mathbf{W}$ in:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{2.13}$$

as defined in Equation 1.2, which factorizes the joint probability distribution of $\mathbf{y}$ by:

$$f(\mathbf{y}) = \prod_{i=1}^{n} f_i(y_i) \tag{2.14}$$

where f is the probability density function (pdf), and $f_i(y_i)$ represents the marginal densities of each of the n variables in the n dimensional random vector $\mathbf{y}$.

A key concept in ICA is that whitening transforms perform part, but not all, of the process of transforming data into a statistically independent representation. By definition, white random variables are uncorrelated and have unit variance. In other words, for the white random vector $\mathbf{x}$:

$$E\left[\mathbf{x}\mathbf{x}^T\right] = \mathbf{I} \tag{2.15}$$

where $\mathbf{I}$ is the identity matrix. Therefore, whitening is the process of transforming a random vector $\mathbf{x}$ by some matrix $\mathbf{B}$ so that $\mathbf{z} = \mathbf{B}\mathbf{x}$ is white. From the earlier discussion on general PCA, it was shown that under a transformation $\mathbf{W} = \mathbf{Q}^T$ of the eigenvectors $\mathbf{Q}$ of the covariance matrix $\mathbf{R}_{xx}$ in:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{2.16}$$

$\mathbf{R}_{yy} = \mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues of $\mathbf{R}_{xx}$. Therefore, a whitening transform can be constructed by simply scaling the transform by $\mathbf{\Lambda}^{-1/2}$. So a whitened $\mathbf{x}$, denoted $\mathbf{z}$ is obtained by:

$$\mathbf{z} = \mathbf{\Lambda}^{-1/2}\mathbf{Q}^T\mathbf{x} \tag{2.17}$$

since $\mathbf{R}_{zz} = \mathbf{\Lambda}^{-1/2}\mathbf{Q}^T\mathbf{R}_{xx}\mathbf{Q}\mathbf{\Lambda}^{-1/2} = \mathbf{\Lambda}^{-1/2}\mathbf{\Lambda}\mathbf{\Lambda}^{-1/2} = \mathbf{I}$

If $\mathbf{x}$ is whitened for the determination of the demixing matrix $\mathbf{W}$:

$$\mathbf{y}_z = \mathbf{W}_o\mathbf{z} \tag{2.18}$$

where $\mathbf{y}_z$ is white and $\mathbf{W}_o$ is a new orthonormal demixing matrix of the whitened data. Thus the ICA problem has been transformed into finding an orthogonal matrix in the whitened space.

Note that for a given set of data $\mathbf{X}$, a whitening transform $\mathbf{\Lambda}^{-1/2}\mathbf{Q}^T$ can be derived directly from the SVD of $\mathbf{X}$ where $\mathbf{\Lambda}^{-1/2}\mathbf{Q}^T = \mathbf{\Sigma}^{-1}\mathbf{U}^T$. Therefore, the columns of:

$$\mathbf{V} = \mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{X} \tag{2.19}$$

are uncorrelated and have unit variance.

For the present, the general representation of the ICA model in batch mode will be discussed. A subsequent section will detail specific methodoligies for determining the demixing matrix. The ICA model can be described in two ways. The difference between the two models occurs as a result of switching the roles of the mixing matrix and the coefficients. Each of these will be described below.

2.2.1 ICA with Statistically Independent Coefficients

From the original batch mode model:

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \tag{2.20}$$

$\mathbf{W}$ is to be found such that the columns of $\mathbf{Y}$ are as statistically independent as possible. In other words, if the columns of $\mathbf{Y}$ are considered as instances of the random vector $\mathbf{y}$ the goal is to find the demixing matrix so that the factorization described in Equation 2.14 is followed as closely as possible for the given data. The concept of "as independent as possible" will be made more rigorous in the subsequent section. If the model is rewritten as:

$$\mathbf{\Sigma}^{-1}\mathbf{Y} = \mathbf{Y}_z = \mathbf{W}_o\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{X} \tag{2.21}$$

$\mathbf{Y}_z$ has white columns. The demixing matrix that needs to be found, $\mathbf{W}_o$, is orthonormal.

$$\mathbf{Y} = \mathbf{W}_o \mathbf{U}^T \mathbf{X} \tag{2.22}$$

This also corresponds to the standard practice of calculating this model by performing ICA on the principal components of $\mathbf{X}$ where $\mathbf{Y} = \mathbf{W}_o \mathbf{\Sigma} \mathbf{V}^T = \mathbf{W}_o \mathbf{U}^T \mathbf{X}$. Thus, this model is equivalent to the PCA model under a rotation of an orthonormal matrix $\mathbf{W}_o$. The rotation seeks to make the the columns of $\mathbf{Y}$ as independent as possible instead of simply decorrelated, as in PCA.

2.2.2 ICA with Statistically Independent Demixing Matrix

In this model, the roles of the mixing matrix and the coefficients are switched. As such:

$$\mathbf{X}^T = \mathbf{S}^T \mathbf{A}^T \tag{2.23}$$

Then:

$$\mathbf{A}^T = \mathbf{S}^{+T} \mathbf{X}^T \tag{2.24}$$

with $(+)$ denoting the pseudo-inverse. The model in Equation 2.20 is rewritten as:

$$\mathbf{W}^{+T} = \mathbf{Y}^{+T} \mathbf{X}^T \tag{2.25}$$

Performing the same whitening process as for the first model:

$$\mathbf{W}_z{}^T \mathbf{\Sigma}^{-T} = \mathbf{Y}_o \left(\mathbf{\Sigma}^{-1} \mathbf{U}^T \mathbf{X} \right)^T = \mathbf{Y}_o \mathbf{X}^T \mathbf{U} \mathbf{\Sigma}^{-T} \tag{2.26}$$

Thus:

$$\mathbf{W}_z{}^T = \mathbf{Y}_o \mathbf{X}^T \mathbf{U} \tag{2.27}$$

where $\mathbf{Y}_o$ is an orthonormal matrix and $\mathbf{W}_z{}^T$ is a matrix with white columns. This model will be used for feature extraction by:

$$\mathbf{Y}_o{}^T = \mathbf{X}^T\mathbf{U}\mathbf{W}_z \tag{2.28}$$

Then:

$$\mathbf{Y}_o = \mathbf{W}_z\mathbf{U}^T\mathbf{X} \tag{2.29}$$

which is the PCA model under a rotation of an orthonormal matrix $\mathbf{W}_z$ with rows that are as statistically independent as possible. Note that $\mathbf{W}_o$ and $\mathbf{W}_z$ are not the same. In the first model, the columns of $\mathbf{Y}$ are as independent as possible. In the second, they are simply decorrelated.

2.2.3 ICA Methods

A full comparison of ICA methods is beyond the scope of this work. However, two main features emerge from an analysis of the differences in methodologies of ICA. The first is the method of estimating the pdf of the sources to be separated. The second is the criterion which is used to provide an objective function to optimize. There have been many reviews on the connection between the objective functions derived from mutual information, negentropy, maximum likelihood, or higher order moments and cumulants (see [30] for an example). The performance, from a mean square of source estimation error point of view, has been similarly reviewed, typically in each paper where an author presents a new algorithm. The major concern in this thesis (and of feature extraction in general), however, is not the mean square error of source estimation, but in the utility of these sources as features extracted from the data. An absolute measure of the effectiveness of these extracted features for the purposes of general pattern recognition is impossible. There is no way to assess a-priori what features are most discriminating for a general class of objects such as those derived from images of an arbitrary scene. As such, the hypothesis is made that the ICA

derived features, irrespective of how they were derived, or how accurately the reflect the original model, are in some way discriminating. The only proviso is that the extracted features closely follow the model that was employed to extract them. This implies that source estimation accuracy is not a useful criterion for selecting an ICA methodology in the feature extraction modality. Consequently, some useful criterion will be now be outlined.

Feature extraction for pattern recognition is often performed on an input space of hundreds or thousands of data samples. Therefore, it is useful to find an ICA method which is relatively fast. Two key points determine the speed of ICA algorithms. The first is the computational cost of estimating the probability density function. The second is the computational cost and the convergence rate of the optimization of the objective function.

Regarding the estimation of the probability density function, two ideas have proven to be useful. The first of these ideas is to directly estimate the density function. This can be accomplished through the use a series expansion of the pdf around the Gaussian pdf. A commonly used expansion, similar in spirit to a Taylor series expansion, utilizing Chebyshev-Hermite polynomials, is the Edgeworth expansion [23]. Another direct estimation method is to use a kernel estimator [70], [71]. Both of these estimation techniques are non-parametric. In methods using non-parametric pdf estimation, the computational cost of estimation is very high [27] making them poor candidates for feature extraction. The second idea is to employ a fixed non-linearity, thus transforming the problem of density estimation into a semi-parametric one. This idea is based on the fact that is possible to get a usable approximation of the densities from a simple family of densities. There is always a component of the densities which cannot be modeled by the density family (hence the name semi-parametric), but it turns out that the errors in density estimation introduced by the aforementioned approximation does not greatly impact the overall independent component estimation. Both FastICA [31] and Bell and Sejnowski's method [27] use this idea, which will be described in the following sections. As a testiment to the accuracy

of the semi-parametric estimation technique, Luo and Lu [71] compared the results of a technique based on the use of a kernel estimator and FastICA and found little (or no) difference in the overall performance. The results, however are compared for only four experiments, making the statistical significance of these experiments rather small.

With respect to the optimization of the objective function, a few options exist. A common optimization method is the gradient descent algorithm. This forms the basis of the Bell and Sejnowski method. This method utilizes the gradient of the objective function and has linear convergence. Although this method is conceptually and computationally simple, it suffers from the slow convergence of a linear method. Newton's method, utilizing the Hessian of the objective function, is another option, which has locally quadratic convergence. Unfortunately, the computational cost of inverting the Hessian matrix can be very high. As a comprimize between convergence speed and computational complexity, a conjugate gradient technique can be applied, which avoids the inversion of the Hessian. This is the method employed by Luo, albeit applied on the Stiefel manifold, which is, for the ICA model, the constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, This constraint corresponds to the orthonormality of the demixing matrix. Unfortunately, this method suffers from a high computational complexity, although has super-linear convergence. It is possible to approximate Newton's method with a fixed point algorithm, which forms the basis of FastICA. Fixed point algorithms have very low computational complexity, and FastICA, under conditions of a symmetrical distribution of the sources (typically the case) has cubic convergence [32]

In this thesis, two methods were used — a natural gradient version of the Bell-Sejnowski algorithm [27] and FastICA [31]. Both of these algorithms represent a good compromise between computational complexity and convergence speed. FastICA was the preferred method for most of the experiments in this thesis due to its speed. However, the Bell and Sejnowski alogrithm was occasionally used to illustrate the effect of not constraining the demixing matrix to be orthonormal in the whitened

space. This had a direct impact on the behavior of the ICA algorithms for feature extraction. A more detailed discussion of this follows in the next chapter.

2.3 Review of Information Theory

The theoretical approach to characterizing random variables which is employed by both FastICA and the Bell and Sejnowski algorithm is based on information theory. As such, a very brief review of information theory, providing definitions for entropy and mutual information for continuous random variables will be provided.

2.3.1 Entropy

The entropy of a continuous random vector $\mathbf{y}$ with a joint pdf $p(\mathbf{y})$ is defined as:

$$H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y} \tag{2.30}$$

This quantity is often called differential entropy. It will be refered to simply as entropy, since all random variables considered herein are continuous. Note that entropy of a continuous random variable can be negative, since probability density functions can be greater than one, given that they fit the definition of a pdf, namely that $\int p(\mathbf{y}) d\mathbf{y} = 1$. This definition of entropy requires that more probable intervals will make entropy smaller (more negative), which fits with the intuitive interpretation of entropy as a measure of randomness. The relative entropy of two densities p_1 and p_2 is defined as:

$$D(p_1||p_2) = \int p_1(\xi) \log \frac{\mathbf{p_1}(\xi)}{\mathbf{p_2}(\xi)} \mathrm{d}\xi \tag{2.31}$$

This is also known as Kullback Leibler (KL) divergence.

A fundamental result of information theory is that zero mean Gaussian random variables have the largest entropy attainable of all zero mean random variables of the same covariance as the Gaussian random variable. This can be seen from the

(easily shown) fact that KL divergence is non-negative. Therefore, considering two densities p and p_{gauss} where p is any density function of a random vector $\mathbf{y}$ with covariance $\mathbf{R}_{yy} = E\left[\mathbf{y}\mathbf{y}^T\right]$ and p_{gauss} is a Gaussian density function with covariance $\mathbf{R}_{yy}$. Therefore:

$$\begin{aligned} 0 &\leq D(p||p_{gauss}) \\ &\leq \int p \log\left[\frac{p}{p_{gauss}}\right] \\ &\leq -H(p) - \int p \log(p_{gauss}) \\ &\leq -H(p) + H(p_{gauss}) \end{aligned} \tag{2.32}$$

Note that the density function p in line 3 of Equation (2.32) can be replaced with p_{gauss} because they would yield the same moments of $\log(p_{gauss})$. This proves $H(p) \leq H(p_{gauss})$.

A quantity called negentropy (J) can be defined as the Kullback Leibler divergence of $p(\mathbf{y})$ and the density of a Gaussian random variable with the same covariance matrix as $\mathbf{y}$, $p_{gauss}(\mathbf{y})$:

$$J(\mathbf{y}) = \int p(\mathbf{y}) \log \frac{p(\mathbf{y})}{p_{gauss}(\mathbf{y})} d\mathbf{y} \tag{2.33}$$

From the properties of the log function, this can be written as:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{2.34}$$

where $\mathbf{y}_{gauss}$ is a Gaussian random variable of the same covariance matrix as $\mathbf{y}$. This can be interpreted as a measure of non-Gaussianity. It is zero for a Gaussian variable and always non-negative, due to the result proven above. This result will be used in the description of the FastICA algorithm.

Mutual information, or the degree of dependence between variables in a random vector $\mathbf{y}$ is defined as:

$$I(\mathbf{y}) = \int p(\mathbf{y}) \log \frac{p(\mathbf{y})}{\prod_i p_i(y_i)} d\mathbf{y} \tag{2.35}$$

where $p(\mathbf{y})$ is the joint probability density of $\mathbf{y}$ and $p_i(y_i)$ are the marginal densities. This definition can be interpreted as a divergence, in the Kullback-Leibler sense, of the pdf of $\mathbf{y}$ from the factorized density $\prod_i p_i(y_i)$. Thus, if the density $\mathbf{y}$ can be factorized as the product of marginal densities, $p(\mathbf{y}) = \prod_i p_i(y_i)$, the log of the ratio $p(\mathbf{y}) \prod_i p_i(y_i)$ is zero, and the mutual information is zero. This actually occurs if and only if the densities are equal, due to the convexity of the negative of the log function and the application of Jensen's inequality, but the details are omitted here for brevity. So, multivariate densities which can be factorized into the product of their marginal densities have statistically independent variables (by the definition of statistical independence) and have $I = 0$. The larger the distance between $p(\mathbf{y})$ and the factorized density, the larger the mutual information between variables.

2.3.2 Minimizimg Mutual Information with Gradient Based Optimization

If a vector valued nonlinear mapping $\mathbf{s} = \mathbf{g}(\mathbf{y})$ (where $\mathbf{y} = \mathbf{W}\mathbf{x}$ from the ICA model in Equation 2.13) is applied such that $\mathbf{s}$ has uniform marginal densities, the product of the marginal distributions of $\mathbf{s}$, $\prod_i p_i(s_i) = 1$ from the definition of the uniform density. Then:

$$I(\mathbf{s}) = -H(\mathbf{s}) = \int p(\mathbf{s}) \log p(\mathbf{s}) d\mathbf{s}. \tag{2.36}$$

The constraint of selecting the nonlinear mapping $\mathbf{g}$ so that $\mathbf{s}$ has uniform marginal densities has an important implication. The relationship between the marginal densities of a random variable $\mathbf{y}$ and its image $\mathbf{s}$ under the mapping $\mathbf{g}$ is known to be:

$$p(s_i) = \frac{p(y_i)}{\left| \frac{\partial g_i(y_i)}{\partial y_i} \right|} \tag{2.37}$$

If s_i has uniform densities:

$$p(y_i) = \left| \frac{\partial g_i(y_i)}{\partial y_i} \right| \tag{2.38}$$

This simply states that the random variable y_i has a probability density function which is equal to the derivative of the nonlinearity g_i. To this end, the nonlinearity should be selected so that its derivative matches the pdf of the random variable y_i (or equivalently that the non-linearity should match the cumulative distribution function). The general assumption that is commonly made for the case of images is that the demixing process yields estimates of sources $\mathbf{y}$ which have a super-Gaussian distribution. Typically, then, for feature extraction from images, the choice of the nonlinearity is made so that its derivative is similar in shape to a super-Gaussian pdf. Functions which are typically employed for this are the hyperbolic tangent and the logistic function $[\frac{1}{1+e^{-x}}]$. In practice, a single function g will be selected to map each of the variables in the multi-dimensional case, since they will all be assumed to have a similarly shaped distribution. While it seems surprising that pdfs of a wide group of supergaussian pdfs can be estimated by a derivative of a fixed non-linear function, it has been shown that a Taylor expansion of the non-linear functions used actually provide statistics higher than the fourth order, thus making separation possible [30]. For derivation purposes, full generality of $\mathbf{g}$ as a vector valued function with multiple non-linearities will be maintained.

Continuing with the derivation of a gradient based opitmization technique for ICA based on minimizing mutual information, a well known formula for the entropy of a random variable under a non-linear mapping can be used. The formula states:

$$H(\Phi_1(\mathbf{w}_1{}^T\mathbf{x}), \cdots, \Phi_i(\mathbf{w}_i{}^T)) = H(\mathbf{x}) + E\{\log|\det\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{W}}|\} \tag{2.39}$$

where $\mathbf{F}(\mathbf{x}) = (\Phi_1(\mathbf{w}_1{}^T\mathbf{x}), \cdots, \Phi_n(\mathbf{w}_n{}^T\mathbf{x}))$ is a vector valued function of n mappings Φ_1 to Φ_n. This directly corresponds to the case of the mapping of each $\mathbf{w}_i{}^T\mathbf{x}$ in $\mathbf{Wx}$ under g_i. Evaluating the derivative:

$$E\{\log|\det\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{W}}|\} = \sum_i E\{\log \Phi_i{}'(\mathbf{w}_i{}^T\mathbf{x})\} + \log|\det \mathbf{W}| \tag{2.40}$$

where Φ' is the derivative of Φ. Thus, by selecting the derivative of the non-linearities g_i to be an approximation to the pdf of the original mixed sources which are approximated by $\mathbf{s} = \mathbf{g}(\mathbf{y})$, an expression for the entropy of $\mathbf{s}$ can be written:

$$I(\mathbf{s}) = -H(\mathbf{s}) = -(H(\mathbf{x}) + \sum_i E\{\log p_i(\mathbf{w}_i{}^T\mathbf{x})\} + \log|\det \mathbf{W}|) \tag{2.41}$$

This expression can be arrived at from a likelihood argument as well [32]. Bell and Sejnowski considered the non-linearities g_i as negative score functions of the distributions of the sources to be estimated:

$$g_i = (\log p_i)' = \frac{p'_i}{p_i} \tag{2.42}$$

Note that this is consistant with the original definition of $\mathbf{g}$. For example, if $-\cosh$ is selected as an approximation for the density function of the mixed sources (along with some constants to ensure that it is a density), $\frac{p'_i}{p_i}$ is the hyperbolic tangent function. In other words, a density family exists that satisfies both definitions for g_i. Using the negative score functions, the derivative of $I(\mathbf{s})$ can be expressed in matrix form as:

$$\frac{\partial I}{\partial \mathbf{W}} = -\left(\frac{\partial \log|\det \mathbf{W}|}{\partial \mathbf{W}} + \sum_i E\left[\frac{\partial}{\partial \mathbf{W}}\log \mathbf{g}(\mathbf{y})\right]\right) = -\left((\mathbf{W}^T)^{-1} + E\left[\mathbf{g}(\mathbf{W}\mathbf{x})\mathbf{x}^T\right]\right) \tag{2.43}$$

This can be minimized by employing gradient descent, which follows the general update rule for $\mathbf{W}$, $\mathbf{W} \leftarrow \mathbf{W} - \alpha\frac{\partial F(\mathbf{W})}{\partial \mathbf{W}}$, where F is an objective function to minimize. Often, as will be done herein, this update rule is written as $\Delta\mathbf{W} \propto -\frac{\partial F(\mathbf{W})}{\partial \mathbf{W}}$ where the positive scalar representing the length of the gradient update is implied. The following update rule for the minimization of I with respect to $\mathbf{W}$ is obtained:

$$\Delta\mathbf{W} \propto ((\mathbf{W}^T)^{-1} + E[g(\mathbf{W}\mathbf{x})\mathbf{x}^T]) \tag{2.44}$$

Steepest descent employs the gradient of a function which points in the steepest

direction in a Euclidean parameter space. Unless the data has been whitened (as mentioned above), the parameter space of $\mathbf{W}$ is not constrained to be Euclidean. It does, however, have a Riemannian metric structure. The notion of distance as defined by Riemannian geometry can then be employed. Distance in Reimannian geometry is defined for two vectors $\mathbf{w}$ and $\mathbf{w} + \Delta\mathbf{w}$ as:

$$d(\mathbf{w}, \mathbf{w} + \Delta\mathbf{w}) = \sqrt{\Delta\mathbf{w}^T \mathbf{G}(\mathbf{w}) \Delta\mathbf{w}} \tag{2.45}$$

where $\mathbf{G}(\mathbf{w})$ is the Riemannian metric tensor, an $(N \times N)$ positive-definite matrix which describes the intrinsic shape of a manifold in N-dimensional space [72]. Of course, if $\mathbf{G}(\mathbf{w}) = \mathbf{I}$ we are referring to a Euclidean space. In a Riemannian space, the steepest descent direction of an objective function F is defined by the direction $\Delta\mathbf{w}$ which minimizes $F(\mathbf{w} + \Delta\mathbf{w})$ under the constraint [72]:

$$d(\mathbf{w}, \mathbf{w} + \Delta\mathbf{w}) = \sqrt{\Delta\mathbf{w}^T \mathbf{G}(\mathbf{w}) \Delta\mathbf{w}} = \epsilon \tag{2.46}$$

where ϵ is a small scalar. This minimization gives the following learning rule, for an objective function F and matrix of parameters $\mathbf{W}$ to be optimized [72]:

$$\Delta\mathbf{W} \propto -\mathbf{G}^{-1}(\mathbf{W}) \frac{\partial F(\mathbf{W})}{\partial \mathbf{W}} \tag{2.47}$$

This form of descent is called natural gradient steepest descent. As such, the only difference between standard steepest descent and natural gradient steepest descent is the multiplication of the gradient by the tensor $\mathbf{G}^{-1}(\mathbf{W})$. It was shown by Amari in [73] that for the descent ICA problem descibed above, the natural gradient update rule is simply the original gradient post-multiplied by $\mathbf{W}^T\mathbf{W}$. This gives a natural gradient update rule:

$$\Delta\mathbf{W} \propto ((\mathbf{W}^T)^{-1} + E[g(\mathbf{W}\mathbf{x})\mathbf{x}^T])\mathbf{W}^T\mathbf{W} \tag{2.48}$$

Therefore:

$$\Delta \mathbf{W} \propto (\mathbf{I} + E[g(\mathbf{y})\mathbf{y}^T])\mathbf{W} \tag{2.49}$$

This offers a considerable advantage over the original steepest descent update rule. The matrix $\mathbf{W}$ no longer needs to be inverted, making the result easier to compute and more stable.

2.3.3 Maximizing Non-Gaussianity using Negentropy and Fixed-Point Optimization — FastICA

The key contribution of the method that was employed by Hyvarinen in FastICA to speed up the convergence of ICA algorithms was the use of fixed point iteration. Fixed point iteration is conceptually very simple. A point x_{fp} of x is called a fixed point of a continuous function $f(x)$ if:

$$f(x_{fp}) = x_{fp} \tag{2.50}$$

The method to find a fixed point by repeated substitution of $x_i = f(x_{i-1})$ for each iteration i until convergence is called fixed point iteration. Note that in general, there can be convergence or divergence, depending on the nature of f.

Fixed point iteration was used by Hyvarinen in the context of ICA optimization by noting that at a stable point of a gradient algorithm, under the constraint of optimizing on the unit sphere, the gradient must point in the direction of the variable for which we are optimizing. In this case, adding the gradient to the variable to be optimized as is done during gradient descent does not change its direction, only its magnitude. If the variable is normalized in each iteration, at convergence, the magnitude of the optimized variable does not change, thus providing the utility of a fixed point method. As proof of the claim, consider the following maximization problem:

Maximize $F(\mathbf{w})$ on the unit sphere under the constraint $||\mathbf{w}|| = 1$ using Lagrange

multipliers. The first step in the maximization involves taking derivatives and setting the result to zero:

$$\begin{aligned} \frac{\partial F(\mathbf{w})}{\partial \mathbf{w}} + \lambda \frac{||\mathbf{w}|| - 1}{\partial \mathbf{w}} &= 0 \\ \frac{\partial F(\mathbf{w})}{\partial \mathbf{w}} + \lambda \frac{\mathbf{w}}{||\mathbf{w}||} &= 0 \\ \frac{\partial F(\mathbf{w})}{\partial \mathbf{w}} &= c\mathbf{w} \end{aligned} \tag{2.51}$$

where c is a scalar and λ is a Lagrange multiplier.

Thus, if $\mathbf{w}$ is optimized with gradient descent on a unit sphere, the gradient must point in the direction of $\mathbf{w}$ at the optimal point. Using this concept for the ICA problem, a fast fixed point iteration version of ICA can be derived from a gradient algorithm based on negentropy. This technique was employed for the FastICA algorithm used in this thesis.

Negentropy is a robust measure of non-Gaussianity, however, the entropies are difficult to find directly, as an estimate of the pdf is required. Higher order cumulants can be used, but these tend to be non-robust. Using non-quadratic functions to approximate negentropy, the following approximation can be derived [32]:

$$J(y) \propto [E\{G(y)\} - E\{G(y_{gauss})\}]^2 \tag{2.52}$$

where $G(\cdot)$ is practically any non-quadratic function, y is a standardized random variable, and y_{gauss} is a standardized Gaussian random variable. Functions that are useful in this regard are $G(y) = \log\cosh(y)$ and $G(y) = -e^{y^2/2}$.

From the ICA model $\mathbf{y} = \mathbf{W}\mathbf{x}$, we can whiten $\mathbf{x}$ to a white random variable $\mathbf{z}$. Then the new model is simply $\mathbf{y} = \mathbf{W}_o\mathbf{z}$ where $\mathbf{W}_o$ is orthonormal. In FastICA, the demixing matrix is found one vector $\mathbf{w}$ at a time, in order to keep $\mathbf{w}$ on the unit sphere and thus apply a fixed-point iteration, as mentioned above. Taking the

gradient of J:

$$J \propto \left[E\{G(\mathbf{w}^T\mathbf{z})\} - E\{G(y_{gauss})\}\right]^2 \tag{2.53}$$

$$\frac{\partial J}{\partial \mathbf{w}} \propto \gamma E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} \tag{2.54}$$

where $\gamma = E\{G(\mathbf{w}^T\mathbf{z})\} - E\{G(y_{gauss})\}$. A gradient descent learning rule for $\mathbf{w}$ can then be written:

$$\Delta\mathbf{w} \propto \gamma E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} \tag{2.55}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{2.56}$$

where g is the derivative of G. The normalization of $\mathbf{w}$ ensures the constraint $E\{(\mathbf{W}^T\mathbf{z})^2\} = ||\mathbf{w}||^2 = 1$ is satisfied. Notice that, for example, if G is chosen to be $\log\cosh(y)$ its derivative is $\tanh(y)$ which is the same function as that which can be used in the Bell and Sejnowski method (see [32] for the connection between negentropy and mutual information ICA methods). Based on the previous discussion of fixed point methods, a simple fixed point iteration step can be written as:

$$\mathbf{w} \leftarrow \{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} \tag{2.57}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{2.58}$$

Unfortunately, the above iteration does not exhibit very fast convergence. Some modifications, shown in [32], based on an approximation of Newton's method, gives a much better iteration step:

$$\mathbf{w} \leftarrow E\{\mathbf{z}g\left(\mathbf{w}^T\mathbf{z}\right)\} - E\{g'\left(\mathbf{w}^T\mathbf{z}\right)\}\mathbf{w} \tag{2.59}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{2.60}$$

where g' is the derivative of g. This is the basic fixed point iteration used in FastICA.

$\mathbf{w}$ is initialized to a random vector and iteration continues until $\mathbf{w}$ in the first line of the iteration changes by less than a pre-defined small amount. Convergence proofs are shown in [32]. These show that FastICA converges (locally) up to the sign, to one of the rows of the inverse of the mixing matrix.

Note that this algorithm estimates only a single vector $\mathbf{w}$. Deflationary orthoganalization using the Gram-Schmidt orthogonalization technique can be used to compute the components one by one, very quickly. In this thesis, symmetric orthogonalization was employed, as it was never necessary estimate the components one at a time. In symmetric orthogonalization:

$$\mathbf{W} \leftarrow \left(\mathbf{W}\mathbf{W}^T\right)^{-1/2}\mathbf{W}. \tag{2.61}$$

In this approach, a single iteration of each $\mathbf{w}$ we wish to estimate is done in parallel. At the end of each iteration, $\mathbf{W} = (\mathbf{w}_1, \cdots, \mathbf{w}_m)^T$ is symmetrically orthogonalized. The implication of the orthogonal constraint on $\mathbf{W}$ will be discussed in the context of classification distance measures in Chapter 3.

