ROBUST POSITION MEASUREMENT IN VISUAL EIGENSPACE

By

PHILIP QUICK, B. Eng. (McMaster University), M. Eng. (McMaster University)

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

McMaster University

Your file Votre référence
ISBN: 0-612-86480-4
Our file Notre référence
ISBN: 0-612-86480-4

ROBUST POSITION MEASUREMENT IN VISUAL EIGENSPACE

DOCTOR OF PHILOSOPHY (2003) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: Robust Position Measurement In Visual Eigenspace

AUTHOR: Philip Quick

B. Eng. (McMaster University), M. Eng. (McMaster University)

SUPERVISOR: Dr. David Capson

NUMBER OF PAGES: xvii, 207

Abstract

A survey and analysis of visual measurement of camera and object position in visual sub-space (eigenspace) is provided leading to several improvements to existing methods, as well as new approaches. Specifically, novel techniques were developed to allow robust measurements in the presence of occlusions and other dynamic scene changes which is known to be a significant challenge in pose measurement methods for important applications such as visual servoing, autonomous robotics in manufacturing and tele-robotics including aerospace, medical operations and others.

Local image information is shown to retain positional information in unoccluded regions that can then be used to determine position in the presence of significant dynamic occlusion. Local information is also shown to be more prone to ambiguity errors due to a lack of salient positional information. A subsectioning and recombination strategy is developed that features the advantages of local eigenspace independence for robustness to occlusion while maintaining the inherent resistance to ambiguity available from global eigenspace analysis. This is achieved by computing modified global projections, while excluding information from occluded sections. A new method for occlusion detection using an eigenspace reconstruction error measure is also developed and evaluated.

A wide variety of experimental measurements are provided to demonstrate the performance of the new methods using an accurate XYZ platform and CCD cameras with metallic, machined parts. Experimental measurements are also performed to

demonstrate improvements for eigenspace position accuracy through the use of multiple cameras. Several techniques are employed to combine and fuse multiple images from decoupled cameras whereby cameras are used for determining position in different directions to to improve accuracy. Subsequently, multiple cameras are applied to achieving three dimensional translational position measurements in visual subspace.

Acknowledgements

I would like to thank my supervisor Dr. David Capson for his patience and unwavering support. I would also like to thank the members of my supervisory committee, Dr. Max Wong and Dr. David Jones for their time and effort on my behalf. Finally I would like to thank my family and my fellow graduate students in the Machine Vision and Image Analysis Laboratory over the years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Determining the position of a camera relative to a scene or determining the position of an object relative to a fixed camera are problems in computer vision that have been difficult to solve. Successfully accomplishing these tasks has many interesting applications. For example, in manufacturing, the ability to determine the position of a part held by a robot or the position of the end-effector relative to a part can be used to increase manufacturing flexibility. Similarly, mobile robot navigation can be achieved through knowing the position of a camera relative to its environment. Virtual reality applications could be enhanced by determining the orientation of a head mounted camera to provide feedback for the appropriate view to transmit. Tele-robotics applications could also benefit by determining the position of robot mounted cameras relative to their environments locally and transmitting this information quickly.

Eigenspace methods (also known as appearance based methods) have been applied successfully to solve positioning problems with a high degree of accuracy. Eigenspace methods have the advantage that they automatically derive visual feature vectors used to determine position based on training image sets for the given application. These feature vectors allow images of objects over a range of motion or images captured over a range of camera motion to be represented in much reduced form. The reduced

representation for images allows position to be determined via a search of a stored set of the reduced image representations over the range of motion. The ability to automatically derive these feature vectors contrasts with many current robot vision techniques that require significant engineering design to develop heuristic solutions for specific applications.

Eigenspace methods originated as solutions to the problem of face recognition and consequently most of the work on eigenspace methods has focused in this area (and to a lesser extent to general object recognition). However research into the use of eigenspace positioning has been scant beyond some important initial advances. Further research in eigenspace positioning is required to facilitate the transfer of these techniques to industry. This thesis investigates and improves upon the basic eigenspace positioning techniques in several aspects.

First, as a prelude, the basic performance of eigenspace positioning in terms of accuracy is investigated. The effects of varying the available eigenspace design parameters are demonstrated experimentally. These results are explained in the context of how they relate to the amount the image features change position within the image over the movement range of the camera and how this position change limits the possible accuracy. Additionally a technique to improve accuracy by filtering the feature vectors in certain situations is described.

The main focus of the thesis is developing techniques to determine position of a camera or object with eigenspace methods when occlusion is present. Occlusion in computer vision is generally considered to consist of one or more objects intruding into the field of vision of the camera altering the resultant image from what would normally be expected. Thus occlusion includes a wide range of possible effects on an image. In terms of applications, potential occlusions include people moving in front of a mobile robot or a partially obscured industrial part that an end-effector equipped with a camera is supposed to maneuver in relation to.

Occlusion presents a vexing problem for eigenspace methods. The eigenspace features reduce images to a global low dimensional format, but in the process all the local image information is lost. Occlusions anywhere in an image alter the low dimensional results such that they are incorrect for determining position accurately. For eigenspace methods, occlusion includes changes to the background as well. This thesis introduces the preservation of local image information in eigenspace positioning to isolate occlusions. Rather than storing only the eigenspace information for the entire image, information is stored corresponding to individual image sections. A dynamic technique termed subsectioning and recombination was developed to combine the information from unoccluded sections to achieve performance similar to the basic unoccluded eigenspace technique even in the presence of significant occlusion. An eigenspace reconstruction measure is demonstrated that is proficient at differentiating between occluded and unoccluded image sections and used to determine which sections to include for subsectioning and recombination. The technique is shown to improve positional accuracy substantially for an occluded metal part positioning experiment and approaches the unoccluded performance level.

The thesis includes an investigation of the use of multiple cameras to improve the accuracy achievable using eigenspace methods for positioning. Certain directions of camera movement are subject to lower positional accuracy due to lower rates of pixel change. Properly arranged multiple cameras can provide additional information for these directions. Several techniques to combine information from multiple cameras with eigenspace methods are proposed and experimentally evaluated with regard to their improved performance. These same multiple camera techniques can be applied to higher dimensional position problems which are of obvious use for manufacturing tasks such as mating parts properly which are not constrained to 2D translational movement. The accuracy of 3D translational positioning for a camera relative to a metal part was improved using the multiple camera techniques. Additionally, an

approach for reducing the number of training images required with multiple cameras through the exploitation of different rates of pixel change was demonstrated successfully. This is an important result because the main implementation difficulty for higher dimensional eigenspace positioning is the vast increase in the number of training images required for each additional dimension.

1.1 Organization of the Thesis

The thesis is organized according to the following summary which lists the main contents and contributions of this work:

Remainder of Chapter One

- Literature survey of eigenspace methods in computer vision.

Chapter Two

- Introduction of general camera/object positioning problems.
- Discussion of Principal Components Analysis for image compression in camera/object positioning.
- Derivation of the eigenvector features from the training images.
- Explanation of the use of eigenspace techniques for camera/object positioning.

Chapter Three

- Discussion of the limiting factors for eigenspace positioning accuracy, including the amount of pixel change versus camera movement.
- Introduction and experimental demonstration of eigenvector filtering as a means to improve accuracy with lowered amounts of training images.

- Experimental analysis of the adjustable parameters for eigenspace positioning and their effects on accuracy.

Chapter Four

- Discussion and experimental demonstration of the effects of occlusion on positional accuracy.
- The concept of local versus global information in regards to eigenspace positioning and preserving unoccluded information.
- Introduction of and experiments with separate eigenspaces, a simple method for incorporating local information for occlusion positioning robustness.
- Discussion and experimental demonstration of ambiguity induced error with the separate eigenspaces technique.
- Introduction of the Subsectioning and Recombination technique for incorporating both global and local information for occlusion robustness and resistance to ambiguity.
- Experimental demonstration of the effectiveness of the Subsectioning and Recombination technique in a highly occluded part positioning problem.
- Experimental demonstration of subsectioning and recombination with severe occlusion.

Chapter Five

- Introduction and experimental demonstration of eigenspace reconstruction as an effective means of detecting occluded sections of an image (a necessary requirement for implementation of Subsectioning and Recombination).

- Use of manifold distance as an alternate measure for determining occluded sections.

Chapter Six

- Explanation of poor accuracy for determining movement along the optical axis of a camera for eigenspace positioning as a function of the amount of pixel change.
- Introduction and rationale for the use of multiple offset cameras to improve accuracy.
- Experimental demonstration and comparison of decoupled cameras, fused images and fused projections techniques for combining multiple camera information to successfully increase positional accuracy.
- Demonstration of multiple cameras techniques as a means for successfully performing 3D translational positioning as opposed to single camera techniques.
- Introduction and rational for the use of the pseudo-invariance property to overcome the issue of training image size, the main implementation difficulty in higher dimensional positioning problems.
- Experimental demonstration of using pseudo-invariance to reduce the number of training images required by 2/3 for 3D translational positioning while maintaining similar levels of accuracy.

Chapter Seven

- Conclusions.
- Future Directions.

1.2 Literature Survey

1.2.1 Early Work on Eigenspaces

Eigenspace techniques in computer vision are a relatively recent area of active research with the earliest work little more than a decade old. Compared with other computer vision techniques, eigenspace techniques use automatically derived global features as compared to previous techniques which use heuristically chosen geometric features such as lines and corners.