2.4 Other Subspaces

It will be convienent for some of the following discussions on kernel learning and support vector machines to use the concept of an inner product space. Given a vector space V is defined over $\mathbb{R}^n$, a function $\langle \cdot, \cdot \rangle$ which maps $V \times V \rightarrow \mathbb{R}$ is an inner product if, $\forall a, b, c \in V$:

- $\langle a, a \rangle \geq 0$
- $\langle a, a \rangle = 0$ if and only if $x = 0$
- $\langle a + b, c \rangle = \langle a, b \rangle + \langle b, c \rangle$
- $\langle sa, b \rangle = s\langle a, b \rangle \forall$ scalars $s \in \mathbb{R}$

For the vector space $\mathbb{R}^n$ the standard inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T\mathbf{y}$.

For the present, the idea of a kernel function will be introduced and used without describing what characterizes them. This will follow in the discussion on support vector machines. A kernel function is a non-linear mapping of the vector space V onto $\mathbb{R}$:

$$k : V \times V \rightarrow \mathbb{R} \tag{2.62}$$

which can be decomposed as:

$$k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{y}) \rangle \tag{2.63}$$

with $\mathbf{x}$, $\mathbf{y} \in V$, and $\mathbf{\Phi} : V \rightarrow \tilde{F} \subseteq F$, where F is an inner product feature space.

2.4.1 Decorrelating In A High Dimensional Space (KPCA)

By defining a very simple map which maps $\mathbf{x}$ to itself, $\mathbf{\Phi}(\mathbf{x}) = \mathbf{x}$, PCA can be formulated as a kernel problem from the SVD of $\mathbf{X}$. Since $\mathbf{U} = \mathbf{\Sigma}^{-1}\mathbf{X}\mathbf{V}$, each left singular vector $\mathbf{u}_j$ can be written as a linear combination of the columns of the right singular vectors of $\mathbf{V}$ by a columnar representation of matrix multiplication:

$$\mathbf{u}_j = {\sigma_j}^{-1} \sum_{i=1}^{n} v_{i,j}\mathbf{x}_i \tag{2.64}$$

where $v_{i,j}$ is the (i,j)th element of $\mathbf{V}$ and $\mathbf{x}_j$ is the jth column of $\mathbf{X}$. Using dual variables $\alpha^j = {\sigma_j}^{-1}\mathbf{v}_j$, the projection of a new data point into the feature space can

be written as:

$$\mathbf{u}_j{}^T\mathbf{x} = \left\langle \sum_{i=1}^{n} \alpha_i{}^j \mathbf{x}_j, \mathbf{x} \right\rangle \tag{2.65}$$

$$= \sum_{i=1}^{n} \alpha_i{}^j \langle \mathbf{x}_i, \mathbf{x} \rangle \tag{2.66}$$

$$= \sum_{i=1}^{n} \alpha_i{}^j k(\mathbf{x}_i, \mathbf{x}) \tag{2.67}$$

where $k(\mathbf{x}_i, \mathbf{x})$ is a scalar function defined by the the inner product $\langle \mathbf{x}_i, \mathbf{x} \rangle$. Note that the matrix, $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, called the kernel matrix, is the covariance matrix of $\mathbf{X}$. As such, the vector $\mathbf{v}_j$ is an eigenvector of the kernel matrix and σ_j is the square root of the corresponding eigenvalue.

Replacing the original map $\mathbf{\Phi}(\mathbf{x}) = \mathbf{x}$ with a general $\mathbf{\Phi}(\mathbf{x})$ the kernel matrix is $\mathbf{K}_{i,j} = k(\mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j))$ and the above representation can be applied for the new mapping. Again, $\alpha^j = \sigma_j{}^{-1}\mathbf{v}_j$ however $\mathbf{v}_j$ is now an eigenvector and σ_j is the square root of the corresponding eigenvalue of the covariance matrix $\langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle$. This mapping defines kernel principal component analysis (KPCA). The projection of a new data point $\mathbf{x}$ into the space defined by KPCA is then:

$$\sum_{i=1}^{n} \alpha_i{}^j k(\mathbf{x}_i, \mathbf{x}) \tag{2.68}$$

under the redefinition of the kernel matrix $\mathbf{K}$. While this provides a definition of KPCA, it does not illustrate the utility of kernel methods. Very briefly, the main utility of kernels is that it is only necessary to use the kernel to calculate a new point in the KPCA space. The original mapping $\mathbf{\Phi}$ is not needed. Kernels which are simple to compute can then replace mappings which are computationally intensive. More details about kernels will be given in the discussion of support vector machines.

2.4.2 Fisher's Linear Discriminant

All of the previously discussed methods of extracting features from the data apply to data that is unlabelled — that is, it has not been assigned a-priori to belong to a particular class. Fisher's linear discriminant (FLD) extracts features by finding directions in data which occur as a result of maximizing the ratio of the between-class scatter and the within class-scatter. As such, the data is assumed to be grouped into classes before the application of FLD. Given a dataset of m-dimensional images formed by a set of N images $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, each of which assumed to belong to one of C classes $\{X_1, X_2, \cdots, X_C\}$, a within class scatter matrix $\mathbf{S}_w$ and a between class scatter matrix $\mathbf{S}_b$ can be defined:

$$\mathbf{S}_w = \sum_{i=1}^{C} \sum_{\mathbf{x} \in \mathbf{X}_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T \tag{2.69}$$

$$\mathbf{S}_b = \sum_{i=1}^{C} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \tag{2.70}$$

where $\boldsymbol{\mu}$ is the sample mean for the entire dataset and $\boldsymbol{\mu}_i$ is the sample mean for class X_i.

A linear transformation $\mathbf{V}$ which maximizes the between class scatter while minimizing the within class scatter can be found from maximizing the ratio (a generalized Rayleigh quotient)

$$\frac{|\mathbf{V}^T \mathbf{S}_b \mathbf{V}|}{|\mathbf{V}^T \mathbf{S}_w \mathbf{V}|}. \tag{2.71}$$

The solution to this maximization problem is found from the generalized eigenvalue problem

$$\mathbf{S}_b \mathbf{v}_i = \lambda_i \mathbf{S}_w \mathbf{v}_i. \tag{2.72}$$

and is the set of eigenvectors of the matrix ${\mathbf{S}_w}^{-1}\mathbf{S}_b$ if $\mathbf{S}_w$ is non-singular. More generally, the solution is the set of generalized eigenvectors of the matrix pencil $(\mathbf{S}_b, \mathbf{S}_w)$. If $\mathbf{S}_w$ is rank deficient, there may be a finite number of eigenvalues, an infinite number

of eigenvalues, or none at all.

Unfortunately, when applying FDA directly to the image vectors produced for pattern recognition problems, the rank of $\mathbf{S}_w$ is determined by the number of training images N, which in all likelihood will be far smaller than the number of pixels in the images, m. Thus $\mathbf{S}_w$ is almost always non-invertible. In [74], the singularity of $\mathbf{S}_w$ was dealt with by reducing the dimensionality of the image set using PCA so that $\mathbf{S}_w$ is full rank and applying FLD on the principal component coefficients. Thus, if the set of eigenvectors produced with ${\mathbf{S}_w}^{-1}\mathbf{S}_b$ is termed $\mathbf{V}_{\text{fisher}}$ and $\mathbf{V}$ corresponds to the eigenvectors produced via PCA on the training images, the Fisher projection coefficients can be produced from an image $\mathbf{x}_i$ via:

$$\mathbf{y}_i = \mathbf{V}_{\text{fisher}}\mathbf{V}\mathbf{x}_i. \tag{2.73}$$

2.5 Feature Selection

PCA techniques intrinsically provide a means of selecting the most significant coefficients which make up the feature vectors. The corresponding eigenvalues indicate the variance represented by each coefficient and variance can be used as a measure of the significance of the coefficent. In other words, dimensionality reduction is performed by selecting the basis images which have coefficients with the largest variance.

2.5.1 Floating Search

When ICA is used for feature extraction, no simple technique exists for selecting the most significant features. The general problem of choosing the best k features out of n total has been examined extensively. Optimal search techniques exist, such as branch and bound algorithms. Branch and bound algorithms rely on the property that for two subsets of the variables X and Y, $X \subset Y \Rightarrow J(X) < J(Y)$ where J is the feature selection criterion. In the case where this cannot be guaranteed, sub-optimal search techniques such as a floating search [47] can be employed. This technique was used

in this thesis to select the k best features which maximize the inter-object Euclidean distance in feature space. Thus, the criterion function to be optimized in the feature search is the sum of the distances between the feature vectors of all of the training objects. Flexibility is built into this search to allow for the selection of previously discarded features and the discarding of previously selected ones. While it is possible to derive a heuristic technique by selecting subsets of features and selecting the best by a majority vote, the technique of floating search is a more rigorous approach. However, it is not guaranteed to find the optimal k features, since for this application the funtion of inter-object Euclidean distance is not guaranteed to be monotonically increasing. A brief, qualitative review of feature selection follows. Quantitative discussions can be found in [75]. Note that both PCA and search techniques can be combined for dimensionality reduction. PCA can be applied a-priori to the dataset to reduce the dimensionality and ICA can be applied on the reduced dimensionality data. The data can then be further reduced in dimensionality by selecting ICA features with a floating search. This technique has been applied in Chapter 5 of this thesis.

In the context of subspace pattern recognition, given a set of features comprised of n dimensional feature vectors which are the coefficients of basis images, the goal of feature selecton is to find the best k coefficients (or correspondingly, the best k basis images). The "best" features are the ones which maximize a criterion function which depends on the features. Sequential forward selection (SFS) adds new features (coefficients from the corresponding basis images) to a feature set one at a time until the criterion function is maximized. In general, the SFS appends the feature from the list of those not selected previously which, when appended to the list of currently selected features, yields the maximum value of the criterion function. This continues until the best feature to add no longer makes the criterion larger or a maximum number of features have been selected. The same process can be applied in reverse by starting with all features and removing one at time. This is called sequential backward selection. A problem with SFS (and SBS) techniques is that once a feature is selected (removed) it cannot be removed (re-selected). There is no guarantee that

there isn't a combination of other features which haven't been selected yet which would provide a larger criterion function value if this feature hadn't been selected.

To avoid this problem, it is advantageous to be able to backtrack in the search. The number of features added and removed per iteration of the search is allowed to float, thus inspiring the name floating search. In a forward version of this type of search, three steps are involved:

1. **Feature Addition** - The feature from the remaining unselected set which increases the criterion the most is added to the selected set.

2. **Feature Test** - The feature from the selected set which reduces the criterion the least is found. If this is the same as the one previously added in step one, keep this feature and return to step one. Otherwise, remove it from the selected set.

3. **Feature Removal** - Continue removing features (in opposite order than they were previously added) from the selected set and testing the criterion until removing a feature produces a smaller criterion than before it was removed. At this point, return to step 1.

Similar steps can be derived for a backward floating search.

Chapter 3

Classification

3.1 PCA vs. ICA and Distance Measures

There has been a large amount of debate in current literature over the relative performance of PCA and ICA for classification. Most often, the performance has been compared under the application of face recognition, as mentioned in Chapter 1. Two issues are standouts for this comparison. The first is the choice of the ICA architecture. The second is the choice of the distance metric. Some simple theory can show that regardless of whether ICA or PCA is used to derive a basis for the purposes of feature extraction, when an ℓ_2 norm or an inner product (cosine angle) metric is used, the classification results will be identical in theory.

It was shown in Chapter 2 that the difference between PCA and ICA is one of a rotation (orthonormal transformation) by $\mathbf{W}_o$ (for ICA with statistically independent coefficients — see Equation 2.22), or by $\mathbf{W}_z$ (for ICA with a statistically independent demixing matrix — see Equation 2.29), provided that ICA was performed on whitened data. Although $\mathbf{W}_o$ and $\mathbf{W}_z$ are different matrices, they are both othonormal. For the remainder of this chapter, the basis derived from ICA will be simply referred to as $\mathbf{W}_o$, to indicate the orthonormal nature of this matrix.

3.2 Euclidean Distance Classification (ℓ_2 norm)

Where the low-dimensional representation of Equation (2.1) is used to represent each image in both the dataset and an unknown test image, we can define $\mathbf{y}_i$ and $\mathbf{y}_t$ to be the low-dimensional representation of the ith object's features and the unknown test image features respectively. The test image can be classified by employing a minimum Euclidean distance metric:

$$\min_i d(\mathbf{y}_i, \mathbf{y}_t) = \min_i (\mathbf{y}_i - \mathbf{y}_t)^T (\mathbf{y}_i - \mathbf{y}_t). \tag{3.1}$$

In the ideal case, the function d has a unique minimum, which occurs when i is the index of the matching object. The function is not necessarily unique in the argument i, since multiple images $\mathbf{y}_i$ may correspond to the same minimum of d. Additionally, classification errors may occur where the minimum occurs at a non-matching object. Such errors may occur for objects of similar appearance or when illumination conditions dramatically change the appearance of an object.

Considering the classification from the perspective of the original data, with $\mathbf{x}_i$ being each image in the dataset, $\mathbf{x}_t$ being an unknown test image and $\mathbf{Q}_o$ and $\mathbf{W}_o$ derived from PCA and ICA on the dataset represented by $\mathbf{x}$, the distance function ℓ_2 norm classification for PCA is:

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_t) &= \mathbf{x}_i{}^T \mathbf{Q}_o \mathbf{Q}_o{}^T \mathbf{x}_i - 2\mathbf{x}_i{}^T \mathbf{Q}_o \mathbf{Q}_o{}^T \mathbf{x}_t + \mathbf{x}_t{}^T \mathbf{Q}_o \mathbf{Q}_o{}^T \mathbf{x}_t \\ &= \mathbf{x}_i{}^T \mathbf{P} \mathbf{x}_i - 2\mathbf{x}_i{}^T \mathbf{P} \mathbf{x}_t + \mathbf{x}_t{}^T \mathbf{P} \mathbf{x}_t \end{aligned} \tag{3.2}$$

where $\mathbf{P}$ is the projector matrix $\mathbf{Q}_o \mathbf{Q}_o{}^T$. The distance function for ℓ_2 norm classification for ICA is:

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_t) &= \mathbf{x}_i{}^T \mathbf{Q}_o \mathbf{W}_o{}^T \mathbf{W}_o \mathbf{Q}_o{}^T \mathbf{x}_i - 2\mathbf{x}_i{}^T \mathbf{Q}_o \mathbf{W}_o{}^T \mathbf{W}_o \mathbf{Q}_o{}^T \mathbf{x}_t + \mathbf{x}_t{}^T \mathbf{Q}_o \mathbf{W}_o{}^T \mathbf{W}_o \mathbf{Q}_o{}^T \mathbf{x}_t \\ &= \mathbf{x}_i{}^T \mathbf{P} \mathbf{x}_i - 2\mathbf{x}_i{}^T \mathbf{P} \mathbf{x}_t + \mathbf{x}_t{}^T \mathbf{P} \mathbf{x}_t. \end{aligned} \tag{3.3}$$

Thus it is shown that PCA and ICA will exhibit identical classification results when an ℓ_2 norm is used as a distance measure for classification and PCA is used for whitening and dimensionality reduction prior to the calculation of ICA. Another interesting note is that the difference in the ℓ_2 metric in classifying dimensionally reduced data and the original data can be reduced to the effect of the projection matrix $\mathbf{P}$ on the inner products. If no dimensionality reduction is used, $\mathbf{P} = \mathbf{I}$.

As mentioned in Section 2.5, PCA can be used to provide whitening and a-priori dimensionality reduction and search techniques can be applied to further reduce the dimensionality of the ICA feature space. It is important to note that when this is done, the PCA and ICA bases span different spaces and are therefore not related by a rotation. Therefore, ℓ_2 norm distance metrics are a possible choice where a difference in recognition performance may be observed.

3.3 Inner Product Classification

In a similar manner to ℓ_2 norm classification, the normalized inner product of the data points can be used as a distance metric. Using $\mathbf{y}_i$ and $\mathbf{y}_t$ as defined above, the test image is classified with normalized inner product by:

$$d(\mathbf{y}_i, \mathbf{y}_t) = \frac{\mathbf{y}_i{}^T \mathbf{y}_t}{\|\mathbf{y}_i\| \, \|\mathbf{y}_t\|}. \tag{3.4}$$

As above, from the perspective of the original data, with PCA:

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_t) &= \frac{\mathbf{x}_i{}^T \mathbf{Q}_o \mathbf{Q}_o{}^T \mathbf{x}_t}{\|\mathbf{Q}_o{}^T \mathbf{x}_i\| \, \|\mathbf{Q}_o{}^T \mathbf{x}_t\|} \\ &= \frac{\mathbf{x}_i{}^T \mathbf{P} \mathbf{x}_t}{\|\mathbf{x}_i\| \, \|\mathbf{x}_t\|} \end{aligned} \tag{3.5}$$

and with ICA:

$$\begin{aligned} d(\mathbf{x}_i, \mathbf{x}_t) &= \frac{\mathbf{x}_i{}^T \mathbf{Q}_o \mathbf{W}_o{}^T \mathbf{W}_o \mathbf{Q}_o{}^T \mathbf{x}_t}{\left\|\mathbf{W}_o \mathbf{Q}_o{}^T \mathbf{x}_i\right\| \left\|\mathbf{W}_o \mathbf{Q}_o{}^T \mathbf{x}_t\right\|} \\ &= \frac{\mathbf{x}_i{}^T \mathbf{P} \mathbf{x}_t}{\|\mathbf{x}_i\| \|\mathbf{x}_t\|}. \end{aligned} \tag{3.6}$$

Again, it is shown that PCA and ICA exhibit identical classification results with an inner product distance metric.

3.4 Differences in Euclidean Distance Classification between PCA and ICA

It is very instructive to note a comment in [44] which illustrated that FastICA alone performed worst (closest to PCA). In light of what was just illustrated, this is not at all surprising. FastICA forces the components to be geometrically orthogonal, since, in the whitened space, statistical orthogonality (decorrelation) is equivalent to geometric orthogonality. Other algorithms, such as infomax [27], do not. From the point of view that independence implies statistical decorrelation and thus statistical orthogonality in the whitened space, the FastICA algorithm is more accurately representing the independent components. However, in practice it is not possible to ensure that all extracted components are exactly independent. FastICA ensures that all moments up to the second are identically zero. The infomax algorithm does not ensure that this is the case. In both cases, however, the remainder of the statistical moments are approximated. As such, the practical methods of approximation for each algorithm determine how accurately each algorithm performs ICA. This key point will be illustrated with a number of applications which use FastICA, where the recognition results will be shown to be identical between PCA and ICA, due to the exact decorrelation that is performed with that algorithm.

3.5 More Representative Distance Measures

Logical choices for distance measures which are not invariant under orthonormal transformations include the ℓ_1 norm and Mahalanobis distance. While the merits of each of these has been argued in [76] and elsewhere, more sophisticated classification techniques are considered in this thesis. Specifically, the Support Vector Machine and projection into a high dimensional space is examined in considerable detail. Since a non-linear transform of the coefficients to be classified is applied, any direct relationship between the subspaces in the linear space cannot be applied in the transformed space.

3.6 Support Vector Classification

To perform classification with a linear SVM, a labeled set of feature vectors $\{\mathbf{x}_i, y_i\}$ is constructed for all l feature vectors in the training dataset. The class of feature $\mathbf{x}_i$ is defined by $y_i = \{1, -1\}$. If the data is assumed to be linearly separable, the SVM attempts to find a separating hyperplane with the largest margin. The margin is defined as the shortest distance from the separating hyperplane to the closest data point. If the training data follows:

$$y_i(\mathbf{x}_i\mathbf{w} + b) - 1 \geq 0 \quad \forall i \tag{3.7}$$

then the points for which Equation 3.7 is an equality lie on one of the hyperplanes $\mathbf{x}_i\mathbf{w} + b = 1$ and $\mathbf{x}_i\mathbf{w} + b = -1$. The margin can be shown to be [10]:

$$\text{Margin} = \frac{2}{\|\mathbf{w}\|}. \tag{3.8}$$

The SVM attempts to find the pair of hyperplanes which gives the maximum margin by minimizing $\|\mathbf{w}\|^2$ subject to the constraints on $\mathbf{w}$ given in Equation 3.7. Reformulating the problem using the Lagrangian, the primal form of the objective function

can be written:

$$L_p = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{l}\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1) \tag{3.9}$$

where the α_i are Lagrange multipliers. Differentiating L_p with respect to $\mathbf{w}$ and b yields:

$$\sum_{i=1}^{l}\alpha_i y_i = 0 \qquad \text{and} \qquad \mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i \tag{3.10}$$

Substituting into Equation 3.9 gives:

$$L_d = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i{}^T\mathbf{x}_j \tag{3.11}$$

This equation is the dual form of the Lagrangian. It is maximized with respect to the α_i subject to the constraints:

$$\alpha_i \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{l}\alpha_i y_i = 0 \tag{3.12}$$

A simple generalization allows the linear case to be extended to the non-linear case. As mentioned briefly in Chapter 2, a kernel function $k(\mathbf{x}, \mathbf{y})$ is an inner product in a feature space where $k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$. Valid kernel functions must be able to be expressed as the aforementioned inner product in feature space. As such, they must satisfy Mercer's conditions, which, stated briefly, are that $k(\mathbf{x}, \mathbf{y})$ must equal $k(\mathbf{y}, \mathbf{x})$ and [75]:

$$\int k(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) d\mathbf{x} d\mathbf{z} \geq 0 \tag{3.13}$$

for all functions $f(\cdot)$ which have finite energy ($\int |f(\mathbf{x})|^2 d\mathbf{x} < \infty$). As such, the kernel function can be expanded as:

$$k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty}\lambda_i \hat{\phi}_i(\mathbf{x})\hat{\phi}_i(\mathbf{y}) \tag{3.14}$$

where λ_i and ϕ_i are the eigenvalues and eigenfunctions satisfying:

$$\int k(\mathbf{x}, \mathbf{y})\phi_i(\mathbf{x})d\mathbf{x} = \lambda_i\phi_i(\mathbf{x}) \tag{3.15}$$

The function $\hat{\phi}_i(\cdot)$ is normalized so that its total energy equals 1.

By applying a non-linear mapping to the feature vectors, a non-linear version of the SVM can be written with the use of the kernel function. Therefore the expression to optimize for a non-linear SVM can be written (in its dual form) as [10]:

$$L_d = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{3.16}$$

where $k(\mathbf{x}, \mathbf{x}')$ is a kernel function satisfying Mercer's conditions. An example kernel function (the one used herein) is the Gaussian radial basis function:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \tag{3.17}$$

where σ is the standard deviation of the kernel's exponential function. The decision function (the function which determines to which class ($+1$ or -1) the feature vector is classified to) of the SVM can be described by:

$$f(\mathbf{y}) = \text{sgn}\left[\sum_{i=1}^{l} y_i\alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right]. \tag{3.18}$$

Note that only for data points $\mathbf{x}$ which lie closest to the optimal decision boundary are the corresponding α_i non-zero, and these are called support vectors. All other parameters α_i are zero. As such, any modification of the data points which are not support vectors will have no effect on the solution. This indicates that the support vectors contain all the necessary information to reconstruct the decision boundary. An estimation of classification error for training with cross-validation can be made

by [10]:

$$E[P(\text{error})] = \frac{\text{SV}}{N} \tag{3.19}$$

where SV is the number of support vectors, N is the number of training data items, and $P(\cdot)$ denotes the probability function. Cross validation determines classification error by using all but 1 of l feature vectors in the training data and testing on the remaining feature vector. This test is repeated for all subsets of feature vectors of size $l-1$. As a result of this estimate of classification error, it can be concluded that reducing the number of support vectors achieves better generalization, since the reduced number of support vectors can still reproduce the same hyperplane or non-linear decision boundary. Error bounds on the number of support vectors and the margin are described in [10].

For data that is non-separable, a modification is made to the original problem definition to allow the margin constraints to be violated. Using the example from the linear SVM, the original margin constraints are rewritten as:

$$y_i(\mathbf{x}_i\mathbf{w} + b) - 1 \geq 0 - \gamma_i \quad \forall i \tag{3.20}$$

where $\xi_i \geq 0$ are slack variables which represent the amount by which the margin constraints can be violated. The new primal representation, including the slack variables becomes:

$$L_p = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{l} \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1) + \frac{C}{2}\sum_{i=1}^{l} {\xi_i}^2 \tag{3.21}$$

with C a scaling parameter on the effect on the slack variables. The dual form of this becomes:

$$L_d = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i\alpha_j y_i y_j {\mathbf{x}_i}^T\mathbf{x}_j + \frac{1}{C}\delta_{ij} \tag{3.22}$$

where δ_{ij} is the Kronecker function defined to be 1 if $i = j$ and 0 otherwise. In the non-linear case, it easy to see that this simply has the effect of changing the kernel

function:

$$k'(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') + \frac{1}{C}\delta(\mathbf{x}') \tag{3.23}$$

This has the effect of adding $\frac{1}{C}$ to the eigenvalues of the kernel matrix $\mathbf{K}$ thus improving the conditioning of the optimization problem.

3.6.1 Detection Of Outliers

It is possible to use a support vector machine to give a description of a set of all feature vectors. More specifically, a general approach for outlier detection and rejection is to find the sphere with minimum volume containing all feature vectors which belong to an object while all others fall outside the sphere. This idea was described in [17], in the context of the support vector classifier. The problem is set up as a one class classification problem, where a set of training data defines the description of an object by virtue of its feature vectors. A decision function is found either accepts or rejects test object feature vectors as members of the training object set. To construct a decision boundary, the following error function is minimized (including slack variable ξ to adjust the number of outliers in the training set):

$$\varepsilon(R, \mathbf{a}, \xi) = R^2 + C\sum_i \xi_i \tag{3.24}$$

where $\mathbf{a}$ is the center of the sphere, R is its radius and C describes the trade-off between the volume of the sphere and the data errors. The solution is constrained so that almost all objects reside in the sphere of radius R:

$$|\mathbf{x}_i - \mathbf{a}|^2 \leq R^2 + \xi_i. \tag{3.25}$$

The Lagrangian can be constructed by incorporating the constraints into the error function ε to obtain a cost function L to minimize:

$$L(R, \mathbf{a}, \alpha_i, \xi_i) = R^2 + C\sum_i \xi_i - \sum_i \alpha_i(R^2 + \xi_i - ((\mathbf{x}_i - \mathbf{a})^T(\mathbf{x}_i - \mathbf{a})^T)) - \sum_i \gamma_i \xi_i \quad (3.26)$$

where $\alpha_i \geq 0$ and $\gamma_i \geq 0$ are Lagrange multipliers. In a manner similar to the previous discussion on support vector classification, the minimization of L can be rewritten as the maximization of L_D where

$$L_D = \sum_i \alpha_i \mathbf{x}_i{}^T \mathbf{x}_i - \sum_{i,j} \alpha_i \alpha_j \mathbf{x}_i{}^T \mathbf{x}_j \quad (3.27)$$

under the constraints $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i = 1$. The decision function is constructed by determining when the distance from the center of the sphere to a test feature vector is less than the radius (expressed in terms of the support vectors). The test feature vector $\mathbf{z}$ is accepted as a member of the training object set if:

$$\mathbf{z}^T\mathbf{z} - 2\sum_i \alpha_i \mathbf{z}^T \mathbf{x}_i + \sum_{i,j} \alpha_i \alpha_j \mathbf{x}_i{}^T \mathbf{x}_j \leq R^2 \quad (3.28)$$

This can naturally be extended to a kernel method by simply replacing the inner product with the kernel function:

$$K(\mathbf{z}, \mathbf{z}) - 2\sum_i \alpha_i K(\mathbf{z}, \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \leq R^2 \quad (3.29)$$

A key point from this result is that the largest non-zero data points for which their support vector coefficients equal C are considered outliers. This data description formalizes the notion that spherical data representations in a linear support vector machine minimize the number of support vectors and minimize the outlier acceptance as described in [17]. In the non-linear case, minimal volume hyperspheres generated by the description boundaries from kernel functions can be used to provide more

flexibility in the decision boundary. It also formalizes a notion that the largest support vector coefficients can be seen to define a set of outliers.

Viewed from the perspective of convex optimization theory, a standard result is that the optimal Lagrange multipliers are the local sensitivities of the optimal value with respect to perturbations in the constraints. The constraints for the linear support vector classifier, given in Equation 3.7 define the position of the separating hyperplanes. Perturbing each constraint (corresponding to each data point) corresponds to small changes in the positions of the data points. The constraint corresponding to the largest Lagrange multipliers (largest support vector values) will have the largest effect on the optimal value. In fact, since only the support vector data points have non-zero Lagrange multipliers, they provide all of the effect on the optimal value. The optimal value of the SVM optimization problem directly determines the margin of the classifier by virtue of Equation 3.8 since the SVM attempts to achieve maximum margin. So the position of the decision boundary is only sensitive to the position of the support vector data points. This high degree of sensitivity to a few discordant data points and insensitivity to inliers is precisely the measure that drives the development of robust statistical techniques. It seems clear, then, that the SVM can be used as an outlier detector.

From the perspective of feature extraction, if a more compact data representation which fits into a hypersphere of reduced volume could be found, this would serve to provide better generalization and outlier rejection by virtue of Equation 3.19. One possible way to achieve this is by modifying the support vectors to move possible outliers in the training data (defined by the support vectors) toward the class mean thus producing a more compact class. This has the direct advantage of making the position of the class boundary much less sensitive to these outlying values. In fact, these outlying training points are simply other examples of the general class and there is little reason to treat them separately from other training examples. This movement of training feature vectors is exploited by performing linear regression on the vectors to find a basis which will optimally (in a least squares sense) transform

the original data into the modified case. This process can be performed iteratively by determining the decision boundary for the training feature vectors, modifying the feature vectors which are support vectors, determining a new basis, calculating new training feature vectors, repeating the determination of the decision boundary and so on. This algorithm will be described in detail in the next section.

3.7 Classification by Modifying the Support Vectors

3.7.1 Feature Scaling By Coefficient Modification

In [17] the support vector data description was developed as a minimum volume containing all objects of the dataset. This minimum volume representation can be exploited provided that the data is rescaled, as in [18]. The general idea for this technique is that by minimizing the volume of feature space, generalization is improved. An algorithm described below presents another option for minimizing the volume of the feature space. It uses the support vectors as an indication of the outer bounds of the feature space and moves the data toward the class mean (by an amount proportional to the support vector coefficients) to shrink the volume in this direction. A subspace that achieves this smaller volume can thus be learned from the modified coefficients. The resulting coefficients are assumed to have resulted from a linear combination of the input data. As such, the basis set is found by linear regression. In the standard least squares model, each coefficient independently comes from the linear combination of the input data. In this case, the regression was performed by canonical correlation [77], thus finding a basis that assumes a multiple output model, where we pool the observations of the coefficients to find a basis. Figure 3.1 shows a block diagram of the method for adjusting the basis (S).

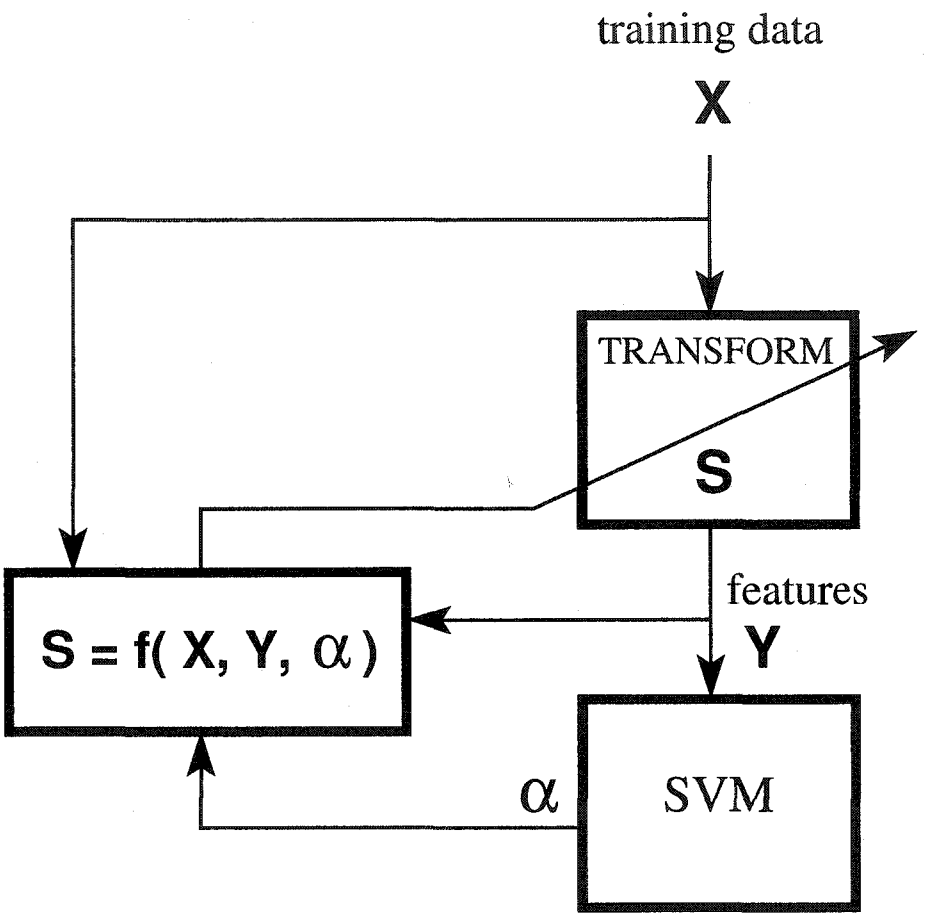


Figure 3.1: Learning a basis from the support vectors

Using the class means for l features total in 2 classes:

$$\mathrm{C}_n = \frac{1}{l/2} \sum_{i=1}^{l/2} \mathrm{Y}_{i,n} \tag{3.30}$$

where $\mathrm{Y}_{i,n}$ are the ith feature in the nth class learned at each iteration with $n = 1, 2$, a matrix of the class means:

$$\mathrm{Y}_{mean} = \left[\begin{array}{cccccc} \mathrm{C}_1^1 & \cdots & \mathrm{C}_1^{l/2} & \mathrm{C}_2^1 & \cdots & \mathrm{C}_2^{l/2} \end{array} \right] \tag{3.31}$$

a matrix scalar of the support vector coefficients α_i,

$$\Lambda = \begin{bmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_n \end{bmatrix} \tag{3.32}$$

and an initial $m \times n$ matrix $\mathbf{S_0}$,

$$\mathbf{S_0} = \begin{bmatrix} \mathbf{I} \\ \\ \mathbf{0} \end{bmatrix} \tag{3.33}$$

where m is the length of the basis vectors and n is the dimensionality of the subspace, boundary feature vectors can be moved toward their class means and basis vectors $\mathbf{S}$ can be learned to fit the new features. Δ refers to the "change in" margin from iteration to iteration, as calculated by a simple difference. The diag$(\cdot)$ function refers to forming a diagonal matrix of elements in the argument.

Algorithm 1 Compact Support Vector Representation (CSVR) Algorithm

1: initialize $\mathbf{S}$ as $\mathbf{S_0}$.
2: initialize:

$$\mathbf{Y}_{train} \leftarrow \mathbf{S}^T\mathbf{X}_{train}$$
$$\mathbf{Y}_{test} \leftarrow \mathbf{S}^T\mathbf{X}_{test}$$

3: initialize Λ to the identity matrix.
4: **repeat**
5: move the support vectors toward the mean by an amount proportional to the support vector α by:

$$\mathbf{Y}_{train} \leftarrow \mathbf{Y}_{train} - \Lambda(\mathbf{Y}_{train} - \mathbf{Y}_{mean})$$

6: recalculate $\mathbf{S}$ by:

$$\mathbf{S} \leftarrow (\mathbf{X}^T\mathbf{X})^{-1})\mathbf{X}^T(\mathbf{Y}_{train}\mathbf{U})\mathbf{U}^+$$

where $+$ denotes pseudo-inverse and $\mathbf{U}$ are the left singular vectors of the generalized SVD of $\mathbf{X}$ and $\mathbf{Y}_{train}$ (canonical correlation regression)
7: calculate:

$$\mathbf{Y}_{train} \leftarrow \mathbf{S}^T\mathbf{X}_{train}$$

8: apply SVM to determine margin, Δ margin and $\Lambda = \text{diag}(\alpha_i)$
9: **until** (Δ margin $< .0001$) or (margin > 1.35).

Upon convergence of the algorithm, the basis $\mathbf{S}$ can be used to find features for test data by:

$$\mathbf{Y}_{test} \leftarrow \mathbf{S}^T \mathbf{X}_{test}$$

To classify the test data, data pairs $(\mathbf{y}_{train_i}, y_i)$ are defined, and the support vector decision boundary for the training data is used to classify $\mathbf{Y}_{test}$.

The steps of the CSVR algorithm are summarized in Algorithm 1. To initialize, the basis vectors $\mathbf{S}$ are set to an n dimensional standard basis. To speed up convergence, the algorithm can be initialized by a basis found from the training data $\mathbf{X}_{train}$ using principal or independent components, Additionally, the low-dimensional data representations are initialized ($\mathbf{Y}_{train}$ and $\mathbf{Y}_{test}$) and $\mathbf{\Lambda}$ is set to the identity matrix (lines 1, 2, and 3).

The first step in the iteration (line 5) moves all training data toward its class mean by an amount proportional to its support vector coefficient α. Support vectors will be set to the class mean for $\alpha = 1$ and the majority of the rest of the training data will be unmodified. The basis vectors $\mathbf{S}$ are then calculated to fit the modified training data set through canonical correlation regression, as shown in line 6. In line 7, the new test $\mathbf{Y}_{test}$ and training $\mathbf{Y}_{train}$ data sets are derived from their projections into the modified basis vectors. In the final step in the iteration (line 8), the newly calculated training subspace coefficients are classified by a SVM, which provides a new $\mathbf{\Lambda}$ for the next iteration.

After each iteration, the classes become more compact, with less effect from outlying points (which have been previously moved toward the mean) and the basis is learned from the regression on the coefficients. The compactness of the classes is illustrated by a steady increase in margin and the simplified shape of the classes is exemplified by a steady decrease in the number of support vectors. At the point where no further improvement in margin occurs, few data points are moved, since the number of support vectors has reached a small value. This condition terminates

the iteration (line 9). The margin change termination threshold was chosen empirically. In a large number of cases, the maximum achievable margin is reached. In this case, the algorithm is terminated slightly early, which provides a significant decrease in iteration time. Chapter 6 will use this algorithm for an experiment in classifying face images and will show that the nature of correlated image sets is such that this technique of volume reduction produces regular shaped data sets which can be readily bounded by a small number of support vectors.

Chapter 4

Position and Orientation Measurement with ICA

4.1 Position Measurement

Over the past ten or so years, PCA has been employed for the application of measuring the position of objects or cameras in one dimensional or two dimensional space. The methodology was described in detail in [37]. An important presupposition of this idea is that the manifold that is generated from the training images is "smooth" (not necessarily in the mathematical sense where constraints could be put on the derivative of the curve) so that interpolation by linear or spline methods could be performed to increase the accuracy. PCA provides just such a manifold. In [37] Nayar, Nene and Murase include three dimensional plots of the first 3 coefficients from a position measurement application. It would appear that the smoothness is an inherent characteristic of the coefficients extracted with PCA. A discussion of this point will be provided in the experiment below designed to illustrate this point.

While the aforementioned PCA based technique has been shown to work quite well when the test images (images which represent the position to be measured) lie close to the test manifold, significant problems occur when this is not the case. This can result

when the test images are occluded or the lighting has changed significantly. It seems reasonable to assume that other subspaces may provide manifolds that are perhaps somewhat "invariant" to occlusion or illumination. This invariance could arise from coefficients which better represent the essential features of the training objects thus making the test images' position closer to the manifold despite changes in imaging conditions. This hypothesis will be tested in detail below. Other than published work from this thesis, to the author's knowledge, no other detailed examination of this supposition exists in the literature. As the experiments will show, regardless of the shape of the manifold and the subspace technique used, there is always a significant degradation of performance when imaging conditions change from training to test and that not much improvement can be expected by simply changing subspaces.

ICA was tested for this application in a modality that should offer little overall difference in performance from PCA. Specifically, the ICA basis was dimensionally reduced with PCA. Indeed, this is shown to be the case, providing experimental verification of the result mentioned in Chapter 2, although there is some variability, likely due to the interpolation between coefficients. The purpose of this experiment in the thesis, however, is to illustrate a specific characteristic of the PCA and ICA basis for this application. Specifically, an important result arises from the examination of the kurtosis of the resulting coefficients which indicate how well the basis images correlate with the training images. Much more will be said about this in the discussion to follow.

4.2 Comparison of Subspaces for Position Measurement

To demonstrate the performance of different subspace methods for determining relative camera position, several experiments were performed. A CCD camera was mounted on a XY table, which is computer controlled with a resolution of 5 μm.

Two objects with differing visual characteristics were placed directly below the camera. The movement of the camera was lateral to the objects to simulate the determination of the position of a camera equipped robot end-effector for performing a task, such as grasping, welding or drilling the object. For simplicity of demonstration, the movement of the camera was limited to planar translation with a range of motion of 20 mm by 20 mm. The original images acquired were 320 by 240 pixels and then downsampled to 80 by 60 pixels.

The first object tested, object A, was a Pentium 3 mainboard, with the view of the camera corresponding to a section relatively rich in a variety of geometric features. The images corresponding to the four corner positions of the camera are shown in Figure 4.1(a)-(d). Note that camera movement producing predominantly horizontal image flow will be termed the x direction; camera movement producing predominantly vertical image flow will be termed the y direction.

The second object used, object B, was a car part that was part of the bumper assembly. The portion of the car part viewable by the camera contained, as the only feature, a hole in the part with both straight and curved edges. The images corresponding to the four corner positions of the camera movement range are shown in Figure 4.2(a)-(d). The simple visual characteristics of this object were chosen as a contrast to those of the previous object.

For each object, a set of 289 training images equally spaced in a 17 by 17 grid (1.25 mm between training images in each direction) over the total camera movement range was acquired. The training images were acquired with a 75 watt light source placed directly above the object, beside the camera. These training images were used in all the subspace methods to derive their projections.

To test the positional accuracy of the different subspace methods, two sets of test images were acquired. The first, consisting of 200 images with the camera moved randomly throughout the proscribed movement range, was acquired with the light in the same position as when the training images were acquired. The second, also consisted of 200 images with the same random camera locations, was acquired with

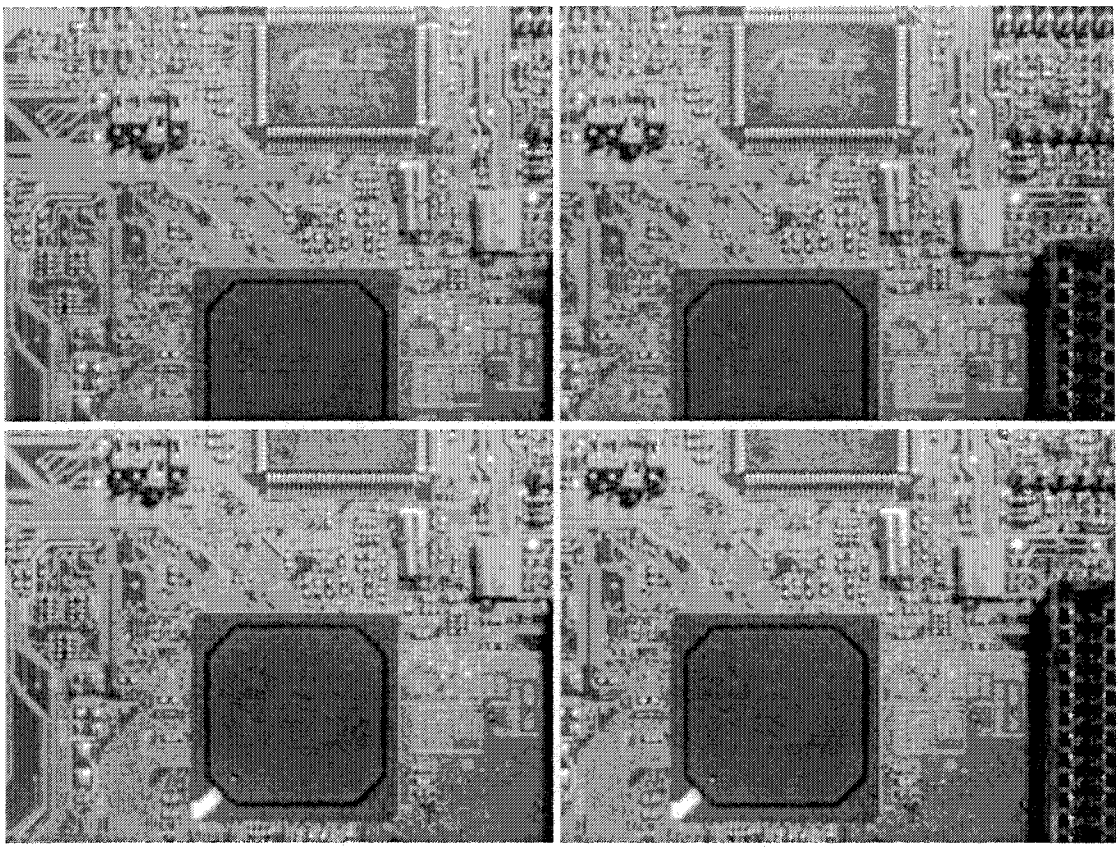


Figure 4.1: Range of Camera Movement, Object A

a light source of the same type as used for the training images emanating from the right side, offset along the x axis at about 45 degrees from the overhead position, to allow comparison of performance with differing lighting conditions.

For pattern recognition, a matrix $\mathbf{T}$ is used as a basis for a low-dimensional representation:

$$\mathbf{y}_i = \mathbf{T}^T \mathbf{x}_i \tag{4.1}$$

thus correlating each data vector $\mathbf{x}_i$ with a subset of the columns of the linear transformation matrix $\mathbf{T}$ which represent significant features in the data. For this experiment, this matrix was determined by using four different methods — PCA, ICA, KPCA and FLD. The details of computing this basis for each of these methods is found in Chapter 2. The vector $\mathbf{y}_i$ is then a set of coefficients which represent the data vector $\mathbf{x}_i$ in a

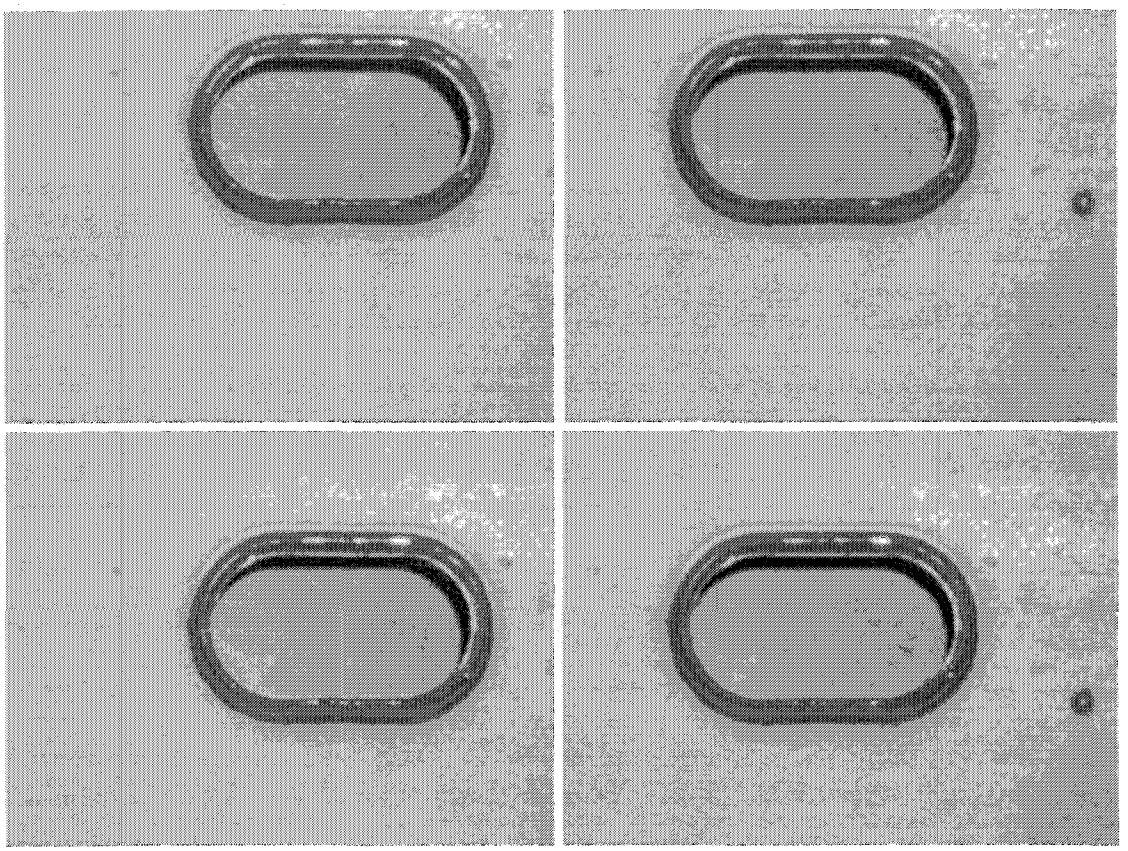


Figure 4.2: Range of Camera Movement, Object B

low-dimensional subspace. ICA was employed in 2 modes — statistically independent basis (ICA 1) and statistically independent coefficients (ICA 2).

To perform the position determination for each subspace method, a set of coefficients was generated for the set of training images. These coefficients were interpolated linearly to provide a second set of coefficients corresponding to 401 by 401 positions. During the initial matching process the current image's coefficients $\mathbf{y}$ are compared with coefficients of the training images $\mathbf{Y}_{training}$ via a Euclidean distance nearest neighbor match. Subsequently, a second Euclidean nearest neighbor match is performed with the interpolated projection coefficients $\mathbf{Y}_{interpolated}$ surrounding the matching training image. Using the position of the XY table, an error was calculated as the difference between the subspace position measurement and the XY table

position. The error was reported separately for the x and y directions.

4.2.1 Using Subspace Information for Determining Camera Position

The aforementioned subspace methods all have in common their ability to reduce images from the predefined camera range to a low dimensional form. If $\mathbf{Y}_{training}$ is defined as a matrix consisting of the projections of a set of m images equally spaced throughout the movement range of the camera:

$$\mathbf{Y}_{training} = [\mathbf{y}_1 \, \mathbf{y}_2 \cdots \mathbf{y}_m] \tag{4.2}$$

then the current position of the camera can be determined by performing a Euclidean nearest neighbor search of the set of projections $\mathbf{Y}_{training}$ and the current projection of the camera $\mathbf{y}_{current}$:

$$\text{Nearest Neighbor} = arg\min||\mathbf{y}_{current} - \mathbf{y}_i|| \tag{4.3}$$

The position is then taken to be the position corresponding to image i, the nearest neighbor match.

4.2.2 Position Error Measurements

For the straight PCA implementation, both 15 eigenvectors and 30 eigenvectors were used to find the camera position. For the FDA implementation, 30 PCA eigenvectors were used for the $\mathbf{V}$ matrix and 15 eigenvectors were used to form the $\mathbf{V}_{\text{fisher}}$ matrix as described in Chapter 2. Two hundred twenty-five different classes were used to form the $\mathbf{S}_w$ matrix corresponding to the individual camera positions that were not on the edge of the camera movement range. Each class consisted of the central image as well as its eight neighbors as mentioned in the earlier section. For ICA,

the images were dimensionally reduced to 30 by PCA prior to the calculation of the independent components. The FastICA algorithm described in [31] was used to find 30 statistically independent basis vectors for the basis ICA 1 and 30 statistically independent coefficients for ICA 2. For kernel PCA, a radial basis function kernel was employed. The kernel σ was varied between 1 and 7 for the first data set and 4 to 10 for the second. Above this, increasing kernel size did not improve performance. The best result (an average of the x and y error) was shown in the table of results.

For the altered illumination case, a LoG filter was applied to the training and test images. For the first data set, histogram equalization was applied to the training and test images before the application of the LoG filter. This was found to improve the results. For the second data set, the histogram equalization was not applied, as the results worsened with its application.

The results for each subspace method on the two different objects with the two different illuminations are illustrated in Tables 4.1 and 4.2. The absolute mean and variance of the errors (in micrometers) for the random image sets in μm are reported separately for both the x and y directions. Additionally, the statistical significance of the results with respect to PCA is shown. Histograms for the x and y errors for the invariant illumination case for object A are shown in Figures 4.3 and 4.4. Table 4.3 shows the kurtosis of the coefficients. This was calculated for each subspace and object from the distribution of all calculated coefficients from the training image set organized as a one dimensional data vector. The purpose of this analysis was to compare an inherent characteristic of the subspaces. Due to this, FLD was left out, since an arbitrary method of clustering was applied, which would not allow a fair comparison of intrinsic characteristics of the method. Tables 4.4 and 4.5 show how PCA and ICA coefficient kurtosis changes with the dimensionality of the subspace. The % eigenvalues represents the percentage of the total sum of the eigenvalues that is retained for a given dimensionality.

Method		μ_x	μ_y	$\sigma^2{}_x$	σ^2_y	Z_x	Z_y
Normal	PCA	17.35	17.09	268.76	233.15		
	ICA 1	17.96	16.93	287.69	224.65	0.365	-0.106
	ICA 2	17.18	15.73	245.18	195.20	-0.106	-0.932
	FLD	13.74	15.14	219.79	236.07	-2.31	-1.270
	KPCA	12.93	15.23	106.41	165.65	-3.230	-1.320
Lighting	PCA	185.61	66.89	16497	1998		
Variation	ICA 1	171.21	68.00	14948	2094	-1.140	0.245
	ICA 2	176.88	66.88	15452	2001	-0.690	0.002
	FLD	938.01	214.15	556190	37680	14.06	10.45
	KPCA	184.20	67.18	16312	2010	-0.110	0.065
Occlusion	PCA	49.34	53.05	1845.6	2369.3		
	ICA 1	39.53	56.44	1567.0	2758.1	-2.370	0.670
	ICA 2	43.86	63.03	1919.3	3620.7	-1.260	1.820
	FLD	168.06	151.33	31858	25977	9.150	8.250
	KPCA	42.40	61.75	1718.8	3437.5	-1.640	1.610

Table 4.1: Table of results showing mean (in micrometers), variance (in micrometers^2) and z scores (w.r.t. PCA) for x and y errors of Object A

Method		μ_x	μ_y	$\sigma^2{}_x$	σ^2_y	Z_x	Z_y
Normal	PCA	10.36	12.69	82.88	111.30		
	ICA 1	10.29	13.41	79.85	129.35	-0.078	0.654
	ICA 2	6.19	12.44	46.23	110.71	-5.210	-0.236
	FLD	22.26	12.18	574.41	134.44	6.570	-0.459
	KPCA	11.08	10.41	98.08	78.19	0.759	-2.330
Lighting	PCA	3205.1	717.1	1.8E6	0.5E6		
Variation	ICA 1	3246.3	798.0	1.8E6	0.7E6	0.307	1.040
	ICA 2	3242.3	717.3	1.9E6	0.5E6	0.273	0
	FLD	4001.6	1336.7	3.7E6	2.9E6	4.800	4.750
	KPCA	3205.1	717.3	1.7E6	0.5E6	0	0
Occlusion	PCA	535.56	269.00	1.1E6	0.4E6		
	ICA 1	534.66	261.05	1.1E6	0.3E6	-0.009	-0.134
	ICA 2	470.84	280.38	1.0E6	0.3E6	-0.632	0.192
	FLD	706.65	433.01	1.4E6	0.4E6	1.530	2.590
	KPCA	542.59	290.35	1.1E6	0.4E6	0.067	0.338

Table 4.2: Table of results showing mean (in micrometers), variance (in micrometers^2) and z scores (w.r.t. PCA) for x and y errors of Object B

Method		Coefficient Kurtosis	
		Object A	Object B
Normal	PCA	4.2080	9.5823
	ICA 1	3.0535	3.1993
	ICA 2	2.3419	2.4544
	KPCA	2.2871	5.5830
Lighting Variation	PCA	3.8995	5.0827
	ICA 1	2.9473	2.8226
	ICA 2	2.6826	2.1282
	KPCA	3.8979	5.0825

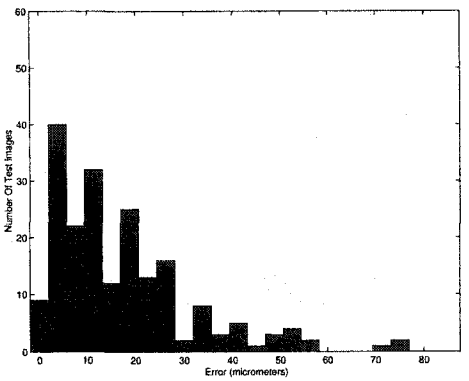
Table 4.3: Kurtosis of the coefficients of the training set for each subspace

	Coefficient kurtosis			
dimension	PCA	ICA 1	ICA 2	% eigenvalues
10	2.6532	2.5951	2.4672	50.706
20	3.5313	3.2967	2.9108	63.884
30	4.2080	3.0195	2.9085	72.518
40	4.8524	2.2871	2.9402	78.431
50	5.5123	2.8618	2.8548	82.529
60	6.1739	2.9183	3.0549	85.571
70	6.8356	2.8989	2.8508	87.927
80	7.4955	2.9152	2.8737	89.824

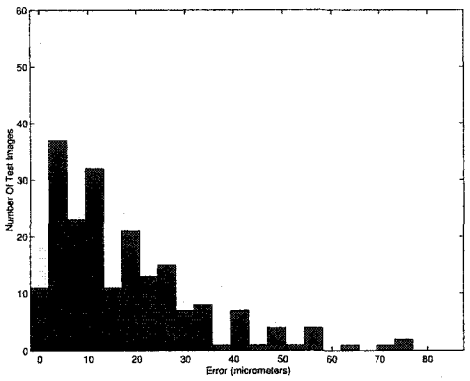
Table 4.4: Change in PCA and ICA coefficient kurtosis with dimension (Object A)

	Coefficient kurtosis			
dimension	PCA	ICA 1	ICA 2	% eigenvalues
10	3.8273	2.9887	2.4273	84.053
20	6.7567	3.2095	2.7243	89.616
30	9.5823	2.8805	2.6899	92.208
40	12.307	3.2560	2.6473	93.975
50	14.991	3.4021	2.7047	95.212
60	17.658	3.3760	2.7374	96.111
70	20.316	3.0746	2.7927	96.787
80	22.975	3.2761	2.8200	97.304

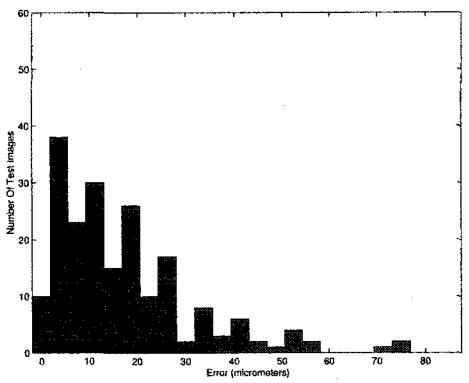
Table 4.5: Change in PCA and ICA coefficient kurtosis with dimension (Object B)