Eigenspace methods were first used to deal with the problem of face recognition and their major applications in computer vision tend to be in that area. Eigenspace techniques for face recognition were a major advance over previous attempts dating to the seventies [19][9] that used individual facial features and perform identification based on the distances between them.

The first work on using eigenspace methods with faces can be traced to the work of Sirovich and Kirby [20]. Rather than trying to perform identification tasks, their work was concerned with finding low dimensional representations of human faces. Since the visual subspace formed by cropped and properly positioned human faces was highly compressible, Principal Components Analysis (PCA) was used to form low dimensional representations of the faces. Principal Components form the basis for eigenspaces, whereby global feature vectors are formed that capture the vast majority of the visual information of the subspace. Using a small number of the appropriate basis vectors allows most of the information from the image to be captured by projecting the images in the visual subspace on the basis vectors, producing a small vector representing the faces. Faces stored in such a manner could be reconstructed by scaling the basis vectors with these projections and adding them together. Note that PCA is the statistically optimal basis for compression in terms of mean square reconstruction error.

Soon after, Turk and Pentland [45] achieved a breakthrough in face recognition by demonstrating that not only could the low dimensional representations of the faces be used for reconstruction, but also for identification. By storing low dimensional representations for a set of individuals, identification could be achieved by finding the low dimensional projection coefficients for an unknown face image and performing a nearest neighbor search with the stored coefficients. This early work illustrated that eigenspace techniques were effective for frontal views of faces with constant illumination.

In the same paper, eigenspace techniques were also first applied to face detection tasks. Deriving an eigenspace from scaled, centered face images will create an eigenspace proficient for reconstructing face images accurately, thus the eigenspace reconstruction could be used as a measure of how well an image represented a face. Reconstructing non-face images with the face specific eigenvectors produces images that differ largely from the original. Thus it can be determined if a region contains a face by using the face eigenvectors to reconstruct the region and measure the difference between the original and the reconstruction.

Moghaddam and Pentland [30] [29] continued to add other advances to the basic eigenspace facial recognition approach. Since eigenspace methods work best where images to be identified fall within the visual subspace formed with the training images, images such as faces viewed at an angle resulted in poorer levels of recognition. Moghaddam and Pentland [41] improved on this by constructing multiple eigenspaces consisting of faces viewed at different angles.

Moghaddam and Pentland [41] also presented an alternate eigenspace technique for face recognition, where the input images were divided into separate sections corresponding to the eyes, nose and mouth regions of the face. Separate eigenspaces were formed for these features and recognition was performed separately for each feature, with results similar to that of the basic eigenspace technique for the entire face.

While face recognition was the most popular use of eigenspace techniques in computer vision, eigenspace methods were soon applied to other applications, among them general object recognition. Object recognition differs from the face recognition problem in that the faces are constrained such that they share similar characteristics and are thus compressible. Object recognition on the other hand has no such constraints and objects to be recognized can be significantly different in appearance. In addition, it is more likely that they will be need to be recognized from a multitude of camera viewing angles. Murase and Nayar [32][33][36] were the first to apply eigenspace techniques to the problem of object recognition. Traditional methods of object recognition concentrated on identifying geometric features such as corners and edges and then recognizing objects based on their relations, with relatively low rates of success. Murase and Nayar performed object recognition by forming a global eigenspace consisting of training images of all the objects to be recognized from various viewpoints and then storing the low dimensional projection coefficients produced by projecting the training images on this eigenspace. Similarly to the face recognition method, a new image of an unknown object was projected on to the eigenspace and the image was identified as being that object whose training image was the nearest neighbor in the eigenspace. As opposed to the face recognition technique of having only one set of projection coefficients per face, Murase and Nayar produced a manifold of projection coefficients of each object in the global eigenspace consisting of the projections of the images of the object rotated in front of the camera to allow pose recognition as well. Murase and Nayar improved the pose recognition of objects once they were recognized by constructing separate eigenspaces for each object. These local eigenspaces allow better pose recognition since their eigenvector features are tuned specifically to the images from one object rotated in front of the camera. They also identified for pose recognition that the projection coefficients from the training images of objects rotated in front of the camera can be interpolated to provide additional positions to match

intermediate to those corresponding to the training images, increasing the possible accuracy of the pose recognition.

Other later research into eigenspaces and object recognition includes that of Campbell and Flynn [6] who applied Murase and Nayar's object recognition techniques to range images. Also Borotschnig [5] et al. used basic eigenspace object recognition and combined it with an active vision system that chose different camera views for ambiguous images.

1.2.2 Previous Work on Camera and Object Positioning

Beyond object recognition, another major application of eigenspace techniques in computer vision has been to determine either the position of a camera within a fixed environment or alternatively, to recognize the position of an object relative to a camera. This area of research has seen less investigation then other eigenspace topics in computer vision.

Again, Murase, Nayar and Nene [35] [37] performed the pioneering work in this area. Their focus was the use of eigenspaces for visual servoing, specifically for determining the position of a camera mounted on a robot end-effector over a printed circuit for the purpose of performing accurate chip insertion. To perform this task, a set of training images was acquired with the end-effector positioned over a range of motion over the circuit board with the correct insertion point as the center. A set of low dimensional projection coefficients was stored and consequently the current position of the end-effector could be obtained by performing a nearest neighbor search. Subsequently the reported position of the camera could be used to move the end-effector to the correct insertion point.

Beyond visual servoing, such techniques were also applied to visual navigation tasks, namely determining the position of a mobile robot within a room or series of rooms. A set of training images was obtained for a robot positioned throughout

the room and position is determined for new positions using Murase and Nayar's technique. Jogan and Leonardis [16][17] accomplished this task using omnidirectional images as the input. This allowed position within a room to be determined despite the orientation of the camera.

Winters et al. [46] implemented a similar eigenspace based method for mobile robot navigation, also using omnidirectional images. To deal with changing illumination, edges images were used with an eigenspace approximation to the Hausdorff Fraction [28]. They did not implement any form of occlusion correction, relying on the fact that with panoramic images, occlusions were only a small percentage of the image, barely effecting the recognition.

Martinez and Vitria [27] also used appearance based methods for robot navigation. PCA and Fischer Discriminant Analysis (FDA) as well as two mixture models produced using Expectation Maximization were used to learn the environment of a research facility and allow the robot to navigate between different locations within the facility.

Jagersand [12] uses eigenspace techniques applied to training images of the motion of robot arms to synthesize simulated images for the purposes of animation of the robot arms. Compressed eigenspace representations of the images are associated with the joint states of the robots. Since eigenspace representation can be used for image reconstruction, images can by synthesized for arbitrary joint positions of the robot. This association of eigenspace representation with the joint states is later used in tele-robotics [14][13], whereby the operator's next movement of the robot can be used to generate predictive images of the robot without the latency of transmitting the actual images over the link.

1.2.3 Previous Techniques for Providing Robustness to Occlusion

The amount of research dealing with eigenspaces and occlusion has not been extensive to this point. Eigenspace methods are especially susceptible to errors with occlusions since the contents of the entire image are reduced to a low dimensional representation. The components of this low dimensional representation are global in nature and thus any occlusion anywhere within the image will alter their values from that associated with the original image.

In face recognition, Martinez [25][26] divides face images into separate sections, each with their own eigenspace and then performs recognition separately for each facial region. Rather than just calculating a nearest neighbor match for each facial region, probabilities are calculated for region and combined. Recognition is based upon adding the local probabilities, thus recognition can be successful even in the presence of occlusion.

For the problem of object recognition, specifically for "bin-picking" operations, Krumm and Ohba et. al. [21] [39] simultaneously derived a method for identifying objects in the presence of occlusion with a modified eigenspace technique. Rather than using global eigenspaces to recognize objects, small "eigenwindows" consisting of approximately 15 by 15 sections of the image were used. For each object, salient features were sought and eigenspaces formed from these features. Objects could be recognized by finding the presence of these features from within a new image. Occlusions can be handled since not all of the features have to be found to identify and find the position of the object.

Techniques for estimating the eigenspace projection coefficients robustly despite occlusion have been proposed by Black and Jepson [4] for the use of eigenspace techniques in tracking applications. Jepson et. al. [15] further advanced this work to use

an EM algorithm to adapt the eigenspace for changing appearance. Leonardis and Bischoff [22][1][2][23] have proposed a more advanced technique for robust estimation of eigenspace projections despite occlusion using subsets of image points rather than the entire image. Using an iterative process governed by the Minimum Description Length (MDL) principle, the final subset of image points consists of unoccluded points. With this technique, both object recognition and pose estimation tasks could be performed accurately despite occlusion. Jogan and Leonardis [16] also used this technique for mobile robot navigation in the presence of occlusion; however position error increased with high amounts of occlusion ($> 50\%$) as the proportion of occluded image points included in the final subset used to estimate the projection coefficients increased.

Huttenlocher et. al [11] provided an interesting eigenspace technique for object recognition for dealing with occlusions by using binary edge maps rather than straight intensity images. Instead of the sum-of-squared differences equations of normal eigenspaces, the Hausdorff distance [28] for comparing binary images was used. This allowed objects to be recognized even with partial occlusion.

1.2.4 Related Eigenspace Research

Several other areas of eigenspace related research are also noteworthy. One of interest is illumination. Eigenspace methods perform poorly when illumination is changed from that under which training occurred. Murase and Nayar [37] suggested that normalizing the images can help with illumination, but it is generally of limited effect. They also suggested the use of multiple sets of training images under varying lighting conditions for better performance with varying illumination [34]. However technique is limited since true illumination invariance requires that the number of training images be large.