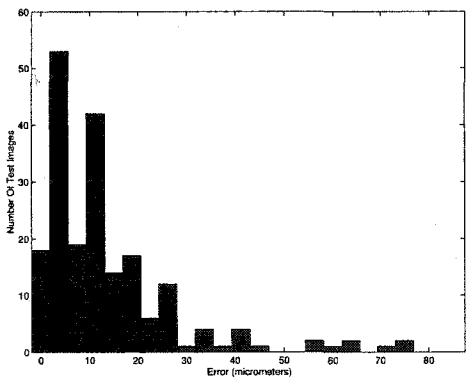
(a) PCA x error



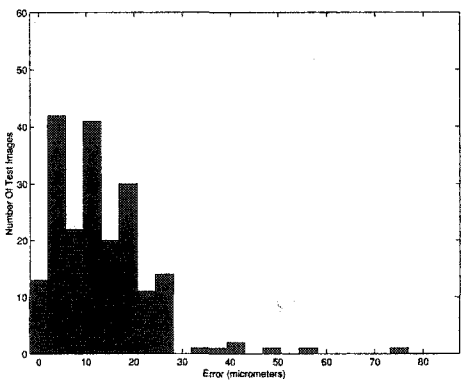
(b) ICA1 x error



(c) ICA2 x error

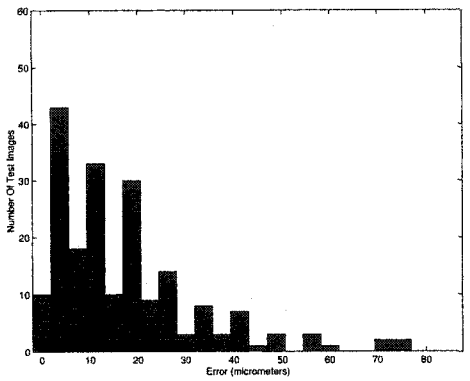


(d) FLD x error

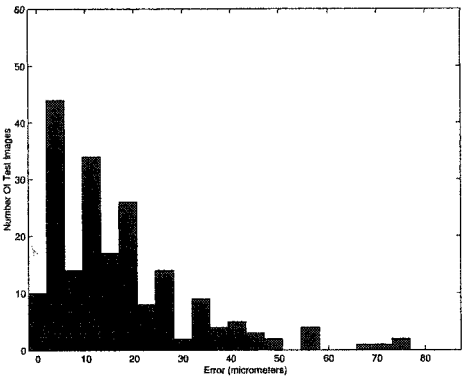


(e) KPCA x error

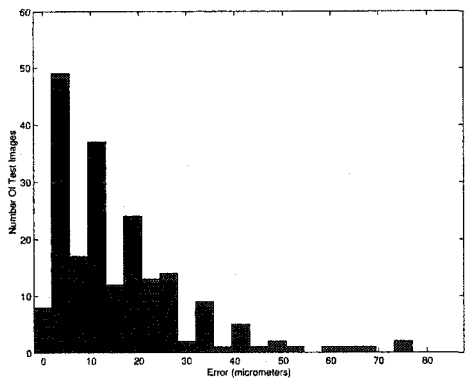
Figure 4.3: Histograms of x errors (in micrometers) for object A with constant illumination



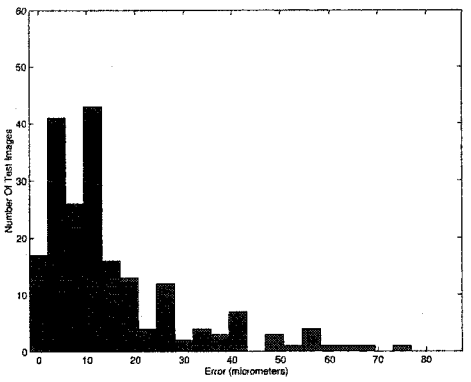
(a) PCA y error



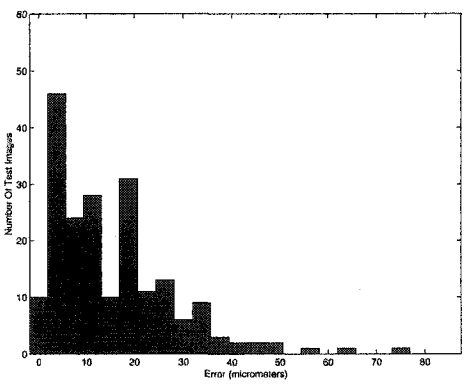
(b) ICA1 y error



(c) ICA2 y error



(d) FLD y error



(e) KPCA y error

Figure 4.4: Histograms of y errors (in micrometers) for object A with constant illumination

4.2.3 Discussion

PCA's performance was equal to all of the other methods tested. While the other methods more often than not offered better results, the differences were not statistically significant ($p < 0.05$), with the exception of a few isolated cases where one direction or the other was significantly better (or worse).The statistically significant differences from PCA's performance from Tables 4.1 and 4.2 are summarized below:

- KPCA normal illumination, x direction, object A (better)
- FLD normal illumination, x direction, object A (better)
- FLD lighting variation, x direction, object A (worse)
- FLD lighting variation, y direction, object A (worse)
- ICA 1 occlusion, x direction, object A (better)
- FLD occlusion, x direction, object A (worse)
- FLD occlusion, y direction, object A (worse)
- ICA 2 normal illumination, x direction, object B (better)
- FLD normal illumination, x direction, object B (worse)
- KPCA normal illumination, y direction, object B (better)
- FLD illumination variation, x direction, object B (worse)
- FLD illumination variation, y direction, object B (worse)
- FLD occlusion, y direction, object B (worse)

Out of the 30 measurements made, 13 had statistically significantly different error with another subspace. Unfortunately, 5 were better and 8 were worse. The 5 better

measurements were equally scattered between the methods. All 8 worse measurements were with FLD. From this it is quite clear that there is no one technique that is significantly better than PCA. This gives a strong indication that for this application, when employing this subspace methodology, PCA is a reasonable choice, given that it is relatively simple and computationally efficient to compute. However, the important unanswered question is why does PCA perform similarly well to the other subspace techniques. Evidently, there is something inherent in the data that makes the PCA representation appropriate. A hypothesis for an answer to this question will be outlined in the next section. Another factor which would tend to make the distribution of the errors similar across all of the tested subspaces is the inherent error in the XY table position, due to mechanical tolerances. This error would appear as a random noise in measurement error and would apply equally to all of the subspace measurements.

Another clear result from the experiment is that lighting change has a significant impact on the accuracy. It is interesting to note that errors in the altered illumination case lie primarily in the direction of the illumination direction (x), with a 3 or 4 fold difference in mean error magnitude between x and y direction for both objects. In the x direction, lighting change caused an order of magnitude increase in mean error for object A and over two orders of magnitude increase for the object B. This difference in behavior is almost certainly due to the very different surfaces that the illumination was reflecting from. For object A, the circuit board, a great many small features were present, as well as a large chip that is matte in appearance (a diffuse reflector). Overall, the circuit board functions as a diffuse surface, scattering light relatively evenly in all directions. Although there are a number of small shadows cast by the low profile features, the overall appearance does not change significantly with the altered illumination direction. Object B had a relatively specular surface, given that it was made of metal. Specular objects' appearance changes with viewing angle, thus as the metal object moves under the camera, its appearance will vary. The amount of variation is a function of the specularity of the object. While this object was not

highly specular, it was much more so than the circuit board.

Position measurement with object A was relatively unaffected by occlusion (a two or three fold increase in position error from normal for all of the subspace methods), while the position measurement accuracy of object B was strongly impacted (much greater than an order of magnitude for all of the subspaces). Here, the difference is most likely due to the feature rich nature of the first object and the feature poor nature of the second. Intuitively, if an object has a great many features, occluding a few will not affect the overall appearance. More specifically, coefficients in the subspace for the objects under test are determined by the correlation of the test image with a basis image. For the first object, a great deal more high frequency content exists. Generally, when PCA is used to reduce the dimensionality of the dataset as was done here (for all of the subspace methods), this has the effect of low-pass filtering, since generally, images' amplitude spectrum varies as 1/frequency [78]. In other words, the majority of the variance in an image set is contained in the first few PCA basis vectors [78]. In this experiment, the same number of basis images was maintained for both objects. Speaking now of PCA exclusively, for object B, most of the energy of the data set will be contained in the first few (low frequency) basis images (those corresponding to the largest eigenvalues). Occluding an image with a featureless patch will have a significant effect on its coefficients resulting from correlations with the low spatial frequency basis images. In general, then (for all of the subspaces), if the basis has more low frequency content, the coefficients will be more strongly affected by occlusions of the type used in this experiment. It is this change in coefficients that cause the error. A hint at the concentration of energy in the low spatial frequency components for object B can be seen in the kurtosis of the coefficients. More about this will be mentioned next. An example will be provided later of occlusions which are not featureless and a very different behavior will be exhibited.

4.2.4 Coefficient Kurtosis

The significance in the distribution of the coefficients resulting from the linear transform can be observed by considering these coefficients as a method of coding the images. Much has been stated in the past regarding effective measures of image coding. The general agreement seems to be that image coding for compression, of course, necessitates no redundancy (statistically decorrelated coefficients) and that some amount of redundancy is advantageous to image recognition. There is far from a consensus, however, about the role that redundancy plays in feature selection for recognition. Barlow introduced the idea in [79] and revisited it in [80] ten or so years later. According to this work, Shannon's model of redundancy — that which wastes channel capacity — is quite inappropriate for coding schemes representing learned information about the environment. From the point of view of object recognition, then, a view of redundancy reduction which focuses on compressive coding will provide a disservice to making hypotheses about the higher level features of images (as a representation of an environment). While this discussion can get quite technical, for the purposes of this thesis, the hypothesis has already been made that sparse coding (super-Gaussian distributions) is effective.

In [53] the kurtosis of the coefficients was investigated for both ICA and PCA for a face recognition application, and it was found that ICA, particularly with statistically independent coefficients, had very much higher kurtosis than PCA. Surprisingly, Table 4.3 clearly shows the opposite effect for a position measurement application. A high kurtosis indicates sparseness in the coefficients, illustrating that for this application, PCA needs very few coefficients to capture the essential statistical characteristics of the data. Clearly this is due to the highly correlated nature of the image set (most of the image set variance is described in the first few eigenvectors), which makes PCA the ideal candidate for representing this type of data. This is in sharp contrast to general object recognition, where the sparse coding of ICA and other techniques tend to outperform PCA.

A hypothesis was made on the basis of this result that PCA coefficient kurtosis would increase as the dimensionality of the data increased. The other part of the hypothesis is that ICA coefficient kurtosis would remain roughly constant. This hypothesis is based on the reasoning that the highly correlated nature of the data meant that the eigenvectors corresponding to the small eigenvalues would be mostly noise and would provide low correlation values (small or zero coefficients). What is critical about this reasoning is that one could guess that the rotation of the ICA basis does not line up with the directions in the data which are caused by low spatial frequency features. In this way, ICA divides up the entire spatial frequency content amongst its basis vectors for this application.

Tables 4.4 and 4.5 then provides a strong indication that this hypothesis is indeed correct. Increasing the dimensionality of the data provides a linear increase in coefficient kurtosis for PCA with a linear increase in dimension and almost no change for ICA. It can be safely concluded that a very few PCA components provide the "sparse" code as defined by Barlow. ICA coefficients end up functioning as a distributed code. The coefficient kurtosis is indicative of a Gaussian distribution (kurtosis = 3) and thus ICA is dividing up the information content of noise. Even more specifically, ICA 2, which tries to maximize non-Gaussianity of the coefficients tends towards sub-Gaussianity, which is yet another indicator that most of the basis vectors are noisy.

4.2.5 Summary

In the preceding experiment a number of subspaces were tested for the application of measuring the 2D position of a circuit board and a stamped metal part. Initially it was proposed that perhaps other subspaces might provide an advantage for lighting variant and occluded images. It was clear from the results that the direct application of these subspaces offered no advantage over the relatively simple technique of

constructing a PCA subspace. By examining the kurtosis of the coefficients, the reason for PCA's relative success in this application was clearly illustrated. The open question, however, remains how to deal with lighting and occlusion with subspace techniques. The next experiment will attempt to provide a general method for dealing with occlusions which only exploits characteristics of the ICA subspace. This is contrasted with methods mentioned in [57] and others which rely on a re-projection of the subspace into the original space for the purposes of testing for occlusion. In the next chapter the use of ICA for lighting variant object recognition will be examined in considerable detail.

4.3 Position and Orientation Measurement with Occluded Images

The problem of measuring the position or orientation of objects with vision when an image is partially occluded is a common one. It is typical that in such applications as autonomous vehicle guidance and visually guided robotics, other objects in the scene that were not present during training, obscure the view of those that were. In a general sense, occlusion is the problem of recognizing an object or image when part of it is no longer the same in appearance as it was in the training phase. A key feature of this type of problem is that only part of the object or image has changed appearance. A portion of it has retained its original appearance. In this way, occlusion is a local phenomenon. In fact, it is this locality that has spawned most of the techniques for handling occlusion — determine which portion of the image is no longer similar to an image in the training set and exclude this portion from further consideration. This determination must be done in the original image space (not subspace). If one attempts to deal with occlusions in subspace, some notion of how the coefficients of the subspace change when the appearance changes — a very difficult problem, since no simple model exists for the appearance of an image (much more will be discussed

about this in the context of statistical learning theory). One might suppose that the situation seems almost hopeless. To make matters worse, general subspace techniques generate coefficients based on a correlation with a whole image — there is no spatial locality imposed. Again, there are specific solutions that have been devised that use locality to produce subspaces of sub-windows (see [57] as mentioned previously) however this does not directly address the global nature of the basis vectors of an entire image. A further worsening of the situation occurs when PCA is used to provide the basis, as it will be seen that it provides a solution which not spatially localized (the trade off between spatial support and frequency selectivity described by the uncertainty principle governing space-frequency resolution). This experiment will attack the problem of spatial locality directly, by employing ICA to provide a spatially localized basis which will be shown to be less sensitive to occlusions. In doing so, the overall goal is not to solve the occlusion problem, but instead is to further illustrate the difference between the nature of PCA and ICA for the purposes of feature extraction.

The general procedure for finding the position of the camera or the orientation of the object was to project each image in the training data set into the PCA and ICA basis set to provide a low-dimensional vector for each image. Each component in the vector was cubic spline interpolated by a factor of 10 across the image set to reconstruct intermediate (between image) low-dimensional vectors in the sub-space. Similarly, each occluded image in each of the two occluded data sets were projected into the same sub-space. The minimum Euclidean distance between an occluded image's low-dimensional representation and the training data set's subspace provided its position with respect to the original training data set. The artificial occlusions were applied to each image in the training set and the average error over all measurements were recorded. The results produced measures of how well the ICA and PCA sub-space techniques localize the position of occluded images in a training set of non-occluded images for translated and panned cameras as well as for the object orientation data set. The ten independent and principal component basis images of

the translation dataset are shown in Figure 4.9.

An experimentally constructed scene was used to provide test image sets. The scene was chosen to have a variety of features at a variety of spatial frequencies. Thirty-one images of the scene were taken by translating a camera horizontally over a distance of 300 mm in 10 mm increments. Twenty-one images of the scene were taken by panning the camera over angles from 85 to 95 degrees from the horizontal axis. Additionally, an orientation recognition experiment was conducted by taking twenty-one images of an object rotated from -50 to +50 degrees. Example image sets are shown in Figures 4.5 and 4.6.

Occluded data sets were produced by selecting 2 translations (160 mm, and 240 mm) and occluding the scene to varying degrees for each translation. Similarly, the 90 and 95 degree panning and the 0 and 25 degree orientation images were occluded to varying degrees. Additionally, a sliding curtain of a blank occlusion and a randomly positioned blank square occlusion were artificially applied to the translation dataset. Example occluded image sets for camera translation and orientation are shown in Figures 4.7 and 4.8.

A set of 10 basis vectors were constructed for each image set using the 10 whitened PCA basis images corresponding to the principal components with the maximum variance. This was done to avoid adding noise to the PCA basis, as mentioned in the previous experiment. This is a reasonable limit, since it was found that for the PCA

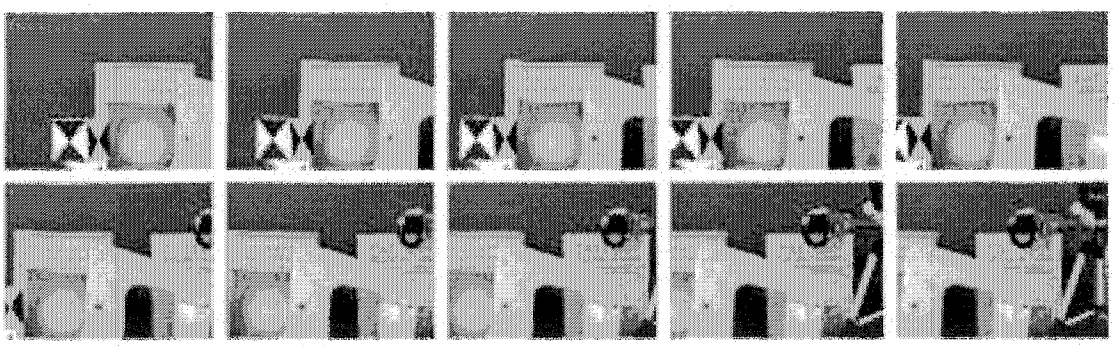


Figure 4.5: Sample images from data set of translated camera training images

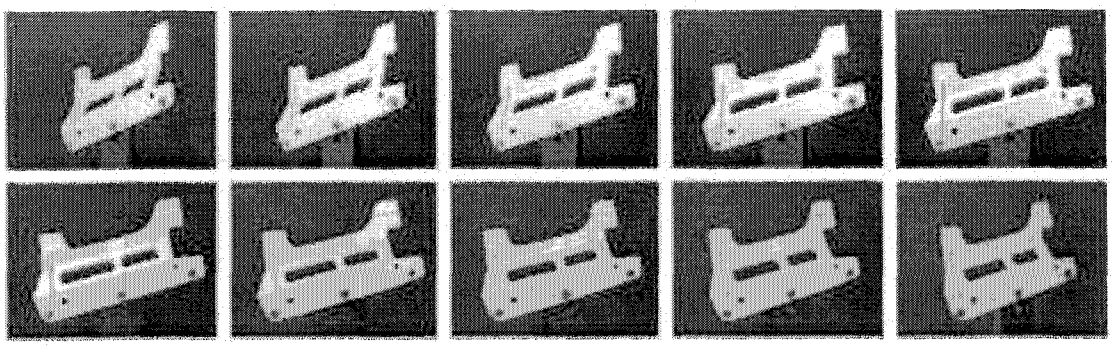


Figure 4.6: Sample images from data set of orientation training images

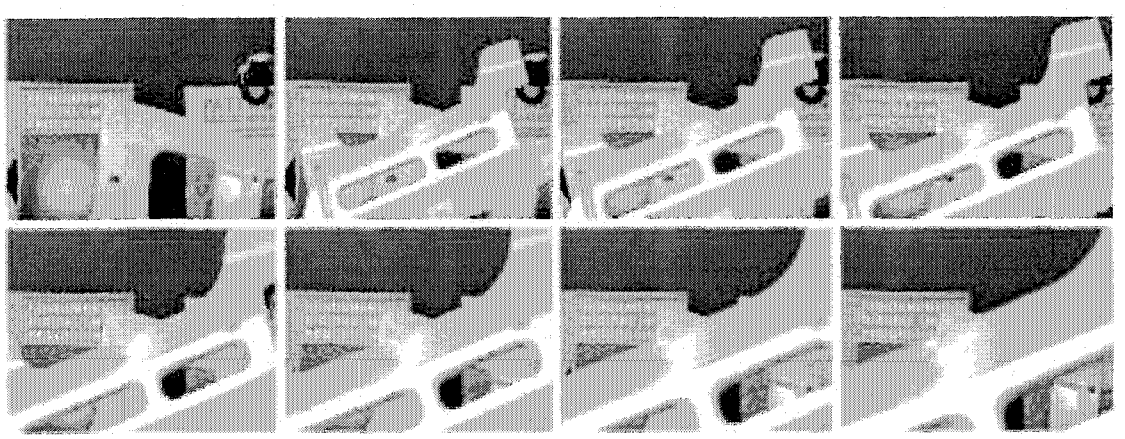


Figure 4.7: Sample images from data set of occluded translated camera images

case, keeping the first 10 components maintained approximately 80 % of the original energy in the data set. For the ICA case, a full set of basis vectors (31 for translation, 21 for panning, and 21 for orientation) were calculated. The best 10 were selected by selecting combinations of 10 out of the full set. A sample of 50 component sets which produced position errors less than that of PCA for the 160 mm translation, 90 degree panning, and 0 degree orientation occlusion data sets were found, and the 10 most common components out of the 50 sets were used in subsequent experiments. This combinatorial method of selecting ICA basis vectors is unique in the literature. It amounts to a supervised selection technique for selection. Later, a more rigorous technique of floating search [47] will be employed. Recall again from Chapter 2 that provided that Euclidean distance is used as a similarity measure, some form

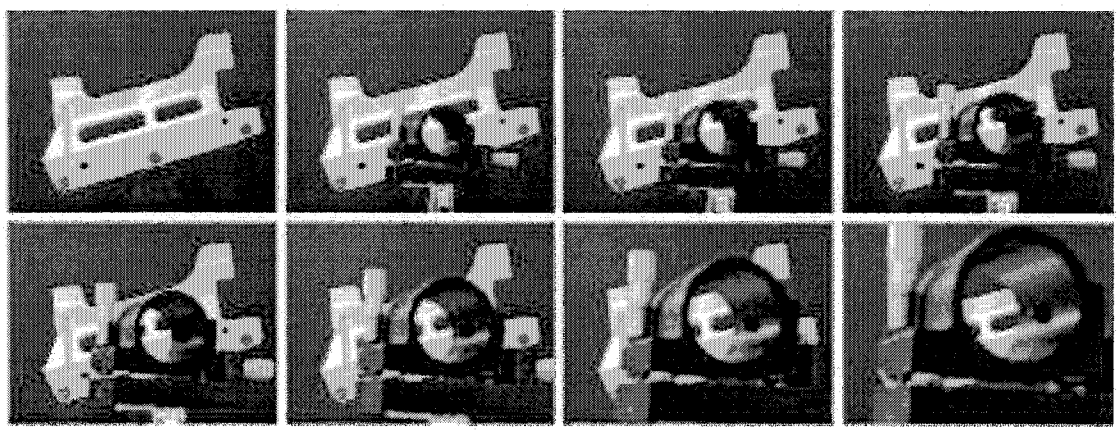
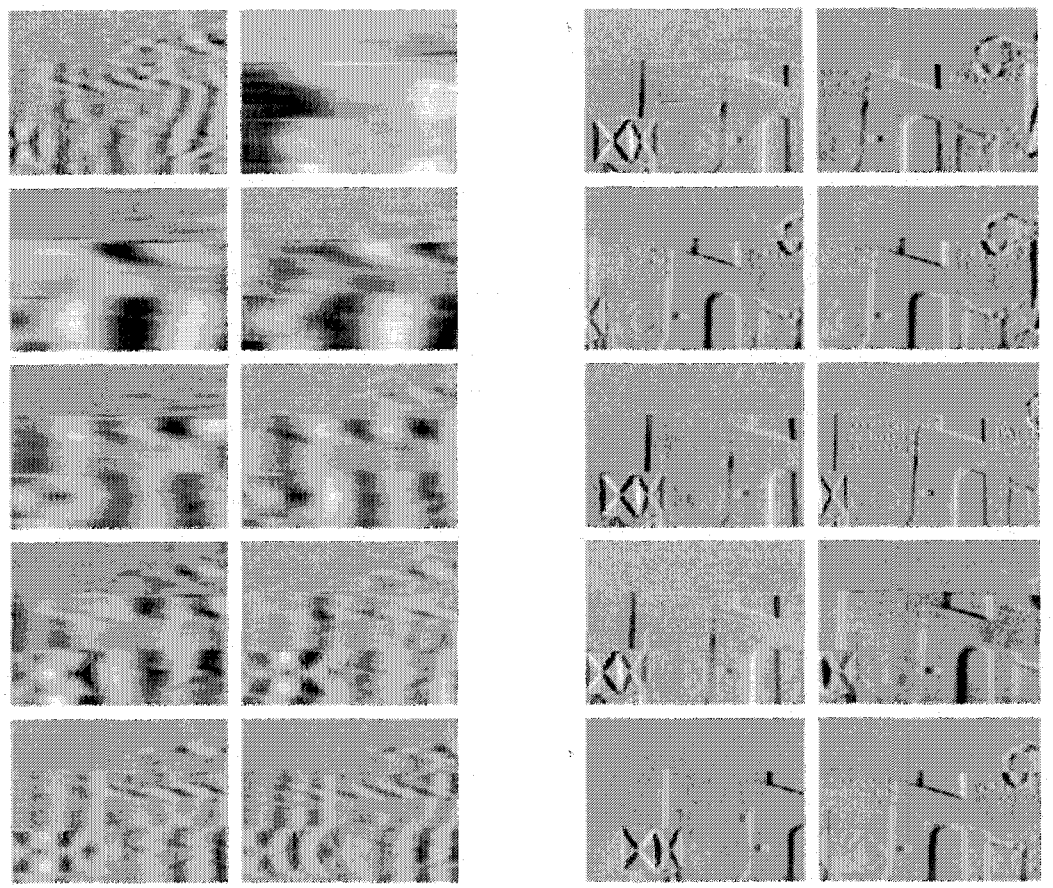


Figure 4.8: Sample images from data set of occluded orientation images

of dimensionality reduction must be employed other than PCA, or the recognition results of PCA and ICA will be almost identical. Unlike the previous experiment, this feature of ICA was addressed directly. Figures 4.10 and 4.11 show the distribution of position errors for a translated and panned camera respectively. Figure 4.12 shows the distribution of orientation measurement error. The absolute error indicates the distance from the correct index in the database in the interpolated sub-space.

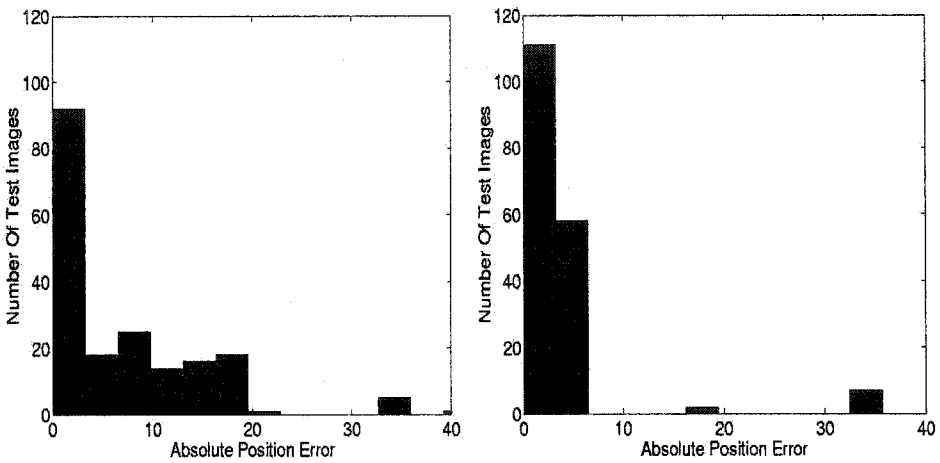
4.3.1 Discussion

In all cases, the majority of the position or orientation errors were clustered around lower values of error for ICA. Indeed, one might suppose that this might be the case, since the ICA basis vectors were in fact selected a-posteriori on the results of test images. However, they were selected on the basis of only one test case for each experiment. Given the relatively small size of the experiments, the errors do not exhibit a Gaussian distribution, so are illustrated graphically rather than by describing mean and variance. The clearest indication of why ICA offers an advantage for this application appears to be shown in Figure 4.9. Analysis of the different basis images leads to a discussion of basis images as spatial filters. Examining the PCA basis images shown, a familiar pattern has emerged. These images are strikingly



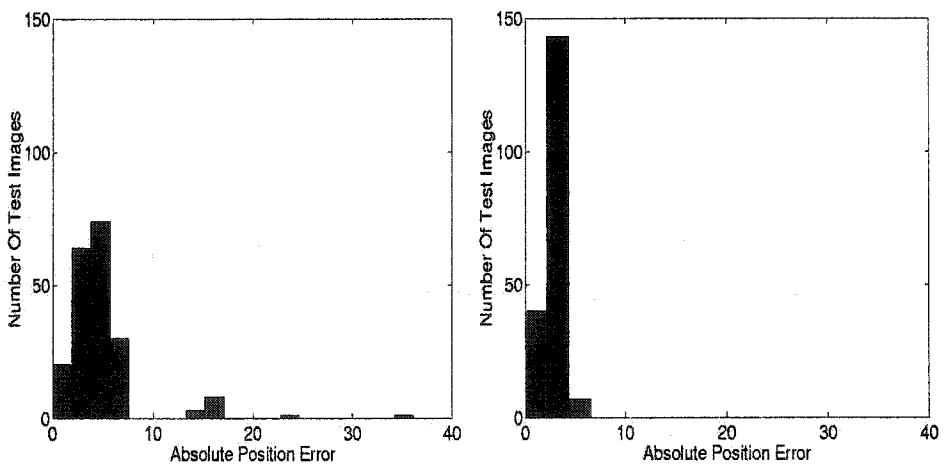
(a) PCA (b) ICA

Figure 4.9: Basis images for the translated camera



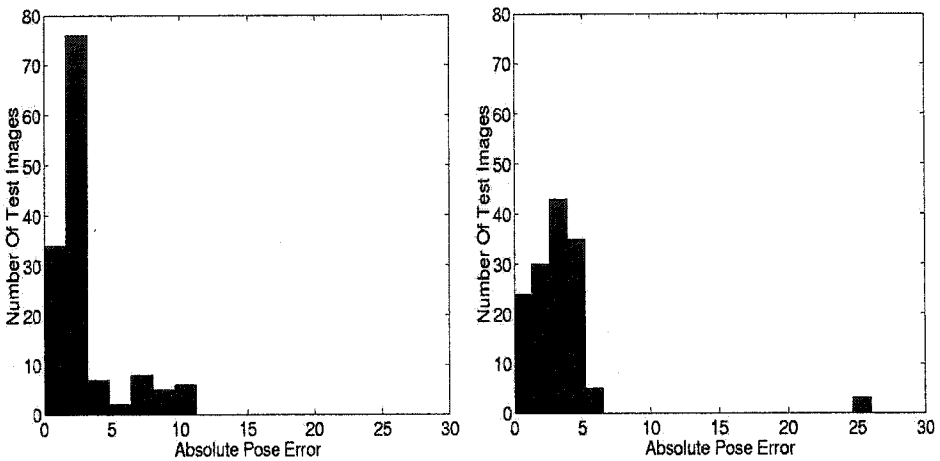
(a) PCA (b) ICA

Figure 4.10: Distribution of position errors (in mm) for translation



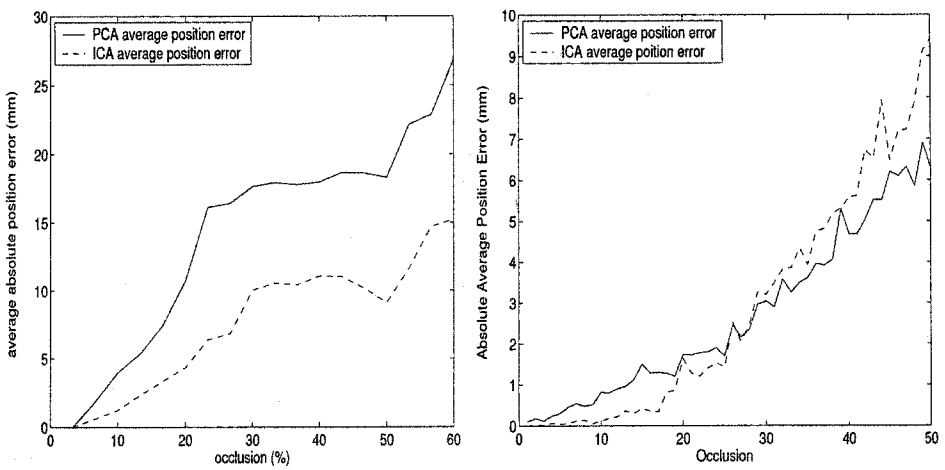
(a) PCA (b) ICA

Figure 4.11: Distribution of position errors (in units of 0.05 degrees) for panning



(a) PCA (b) ICA

Figure 4.12: Distribution of orientation errors (in units of 0.5 degrees)



(a) left to right sliding occlusion (b) random position square occlusion

Figure 4.13: Average position error

similar to a basis calculated by the DCT — light and dark bands that extend over the entire width of the image. Basis image 2 has only one such cycle, while basis images 2 and 3 have two cycles. Images 3 and 4 have three cycles, and 7 and 8 have six cycles. This is not coincidental nor data dependent. It is simply due to the retention of only low spatial frequency basis vectors which occurred because a dimension of 10 was selected. The fact that PCA basis images bear any relation to a DCT basis is a bit more detailed.

If each pixel's change across an image set is modeled as a 1st order Markov process, it can be shown that the DCT and PCA provide similar bases. By definition, a random sequence $u(n)$ is called Markov-p or pth order Markov if the conditional probability of $u(n)$ given the entire past is equal to the conditional probability of $u(n)$ given only $u(n-1), ..., u(n-p)$.

$$P(u(n)|u(n-1), u(n-2), ...) = P(u(n)|u(n-1), ..., u(n-p))$$

This effectively states that locally the value of a pixel in one of the images in the set could be predicted given only corresponding pixels from p previous images in the set. Since the image sets used here are highly correlated (don't change much from image to image) the case roughly corresponds to a low order Markov process. The covariance function of a stationary first order Markov sequence $u(n)$ is $r(n) = \rho^{|n|}$ so its covariance matrix is:

$$\mathbf{R} = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{N-1} \\ \rho & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \rho^2 \\ \vdots & & \ddots & \ddots & \rho \\ \rho^{N-1} & \cdots & \cdots & \rho & 1 \end{bmatrix}$$

Now, the DCT has the property that the basis vectors of the cosine transform are

eigenvectors of the matrix $\mathbf{Q}_c$ where:

$$\mathbf{Q}_c = \begin{bmatrix} 1-\alpha & -\alpha & & \mathbf{0} & \\ -\alpha & 1 & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \\ \mathbf{0} & & \ddots & 1 & -\alpha \\ & & & -\alpha & 1-\alpha \end{bmatrix}$$

The covariance of a first order Markov sequence can be described as:

$$\beta^2\mathbf{R}^{-1} = \begin{bmatrix} 1-\rho\alpha & -\alpha & & \mathbf{0} & \\ -\alpha & 1 & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \\ \mathbf{0} & & \ddots & 1 & -\alpha \\ & & & -\alpha & 1-\rho\alpha \end{bmatrix}$$

where $\beta^2 \triangleq (1-\rho^2)/(1+\rho^2)$ and $\alpha \triangleq \rho/(1+\rho^2)$. $\beta^2\mathbf{R}^{-1} \cong \mathbf{Q}_c$ for $\rho \cong 1$ therefore their eigenvectors will be close. Additionally, the eigenvectors of $\beta^2\mathbf{R}^{-1}$ are identical to those of $\mathbf{R}$. Of course, the image set is not exactly first order Markov, but to the extent that it is close, this result is valid.

In any case, the above argument gives a strong hint of the behavior of PCA basis vectors for highly correlated image sets. The first few basis vectors will bear a strong resemblance to those of the DCT — that is, they are sinusoidal and have global support. The same simple analysis cannot be provided for ICA. However it has been shown in [81] and elsewhere that ICA basis closely resemble oriented wavelet type filters. When one considers that the set of coefficients resulting from the linear transformation provided by the basis vectors are effectively filter coefficients, with the basis vectors as filters, a general claim can be made about the difference between PCA and ICA bases for this application. PCA coefficients, then, are the filter coefficients resulting from a filtering operation with the low pass operation of

the first few sinusoidal basis vectors. This implies that the PCA basis vectors are "tuned" to low spatial frequency features in the images and are of global spatial support. ICA basis vectors are tuned to be oriented and bandpass by the orientation and spatial frequency content of the images and are of local spatial support. In this way, the ICA filters are "optimized" by the trade off between spatial locality and frequency selectivity and are oriented with the primary direction of features in the image. The degree to which ICA resembles wavelet filtering has been examined in detail elsewhere and will not be included here.

What is of importance here is the significance of the difference in these filters for the application of recognition in the presence of occlusion. It was mentioned previously that PCA functions well when the test images are not occluded. In light of the above discussion, the question then becomes how well the basis vectors are tuned to ignore the spatial frequency and spatial locality of the occluding features in the image. From the spatial locality point of view, PCA performs poorly. The filters are of global support, thus any change in any part of the image will be applied to the filter. One can then only hope that the frequency selectivity is such that the occluding feature is not passed through the filter. The direct implication is that occlusions composed predominantly of low spatial frequency content will have a large impact on the PCA coefficients. In fact, perhaps the worst case is a blank occlusion (zero spatial frequency) such as the type applied to achieve the results in Figure 4.13. Indeed, it is seen that even a small percentage occlusion has a significant effect on PCA.

ICA seems to be more promising with respect to both spatial frequency and spatial locality tuning. The bandpass nature should provide some invariance to the coefficients under the effect of blank occlusion. Indeed, in Figure 4.13a when a curtain of the entire vertical extent of the image is applied to varying degrees from left to right, a significant improvement occurs from the use of ICA. Things start well with ICA for randomly positioned occlusions in Figure 4.13b, however beyond 30 % occlusion,

PCA actually does marginally better. In light of all that has been mentioned previously, this is a difficult result to explain. The current working answer is that because the ICA basis spans a different space to that of PCA, some of its basis images could be very sensitive to occlusions at specific places with specific images. In other words, while the linear addition of the PCA basis will provide a (blurry) reconstruction of the entire image set, the ICA basis will not. There will be places where the energy is focused, thus increasing its sensitivity to occlusions at these locations with certain images. Of course, the sensitivity would be reduced at other locations, so it is hard to say how that trade off works. Generally, beyond 30 % occlusion, both techniques exhibit significant error in any case. Overall, however, the spatial and frequency selectivity seems to be advantageous.

Finally, this experiment illustrated the more realistic case of specific occlusions provided by an object which is not featureless. Here again, the tuning of the ICA basis seems to provide an advantage, lowering the variance of the position errors. This is expected, particularly since the orientation specificity of the filters could potentially offer a significant improvement in selective insensitivity to occluding features. Figures 4.10, 4.11 and 4.12 provide a good indication that ICA offers an advantage for this application, despite the fact that the bandpass filtering might be tuned to occluded objects' features.

4.3.2 Summary

In this experiment, independent component bases were chosen in a supervised manner, using minimum measurement error of selected test images. Approximately 250 selections were needed to find 50 basis vectors which provided lower position, panning and orientation measurement error than that of PCA in the presence of occlusions. Therefore, approximately 20 % of the basis combinations were more occlusion invariant than the eigenfeatures. The best independent component basis were selected based on the results of one of the occluded data sets and were observed to exhibit

equally good performance over the other data sets with significantly different occlusion appearance. It was shown that PCA and ICA basis images vary significantly in their spatial, spatial frequency and orientation selectivity. For applications of highly correlated image sets, ICA can, to some extent, extract important features from the underlying unoccluded data set which were invariant to occlusion.

Chapter 5

Recognition Under Varying Illumination

5.1 Specular Objects

In Chapter 1, the literature survey recounted prior work on characterizing the set of images under all possible lighting conditions. This lead to the conclusion that provided the surfaces are Lambertian reflectors, a subspace of relatively small dimension can reconstruct all possible lighting conditions. While this idea lends itself well to face recognition (because of the roughly Lambertian reflectance characteristic of skin) it does not apply at all to specular objects. In fact, due to the non-linear nature of the reflectance of these surfaces, a linear subspace cannot accurately model this situation. In this chapter, ICA is examined for the application of recognizing specular objects despite the inappropriate nature of the model. It will be seen that applying some very simple LoG filtering to the training and/or test images has a dramatic effect on the ability of the linear model to characterize specular objects for the purposes of recognition. The experiments will illustrate the importance of image edges as clues to the identity of objects. As a motivation for LoG filtering, a brief discussion will be provided about the early vision process in the human visual system. This will be

shown to motivate the use of ICA as an appropriate representation for general object recognition.

Additionally, focus will be placed on the differences in behavior of PCA and ICA bases. The application tested in these experiments is of a more general nature than that of the previous chapter. The measurement application of Chapter 4 was specific in the sense that the training images were highly correlated. In the case of general object recognition, this is not the case, as the training images are composed of a variety of unrelated objects. This fact provides motivation for the use of ICA as a more appropriate basis to represent the data set. A floating search technique is applied to select ICA basis vectors.

5.1.1 Classification

If the low-dimensional representation described in Chapter 2 and applied in Chapter 4 is used to represent each image in both the dataset and an unknown image, $\mathbf{y}_i$ and $\mathbf{z}$ can be defined to be the low-dimensional representation of the mean of the ith object's features and the unknown image features respectively. The unknown image was classified by employing a minimum Euclidean distance metric:

$$\min_i d(\mathbf{y}_i, \mathbf{z}) = \min_i (\mathbf{x}_i - \mathbf{z})^T (\mathbf{x}_i - \mathbf{z}) \tag{5.1}$$

This differs from the application in Chapter 4, as there is no interpolation between points in subspace. In the ideal case, the function d has a unique minimum, which occurs when i is the index of the matching object. In general, however, classification errors may occur where the minimum occurs at a non-matching object. Classification errors may occur for objects of similar appearance or when illumination conditions dramatically change the appearance of an object. In fact, there is no guarantee that the function will have a unique minimum. As a simple example of this, consider that an object's visible surface $f(x, y)$ is not distinguishable from a transformation $\hat{f}(x, y) = \lambda f(x, y) + \mu x + \nu y$ for all λ, μ, and ν (see [82]). If their visible surface

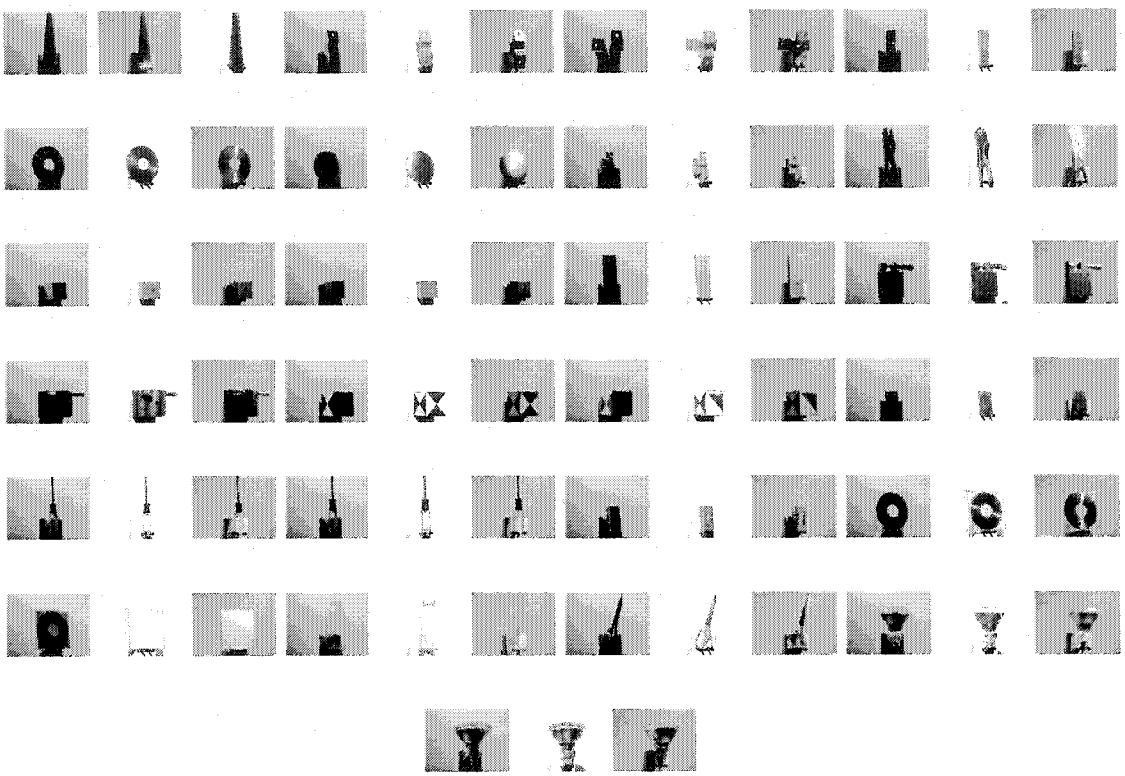
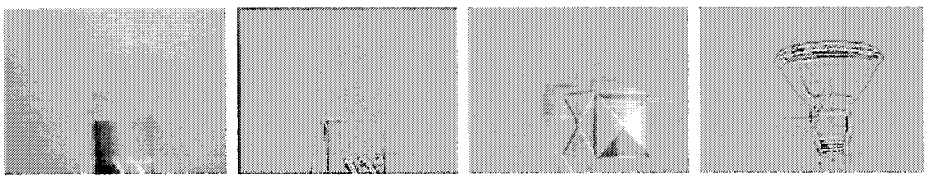


Figure 5.1: Training images for all 25 objects. Objects are grouped in the three illumination conditions: left, center, and right. These images were used to create both the PCA and ICA subspaces.

is indistinguishable, obviously the subspace representation is indistinguishable. The transformation can result from lighting position change or physical motion of the object. In fact, two different surfaces illuminated from different angles can appear similar simply due to this ambiguity. The situation is worsened by the dimensionality reduction of the subspace, leaving much more room for ambiguity. In practice, however, this effect is minimal. Were it not, subspace recognition would be an impossibility.



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

Figure 5.2: First basis vectors for (a) PCA with no pre-filter (b) PCA with LoG pre-filter (c) ICA with no pre-filter and (d) ICA with LoG pre-filter. (Brightness and contrast have been enhanced)

5.1.2 Recognition Experiment

An object recognition experiment was constructed using a standard RS-170 greyscale camera connected to a PC framegrabber card. Lighting was varied by selectively turning on various lights arranged in a semicircle around the object to be classified. Training images were captured for each object in 320x240 pixel resolution for three lighting conditions which were essentially left side illumination, front-side illumination, and right side illumination (see Figure 5.1).

The set of 25 objects included items with high specularity such as aluminum parts and a CD-ROM. The set also included objects whose appearances were similar such as a CD-ROM and a CD-ROM in a clear jewel case. Next, two test images were captured for each object under two new lighting conditions not included in the training set (see Figure 5.3).

5.1.3 Use of LoG Pre-Filtering

Seven different variations of PCA and ICA were explored for object recognition with and without the use of a Laplacian of Gaussian (LoG) pre-filter. ICA was used to provide a basis where the columns are as statistically independent as possible. ICA dimensionality reduction was accomplished by either using only PCA a-priori

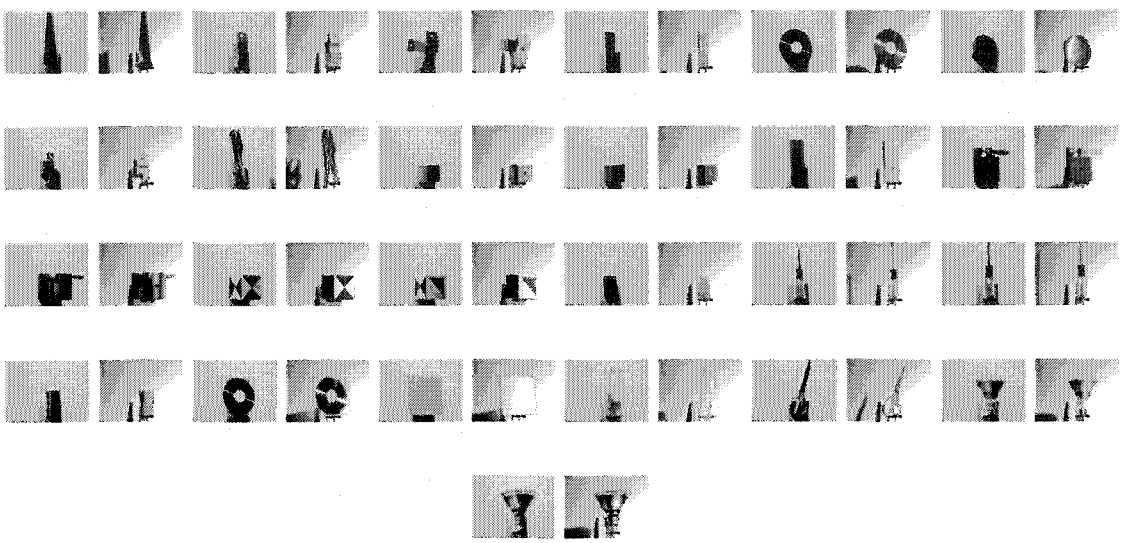


Figure 5.3: Set of 2 test images for all 25 objects under unique illumination conditions used to test PCA and ICA object recognition.

or by only applying a floating search feature selection method. The floating search technique is described in Section 2.5. Three scenarios were explored regarding the use of the LoG pre-filter. First, no LoG filter was applied during training or recognition. Second, the LoG filter was applied to the training images during the training phase and the test data sets during the recognition phase. Third, the LoG filter was applied only during the training phase for the purposes of finding the basis set. All images were mean-adjusted prior to training or recognition. Classification proceeded via a minimum Euclidean distance metric to each object's mean in subspace. To summarize the experimental variations:

- No LoG Filter
  - 'Classic' PCA using standard eigenspace techniques and no pre-filtering
  - ICA using floating search (FS) algorithm and no pre-filtering
- LoG Training and LoG Test Images

  - PCA with the training set and the test images filtered with a LoG filter
  - ICA using floating search with the training set and the test images filtered with a LoG filter

- LoG Basis Images Only
  - PCA with the basis training set pre-filtered with a LoG filter
  - ICA using floating search and pre-filtering the basis training set with a LoG filter
  - ICA/PCA where the ICA dimensionality is reduced using PCA and pre-filtered with a LoG filter

5.1.4 Recognition Rates

The subspaces were computed to produce the basis vectors, a sample of which are shown in Figure 5.2. The final results are summarized in Table 5.1 where the recognition rates are reported for PCA and ICA with and without pre-filtering and for subspaces ranging from a dimension of 10 to 30. The recognition rate was determined as the percentage of test images in Figure 5.3 for which the object was correctly recognized. The results indicate that ICA using a Laplacian of Gaussian pre-filter with a floating search algorithm performed best. The recognition rates for the best PCA and ICA approach are plotted in Figure 5.4. The results also show that at least a dimension of 20 is required for reasonable performance.

5.1.5 Discussion

The lowest levels of the human visual system (even before the electrically transmitted information arrives at the visual cortex) are characterized by cells that serve the purpose of edge detection. In fact, at the back of the retina, providing the output to the optic nerve are ganglion cells. A key feature of these ganglion cells is that they

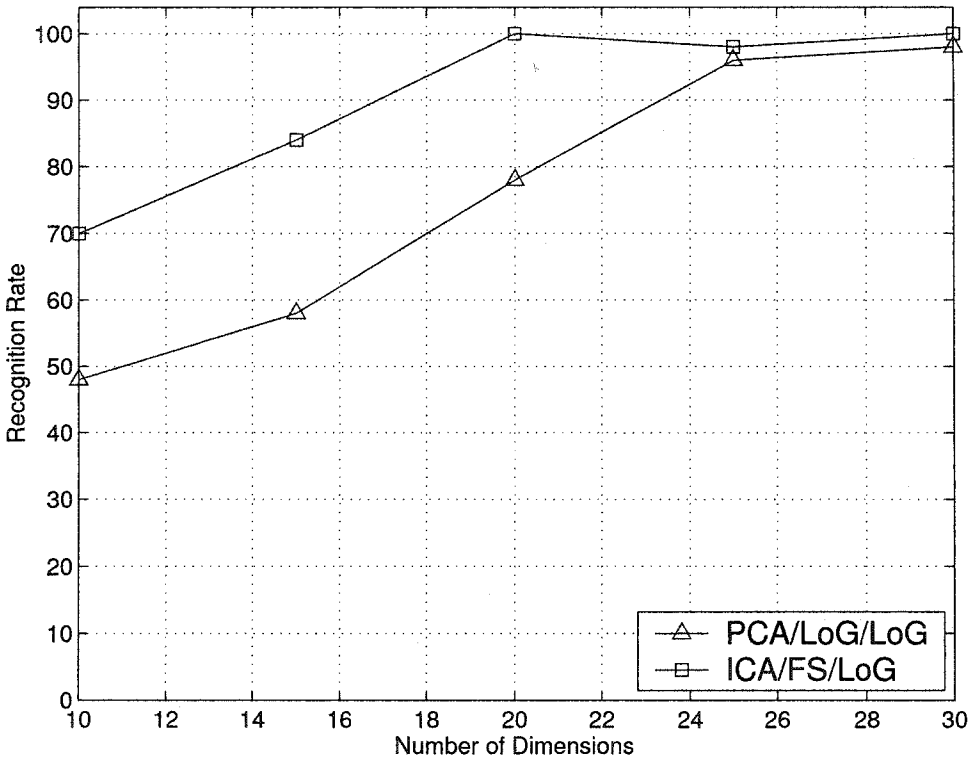


Figure 5.4: Plot of recognition rate for best ICA and PCA results.

have a receptive field (the input from neighboring photoreceptors in a circular area in the retina) which is center surround. Center surround describes a receptive field which has a circular zone at the center and a surround making up the remaining field. These regions are of opposite effect — either center excitatory/surround inhibitory (on-center cells) or center inhibitory/surround excitatory (off-center cells). This implies that the cells respond most to differential illumination of the center and the surround. Importantly, illumination which excites each region equally will almost cancel — providing very weak output from the cell. Thus, a main feature of the human visual system is that it only provides information on the differences in an image. Featureless regions are not "seen" as they provide no output from the retina. Why is the human visual system organized in this way? It is for the simple reason that the useful information in a visual scene is the contrast features, or edges. The absolute reflected

dimension		*30*	*25*	*20*	*15*	*10*
Method		*Recognition (%)*				
No LoG	PCA	76	74	66	56	50
	ICA/FS	32	38	32	26	26
LoG Training	PCA	98	96	78	58	48
& LoG Test	ICA/FS	98	92	86	74	50
LoG Basis	PCA	98	94	82	56	48
Only	ICA/FS	100	98	100	84	70
	ICA/PCA	98	94	82	56	48

Table 5.1: Table of results showing the recognition rate for each technique using subspace dimensions ranging from 10 to 30.

light from a scene is generally uninformative, since it is determined by the intensity of the illumination source. Based on this intelligent and workable design, it seems reasonable to emulate this as a first step in illumination insensitive recognition. A LoG function provides exactly this emulation. Its "Mexican hat" shape is such that the positive region provides the center excitatory region and the negative region conforms to the surround inhibitory region. The integral of these regions is roughly equal, providing a response of near zero when uniformly excited across the support of the function. In the human visual system, the receptive fields are of variable size, from only a few minutes of arc in the foveal region, to 3 to 5 degrees of arc in the periphery. To respond to different spatially sized features, then, there is a need for variable sized receptive fields. More will be mentioned of this in the next experiment, where the tuning of the size of this receptive field is investigated To simplify matters for the current case, the scale of the LoG function (σ) fixed at a reasonable value for the scale of the objects to be recognized. However, this tuning needed to be performed experimentally a-priori.

Figure 5.2 provides a visual demonstration of the effect of the application of the LoG filter to the training images. The PCA and ICA basis images are thus calculated from LoG filtered images. When the LoG filter was not applied, large areas of uniform illumination are visible. When the LoG filter was used, only object edges are visible. An important point here is that the image set is no longer highly correlated as it

was in the previous chapter, thus we no longer get the DCT type basis. The PCA basis is no longer sinusoidal in nature and of global support. The use of the LoG filter effectively removed the low spatial frequency features the high frequency energy is divided up amongst the eigenvectors. However, PCA will still partition the basis so that the small eigenvalues will represent basis images which are predominately high-frequency noise and those edge features that repeat at a low spatial frequency will provide the large eigenvalues and their representative eigenvectors.

When LoG filtering was performed on both training and test images, an interesting difference in the performance between PCA and ICA was observed. For PCA, recognition rates fell below the best unfiltered rate (approximately 75 %) below a subspace dimensionality of 20. For ICA, this did not occur until a subspace dimensionality of 15 (see Figure 5.4). This seems to show that the ICA with a floating search has selected more discriminating features than PCA does using variance of the coefficients as a selection technique. While PCA could be used with a floating search technique, PCA uses variance alone as a measure for finding directions (eigenvectors) in the data. Selecting features with small variance in favor of those with larger variance would directly correspond to favoring noisy directions in the data, which would be counter-productive. There was a slight drop in the recognition rate for ICA with a dimensionality of 25 shown in Figure 5.4. This could be due to the use of a sub-optimal search technique to select features and the fact that the feature selection criterion does not directly maximize recognition rate.

This simple experiment provides strong evidence that ICA provides more discriminating features for the purpose of recognizing patterns in image data than does PCA. This experiment also shows, however, that for this experiment the best features provided by ICA are not better than the best features provided by PCA. This can be easily explained by the cardinal rule of feature selection - the N best features are not necessarily the best N features, even if the features are decorrelated. In other words, a definitive statement of the discriminating power of coefficients in a subspace cannot

be determined by a single set of features. It is necessary to look at all groups of coefficients. As N gets smaller, the N best features and the best N features will converge. As such, smaller subsets of features give more direct evidence of how discriminating the features are. With PCA, reducing N reduces the discrimination power of the coefficients more rapidly than in ICA, illustrating the claim that ICA derived features are more effective. While in practice, this might not seem all that useful, since provided enough features are used, PCA and ICA performance are similar for this experiment. However, if speed of computation is a concern, as might be the case in real-time applications, the tradeoff between classifier performance and dimensionality becomes all important.

Another interesting result arose from the LoG filtering of the basis images only. When PCA was used to reduce the dimensionality of the space over which ICA was applied (no floating search was used), identical recognition results between ICA and PCA occurred. This is to be expected, due to the rotation between the PCA and ICA basis. The result of particular interest, however, is in using ICA with feature selection provided by a floating search. In this case, a significant improvement occurs in the performance of ICA. It would appear that is no formal way of describing why this improvement occurs. However, it can be hypothesized that the use of PCA provides a basis which is much more noisy than the ICA basis. This is due to the calculation of PCA on images which do not have much low frequency content. In the case where both training and test images were pre-filtered with LoG filters, this would not matter, as the bandwidth of the filtered input images matches the basis. In this way, any noise in the basis outside of the bandwidth of the filtered training and test images would not affect the resulting coefficients. The situation is different when the original, unfiltered images are applied to a filtered basis. In this case, large areas of low-frequency content are filtered by high frequency, noisy filters, resulting in noisy coefficients. Perhaps the most interesting question is why the basis images, when constructed in this way, construct an effective basis for the original, unfiltered images. This is a question for further research.