Ohba et al. [39] provided a technique for eigenspace recognition to deal successfully with illumination. In their technique, rather than using raw color intensity images, they divided the image into lumina and chrominance components. For illumination invariant recognition, they used only the chrominance component, which is resistant to change under varying illumination. It was relatively successful although monochromatic objects cause difficulties.

Bischoff and Leonardis et al. [3] used an approach similar to their previous methods for dealing with occlusion to provide illumination invariance. A set of gradient filter banks was incorporated into the eigenspace basis. This allowed eigenspace coefficients invariance to illumination. These coefficients were then robustly recovered using a competitive algorithm to select illumination invariant pixels.

Nene and Nayar [38] have also developed a method that significantly cuts down on the search time. Using a novel data structure, they ordered all the stored points for the manifold based on their value of the first projection coefficient. Thus a limited range of potential matches could be obtained by searching the first set of projection coefficients around the first projection coefficient. These matched coefficients were linked to their second coefficient values and so on, allowing a significant amount of calculation to be avoided.

1.3 Scientific Summary

The following is a summary of the scientific contributions of the thesis. These contributions to the enhancement of eigenspace positioning can be grouped into three areas: an analysis of the sources of error in eigenspace positioning, the use of local information for providing robustness to occlusion, and the use of multiple cameras for improving positional accuracy.

The first area, an analysis of the sources of error in eigenspace positioning consists of several aspects. A relationship between accuracy and the formation of the eigenspace feature vectors from the training images is demonstrated. A smoothing technique for these feature vectors is proposed and demonstrated to be effective for improving accuracy for situations where the number of training images across the movement range of the camera is limited. An upper limit on the accuracy possible for eigenspace positioning methods based on the movement of image features within the training images across the movement range of the camera is proposed. This idea is used to explain the experimental behavior of accuracy with regards to the number of training images used to provide the set of low dimensional eigenspace representation.

In the second area, the use of local image information within the eigenspace paradigm is advanced as a solution for providing robustness to occlusion. A naive application of this idea is shown to suffer from errors brought about by ambiguity between sections of images. Subsectioning and recombination, a method combining local robustness to occlusion with resistance to ambiguity error is demonstrated to be successful for greatly reducing occlusion error. Eigenspace reconstruction error is successfully demonstrated as a method for deciding which section of an image are occluded for the above mentioned occlusion robustness techniques.

In the third area, multiple cameras are shown to be useful for obtaining additional positional information in certain directions of camera movement. Several methods for combining this information within the eigenspace rubric are proposed; a simple combination of images into a set of super-images is judges to be the best. These multiple camera techniques are shown to be effective for performing 3D translational positioning. An approach for limiting the number of training images for multiple cameras is proposed and demonstrated.

Chapter 2

General Eigenspace Positioning

In this chapter, the general idea of camera and object positioning problems are defined. This is followed by a discussion of the theoretical basis of eigenspace methods, namely Principal Components Analysis (PCA). This leads to the discussion of why eigenspace methods can be applied to positioning problems and a general description of eigenspace based positioning. These basic techniques are then used as the basis for exploration in the remainder of the thesis.

2.1 General Positioning Problems

The basic task of camera positioning problems is to define an *a priori* range of motion for the camera within a known environment and to then be able to determine the position of the camera within this range of motion via the image obtained by the camera. The range of motion of the camera may be translational, rotational or a combination of both. A simple example is the use of image based positioning for determining the position of a mobile robot within a room. With a camera mounted on a mobile robot, the range of camera motion within the room could be described as a two dimensional translational range as the camera moves with the robot. Similarly, if

the camera is fixed on the robot, the rotational direction of the robot (and hence the camera) could be added to the range of motion of the camera and also a parameter to be determined along with the translational position of the camera.

Alternatively, a different but related set of problems may be addressed by fixing the camera and moving an object or target within the field of view. This would be of obvious use in visual servoing problems; i.e. the determination of the position of a part attached to a robot gripper over a predefined range of translational and/or rotational movement.

2.2 Principal Components Analysis for Visual Subspace Compression

For a defined range of camera motion, the images (assuming that the image rows are concatenated to form image vectors) provided by a camera in this range will determine a visual subspace (also known as an eigenspace).

A simple but flawed method for solving the camera positioning problem for a preset range would be the use of template matching. A large set of images across the movement range of the camera could be collected. The position of a camera could potentially be determined by comparing the new image with the stored set of images via template matching and choosing the position of the image with the closest match.

This method will give accurate positional information with extreme impracticality. The template matching is well known to be computationally inefficient and would need to be performed with every image in the set. To have a high level of accuracy the set would also have to be large. A solution to this problem while still using the basic method would be the judicious use of image compression.

Indeed, the visual subspace consisting of all the possible images within the defined

range of camera motion is highly compressible. Compared to the total possible space for an image for a set number of pixels, in general the images from the visual subspace will occupy a tiny fraction of the total possible images.

The images for nearby camera positions will in general be highly correlated, thus the images throughout the visual subspace are highly compressible. A question is what is the best method for compressing an *a priori* known visual subspace. Optimal compression may be achieved using the Principal Components of the visual subspace.

2.2.1 Principal Components for High Dimensional Pattern Vectors

Principal Components Analysis [18] (also known as the Karhunen-Loeve Transform [24]) is a statistically based orthogonal expansion primarily used to vastly reduce the number of variables required to represent high dimensional pattern vectors (such as images). This dimensionality reduction simplifies pattern recognition problems and the reduced dimensionality is the critical feature in the use of Principal Components Analysis for camera position determination problems.

We will assume that $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ is a pattern vector of size n drawn randomly from an underlying but unknown probability distribution (we assume that $\mathbf{x}$ has a mean equal to zero). If n is large and the data is highly redundant (such as is often the case for images), it is possible to represent the pattern vectors drawn from the distribution in a reduced form. Ideally, this reduced form should produce an accurate reconstruction $\mathbf{x}'$ with as few expansion terms as possible.

PCA produces vectors $\mathbf{e}_i$, which form an orthonormal basis. Any orthonormal basis comprised of n vectors can be used to completely reconstruct a vector $\mathbf{x}$ with a dimensionality n:

$$\mathbf{x} = \mathbf{x}' = \sum_{i=1}^{n} y_i \mathbf{e}_i \tag{2.1}$$

$$y_i = \mathbf{x}^T \mathbf{e}_i \tag{2.2}$$

From basic linear algebra, the orthonormal basis vector coefficients y_i are the inner products of the original vector $\mathbf{x}$ and the basis vectors $\mathbf{e}_i$. Using n vectors to reconstruct $\mathbf{x}$ is obviously of no benefit in terms of dimension reduction. However, by truncating the number of basis vectors used to represent $\mathbf{x}'$ instead of using the full n, a dimensionality reduction can be achieved. Unfortunately, for an arbitrary orthonormal basis, valuable information will be lost from the missing basis vectors and more than likely $\mathbf{x}'$ and $\mathbf{x}$ will be significantly different. The aim of PCA is to construct optimal basis vectors such that only a few contain most of the information required to reconstruct $\mathbf{x}$.

Intuitively, high dimensional vectors drawn from some underlying probability distribution are often highly redundant and thus it is likely that some sort of compression is possible. If we can acquire a set of training vectors drawn from an underlying probability distribution, we can construct the PCA vectors to minimize the mean square error (MSE) between the original vector drawn from this distribution and the truncated reconstruction using only k PCA vectors:

$$MSE = E[\|\mathbf{x}' - \mathbf{x}\|^2] \tag{2.3}$$

$$\mathbf{x}' = \sum_{i=1}^{k} (x^T \mathbf{e}_i)\mathbf{e}_i \tag{2.4}$$

These optimal representation vectors are calculated by finding the eigenvectors with the largest eigenvalues corresponding to the covariance matrix of the pattern

vector $\mathbf{x}$. The fact that the eigenvectors of the covariance matrix minimize equation 2.3 as opposed to any other orthonormal basis, can be found using the methods of Lagrange multipliers [7]. If $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_m]$ corresponds to m samples of the underlying probability distribution of $\mathbf{x}$, the an estimate of the covariance matrix $\mathbf{C}$ can be formed by:

$$\mathbf{C} = \mathbf{X}\mathbf{X}^T \tag{2.5}$$

Thus the PCA vectors $\mathbf{e}_i$ solve the eigenvector decomposition:

$$\mathbf{C}\mathbf{e}_i = \lambda_i \mathbf{e}_i \tag{2.6}$$

Since the covariance matrix is a symmetric matrix, a variety of solutions exist to find the eigenvectors for optimal reconstruction error including the power method, the symmetric QR algorithm and Jacobi methods [10]. Section 2.3.1 explains an approach for using the singular value decomposition for faster computation of the covariance matrix eigenvectors, specifically with regards to the camera positioning determination problem.

For the camera positioning problems that are of interest to this thesis, the range of motion of the camera will be known and will not change. Thus the statistics of the underlying image pattern vector $\mathbf{x}$ will remain the same, allowing the derivation of specific PCA vectors $\mathbf{e}_i$ for each application. This will also allow the representation of any image $\mathbf{x}$ from the movement range of the camera to be represented in low dimensional form by the projection coefficients $\mathbf{x}^T\mathbf{e}_i$ of the basis vectors, since in equation 2.4 the $\mathbf{e}_i$ vectors are already known. This low dimensional representation is critical for eigenspace positioning as will be seen in the next section. As will be seen in the following chapters, image vectors consisting of 40000 data points can often be represented by twenty eigenvector coefficients for the purposes of camera position determination.