5.1.6 Summary

In the preceding experiment, a number of features of the use of ICA for the recognition of specular objects were illustrated. The foremost of these is that ICA, due to the nature of the basis, is a good choice for representing LoG filtered images. It was shown in this and previous experiments that the ICA basis, due to its edge-type character, provides a natural choice for representing LoG filtered images. It was also shown that LoG filtering provides a degree of lighting independence to the recognition of specular objects. This represents a simple and effective solution to the problem of varying illumination when the lighting model is not Lambertian and thus cannot be modeled by a subspace of low dimension. It was also shown that, in contrast to the previous experiment, where the images were highly correlated, an improvement can be achieved by the use of ICA and the use of moments higher than second order which exist when the data set is not as correlated (since it is comprised of dissimilar looking objects). In the next experiment, ICA will be used to derive filters of a type similar to LoG filters and recognition results will be examined on a face database.

5.2 Faces

5.2.1 ICA vs. LoG Pre-Filters

To demonstrate the performance of ICA derived pre-filters for recognition under conditions of lighting variance, a face recognition experiment was conducted which compared the use of such filters against the use of a LoG filter and no filtering using the Yale Face Database B [74]. The database contains 10 subjects imaged under 9 different poses and 64 lighting positions. Eight sets of results were obtained (one for each pose) by constructing training data sets for each of 8 poses from the first 8 lighting positions of 9 subjects for a total of 72 training images per pose. The test data sets comprised the same subjects imaged under the last 56 lighting positions from the same poses as the training set, creating 504 test images per pose. The training and

test images were histogram equalized and mean centered before subspace calculation and classification. Additionally, all images were cropped to a 200 by 200 pixel square around the image center as supplied by the database, and re-sampled to 50 by 50 pixels.

For the ICA pre-filtering case, classification proceeded as shown in Figure 5.5. The set of 72 database images were formulated as a matrix **I** and the synthesis model

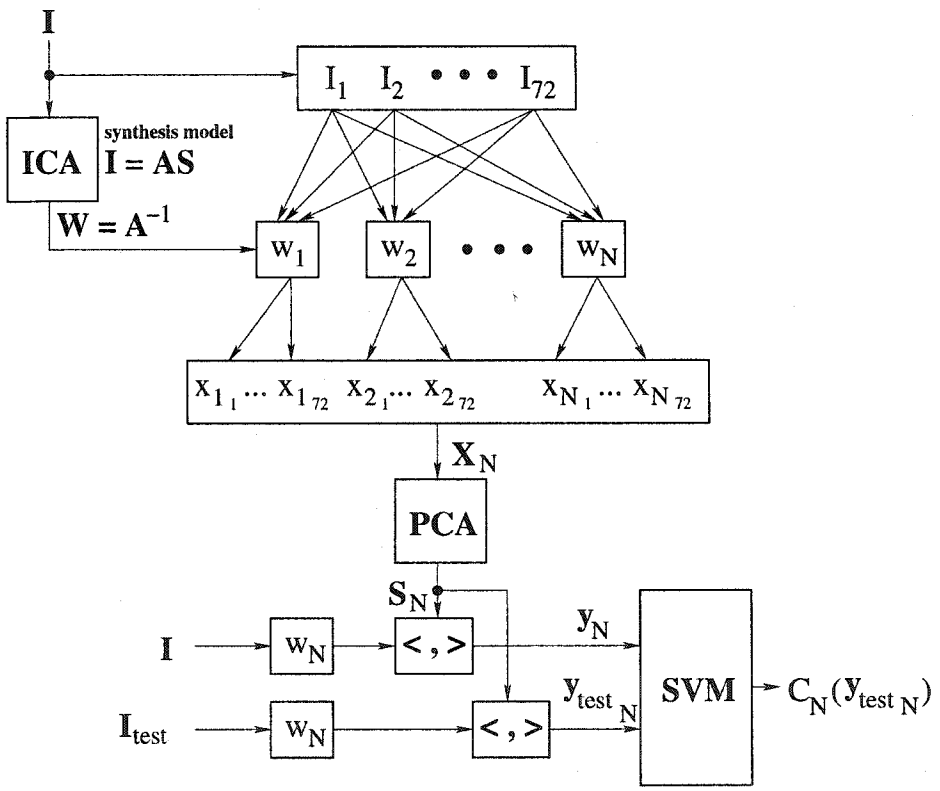


Figure 5.5: ICA pre-filtering

of ICA was used to derive a set of $N = 32$ (reduced in dimension from 64 with PCA) basis images **A** which were inverted ($\mathbf{W} = \mathbf{A}^{+}$) to create a set of pre-filters w_1 to w_N. To find the basis, 10,000 randomly positioned 8 by 8 pixel patches were extracted from the database images. Each filter was applied to both test and training images. All results shown are for the single filter which exhibited the best performance (determined by maximum margin) on the test images. As such, 8 optimal filters were

selected — one for each pose. The SVM kernel ranges were determined a-posteriori on the test images by finding the kernel σ corresponding to the maximum margin and roughly centering it in a range of 10 values in increments of 0.5, thus giving a kernel σ range of 4.5 (inclusive of the end points).

For the LoG pre-filtering case, the use of the synthesis model of ICA for deriving pre-filters was replaced by the use of a LoG filter with a variable σ. For each filter σ, a range of SVM kernel σ values were tested, determined as described above. In all cases, faces (filtered or non-filtered) from the training database were reduced in dimensionality to 25 with PCA to provide a basis and the principal component coefficients of the test images were classified by a SVM. A soft margin support vector machine was used with the parameter C fixed at 100, which produced generally good results over all of the experiments. The one-per-pair of classes SVM method (often called one against one) was used in this experiment (see [83] for a review of multi-class SVM methodologies). In the one-per-pair of classes SVM, $C(C-1)$ two class SVM's are used, where C is the number of classes. Each test pattern is assigned to a class using each classifier in turn and a majority voting scheme determines to which class the test example will be classified.

5.2.2 Classification Results

The 32 ICA pre-filters for pose 1 are shown in Figure 5.6. A sample of 25 basis images for the LoG and ICA pre-filtering case is shown in Figure 5.9, calculated from the pose 1 database. The final SVM classification results are shown in Table 5.2 which describes the mean margin, number of support vectors (NSV) and average number of errors (out of a total of 504) across all 8 poses. Although average errors were shown in the table, at the optimal margin point for both the LoG and ICA cases, the number of errors was zero for all poses. Figure 5.7 shows the distribution of margin across the range of kernel σ tested for each method.

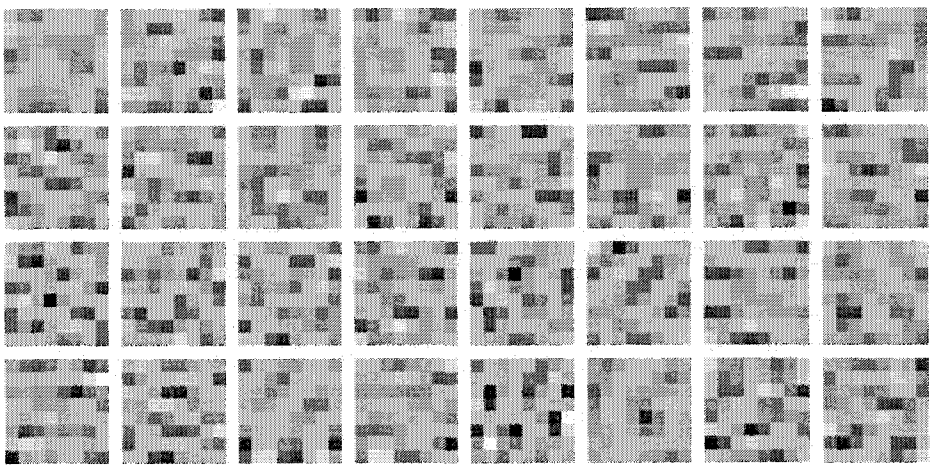


Figure 5.6: ICA pre-filters for pose 1

Method		*Margin*	*NSV*	*# of Errors*	*Kernel σ*
no filter		0.5739	13.20	21.30	5.5-10
ICA		0.7521	12.19	0.125	5.5-10
LoG	$\sigma = 0.4$	0.7643	12.59	0.125	27.5-32
	$\sigma = 0.5$	0.7553	12.08	0.075	10-14.5
	$\sigma = 0.6$	0.7322	11.19	0	5-9.5
	$\sigma = 0.7$	0.7035	12.72	0.150	1.5-6
	$\sigma = 0.8$	0.6817	12.07	0.163	1-5.5

Table 5.2: Means of margin, number of support vectors (NSV) and number of errors for the optimal kernel σ ranges indicated across all 8 poses

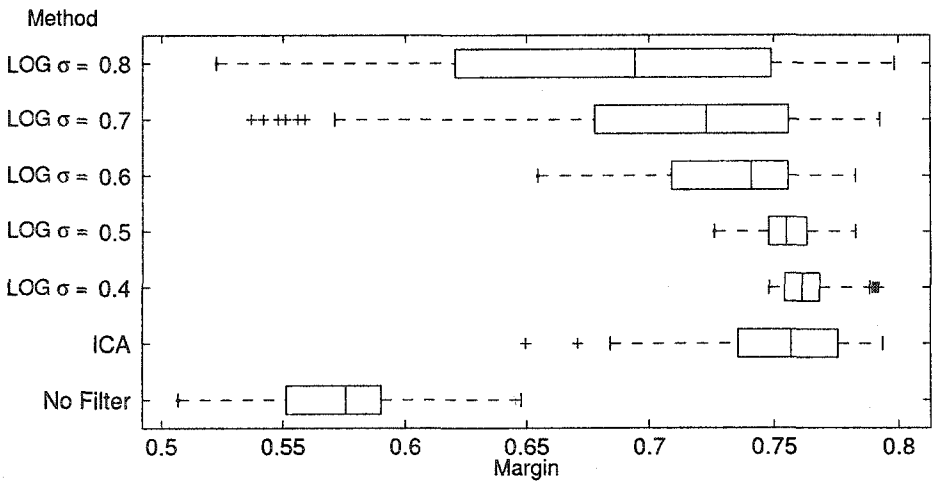
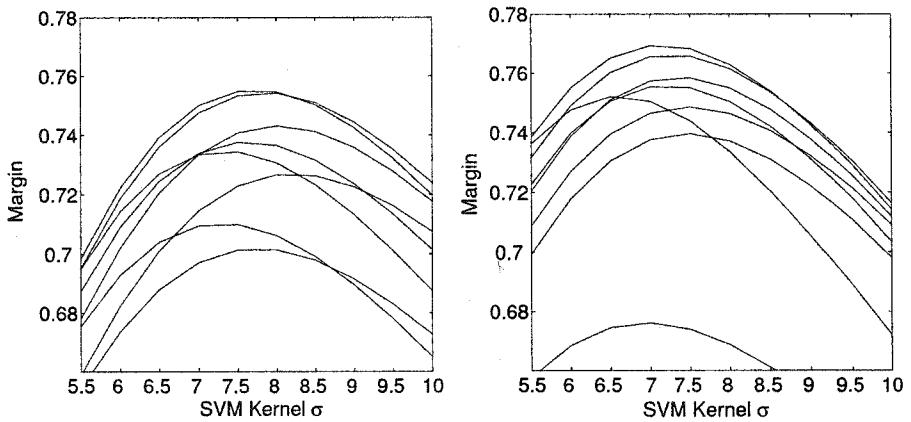
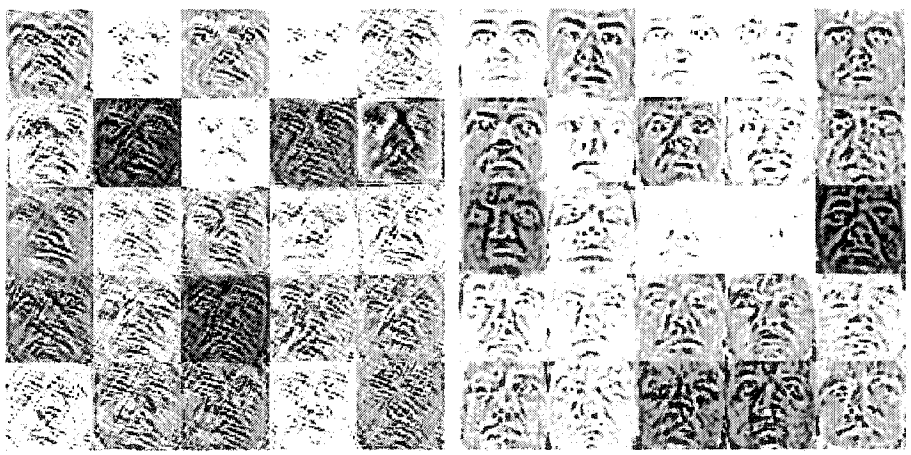


Figure 5.7: Margin across all 8 poses and all SVM kernel sigmas



(a) (b)

Figure 5.8: Margin for 8 of the 32 filters for (a) Pose 1 (b) Pose 8



(a) (b)

Figure 5.9: Pre-filter basis images (contrast enhanced) for pose 1 (a) ICA (b) LoG

5.2.3 Choice of the Kernel

An important consideration when working with support vector machines is the choice of the kernel function. A reasonable hypothesis for this experiment is to assume a multivariate Gaussian distribution for the class-conditional density of the features. That is to say:

$$p(\mathbf{y}|\omega_j) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right] \tag{5.2}$$

with ω_j representing the jth class, μ, $\boldsymbol{\Sigma}$ the mean and covariance of the class features and p being the feature vector dimensionality. This is a reasonable assumption, since the images within each class are quite correlated (they are all of the same face). In other words, most of the differences in the images can be encoded with moments up to the second order. Recall from basic probability theory that Bayes solution for minimum error in a two class problem is that one should assign $\mathbf{y}$ to class ω_i if:

$$p(\mathbf{y}|\omega_i)p(\omega_i) \geq p(\mathbf{y}|\omega_j)p(\omega_j) \tag{5.3}$$

After applying Bayes rule, the following discrimination function results:

$$f_{ij}(\mathbf{y}) = p(\omega_i|\mathbf{y}) - p(\omega_j|\mathbf{y}) = 0 \tag{5.4}$$

If we were to perform kernel discriminant analysis, that is, model the class-conditional density with a kernel (the Parzen method) can be written:

$$p(\omega_i|\mathbf{y}) = \frac{p(\omega_i)}{p(\mathbf{y})}\frac{1}{n_i h^p}\sum_{k=1}^{n} K\left(\frac{1}{h}(\mathbf{y}-\mathbf{y}_k)\right) \tag{5.5}$$

where $K(\mathbf{y})$ is a kernel function satisfying $\int K(\mathbf{y})d\mathbf{y} = 1$, $\mathbf{y}_k$ is a set of n_i p-dimensional sample features in class ω_i and h is a kernel smoothing parameter. Therefore, the term in the discrimination function for the *ith* class is of the form:

$$\sum_{k=1}^{n} w_i \phi_k(\mathbf{y} - \mathbf{y}_k) \tag{5.6}$$

where $\phi_i(\mathbf{y} - \mathbf{y}_k) = K((\mathbf{y} - \mathbf{y})/h)$ and $w_i = \frac{p(\omega_i)}{n_i}$. The term $p(\mathbf{y})h^p$ can be ignored since it is independent of i. What this has shown is that for the Bayes decision rule, a discriminant function has terms in the form of a radial basis function with a center at each data point and weights determined by class priors. If the kernel function is chosen to be a Gaussian RBF, we are using a sum of Gaussian kernels with centers at each data point and we are actually approximating the density $p(\omega_i|\mathbf{y})$ or $p(\mathbf{y}|\omega_i)$ through Bayes rule. Needless to say, it is reasonable to model a Gaussian density with a Gaussian RBF, although it is not necessary. Other kernels could be used. The Gaussian RBF, however, can find a very simple and accurate density estimation function.

The above describes the choice of a kernel for kernel discriminant analysis and a support vector machine was employed herein. However, the discriminant functions resulting from the use of a support vector machine with a Gaussian RBF kernel and the Gaussian RBF classifier as described above are identical in form. However, the meaning of the weights is very different. In the support vector machine, the weights are coefficients that make an optimal separating hyperplane and in the RBF classifier they are class priors. The support vectors correspond to the data point centers in the RBF classifier. None the less, if the SVM classifier decision boundary follows the optimal Bayes decision boundary, the results between the two methods will be the same and the SVM classifier has effectively modeled the class-conditional densities. This provides a direct justification for using a Gaussian RBF kernel with the SVM for classifying data which has Gaussian class-conditional densities. In fact, the average number of support vectors for perfect classification in this experiment

was low (approximately 11 out of 16 or about 69 % of the data points were support vectors).

5.2.4 Discussion

From Table 5.2 it is apparent that the σ_{LoG} of the LoG filter significantly affects the optimum kernel σ_{SVM}. This implies that the LoG filter must be tuned to the spatial resolution of the dataset images for optimum performance of the SVM. Unfortunately, this is not known a-priori. The ICA filters, on the other hand, exhibit maximum performance over precisely the same range as the best LoG filter σ_{LoG} by acquiring the characteristic spatial resolution of the dataset. The distribution of margin across the tested range of kernel σ_{SVM} in Figure 5.7 is particularly revealing of the performance of each method. First, no filtering obviously offers poor performance. More interesting, however, is the variation in the spread of the margin for each LoG filter σ_{LoG}. For large filter σ_{LoG} (corresponding to a small kernel σ_{SVM}, see Table 5.2) a large spread in margin is observed. This occurs because the *range* of the kernel σ_{SVM} was fixed at 4.5. The implication here is that the optimal kernel σ_{SVM} is, of course, more sensitive at smaller values. All of this makes it quite difficult to hunt for the best filter σ for small values of kernel σ_{SVM}. Overall, the two degrees of freedom (filter σ_{LoG} and kernel σ_{SVM}) result in a difficult tuning problem.

Some interesting characteristics of ICA derived filters are illustrated in Figure 5.9. Clearly, the filters are of the same nature as the LoG filters. More subtly, the ICA filters exhibit some direction specificity. In a few obvious cases, for example, diagonal directions seem to be preferred, for example, in the upper left face, where diagonal lines of noise have been enhanced by the filter. A glance at the filters from Figure 5.6 show some interesting dominant directions. The filters also have limited spatial support. Filter 22, (counting left to right and down from the top left) for example, is diagonally oriented (top left to bottom right) and has a width of only about 3 pixels along the diagonal direction. Additionally, a pattern of alternating dark and light

bands of a single pixel width runs the length of the diagonal. While it is impossible to specifically know which physical features generated these specific filter patterns, Chapter 1 gives some research examples where ICA is compared to low-level feature detectors in the human visual system. It seems likely that the patterns observed in this experiment are the same low-level features such as lines and oriented bars that are mentioned in the work of Hyvarinen, Bell and Sejnowski and others in the context of ICA feature extraction. This experiment, then, provides more evidence of the fundamental utility of edge detection in the process of vision. What is critical, however and not intuitive is that sparsity is the mechanism behind the extraction of oriented edge-like filters. When the the ICA coefficients are made as statistically independent as possible, as they were in this experiment, the resulting distributions of the coefficients are super-Gaussian (most coefficients are very near zero - the definition of sparsity).

One significant problem with the use of ICA filters is that while the optimal SVM kernel σ remains relatively fixed for each of the 32 filters across all poses (Figure 5.8 shows the margin variation with kernel σ for 8 of the 32 filters and 2 poses), the performance of each filter does not. This necessitates the selection of the best filter (or combination of filters). A couple of options have been examined in this regard. One is to select the best filter a-priori by examining the statistical characteristics of the filters with kurtosis seeming to be a promising measure. Additionally, the best filter could be selected by cross-validation on the training images.

5.2.5 Summary

With respect to the previous experiment, LoG filters alone were used to provide a measure of lighting invariance to the recognition of specular objects. In this experiment, it was shown that ICA could be used to derive filters of the same type as LoG, which are effective for removing the effects of lighting variation in image recognition.

The importance of tuning LoG filters, particularly in the case of using a kernel classification technique such as a SVM was illustrated. As an alternative to using LoG filters of multiple spatial resolutions, ICA was used to derive filters of spatial resolutions and orientations that are exhibited by the features in the database images. The important features in the images, for the purposes of lighting invariant recognition are the edges and these are precisely the types of features that are extracted by ICA and its sparse representation.

Chapter 6

Improved SVM Classification

6.1 Modifying PCA and ICA

6.1.1 Synthetic Example: Gaussian Mixture

To illustrate the relationship between PCA and ICA and to illustrate the effectiveness of iterative modification of coefficients, a synthetic example of 2 classes, each comprising a mixture of 3 Gaussian random variables, shown in Figure 6.1(a), is used to calculate principal and independent component subspaces. In this example, $n > p$ so we use $\mathbf{X}\mathbf{X}^T$ to compute the eigenvectors. The mixture of Gaussian data points:

$$\mathbf{X} = \left[\begin{array}{cc} \mathbf{X}_{C1} & \mathbf{X}_{C2} \end{array} \right] \tag{6.1}$$

are defined by:

$$\begin{aligned} \mathbf{X}_{C1} &= \sum_{n=1}^{3} \frac{1}{|\Sigma_n|} \exp\{(\mathbf{x}-\mu_n)\Sigma_n(\mathbf{x}-\mu_n)^T\} \\ \mathbf{X}_{C2} &= \sum_{n=4}^{6} \frac{1}{|\Sigma_n|} \exp\{(\mathbf{x}-\mu_n)\Sigma_n(\mathbf{x}-\mu_n)^T\} \end{aligned} \tag{6.2}$$

where

$$\begin{aligned} \mu_1 &= \begin{bmatrix} -4 \\ -2 \end{bmatrix} & \mu_2 &= \begin{bmatrix} -6 \\ 0 \end{bmatrix} & \mu_3 &= \begin{bmatrix} -3 \\ 3 \end{bmatrix} \\ \Sigma_1 &= \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix} & \Sigma_2 &= \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} & \Sigma_3 &= \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \\ \mu_4 &= \begin{bmatrix} 4 \\ -3 \end{bmatrix} & \mu_5 &= \begin{bmatrix} 3 \\ 1 \end{bmatrix} & \mu_6 &= \begin{bmatrix} 5 \\ 4 \end{bmatrix} \\ \Sigma_4 &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} & \Sigma_5 &= \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix} & \Sigma_6 &= \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \end{aligned} \tag{6.3}$$

6.1.2 PCA and ICA comparison

The resulting PCA and ICA basis vectors are related by an orthonormal transformation. Additionally, the PCA and ICA basis vectors are orthonormal (an example can be seen in Figure 6.1(a)). As such, distances in the original space are preserved under a transformation using either PCA or ICA bases (see Figures 6.1(b) and 6.1(c)). In the example of Figure 6.1(a) the axes chosen for the principal components are clearly not optimal for discrimination, however the independent axes better represent the axes along which the classes can be separated. Again, if Euclidean distance is used as a classifier, the distances in the PCA space are preserved in the ICA space as can be verified by measuring some distances between data points.

6.1.3 Iterative Components

The CSVR algorithm described in Chapter 3 is tested using the synthetic example of 2 classes described above. Thirty data points per class are used as training, and

30 data points per class are used as test. The margin, number of support vectors and error rates are collected from the SVM for each of 5 different widths of the kernel. The experiment is run with 50 instances of the Gaussian random variables. Iteration is terminated when the margin change was less than 0.0001. The original data, principal and independent coefficients and the result of the iterative adaptation are shown in Figure 6.1 for one of the example data sets. Box plots of the SVM output are shown in Figure 6.2 showing quartiles of the number of support vectors, margin and recognition rate. Table 6.1 shows the mean margin, number of support vectors N_{sv} and recognition rate, along with their Z scores (with respect to PCA) for the entire mixture of Gaussian dataset, averaged across all values of kernel σ from 1 to 5.

6.1.4 Face Database - Pose and Lighting Variance

To demonstrate the algorithm for images, Yale Face Database B is employed. The database contains 10 subjects imaged under 9 different poses and 64 lighting positions. For this experiment, multiple 2 class recognition experiments are performed over 36 pairs of subjects. For each pair of subjects, a training data set is constructed from the first 32 lighting positions for the poses 1 and 2 of each subject. The test data set comprised the same pair of subjects imaged under the last 32 lighting positions from the poses 7 and 8. As such, the recognition will therefore require some degree of lighting and pose invariance. The training and test images were histogram equalized and mean centered before subspace calculation and classification. For this example, $p > n$ so we use $\mathbf{X}^T\mathbf{X}$ to compute the eigenvectors. Recognition performance (margin, number of support vectors and error rate) is tested for each subject pair for kernel σ ranging from 1 to 5.

The dimensionality of the training subspace is reduced to 25 prior to recognition. For PCA, this is done by selecting the 25 basis images with the largest variance. For ICA, the dimensionality is reduced to 40 a-priori using PCA and further reduced to

	Margin		N_{sv}		Recognition Rate (%)	
Method	mean	Z	mean	Z	mean	Z
No Subspace	0.3385	0	17.3560	0	97.6200	0
PCA	0.3385		17.3560		97.6200	
ICA	0.3381	-.0275	17.2700	0.0662	97.5933	-0.1082
PCA Iterative	0.3911	2.8399	12.5480	4.3525	97.7933	0.7247
ICA Iterative	0.3894	2.7506	12.7800	4.1019	97.7800	0.6698

Table 6.1: Classification results for mixture of Gaussian dataset showing mean and Z scores (with respect to PCA)

	Margin		N_{sv}		Recognition Rate (%)	
Method	mean	Z	mean	Z	mean	Z
No Subspace	0.2178	-4.7069	125.1556	-8.0264	93.0469	-0.6230
PCA	0.2408		116.8556		93.4679	
ICA	0.2664	3.9515	108.8556	4.6647	93.0452	-0.5875
PCA Iterative	1.2509	89.8193	12.7833	103.4372	95.2300	2.4941
ICA Iterative	1.2475	89.5323	13.2500	103.2759	95.2083	2.4772

Table 6.2: Classification results for Yale Face Database showing mean and Z scores with respect to PCA

25 using a floating search to select the optimum features, based on maximizing the mean inter-class Euclidean distance for all training points in the subspace. When the iterative algorithm is applied, the basis images are initialized to those found by PCA and ICA. Iteration is terminated as in the previous experiment.

Figures 6.3(a) and 6.3(b) show the training images for 2 of the faces (selected randomly) from the dataset. Figures 6.3(c) and 6.3(d) show the test images for the same 2 faces. Figure 6.4 shows the resulting principal, independent and iterative basis images for the training images shown in Figures 6.3(a) and 6.3(b). Box plots of the SVM output for the face database are shown in Figure 6.5 showing quartiles of the number of support vectors, margin and recognition rate. Figure 6.6 shows the average margin, average number of support vectors and average recognition rate as the kernel σ is varied from 1 to 5. Table 6.2 shows the average number of support vectors, margin and recognition rate, along with their Z scores (with respect to PCA) for the entire dataset, averaged across all values of kernel σ.

6.1.5 Discussion

A number of significant results are illustrated by these experiments:

- Principal and independent components have an implicit relationship and Euclidean distance classifiers are ineffective at illustrating the difference between these two data representations.
- When a support vector classifier is employed, independent component representations consistently exceed the margin and reduced the number of support vectors over both the principal component representation and the raw data.
- Enhanced generalization performance and lower error rates can be achieved by using the support vector coefficients to modify the PCA and ICA representation. The classes became more compact creating a corresponding decrease in the number of support vectors and increased margin.
- During each iteration of the iterative components algorithm, a rapid decrease in the number of support vectors rapidly decreases the number of modified features providing an exponential increase in margin (Figures 6.5(e) and 6.5(d)).
- Pose and lighting variances in images, when treated as outliers in the dataset can be effectively classified by a support vector classifier.
- The small reduction in recognition performance between PCA and ICA is not statistically significant, so each method performed about equally. The improved recognition performance of the iterative technique is statistically significant.
- The improvement in generalization for ICA and the iterative techniques illustrated by improved margin and reduced number of support vectors is statistically significant for both results on the face database.
- Iterative PCA and iterative ICA exhibit similar performance with respect to margin, number of support vectors and recognition rate for both the Gaussian

mixture and the face database.

- Recognition rate improves slightly for increasing kernel σ for no subspace, PCA and ICA, with little change for the iterative technique.
- Margin improves almost linearly with increasing kernel σ for the no subspace, PCA and ICA cases, as is evident in Figure 6.6(a). The number of support vectors also decreases linearly with increasing kernel σ, as shown in Figure 6.6(b). The iterative technique shows the opposite effect for both margin and number of support vectors although the change is not as significant.

There have been a number of pattern recognition results published in the past where PCA and orthonormal ICA basis are compared under a Euclidean distance classifier. Such comparisons are valid when the PCA and ICA basis do not span the same space, such as when a subset of the ICA components are selected by a floating search or branch and bound techniques. Due to the difficulty of the feature selection problem (dimensionality reduction) inherent in pattern recognition, the support vector classifier allows the performance of PCA and ICA to be reliably compared. Additionally, both the principal and independent subspaces can be de-noised by dimensionality reduction with PCA and the uniqueness of the classification results for the two techniques will be maintained.

Considering the support vectors to define the data outliers is shown to be a useful idea when classifying datasets with widely varying classes. Pose variance in images creates large changes in object appearance and thus complex class shapes. Lighting variance in an image dataset is of limited dimensionality for Lambertian surfaces [50] however can still create classes with high variance. The algorithm employed herein offers a general solution to creating compact classes for support vector classification.

An open question remains as to why ICA significantly outperforms PCA in generalization for the pose and lighting variant images under support vector classification. Typically, independent component bases comprise the image edges, which are significant features in the context of pattern recognition. This is equivalent to stating that

the defining features of images are described in the higher order statistical relationships and that image datasets contain highly non-Gaussian statistics.

6.2 Generating Optimal Features

6.2.1 Two Class Recognition with CSVR

Figure 6.7 shows training and test images for the first two objects and faces from the COIL [84] and Yale databases. The resulting basis images for the training images are shown in Figure 6.8 for both the COIL and Yale experiments. For this experiment, the basis search was initialized by an identity matrix. This permitted the observance of the convergence characteristics of the algorithm from a common starting point on all data sets. As a result, convergence could be averaged over all of the test cases.

6.2.2 Object Database - Pose Variance

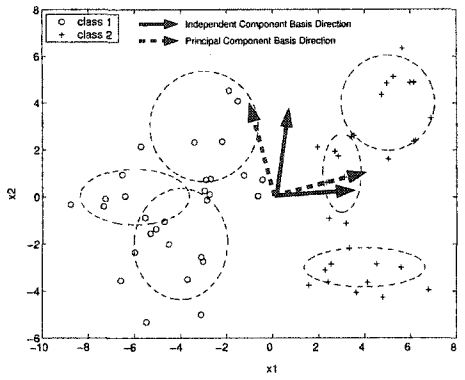
The CSVR algorithm was tested on general objects from the COIL database. 30 objects under poses ranging from 0 to 355 degrees were classified with a soft margin support vector classifier with a large value of $C = 100$. This yielded 435 two class classification examples. The training data consisted of objects at poses taken every 10 degrees starting from 0 degrees. The test data used objects at poses taken every 10 degrees starting from 5 degrees. The dimensionality of the learned subspace was 25. Recognition performance (margin, number of support vectors and error rate) is tested for the raw data for each subject pair for kernel σ ranging from 1 to 50. The recognition results for the largest margin case and the results after the termination of the CSVR algorithm are shown in Figure 6.9.

6.2.3 Face Database - Pose and Lighting Variance

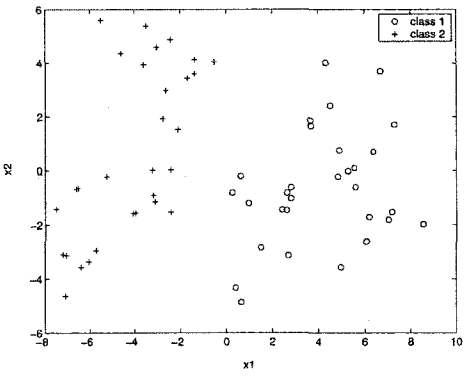
To demonstrate the CSVR algorithm for face images, Yale Face Database B is employed. The database contains 10 subjects imaged under 9 different poses and 64 lighting positions. For this experiment, multiple 2 class recognition experiments are performed with the SVM ($C = 100$) over 36 pairs of subjects. For each pair of subjects, a training data set is constructed from the first 32 lighting positions for the poses 1 and 2 of each subject. The test data set comprised the same pair of subjects imaged under the last 32 lighting positions from the poses 7 and 8. As such, the recognition will therefore require some degree of both lighting and pose invariance. The training and test images were histogram equalized and mean centered before subspace calculation and classification. Recognition performance for the raw data (as described above) and the CSVR algorithm with a subspace dimension of 25 is shown in Figure 6.10.

6.2.4 Convergence

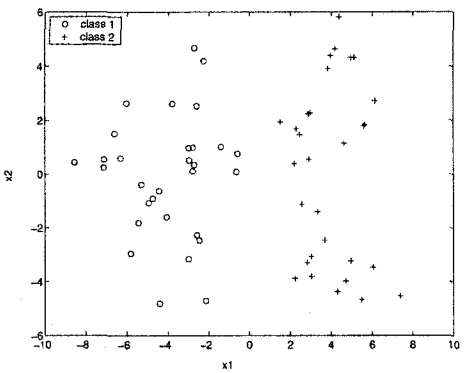
To illustrate the convergence of the algorithm, the volumes of the classes, margin, and number of support vectors were plotted as an average across all test cases. Convergence, of course, occurred at a different number of iterations for each test. Thus, to find an average, the termination condition was fixed at 401 iterations for the COIL database and 72 iterations for the Yale database. Volume for each class was estimated by sum of the the absolute distances of each subspace data point to its class center. The average convergence characteristics for the COIL test is shown in Figure 6.11 and for the Yale test in Figure 6.12. The maximum achievable geometric margin, $\sqrt{2}$ (see Equation 3.8), is shown as a dashed line on the mean margin plots.



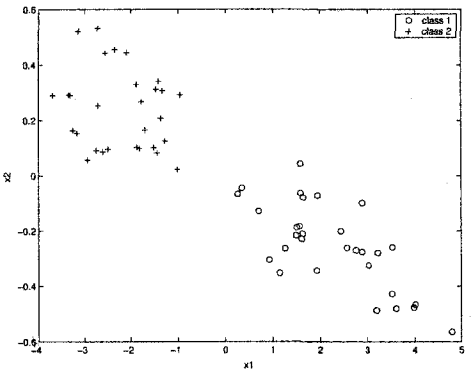
(a) Raw Data (Dashed ellipses represent the 50 percentile probability contour)



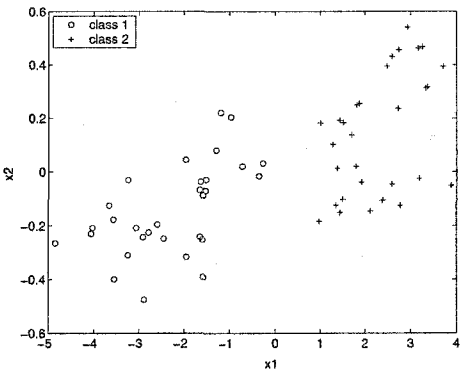
(b) Principal Component Coefficients



(c) Independent Component Coefficients

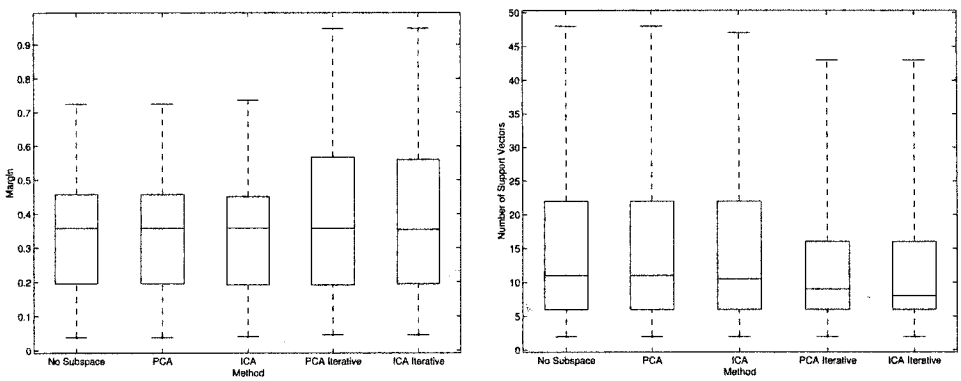


(d) Modified Principal Component Coefficients



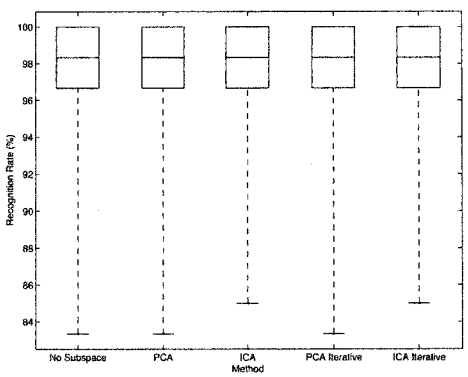
(e) Modified Independent Component Coefficients

Figure 6.1: Example Mixture of Gaussian Data Set



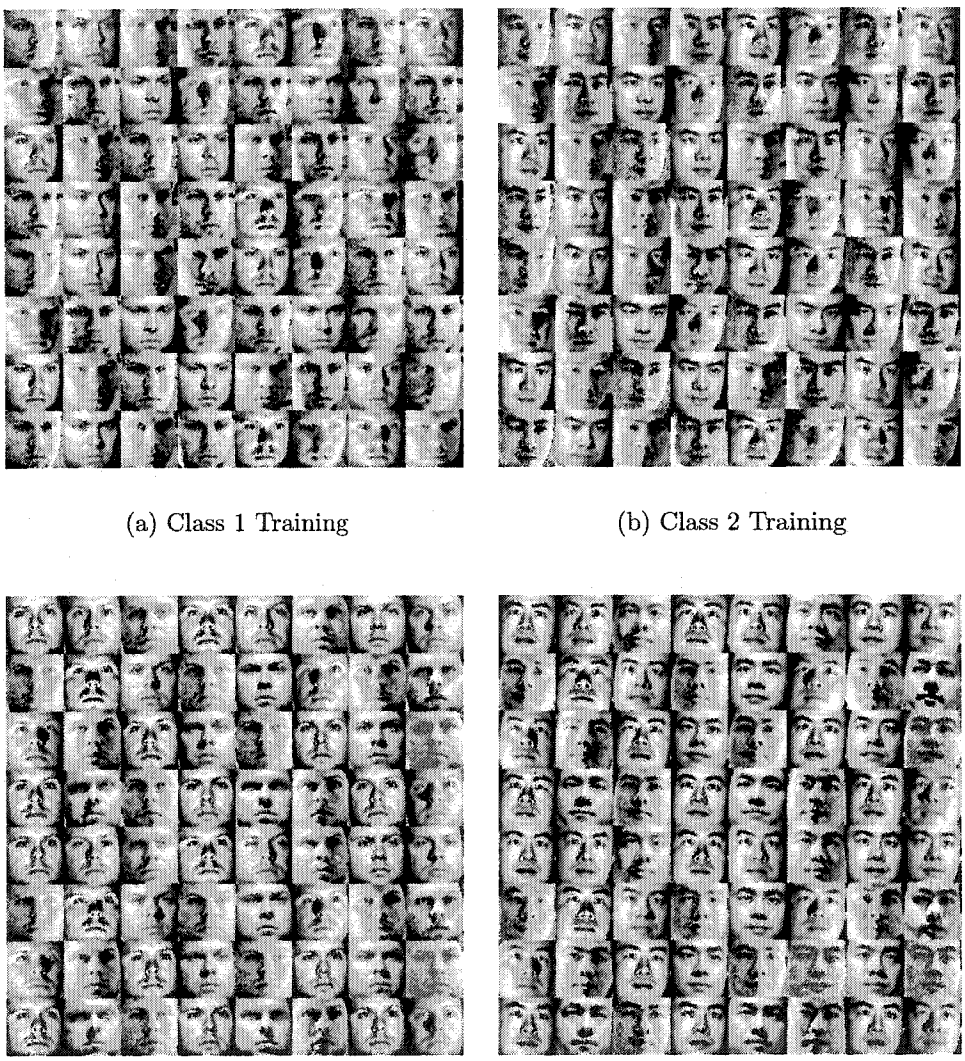
(a) Margin

(b) Number of Support Vectors



(c) Recognition Rate

Figure 6.2: SVM classification of Gaussian Mixture



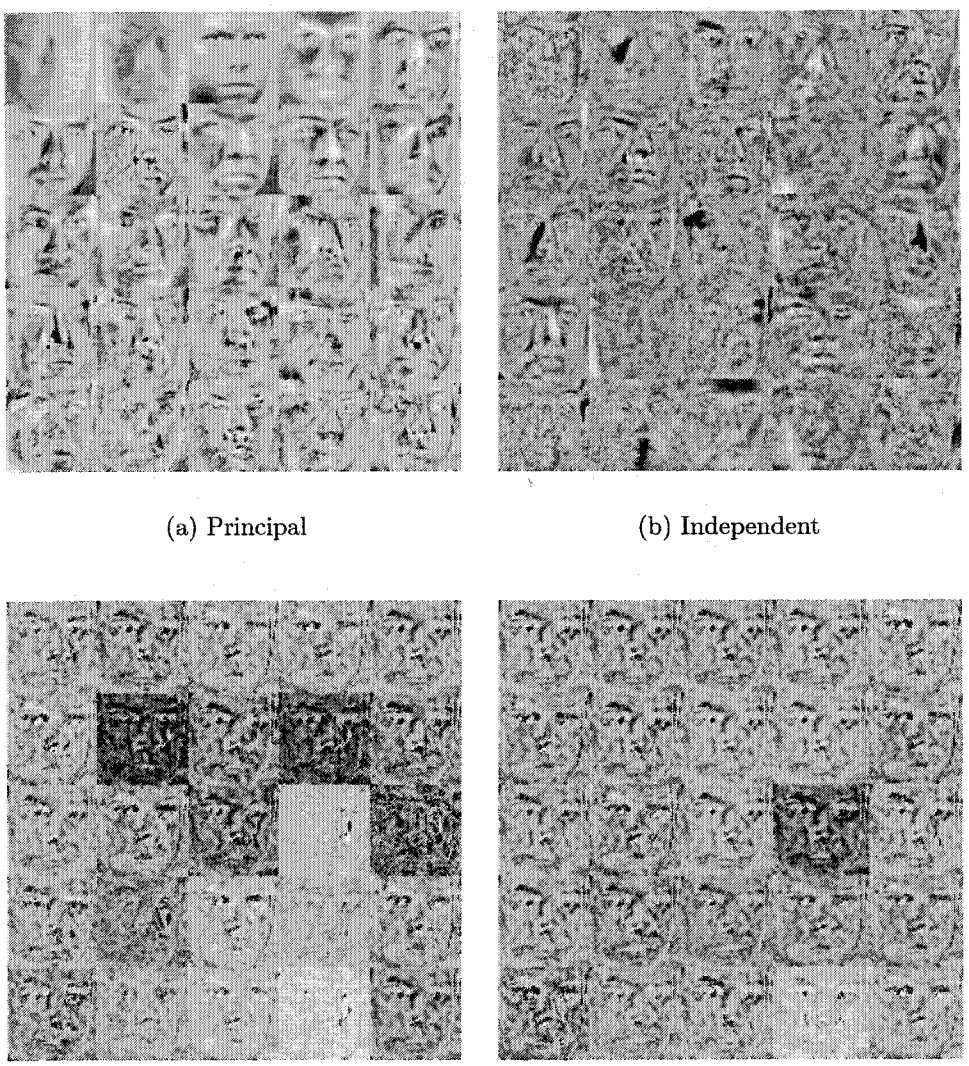
(a) Class 1 Training

(b) Class 2 Training

(c) Class 1 Test

(d) Class 2 Test

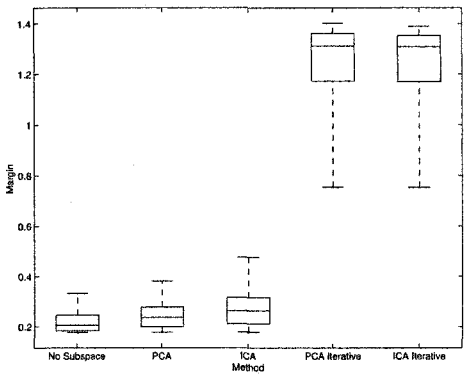
Figure 6.3: Example of Training and Test Images



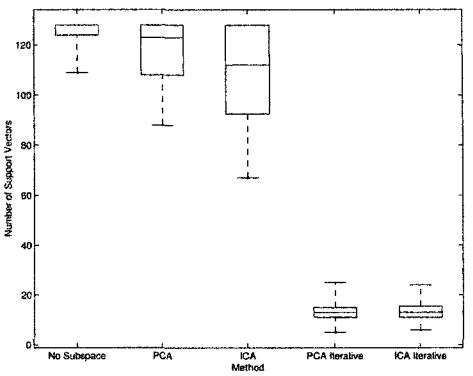
(a) Principal (b) Independent

(c) Iterative PCA (d) Iterative ICA

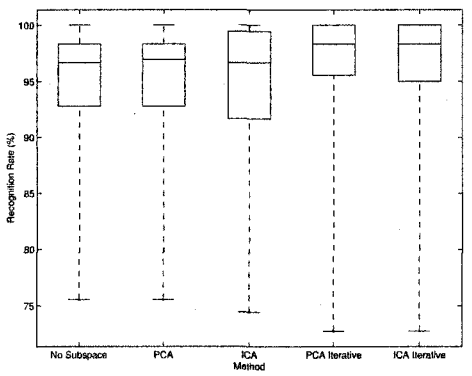
Figure 6.4: Example Components (contrast enhanced)



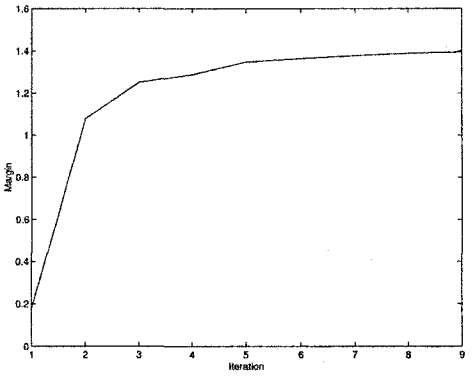
(a) Margin



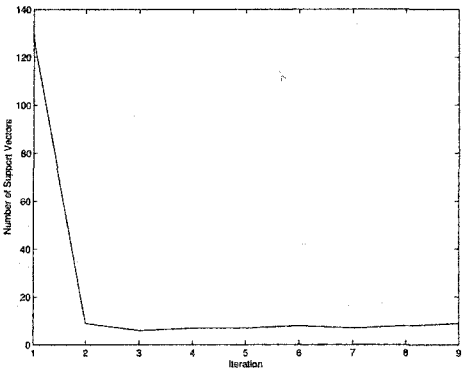
(b) Number of Support Vectors



(c) Recognition Rate

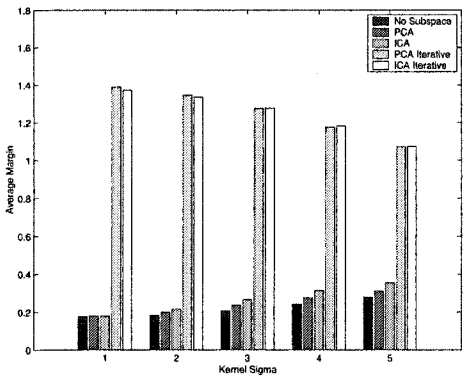


(d) Margin Change Over Iteration

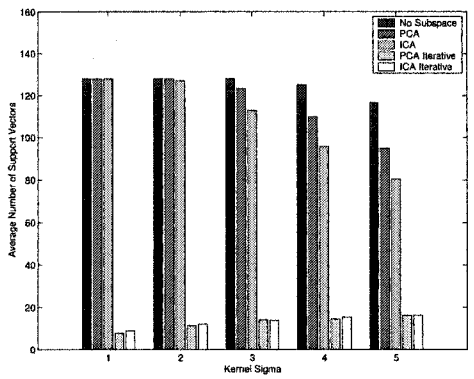


(e) Number of Support Vector Change Over Iteration

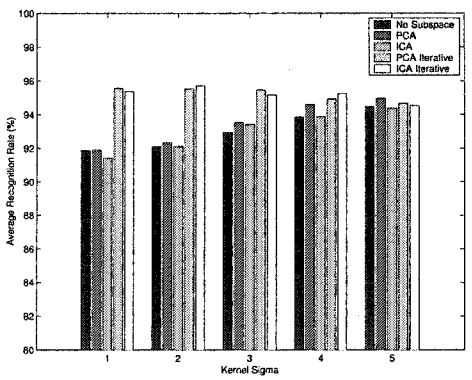
Figure 6.5: SVM classification of Face Database



(a) Average Margin



(b) Average Number of Support Vectors



(c) Average Recognition Rate

Figure 6.6: Average Performance vs Kernel Sigma for Face Database

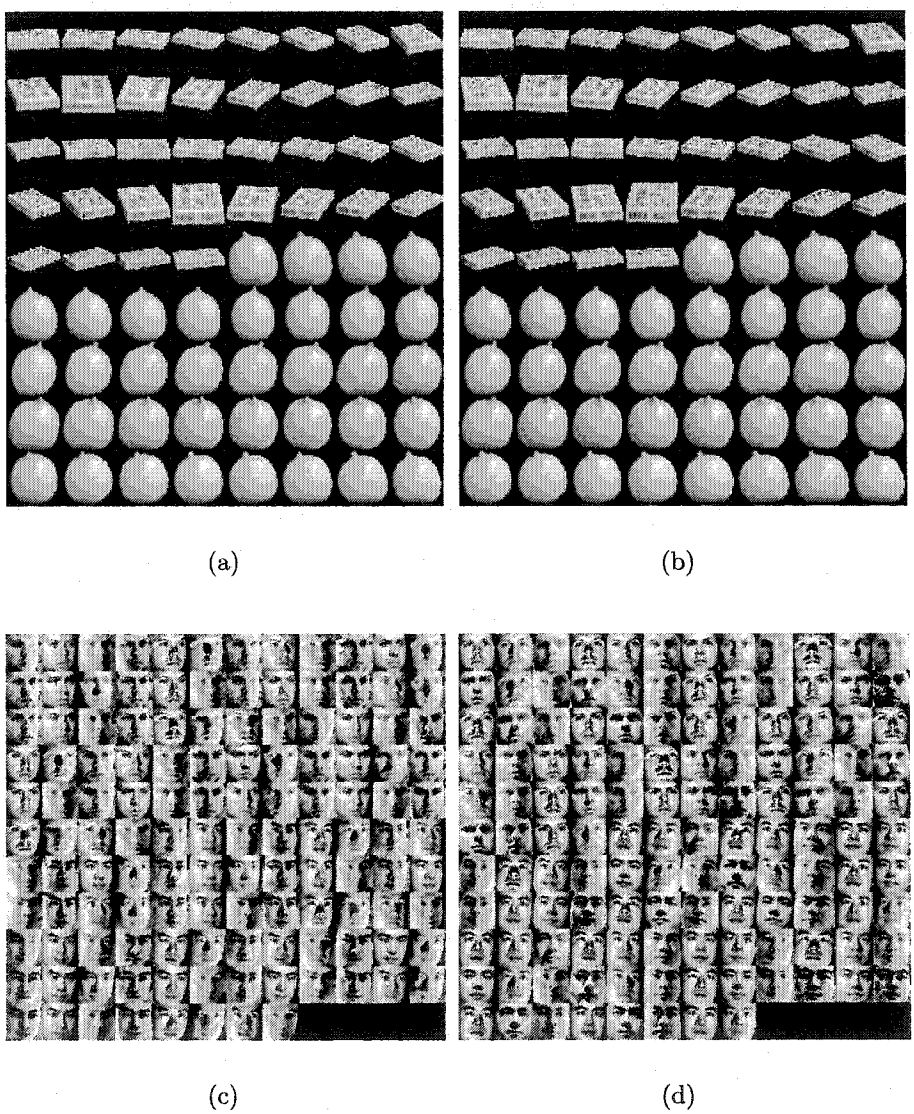
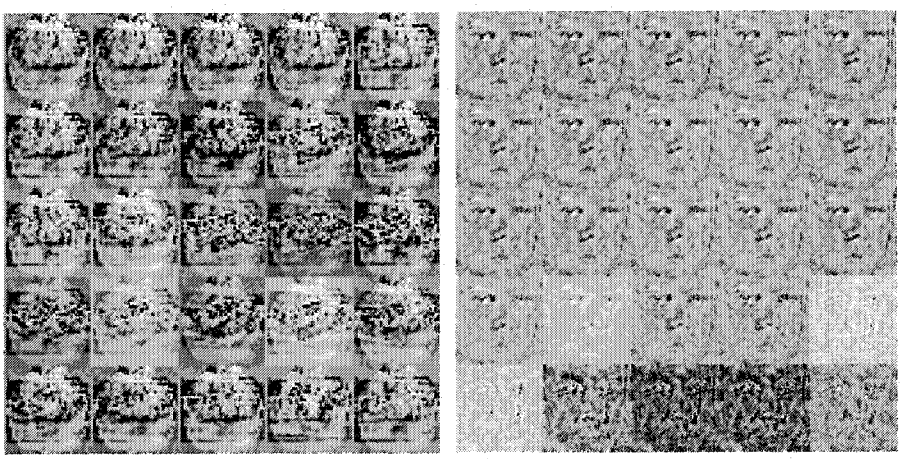


Figure 6.7: Example images of (a) COIL Training (b) Coil Test (c) Yale Training and (d) Yale Test.



(a) (b)

Figure 6.8: Basis images for above dataset for (a) COIL (b) Yale (Brightness and contrast have been enhanced)

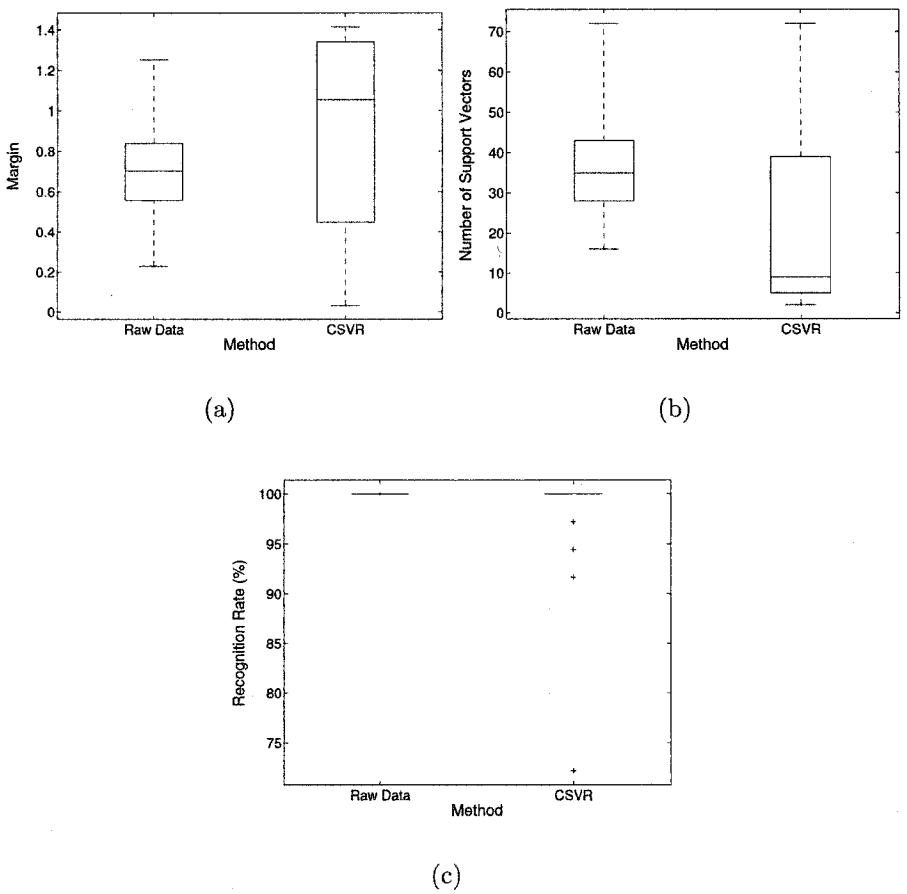
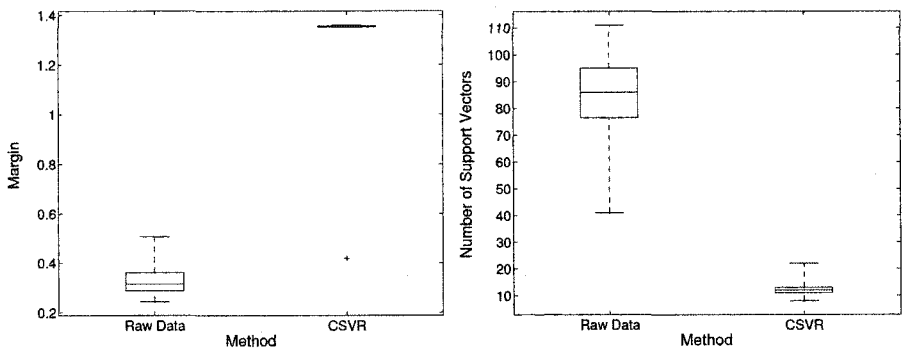
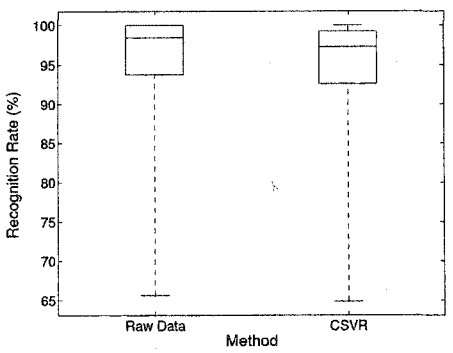


Figure 6.9: Box Plots for COIL results (a) Margin (b) Number of Support Vectors (c) Recognition Rate.



(a) (b)



(c)

Figure 6.10: Box Plots for Yale results (a) Margin (b) Number of Support Vectors (c) Recognition Rate.