Two more useful properties of PCA beyond the minimization of mean square error are important with regards to the use of PCA for pose estimation problems. The first is that the coefficients $\mathbf{x}^T\mathbf{e}_i$ of the different eigenvectors are mutually uncorrelated [40]:

$$E[(\mathbf{x}^T\mathbf{e}_i)(\mathbf{x}^T\mathbf{e}_j)] = 0, i \neq j \tag{2.7}$$

This is an important property for pattern recognition and ensures features with redundant information are not utilized.

The other important property is that the variances of the projection coefficients $\mathbf{x}^T\mathbf{e}_i$ are maximized:

$$\max E[(\mathbf{x}^T\mathbf{e}_i - E[\mathbf{x}^T\mathbf{e}_i])^2], i = 1, ..., k \tag{2.8}$$

Generally, a feature with larger variance will be less likely to be corrupted by noise when judging between two positions.

The effectiveness of PCA for compression depends on the data underlying the pattern vector to be highly correlated. If each element of the pattern vector is independent of the others, the reconstruction error of $\mathbf{x}'$ using the PCA vectors will relatively large (this is true for any orthonormal basis in this case). This is because the number of PCA vectors required to accurately represent the pattern vector will be the same as the number of elements of the vector and thus using a truncated representation will result in a large reconstruction error. Thus PCA is best used for specialized applications where the pattern vector will be highly correlated. For example, the pattern vector underlying images of scaled human faces will be more restrictive than images of faces at different scales. Thus the PCA reconstruction error

on average for the scaled faces will be less than that of the unscaled faces. For the camera positioning problems considered in this thesis, the resultant pattern vectors are highly correlated thus PCA applied to these images will perform well in terms of both reconstruction error and consequently accuracy as well.

Fortunately, there is a measure of how correlated and hence how well suited PCA is for a given data set in terms of reconstruction error, namely the eigenvalues λ_i corresponding to the eigenvectors $\mathbf{e}_i$. The relative sizes of the eigenvalues corresponding to the PCA vectors are actually equivalent to the variance of the projection coefficients $\mathbf{x}^T\mathbf{e}_i$ of the PCA vectors [7]:

$$\lambda_i = E[(\mathbf{x}^T\mathbf{e}_i - E[\mathbf{x}^T\mathbf{e}_i])^2], i = 1, ..., k \tag{2.9}$$

The fact that the eigenvalues are equivalent to the variance of the projection coefficients is important in particular for eigenspace positioning as will be seen in the next section, however, in addition, examining these values can add insight into the behaviour of PCA for a specific instance. Generally speaking, for PCA with a set number of eigenvectors, a greater information compression will be achieved if the covariance matrix has a small number of large eigenvalues (and the rest negligible) as opposed to the eigenvalues being more uniform in distribution [7]. For the case of image vectors with a few large eigenvalues combined with the rest significantly smaller, a smaller reconstruction error will result with a limited number of eigenvectors compared to the case of more uniform eigenvalues. To demonstrate this, if the PCA reconstruction vector is $\mathbf{x}'$, the reconstruction error vector is defined by:

$$\hat{\mathbf{x}} = \mathbf{x} - \mathbf{x}' \tag{2.10}$$

This value can also be determined by:

$$\hat{\mathbf{x}} = \sum_{i=k+1}^{n} (\mathbf{x}^T \mathbf{e}_i)\mathbf{e}_i \tag{2.11}$$

where k is the number of eigenvectors used for the reconstruction and n is the dimensionality of the vectors. The mean of its norm can be calculated by:

$$E||\hat{\mathbf{x}}||^2 = \sum_{i=k+1}^{n} E(\mathbf{x}^T \mathbf{e}_i)^2 \tag{2.12}$$

which through matrix manipulation can be reformed into:

$$E||\hat{\mathbf{x}}||^2 = \sum_{i=k+1}^{n} \mathbf{e}_i^T E[\mathbf{x}\mathbf{x}^T]\mathbf{e}_i \tag{2.13}$$

where $E[\mathbf{x}\mathbf{x}^T]$ is equivalent to the covariance matrix $\mathbf{C}$:

$$E||\hat{\mathbf{x}}||^2 = \sum_{i=k+1}^{n} \mathbf{e}_i^T \mathbf{C}\mathbf{e}_i \tag{2.14}$$

and subsequently via equation 2.6 and the fact that $\mathbf{e}_i^T \mathbf{e}_j = \delta_{ij}$:

$$E||\hat{\mathbf{x}}||^2 = \sum_{i=k+1}^{n} \lambda_i \tag{2.15}$$

Thus if the remaining eigenvalues beyond the *kth* are small in relation to the first few eigenvalues, the average reconstruction error will also be small. Eigenspace positioning depends on PCA achieving good compression and small average reconstruction errors.

2.3 PCA Applied to Camera Positioning Problems

The ability to provide low dimensional representations of high dimensional spaces via PCA makes it a natural for applying it to camera and object positioning problems.

In positioning problems, the range of camera motion forms a visual subspace. An image $\mathbf{x}$ acquired from within this range of camera motion can be thought of as being drawn from a high dimensional probability distribution, if it is assumed that all camera positions are equally likely. Since images from nearby camera positions are usually highly correlated, the visual subspace corresponding to these images will be highly compressible via PCA.

Thus a high dimensional image[1] $\mathbf{x}$ can be represented via PCA in a low dimensional form $\mathbf{y} = [y_1, y_2, ..., y_k]^T$ (herein referred to as the projection coefficients of the image) via:

$$y_i = \mathbf{e}_i^T \mathbf{x} \tag{2.16}$$

$$\mathbf{y} = \mathbf{E}^T \mathbf{x} \tag{2.17}$$

where $\mathbf{E}$ is composed of a small number (typically 10 to 20) of the eigenvectors $\mathbf{e}_i$ with the largest eigenvalues λ_i corresponding to the covariance matrix of the visual subspace:

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k] \tag{2.18}$$

If we store a set of these projection coefficients $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m]$ corresponding to a set of m training images $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$ spaced equally throughout the range of motion of the camera, we determine a low dimensional manifold corresponding to the movement of the camera in the eigenspace. Thus we have associated camera position with a low dimensional representation.

[1]Note that it is assumed that the original image vector has had the estimated average vector across the eigenspace removed to produce $\mathbf{x}$, thus $E[\mathbf{x}] = 0$

Figure 2.1 shows an example of a visualized eigenspace manifold produced with the first three eigenvector projection coefficients y_1, y_2, and y_3, for a set of images corresponding to a camera moving in a square plane over a metal test object (the example in the next chapter). Adjacent points in the manifold correspond to images obtained from adjacent camera positions. Thus with PCA we can perform a fast comparison operation between images via the projection coefficients as opposed to the previously mentioned template matching approach to position determination.

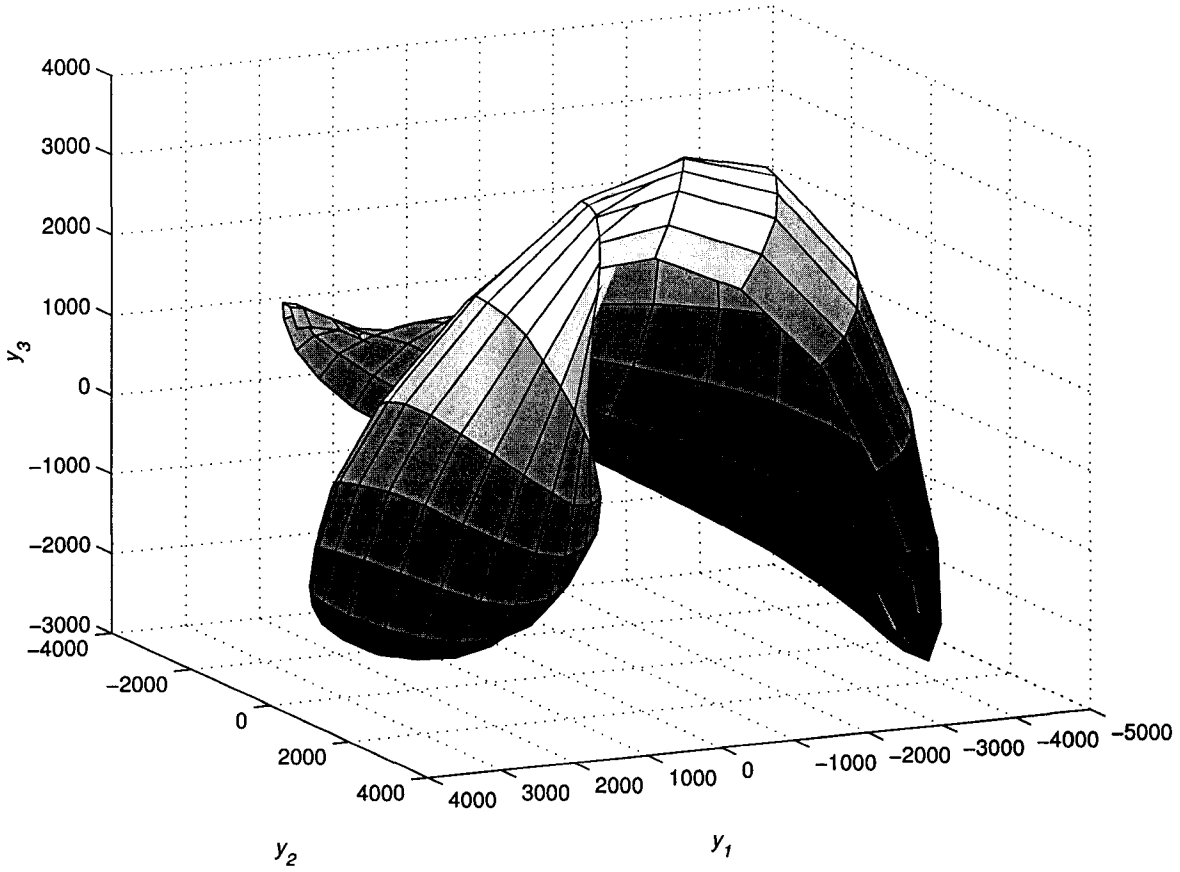

Figure 2.1: Example Eigenspace Manifold

To determine the current position of a new image $\mathbf{x}_{new}$ corresponding to a camera position within the preset range, the stored set of projections $\mathbf{Y}$ corresponding to the training images can be used. Calculating the projection coefficients corresponding to