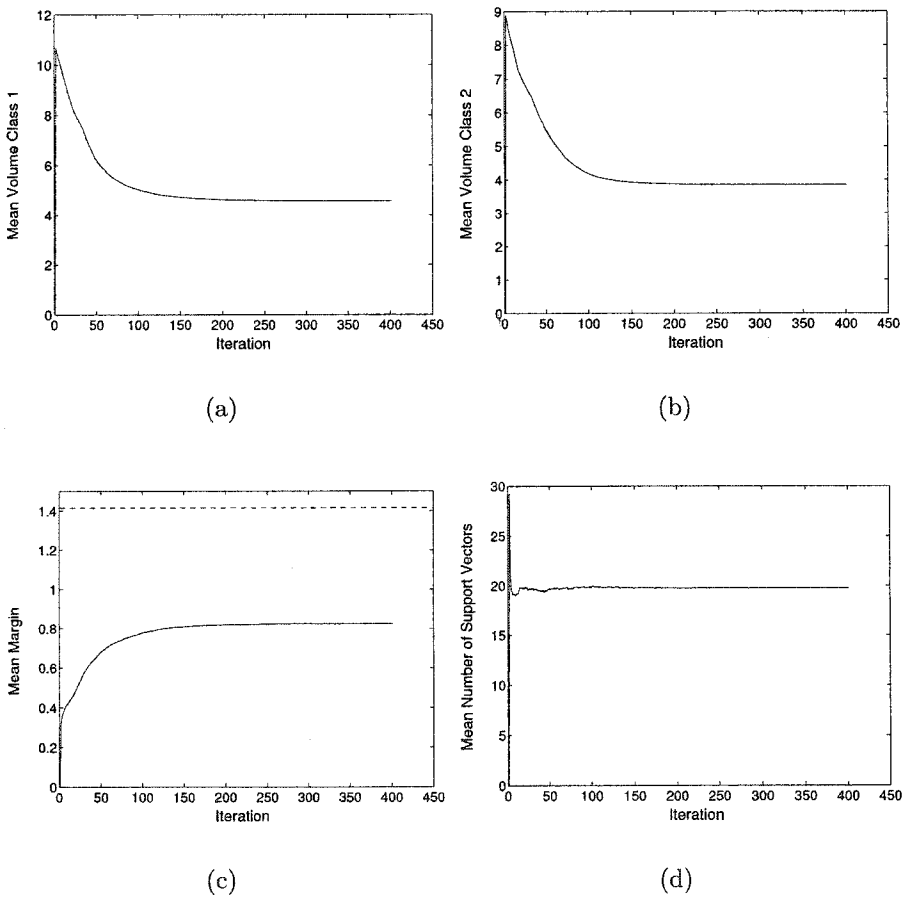


Figure 6.11: COIL averages per iteration of (a) Volume of Class 1 (b) Volume of Class 2 (c) Margin (d) Number of Support Vectors.

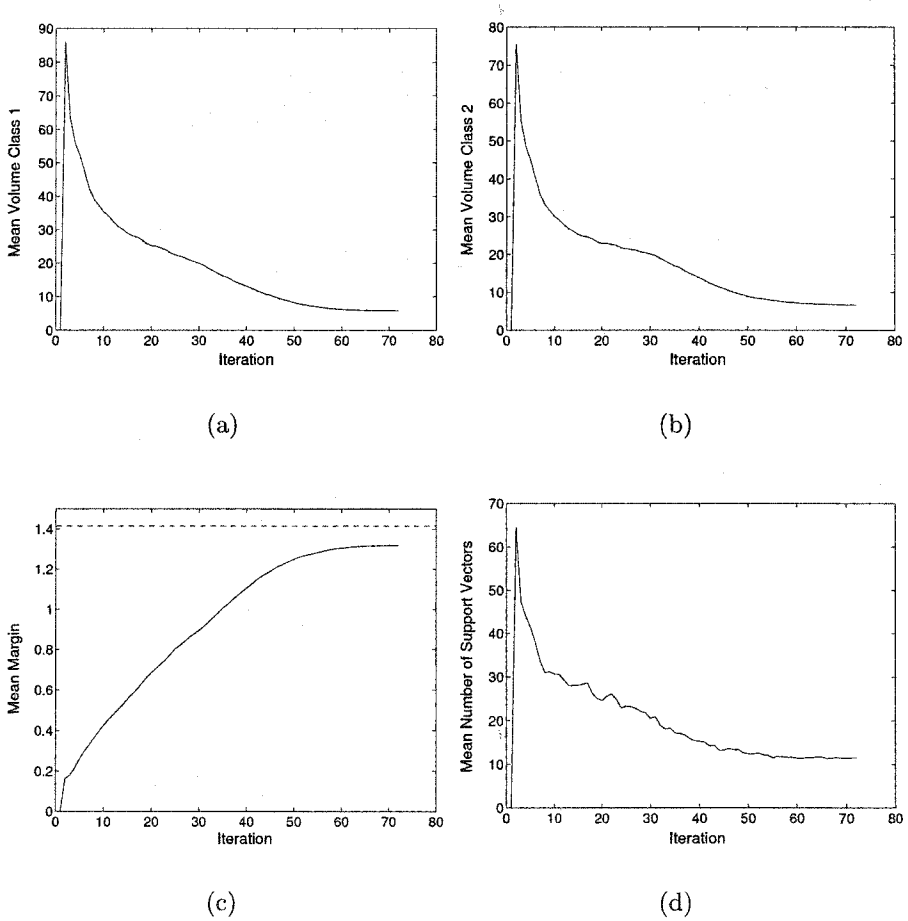


Figure 6.12: Yale averages per iteration of (a) Volume of Class 1 (b) Volume of Class 2 (c) Margin (d) Number of Support Vectors.

6.3 Discussion

6.3.1 Choice Of SVM Parameter C

C was chosen as 100 empirically, however there was no significant change in the results (error rate, margin, or number of support vectors) over a very wide range of values, from $C = 0.1$ to $C = 100$ for both datasets. The largest value of C was chosen under the assumption that a heavy penalty for errors creates more complex decision boundaries making it easier to illustrate the reduction of complex decision boundaries into simpler ones. In any case, the same value of C was used for the raw data SVM and the CSVR SVM.

6.3.2 Raw Data and CSVR Results

For both databases, the average recognition rate was almost identical between the raw data and CSVR classification. However, substantial increases in margin and decreases in the number of support vectors resulted from the use of CSVR. This is a direct indication that the CSVR's ability to generalize effectively for data with characteristics typical of image databases. The basis images indicate a highly redundant coding, with a lot of the basis images exhibiting similarity. This is in sharp contrast to PCA, which provides a set of decorrelated bases. While this type of coding would be highly inefficient for image coding applications, redundant coding is gaining ground for applications in image recognition.

6.3.3 Volume, Margin and Number of Support Vector Convergence

The averages of volume, margin and number of support vectors over multiple iterations show a strong relationship between these quantities. For the case of image databases, it appears that reduction in class volume in the direction of the class means

provides an effective and well behaved way to regularize the class shapes. Average convergence occurred quite rapidly for both databases, after about 50 iterations for Yale and about 100 iterations for COIL . However, for a number of recognition tasks, where the class distributions may be highly irregular, or strongly multi-modal, it is possible that such regular shapes may not occur after iteration. It appears that the nature of the Yale and COIL correlated image sets is particularly amenable to the CSVR representation.

6.4 Conclusion

The CSVR provides a stable technique for generating simple class shapes from image databases which provide good generalization over variations in lighting and pose. As expected, the use of this type of an approach for a mixture of gaussian dataset provides an overly simplistic class shape which is unable to capture the important structure in the data. What is clear from the results is that image datasets such as face or general object databases, due to a relatively high amount of correlation between images, can be effectively modeled by simple class shapes. It is important to note, however, that although the overall number of support vectors has been dramatically reduced, the classes still cannot be described by a simple Gaussian distribution. The support vectors still define a class boundary which is more complex than the quadratic decision function that would result from the use of Bayesian techniques with a Gaussian class-conditional density function.

The CSVR algorithm is a step in the direction of deriving optimal features for use with a support vector classifier. The technique unifies feature extraction and classification into one process. As a result, for any given kernel function (and associated kernel parameters), features will be extracted which are dependent on the classifier. Linking feature extraction and classification thus offers interesting research possibilities in optimizing kernel parameters as well as features for any given data set.

Chapter 7

Conclusions and Future Work

7.1 ICA for Feature Extraction in Image Recognition

In this thesis, ICA was used to extract features from training image datasets for the purpose of recognizing images similar to those in the training data. The sparseness of activation patterns that have been observed in the lowest levels of the human visual system provide a direct motivation for the use of statistical independence in guiding feature extraction in image recognition. This biological evidence has been used to support the notion that the important features in images for the purposes of recognition are precisely those that are extracted from enforcing a sparse coding strategy.

In order that ICA be used effectively for the application of pattern recognition, it was shown that care had to be taken in selecting the extracted features. Specifically, due the implicit orthogonal relationship with PCA features, ICA in theory will exhibit identical recognition results when the distance between features was measured in a Euclidean sense. However, due to the approximations employed in ICA algorithms, there are small variations in recognition results which depend on the choice of algorithms. The FastICA algorithm, with its explicit orthogonalization of the mixing

and demixing matrices, results in identical performance to PCA. The gradient descent algorithm of Bell and Sejnowski, which does not constrain the search space of the mixing and demixing matrices to that of orthogonal matrices can offer a slightly different recognition result from PCA. To the extent that this effect is algorithm dependent and is based on an approximation, a comparison between the recognition performance of PCA and ICA with a Euclidean distance measure is ill-advised. To avoid this issue, selection of ICA basis images, when Euclidean distance measures were used, was performed by a combinatorial selection of features.

A variety of experiments were performed to evaluate the comparative performance of ICA at extracting features from images. An experiment was conducted wherein a number of subspaces were tested for the application of measuring position in two dimensions. The results indicated that the ICA and KPCA offered no advantage over the relatively simple technique of constructing a PCA subspace. PCA's success seemed to arise from the high degree of correlation in the database images from which the basis images were derived. For this application, the kurtosis of the PCA features was much higher than with any other subspace. This indicated that PCA was inherently well suited to represent this particular dataset. For the application of position or pose measurement in one dimension, ICA was shown to offer an advantage when occlusion occurred in images for test positions or poses to be measured. It was hypothesized that the edge type basis images which result the use of ICA provide feature localization which reduces the effect of local occlusions which do not obscure extracted features. Combinations of the ICA basis images showed that approximately 20 % of the combinations offered more accurate position measurement than eigenfeatures.

For the case of general object recognition in the presence of changing lighting conditions, it was shown that ICA, due to the localized nature of the basis, is appropriate for representing LoG filtered images. LoG filtering can be thought of as a simple model of the lighting invariance which seems to be applied by the low levels of the human visual system. LoG filtering provides a methodology for dealing

with varying illumination for object surfaces which cannot be modeled by a Lambertian reflectance model. For databases where the images are not highly correlated (as the objects are of a general nature) ICA can provide significant improvement in recognition rates over PCA. Previous work in the literature has shown that for this application, the kurtosis of ICA coefficients is significantly higher than that of PCA, providing an indication that ICA is well suited to general object recognition, particularly when lighting variance occurs.

When small random image patches were taken from a dataset, ICA was used to derive LoG type filters, which were compared to fixed scale LoG filters with support vector classification. ICA was shown to provide filters of the spacial resolution and orientation of the features in the database images. This made these filters particularly useful, since they did not need to be tuned to the particular application. This had an important advantage for classification with a SVM with kernel parameters which was very sensitive to the tuning of the filters. The ICA filters provided an alternative to using fixed LoG or Gabor type filters at multiple scales and orientations to provide lighting invariance over images which have features that exhibit a variety of sizes and rotations.

7.2 Modifying Features with SVM Classification and the Compact Support Vector Representation (CSVR)

The CSVR algorithm was developed to provide a link between the selection of features from a support vector machine and the extraction of features from the raw data. When training features labeled as support vectors are modified to be inliers, simple class shapes result from image databases which provide good generalization over lighting and pose variations. Since image datasets such as face or general object databases are somewhat correlated between images, this data can be effectively modeled by simple

class shapes. The class boundaries which resulted from the CSVR algorithm were greatly simplified (a rapid decrease in the number of support vectors occurred during iteration).

The experiments conducted with the CSVM algorithm summarized a number of the key points illustrated by this thesis. It was again illustrated that principal and independent components have an implicit orthogonal relationship and Euclidean distance measures for classification provide no statistically significant difference between these two data representations. Support vector classifiers combined with features derived with independent component analysis consistently produced larger margins and smaller numbers of support vectors than both the principal component and raw data representations. There exists no definitive work on the advantage of ICA features in their ability to generalize over image variants. It can be hypothesized that because independent component bases comprise the image edges, these are significant features which aid in the recognition of objects despite changes in illumination or pose.

Pose variance or illumination change in images creates large changes in object appearance. Improved generalization of the classifier over variation in object pose or illumination was accomplished by using the CSVR algorithm to modify the PCA and ICA representation. The algorithm worked by making the classes more compact by decreasing the class volume. This in turn created a corresponding decrease in the number of support vectors and increased margin. Each iteration produced a rapid decrease in the number of support vectors and thus was self-regulating. As the number of support vectors decreased, the number of features which were modified decreased and the volume change decreased. The algorithm can be initialized with an identity basis although faster convergence can be obtained by initializing with a PCA or ICA derived basis. Regardless of which basis is used for initialization, the algorithm was observed to converge to a similar performance result. The basis images which resulted after the algorithm converged were very similar across the basis set. This seemed to imply that redundancy was exploited to allow for the improved generalization. This redundant basis was very different from an orthogonal basis such as PCA, illustrating

the large differences in representations between the applications of pattern recognition and image coding.

It is important to note that the algorithm is most effective for the case of image databases. For this type of data, reduction in class volume in the direction of the class means is a logical choice to simplify the class shape. However, for an arbitrary class distribution this strategy might not be appropriate. What was more significant for the purpose of this thesis was that image datasets *are* suitable for the application of an algorithm of this type. This gave some insight into the nature of variations in illumination and pose in image databases. Also significant was the fact that even a simple connection between feature extraction and classification can provide significant improvement in overall recognition performance. Some simple extensions for this algorithm will be proposed in Section 7.4.

7.3 Multidimensional Features

A natural extension to basic ICA for feature extraction is to relax the assumption of independence. The idea behind multidimensional ICA is to assume that the independent components can be divided into k-tuples, so that the coefficients in a given k-tuple may be dependent, but other k-tuples are not permitted to be dependent. If J denotes the number of feature subspaces which are independent and $S_j, j = 1, \ldots, J$ denotes a set of indices of the subspace coefficients $\mathbf{y}$ which belong to the subspace of index j, the probability density of the jth k-tuple of $\mathbf{y}_i$ is $p_j(\mathbf{w}_{iT}\mathbf{x}(n), i \in S_j)$. For T observed data points $\mathbf{x}(t), t = 1, \ldots, T$, the likelihood L of the data in the multidimensional model is [32]:

$$L(\mathbf{x}(n), n = 1, \ldots, T; \mathbf{b}_i, i = 1, \ldots, m) = \prod_{T=1}^{T} \left[|\det \mathbf{W}| \prod_{j=1}^{J} p_j(\mathbf{w}_{iT}\mathbf{x}(n), i \in S_j) \right] \tag{7.1}$$

In order to use this model, an estimate of the probability density function for each of the J k-tuples is necessary. While this is potentially a very difficult problem, it has been accomplished in the past by kernel density estimation techniques. A kernel estimate of the probability density at the point x_0 has the form (in the one dimensional case) for a random sample $x_1, \ldots, x_N$:

$$p(x_0) = \frac{\#x_i \in \mathcal{N}(x_0)}{N\lambda} \tag{7.2}$$

with $\mathcal{N}$ a small neighborhood around x_0 of width λ and $\#x_i$ the number of samples in the neighborhood. This estimate can be smoothed through the use of a kernel function:

$$p(x_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K(x_0, x_i) \tag{7.3}$$

For the multidimensional case the probability density function can be written as:

$$p(\mathbf{x}) = \frac{1}{N\lambda} \sum_{i=1}^{N} K(\mathbf{x}, \mathbf{x}_i) \tag{7.4}$$

Here we now have a link between the extracted features and the classification of them with a SVM according to the kernel function. This is but one simple example of how the process of feature extraction can be coupled with SVM classification. An important characteristic of this idea is that no a-priori information about class density functions is employed in either classification or feature extraction, yielding a fully automated process.

When there is a dependence between not only the k-tuples but among neighboring components a topographic model of ICA can be employed [32]. A neighborhood function defines the strength of the connection between units within the k-tuples and between them. Again, this can be linked directly to the classification by defining the neighborhood function as a kernel function. This technique is similar to the self-organizing map [85].

7.4 Optimal CSVM Features

An important extension of the CSVR algorithm is to improve on the iterative nature of the update of the basis. One simple way to approach this is to add an additional term in the SVM optimization problem to penalize support vectors that are a large distance from the class means:

$$L_d(\alpha, \mathbf{S}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{S}^T \mathbf{x}_i, \mathbf{S}^T \mathbf{x}_j) - \sum_{i=1}^{l} \alpha_i |\mathbf{S}^T \mathbf{x}_i - \mathbf{y}_{mean}|^2 \quad (7.5)$$

Unfortunately, this is no longer a simple quadratic programming problem, as we are now optimizing with respect to the basis $\mathbf{S}$ as well as the support vectors and as such there is likely no unique maximum. More examination needs to be undertaken to understand the optimization landscape for this problem.

Returning to the original iterative update, a more complex scheme could be used to determine the direction of movement for the support vectors, to make the algorithm more general. The fundamental problem is that the class mean may not be the ideal center of the distribution of the class. In fact, the mean may not reside inside the cluster of points which defines the class. The mixture of Gaussian example in Chapter 6 illustrates an example of this type. To resolve this problem, the mean center $\mathbf{m}_c$ can be used in place of the mean of the class. For the mean center:

$$\sum_{\mathbf{y} \in C} d(\mathbf{m}_c, \mathbf{y}) \leq \sum_{\mathbf{y} \in C} d(\mathbf{z}, \mathbf{y}), \forall \mathbf{z} \in C \quad (7.6)$$

where d is the distance between two points. This relation describes the idea of summing (for each point) the distance to every other point. The point with the lowest distance sum is the mean center of C.

Another possibly better location for the class center might be to take the region of the class with the highest probability. For this, a kernel density estimate can be made for the points in the class, as described in the previous section. This has the

advantage of being robust to outliers, as they have no effect on the point of the class with the highest probability. However, an estimate of the class density function can be difficult to find in multiple dimensions. The density estimation is not as problematic for this case, since only a maximum of the density function is needed.

Another interesting approach would be to assume that the class itself contains clusters of data and to move the support vectors in a direction towards the closest cluster mean or mean center. There are a large number of clustering techniques, but herein a simple example of nearest neighbor hierarchical clustering will be briefly described. In this method, the distance between two clusters A and B is defined by the minimum distance between a point in A and a point in B. At each step in the method, a distance is found for every pair of clusters and the two clusters with the smallest distance are merged. The process is then repeated with one less cluster. Initially, each cluster consists of a single data point. The method continues until there are just two clusters. The result of this process is a tree which groups the data points into groups. At any point in the tree, a number of clusters can be selected, and the tree can be traced for each cluster to find the component data points. To fully automate the process, an estimate of the number of clusters c must be made. One simple method that can be used, based on the between and within cluster scatter matrices is to define an index f:

$$f = \frac{tr(\mathbf{S}_b/(c-1)}{tr(\mathbf{S}_w)/(n-c)} \tag{7.7}$$

with n defined as the number of data points, a within class scatter matrix $\mathbf{S}_w$ and a between class scatter matrix $\mathbf{S}_b$. The value of c is chosen which maximizes f [86]

Bibliography

[1] S. Ullman. *High-Level Vision.* The MIT Press, 1996.

[2] D. Hoffman. *Visual Intelligence.* W. W. Norton & Company Inc., 1998.

[3] E. Kandel, J. Schwartz, and T. Jessell. *Principles of Neural Science.* The McGraw-Hill Companies Inc., 2000.

[4] D. Hubel and T. Wiesel. Brain mechanisms of vision. *Scientific American*, pages 150–162, 1979.

[5] P. Hancock, R. Baddeley, and L. Smith. The principal components of natural images. *Network*, 3:61–72, 1992.

[6] B. Olshausen and D. Field. Wavelet-like receptive fields emerge from a network that learns sparse codes from natural images. *Nature*, 381:607–609, 1996.

[7] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[8] A. Bell and T. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1999.

[9] V. Vapnik. *Statistical Learning Theory.* John Wiley & Sons Inc., 1998.

[10] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

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

[12] P. Phillips. *Support vector machines applied to face recognition.* Advances in Neural Information Processing Systems II. MIT Press, 1999.

[13] Y. Qi, D. Doermann, and D. DeMenthon. Hybrid independent component analysis and support vector machine learning scheme for face detection. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 1481–1484, 2001.

[14] X. Zhang, V. Ramani, Z. Long, Y. Zeng, A. Ganapathiraju, and J. Picone. Scenic beauty estimation using independent component analysis and support vector machines. In *Proceedings of IEEE Southeastcon*, 1999.

[15] S. Chang, B. Kirshnapuram, P. Runkle, L. Carin, S. Der, and N. Nasrabadi. Feature extraction and support-vector classification of FLIR imagery. *IEEE Transactions on Image Processing (submitted).*

[16] B. Scholkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *Proceedings of the 1st International Conference on Knowledge Discovery & Data Mining*, pages 252–257, 1995.

[17] D. Tax and R. Duin. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

[18] P. Juszczak, D. Tax, and R. Duin. Feature scaling in support vector data description. In *Proceedings of the 8th Annual Conference of the Advanced School for Computing and Imaging*, pages 95–102, 2002.

[19] C. Jutten and J. Herault. Blind separation of sources, part 1: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[20] J. Cardoso. Sources separation using higher order moments. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 2109–2112, Glasgow, Scotland, 1989.

[21] J. Cardoso. Super-symmetric decomposition of the fourth-order cumulant tensor, blind identification of more sources than sensors. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 14–17, Toronto, Canada, 1991.

[22] L. Tong, V.C. Soon, R. Liu, and Y. Huang. AMUSE: a new blind identification algorithm. In *Proceedings of the ISCAS*, pages 1784–1786, New Orleans, Louisiana, 1990.

[23] P. Comon. Independent component analysis - a new concept? *Signal Processing*, 36:287–314, 1994.

[24] R. Vigario, J. Sarela, V. Jousmaki, M. Hamalainen, and E. Oja. Independent component approach to the analysis of EEG and MEG recordings. *IEEE Transactions on Biomedical Engineering*, 47(5):589–593, 2000.

[25] A. Barros, R. Vigario, V. Jousmaki, and N. Ohnishi. Extraction of event-related signals from multi-channel bioelectrical measurements. *IEEE Transactions on Biomedical Engineering*, 47(5):583–588, 2000.

[26] Y. Schechner, J. Shamir, and N. Kiryati. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *Proceedings of the International Conference on Computer Vision*, pages 814–819, 1999.

[27] A. Bell and T. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[28] J.F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4:109–111, 1997.

[29] J. Karhunen and J. Joutsensalo. Representation and separation of signals using nonlinear PCA type learning. *Neural Networks*, 7:113–127, 1994.

[30] T. Lee, M. Girolami, A. Bell, and T. Sejnowski. A unifying information-theoretic framework for independent component analysis. *Computers and Mathematics with Applications*, 39:1–21, 2000.

[31] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[32] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley & Sons, Inc., 2001.

[33] E. Oja. *Subspace Methods of Pattern Recognition.* Research Studies Press, Letchworth, Hetfordshire, England, 1983.

[34] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):4–36, 2000.

[35] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):780–788, 2002.

[36] H. Murase and S. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[37] S. Nayar, S. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Transactions on Robotics and Automation*, 12(5):750–758, 1996.

[38] M. Jogan and A. Leonardis. Robust localization using an omnidirectional appearance based subspace model of environment. *Robotics and Autonomous Systems*, 45(1):51–72, 2003.

[39] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898, 2000.

[40] A. Martinez and J. Vitria. Clustering in image space for place recognition and visual annotations for human-robot interaction. *IEEE Transactions on Systems, Man and Cybernetics B*, 31(5):669–682, 2001.

[41] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of the 1991 Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[42] M. Bartlett and T. Sejnowski. Independent components of face images: A representation for face recognition. In *Proceedings of the 4th Annual Joint Symposium on Neural Computation*, Pasadena, California, 1997.

[43] K. Baek, B. Draper, R. Beveridge, and K. She. PCA vs. ICA: A comparison on the feret data set. In *Proceedings of the Joint Conference on Information Science*, pages 824–827, 2002.

[44] B. Draper, K. Baek, M. Bartlett, and R. Beveridge. Recognizing faces with PCA and ICA. *Computer Vision and Image Understanding*, 91:115–137, 2003.

[45] C. Liu and H. Wechsler. Comparative assessment of independent component analysis (ICA) for face recognition. In *Proceedings of the Second Int'l Conf. on Audio and Video-based Biometric Person Authentication*, 1999.

[46] K. Messer, D. de Ridder, and J. Kittler. Adaptive texture representation methods for automatic target recognition. In *Proceedings of the British Machine Vision Conference*, pages 443–452, 1999.

[47] P. Somol, P. Pudil, J. Novovicova, and P. Paclik. Adaptive floating search methods in feature selection. *Pattern Recognition Letters Letters*, 20:1157–1163, 1999.

[48] M. Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel eigenfaces. In *Proceedings of the 2000 IEEE International Conference on Image Processing*, pages 37–40, 2000.

[49] K. Kim, K. Jung, and H. Kim. Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, 9(2):40–42, 2002.

[50] S. Nayar and H. Murase. Dimensionality of illumination manifolds in eigenspace. *CUCS-021-94 Technical Report*, 1994.

[51] G. Brelstaff and A. Blake. Detecting specular reflections using lambertian constraints. In *Proceedings of 2nd International Conference on Computer Vision*, pages 297–302, 1988.

[52] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions. *International Journal of Computer Vision*, 28(3):245–260, 1998.

[53] M. Bartlett. *Face Image Analysis by Unsupervised Learning.* Kluwer Academic Publishers, 2001.

[54] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[55] P. Hoyer and A. Hyvarinen. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, 42(12):2412–2423, 1991.

[56] H. Bischof, H. Wildenauer, and A. Leonardis. Illumination insensitive recognition using eigenspaces. *Computer Vision and Image Understanding*, 95:86–104, 2004.

[57] J. Krumm. Eigenfeatures for planar pose measurement of partially occluded objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 55–60, San Francisco, California, 1996.

[58] K. Ohba and K. Ikeuchi. Detectability, uniqueness and reliability of eigen windows for stable verification of partially occluded objects. *Computer Vision and Image Understanding*, 19(9):1043–1048, 1997.

[59] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition*, pages 453–458, 1996.

[60] J. Cardoso. Multidimensional independent component analysis. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 1941–1944, 1998.

[61] A. Hyvarinen, P. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001.

[62] S. Li, X. Lv, and H. Zhang. View-subspace analysis of multi-view face patterns. In *Proceedings of the IEEE ICCV Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, pages 125–132, 2001.

[63] S. Li, X. Lv, H. Zhang, Q. Fu, and Y. Cheng. Learning topographic representations for multi-view image patterns. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 1329–1332, 2001.

[64] F. Salam and G. Erten. Sensor fusion by principal and independent component decomposition using neural networks. In *Proceedings of the 1999 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 211–215, 1999.

[65] J. Stone, J. Porrill, C. Buchel, and K. Friston. Spatial, temporal and spatiotemporal independent component analysis of fMRI data. In *Proceedings of the 18th Leeds Statistical Research Workshop on Spatial-Temporal Modeling and its Applications*, pages 23–28, 1999.

[66] M. Lewicki and T. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12:337–365, 2000.

[67] G. Burel. Blind separation of sources: A nonlinear neural algorithm. *Neural Networks*, 5(6):937–947, 1992.

[68] B. Koehler T. Lee and R. Orglmeister. *Blind source separation of nonlinear mixing models.* IEEE Press, neural networks for signal processing vii edition, 1997.

[69] P. Pajunen, A. Hyvarinen, and J. Karhunen. Nonlinear blind source separation by self-organizing maps. In *Proceedings of the International Conference on Neural Information Processing*, pages 1207–1210, 1996.

[70] D. Pham. Blind separation of instantaneous mixture of sources via an independent component analysis. *IEEE Transactions on Signal Processing*, 44(11):2768–2779, 1996.

[71] Z. Luo and J. Lu. On blind source separation using mutual information criterion. *Mathematical Programming*, 97:587–603, 2003.

[72] S. Haykin, editor. *Unsupervised Adapive Filtering - Blind Source Separation*, volume 1. John Wiley & Sons, Inc., 2000.

[73] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

[74] N. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *Proceedings of the European Conference on Computer Vision*, pages 45–58, 1996.

[75] A. Webb. *Statistical Pattern Recognition.* John Wiley & Sons, Ltd., 2 edition, 2002.

[76] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley Interscience, 2 edition, 2000.

[77] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer-Verlag, 2001.

[78] D. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987.

[79] H. Barlow. Unsupervised learning. *Neural Computation*, 1:295–311, 1989.

[80] H. Barlow. Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12:241–253, 2001.

[81] J. Hurri, A. Hyvarinen, and E. Oja. Wavelets and natural image statistics. In *Proceedings of the Scandanavian Conference on Image Analysis*, 1997.

[82] P. Belhumeur, D. Kriegman, and A. Yuille. The bas-relief ambiguity. *International Journal of Computer Vision*, 35(1):33–44, 1999.

[83] C Hsu and C Lin. A comparison on methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.

[84] S. Nene, S. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). *CUCS-006-96 Technical Report*, 1996.

[85] T. Kohonen. *Self-Organizing Maps*, volume 1. Springer, 2000.

[86] A. Rencher. *Methods of Multivariate Analysis*, volume 1. John Wiley & Sons, Inc., 2002.

[87] S. Haykin and B. Kosko, editors. *Intelligent Signal Processing.* IEEE Press, 2001.

[88] T. Lee, M. Lewicki, and T. Sejnowski. *Unsupervised classification with non-Gaussian mixture models using ICA.* Advances in Neural Information Processing Systems II. MIT Press, 1999.

[89] D. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):831–837, 1996.

[90] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[91] B. Scholkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[92] A. Labbi, H. Bosch, C. Pellegrini, and W. Gerstner. Sparse-distributed codes for image categorization. In *Proceedings of the International Conference on Neural Information Processing*, pages 336–341, 1999.