the new image $\mathbf{y}_{new}$ via eq. 2.8, a nearest neighbor search of the stored projection coefficients can be performed:

$$\text{Nearest Neighbor } nn = arg\min \|\mathbf{y}_{new} - \mathbf{y}_i\|, \forall i = 1, \ldots, m \tag{2.19}$$

The current position of the camera can then be ascertained as that corresponding to the position of the camera where the projection coefficients $\mathbf{y}_{nn}$ were obtained from training image $\mathbf{x}_{nn}$ [37]. Thus comparison with the stored projection coefficients corresponding to the training images allows a computationally fast method of obtaining camera position.

An important consequence of this technique is that the accuracy obtainable is limited by the number of training images, as camera positions intermediate to those corresponding to the training images cannot be identified. One way to increase this accuracy is to use more training images. An alternate technique, (as first proposed by Nayar et. al [37]) is to instead increase the number of camera positions represented by $\mathbf{Y}$ by deriving interpolated projection coefficients corresponding to positions intermediate of the training images. Thus for the example of planar camera position, a denser grid of camera positions can be formulated and then the projection coefficients derived from those corresponding to the actual training images. The interpolation could be linear or alternatively spline based. The interpolation is performed independently for each individual projection coefficient y_g. As an example for forming the interpolated projection coefficients, consider four images: $\mathbf{x}_a$, $\mathbf{x}_b$, $\mathbf{x}_c$, and $\mathbf{x}_d$. These images correspond to the camera at the four corners of a square within a two dimensional plane. The interpolated projection coefficient value for the camera position in the center of the square (assuming linear interpolation) for the gth eigenvector can be calculated by:

$$y_{g_{center}} = (y_{g_a} + y_{g_b} + y_{g_c} + y_{g_d})/4 \tag{2.20}$$

The complete $\mathbf{y}_{center}$ projection coefficients vector can be formed from equation 2.20 for each individual eigenvector. For positions other than the center, the $\mathbf{y}_a$, $\mathbf{y}_b$, $\mathbf{y}_c$, and $\mathbf{y}_d$ vectors need to be weighted appropriately. Spline based procedures would require additional surrounding projection coefficients. Higher dimensions of camera movement can be similarly interpolated.

An important issue for accuracy is that the interpolated projection coefficients will differ slightly from those corresponding to the actual training images thus accuracy cannot be increased indefinitely with more interpolated projection coefficients. We will speak more of the limitations of the method in future sections.

The eigenspace method is interesting compared to previous methods of camera and object position determination that depended on the heuristic use of geometric features such as corners or edges. An advantage of eigenspace methods is that the eigenvector features are generated automatically based on the statistics of the visual subspace without requiring any heuristics decisions on the part of the designer for a particular application. All that is required for an application is to define the range of motion, whether it be the camera moving relative to the object or an object relative to the camera, acquire the necessary training images and position can then be accurately ascertained.

2.3.1 Nearest Neighbor Search Time

One issue with the above mentioned method is the computational requirements of the nearest neighbor search phase, which requires $\mathbf{y}_{new}$ to be compared with all the projection coefficients in $\mathbf{Y}$.

Nene and Nayar [38] have proposed a binary search algorithm to considerably

speed up the search for the nearest neighbor match by preprocessing the set of projection coefficients $\mathbf{Y}$. Starting with the first projection coefficient y_1 for each $\mathbf{y}$, $\mathbf{Y}$ is ordered from first to last with respect to this y_1 rather than by camera position. For a new image, its y_1 value can be matched to the closest y_1 value in the ordered set $\mathbf{Y}$. Only $\mathbf{y}$ vectors from within $\mathbf{Y}$ are considered if their y_1 values are within a preset amount ϵ on either side of the new images's y_1 value. This procedure can be repeated with the other components of $\mathbf{y}$, leaving only a small portion of the total set $\mathbf{Y}$ to be evaluated for the nearest neighbor search. For an eigenspace based object recognition algorithm, search time using the algorithm was reduced to 1/75 of the exhaustive search of the total set $\mathbf{Y}$.

As an alternative to this approach of preprocessing the projection coefficients, another technique for limiting the number of comparisons for the nearest neighbor search is to perform a coarse search of the projection coefficients corresponding to the actual training images. In most cases, the interpolated projection coefficients will significantly outnumber those corresponding to the training images (ie. 1,000,000 to 300 for the example in Chapter 3). This can be followed by a truncated nearest neighbor search of the interpolated projection coefficients of $\mathbf{Y}$ within a set distance of the position of the matched training image projection coefficients $\mathbf{y}_{nn}$. The set distance will be dependent on the application, but in many cases can be set to reduce the number of interpolated training projection coefficients to be evaluated to less than 5% of the total interpolated projection coefficients with no loss of accuracy. This method is suggested as an alternative to Nene and Nayar's method because the occlusion robust approaches to eigenspace positioning described in Chapter 4 do not allow preprocessing of the set of interpolation projection coefficients $\mathbf{Y}$. Potentially an approach could be developed that performed interpolation of the training image projection coefficients after the nearest neighbor search phase. Thus the amount of interpolation could be increased dynamically.

2.3.2 Calculation of the Eigenvectors

As stated earlier, the eigenvectors (corresponding to the largest eigenvalues) of the covariance matrix for the visual subspace are the features used to provide low dimensional representations allowing the use of eigenspace methods for object or camera positioning. However the calculation of the eigenvectors directly from the covariance matrix corresponding to the visual subspace is generally computationally intractable. The covariance matrix for the visual subspace can be estimated by acquiring a training set of images spaced equally throughout the movement range of the camera. As stated in section 2.2.1, if $\mathbf{x}_i$ represents one of the image vectors from the set we can construct a matrix $\mathbf{X}$ consisting of these m image vectors collected columnwise:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \tag{2.21}$$

Consequently we can calculate an estimate of the covariance matrix $\mathbf{C}$ via:

$$\mathbf{C} = \mathbf{X}\mathbf{X}^T \tag{2.22}$$

This matrix will have dimensions equal to the length of the image vectors themselves. Thus for images of size 320 by 240 pixels, the dimensions for the covariance matrix would be 76800 by 76800. Calculating the eigenvectors from such a large image is computationally intractable.

Fortunately, an alternative method exists to estimate the eigenvectors of the covariance matrix (outlined by Murakami and Kumar [31]) if the number of images m used to estimate the covariance matrix is significantly smaller than the number of pixels n in the images. This is based on the well known singular value decomposition:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \tag{2.23}$$

where the column vectors of $\mathbf{U}$ correspond to the normed eigenvectors of $\mathbf{A}\mathbf{A}^T$ and the column vectors of $\mathbf{V}$ correspond to the normed eigenvectors of $\mathbf{A}^T\mathbf{A}$. Noting that $\mathbf{A}\mathbf{A}^T$ is equivalent to our estimate of the covariance matrix, we term $\mathbf{C}_{implicit}$ as the implicit covariance matrix:

$$\mathbf{C}_{implicit} = \mathbf{X}^T\mathbf{X} \tag{2.24}$$

which has dimensions the same size as the number of training images m. A well known relationship for the SVD is that the column vectors $\mathbf{v}_i$ of $\mathbf{V}$ are related to the column vectors $\mathbf{u}_i$ of $\mathbf{U}$ via:

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i \tag{2.25}$$

where the σ_i are the singular values of Σ and σ_i^2 corresponds to the eigenvalues of $\mathbf{A}^T\mathbf{A}$. This relation allows the eigenvectors of $\mathbf{A}\mathbf{A}^T$ to be estimated via the eigenvectors of $\mathbf{A}^T\mathbf{A}$. For our purposes this allows calculation of the eigenvectors $\mathbf{e}_i$ of the covariance matrix estimate $\mathbf{C}$ via the eigenvectors $\tilde{\mathbf{e}}_i$ of the much smaller implicit covariance matrix $\mathbf{C}_{implicit}$:

$$\mathbf{e}_i = \lambda_i^{-1/2}\mathbf{X}\tilde{\mathbf{e}}_i \tag{2.26}$$

The calculation of the eigenvectors of $\mathbf{C}_{implicit}$ can be performed via a commercial software package such as Matlab without computation time being a significant issue.

This approach to deriving the eigenvectors has some interesting consequences for the accuracy for eigenspace positioning and will be discussed in the next chapter. Note that the training images $\mathbf{x}$ used to calculate the projection coefficients $\mathbf{Y}$ for eigenspace positioning do not necessarily have to be the same as those used to derive

the eigenvectors. These two steps are in fact independent of one another. Potentially one could calculate feature vectors independently of the image statistics such as using a Fourier or a Discrete Cosine transform.

2.4 Chapter Summary

General positioning camera problems such as those encountered for mobile robots and part positioning were described as well as the visual subspaces formed corresponding to the movement range of cameras. The use of Principal Components Analysis for representing images from these visual subspaces in reduced form was shown. Obtaining the eigenvectors corresponding to the Principal Components via the implicit covariance matrix was explained. Subsequently, the basic technique of determining camera position over a predefined movement range using eigenspace projections was explained:

- Obtain a set of training images spaced equally throughout the movement range of the camera
- Using the implicit covariance matrix, obtain the eigenvectors corresponding to the covariance matrix of the training images
- Collect a set of projection coefficients over the movement range of the camera by projecting the training images on the eigenvectors
- Interpolate the projection coefficients for camera positions intermediate to the training images for additional accuracy
- For a new image, find its projection coefficients and perform a nearest neighbor search with the stored set of interpolated projections

- The current camera position is determined as that with the nearest neighbor projection coefficients

Chapter 3

Eigenspace Positioning and Accuracy

3.1 Accuracy and Eigenspace Positioning

The accuracy that can be achieved by eigenspace based position determination is an important but not oft studied issue. If eigenspace positioning is to be used in industrial applications, especially as a tool for high accuracy positioning of robot end-effectors, the factors that influence accuracy must be well known. Typically there are two areas that affect the accuracy for a given application:

1. The visual and distance parameters of the visual subspace of the problem.
2. The controllable parameters available such as the number of training images, amount and types of interpolation and the number of eigenvectors.

This chapter explores the effects on accuracy of these two problems, the limits of accuracy achievable and recommendations for improving accuracy via the controllable parameters. Also included is a new technique for filtering the eigenvectors to improve accuracy in certain instances.

3.2 General Discussion of Accuracy

Positional accuracy relies upon many related factors. One simple way to approach accuracy is the observation that accuracy can be said to primarily depend upon how close the interpolated projection coefficients are over the movement range of the camera to the actual projection coefficients at the interpolated locations.

As mentioned in section 2.3, using only the projection coefficients corresponding to the training images will limit the accuracy because only the positions of the camera for the training images can be returned as the new position of the camera. Projection coefficients corresponding to positions intermediate to the training images can be generated by interpolating the projection coefficients for each eigenvector. Accuracy can be increased by using more interpolated projection coefficients until a limit is reached in part due to the inaccuracy of the interpolation projection coefficients themselves.

If the interpolated projection coefficients vary significantly from the actual values, then at the nearest neighbor matching phase, the projection coefficients of a new image will likely not match the interpolated projection coefficients at the actual position. Larger variance from the actual projection coefficients will increase the possibility of even larger errors. The exact effect on accuracy is difficult to predict, since potentially the projection coefficients of any nearby position could be matched. This topic is discussed in more depth in Chapter 4, however the important fact to consider here is that improving the accuracy of the interpolated projection coefficients is paramount for improving positional accuracy.

In terms of improving accuracy the question is then what can be done to improve the accuracy of the interpolation as well as how visual characteristics of visual subspaces affect the accuracy of the interpolation.

In Chapter 2, we described the projection coefficients as forming a manifold corresponding to a range of camera positions. For explanative purposes we will also refer

to the plot of the projection coefficients for each eigenvector versus camera position as a projection plot or as a plot of the manifold versus position for each eigenvector. Figure 3.5 shows an example of this concept, it is explained in more depth in section 3.3.1.

With this concept of the projection coefficients of an eigenvector forming a manifold surface versus camera position it can be said ideally for any application to obtain good interpolation, the eigenvectors should produce a smooth projection manifold for each eigenvector that with no discontinuities or regions of rapid change that could result in regions of camera movement with inaccurate interpolation.

The controllable parameters available to the designer of a position determination application can have important effects on improving the accuracy of the interpolation, especially the number of training images used to calculate the projection coefficients and the resulting interpolated projection coefficients. Their effects on accuracy will be discussed later on in the chapter.

The visual characteristics of the applications visual subspace can also have important effects. Especially important, in terms of the limits of the accuracy achievable is the relationships between the amount of image feature change within the images versus the range of camera movement.

Before these ideas are presented, an experimental example of position determination will be illustrated in the next section to provide a basis of evaluating the ideas in this chapter.

3.3 Experimental Example

A simple example was chosen for exploring eigenspace positioning accuracy. The application was chosen to be determining the two dimensional translation position of a camera relative to a metal part and its background. Conceivably such situations

could arise for visual servoing applications where the task is to properly position an end-effector with a camera mounted on it relative to the part. The camera was mounted on an XY table to allow precise positioning; the positional resolution of the table was 10 μm in both directions.

For all experiments in the thesis except where explicitly described, the size of the images used was 320 by 240 pixels, acquired using a grayscale 8-bit Matrox Meteor II frame grabber card. In all cases the eigenvectors were calculated with Matlab. All other processing was performed in C++ on a 600 MHz Pentium III computer.

For this particular experiment, the range of camera motion was confined to a 4 cm by 4 cm square. To allow a wide latitude in terms of selecting possible sets of training images, 4225 images were acquired of the object over a 65 by 65 grid covering the movement range of the camera. Thus the images were 0.625 cm apart in each direction. Note that the movement of the camera was perpendicular to its optical axis, the significance of which will be addressed in the chapter on multiple cameras.

Figure 3.1 shows four images from the entire set of images, corresponding to the four corners of the movement range. Note that the object was approximately 25 cm from the lens of the camera. Examining the four extreme position images, the distance in pixels the image features (such as the hexagonal recess for example) move within the images over the camera movement range is 168 pixels in both directions.

A set of 200 images with the position of the camera at random positions throughout its prescribed movement range was also acquired. This allows accuracy to be estimated via this testing set because the random positions of the camera were recorded for comparison to the position determined by the eigenspace positioning.

To illustrate the accuracy achievable, a subset of 289 images consisting of images spaced 2.5 mm apart in each direction from the entire image set was used to estimate the eigenvectors of the visual subspace with the implicit covariance approach as described in the previous chapter. The same set of 289 training images was used to

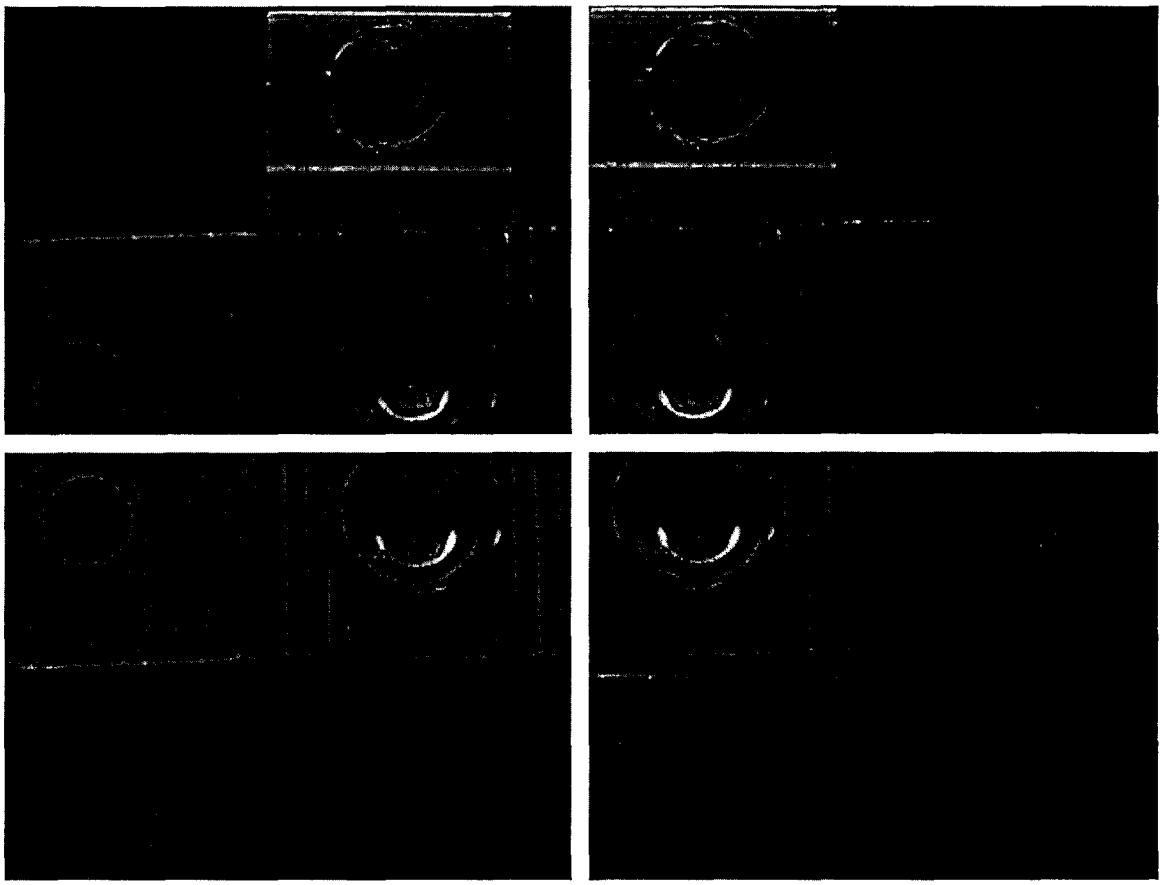

Figure 3.1: Corner Images of Camera Movement Range

calculate a set of projection coefficients covering the movement range of the camera. This set of projection coefficients was interpolated linearly as described in section 2.3 to provide 64 positions in each direction between the original 289 positions, a set of 1041 by 1041 camera positions are then available to match. Such a large amount of interpolation is required to ensure the full accuracy capability of the eigenspace positioning method is evaluated. For all experiments, interpolation above this level resulted in no improvement in accuracy. Unless explicitly mentioned, an exhaustive search for the nearest neighbor matching phase was used to thoroughly evaluate the accuracy possible.

Using the random set of 200 images that was also acquired in addition to the

training set of images, each had its projection coefficients calculated from the 289 set eigenvectors and a nearest neighbor search was performed between those projection coefficients and that of the interpolated set. The error for each random image was determined as the difference in position between the random image and that of the interpolated matched projection coefficients. The error was reported in two parts corresponding to the two directions of movement, the vertical error consisting of the difference in position for camera movement up and down with regards to the images and similarly the horizontal error corresponding to the side to side movement of the camera. Thus for reference purposes, camera movement along one axis of the XY table will be referred to as vertical movement and horizontal for the other axis. The camera is roughly aligned for simplicity such that this vertical camera movement produces mostly vertical image flow with a small horizontal component. Note the camera could be rotated such that movement along one of the XY axes would produce diagonal movement of the image features within the images.

The mean absolute error in the vertical direction reported for the random set was 69.6 μm and in the horizontal direction it was 191 μm. The error as a percentage of the distance between training images was 2.78 % for the horizontal direction and 7.64 % for the vertical direction. To illustrate the distributions of the errors in both directions, Figure 3.2(a) shows a histogram of the absolute error for the vertical direction. Similarly Figure 3.2(b) shows a histogram of the absolute error for the horizontal direction (for the remainder of this chapter, part (a) of an error histogram represents the vertical absolute error).

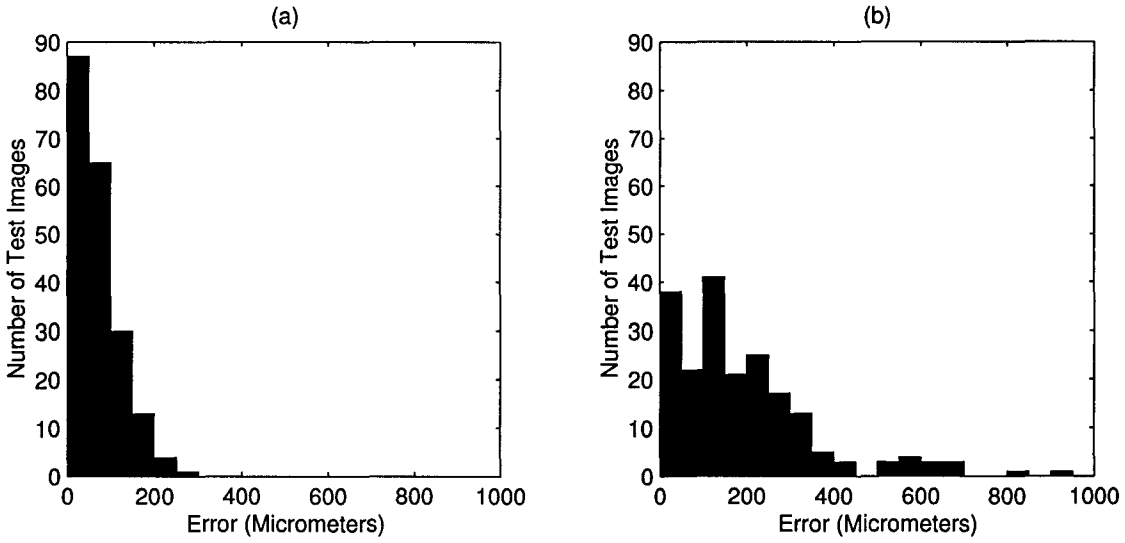

Figure 3.2: Histograms of Absolute Position Error, (a) Vertical, (b) Horizontal

3.3.1 Tools for Analysis of Visual Subspaces for Eigenspace Positioning

Understanding the relationships between visual subspaces and the resultant positional accuracy achievable can be quite complicated. The following two techniques can be used to enhance this understanding, demonstrated with the example visual subspace.

A useful tool for examining the behaviour of eigenspace methods is the visualizations of the eigenvectors $\mathbf{e}_i$ used to form the projection coefficients $\mathbf{y}$ for the visual subspace. In some instances, the performance in terms of accuracy can be partially explained with regards to the visualized eigenvectors and the images from the visual subspaces. Visualized eigenvectors can be constructed by noting the eigenvectors are in row vector format but their values correspond to specific positions in the original images. Thus performing the opposite of the concatenation operation performed on the training images to form image vectors, the eigenvector values can be assigned to an image corresponding to the original image dimensions and quantizing the floating

point values equally to values between 0 and 255 to form an 8 bit grayscale image. Figure 3.3 shows the visualized eigenvectors of the example visual subspace corresponding to the highest four eigenvalues. Note that these first eigenvectors are for the most part low frequency in both directions, as would be expected for reconstruction optimized basis vectors. Note that using only 289 images results in a pattern of grid like formations. To illustrate the contrast between the early eigenvectors, Figure 3.4 shows the visualized eigenvectors corresponding to the 13th through 16th eigenvectors. In comparison to the first through fourth visualized eigenvectors, these visualized eigenvectors are of a higher frequency in nature. These high frequency characteristics again follows the patterns one would expect of using reconstruction optimization based vectors.

A similar tool for analyzing visual subspace eigenspace behaviour is plots of the individual projection coefficients y_i of each eigenvector $\mathbf{e}_i$ versus camera position. Simply for each eigenvector, the projection coefficient value is plotted versus the training image camera positions. Figure 3.5(a) through 3.5(d) shows the plots versus the 4 cm by 4 cm camera motion range for the first four eigenvectors using 289 training images. The plots are drawn as mesh plots to reinforce the concept of the projection coefficients as being part of a manifold. For analysis purposes, we will refer to these individual projection plots as forming a manifold of projection coefficient values versus position and we are interested in the shape these individual manifolds exhibit for each eigenvector. Note that these first few manifolds change relatively little and smoothly over the movement range of the camera as would be expected with the corresponding low frequency eigenvectors. To illustrate different manifold properties of the later eigenvectors, Figure 3.6(a) through 3.6(d) shows the projection plots corresponding to the eigenvectors with the thirteenth through sixteenth largest eigenvalues. Note that the manifolds of the later eigenvectors are considerably less smooth and high frequency. Also note that interpolating accurately for the projection coefficients of

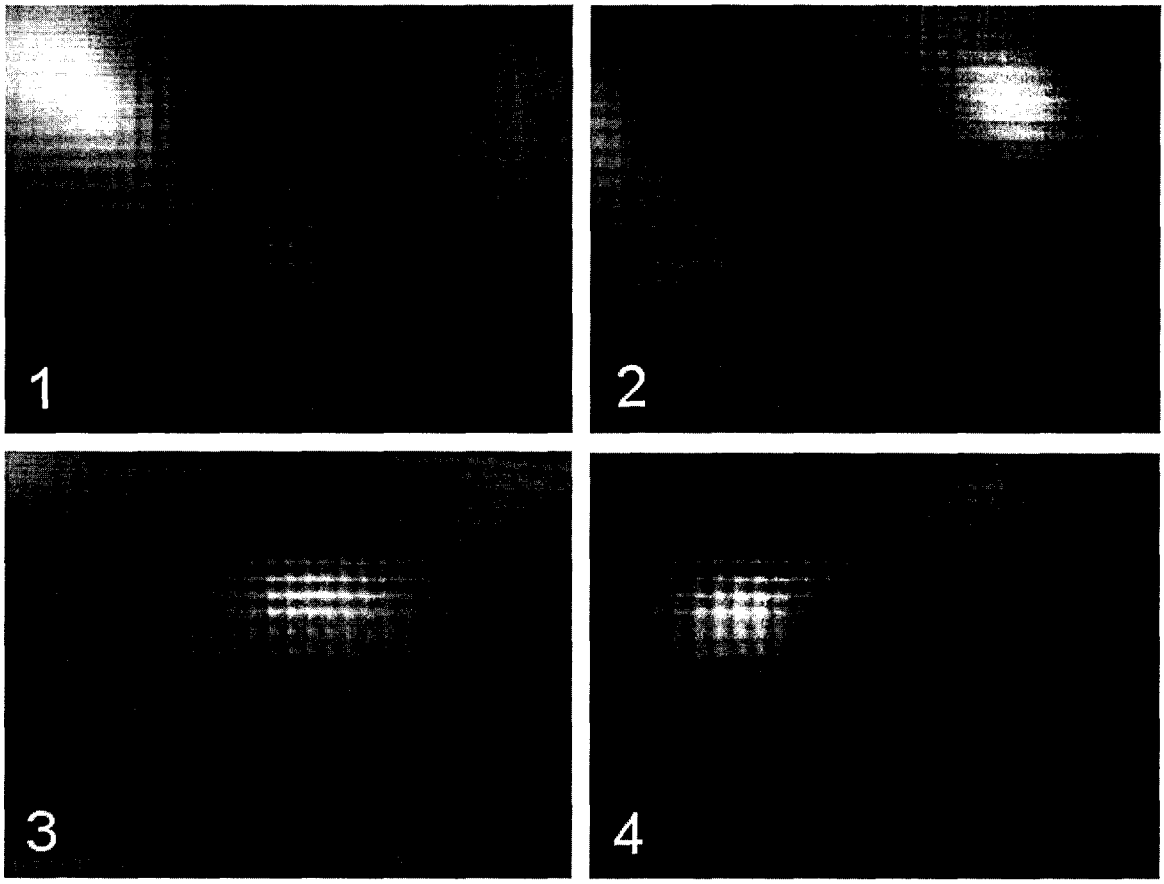

Figure 3.3: First Four Eigenvectors

the later eigenvectors will be more difficult than for the earliest eigenvectors with a set number of training images.

3.4 Image Feature Change and the Limits of Accuracy

The second half of this chapter discusses the effects of the adjustable training parameters on accuracy. More importantly in some ways however is the nature of the information presented by the images in the targeted visual subspace itself in terms of

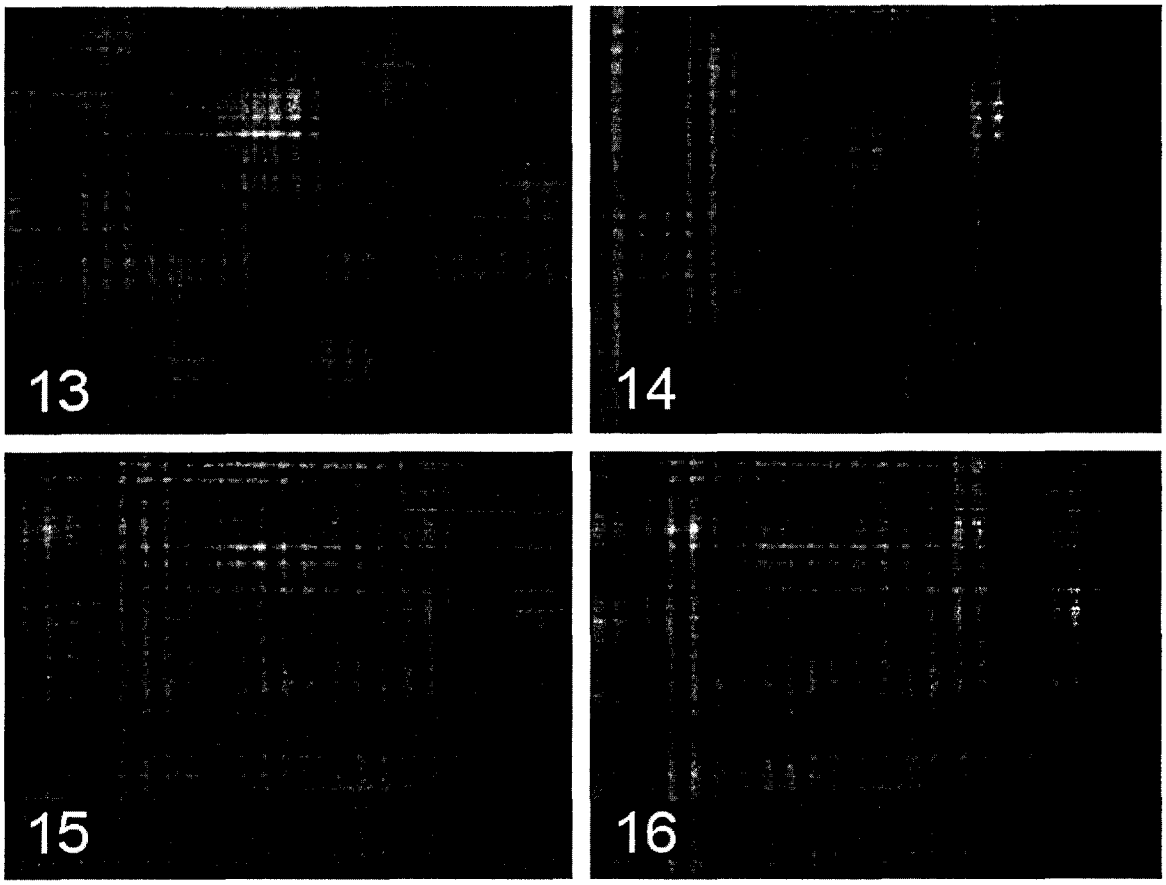

Figure 3.4: Thirteenth through Sixteenth Eigenvectors

the ability to determine position.

The ability to determine position with eigenspace methods depends on the smoothness of the projection coefficients manifolds versus position and thus the ability to correctly interpolate intermediate positions. If for a manifold corresponding to the projection coefficients of one eigenvector versus camera position the coefficient values change rapidly and discontinuously with camera movement, determining position with such interpolated projection coefficients will be difficult. However, an important idea that underlies these manifolds beyond the eigenspace paradigm is the idea of the image feature change over the movement range of the camera. Informally, this refers to the change in pixel position of image features such as a corner within the images

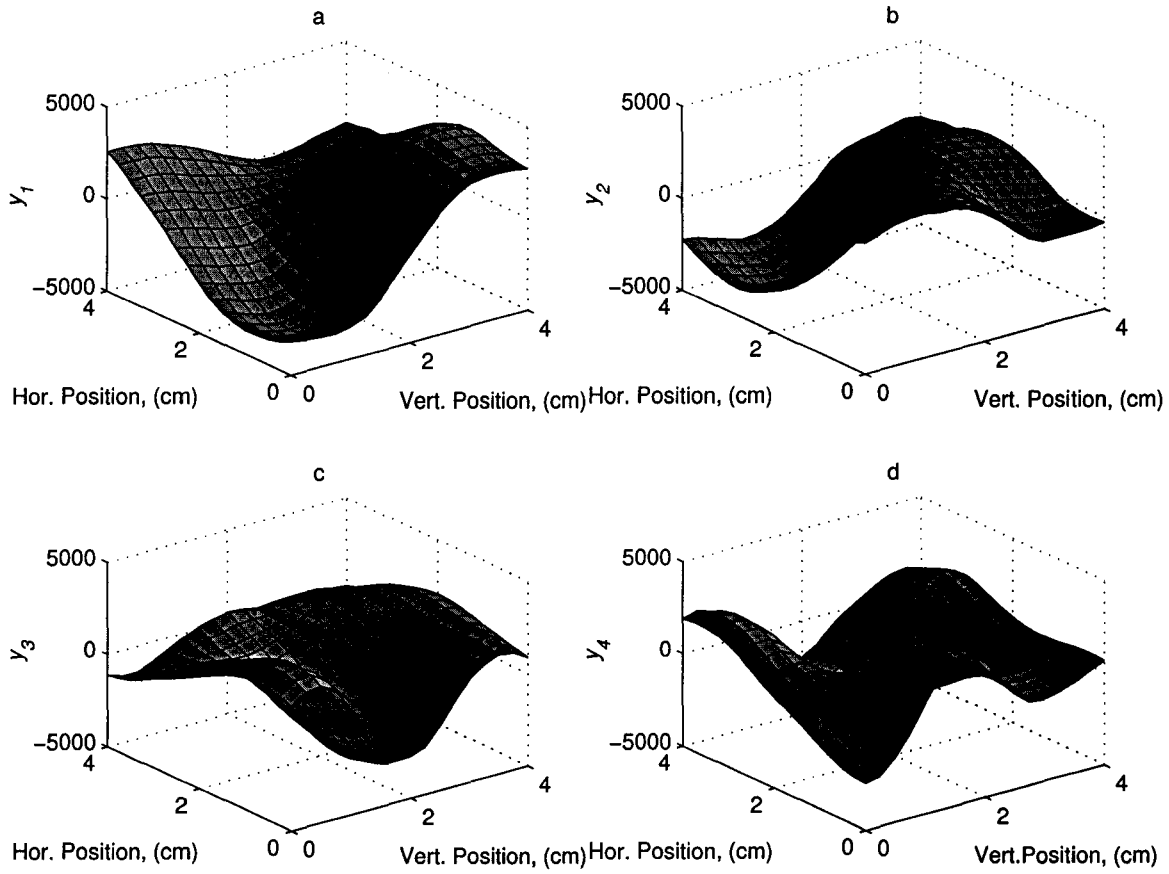

Figure 3.5: Projection Coefficient Value versus Position, (a) First Eigenvector, (b) Second Eigenvector, (c) Third Eigenvector, (d) Fourth Eigenvector

as the camera moves.

For the remainder of this chapter, this idea will be explored with regards to two dimensional translational positioning for simplicity, however it also has important implications to higher dimensional positioning problems which are considered in Chapter 6.

Basically the idea of image feature change with regard to positioning is the limit of the accuracy achievable for a set amount of image feature change across the movement range of the camera. Referencing Figure 3.1 which shows the four images corresponding to corner positions of the camera for the main experiment in this chapter, it is easy

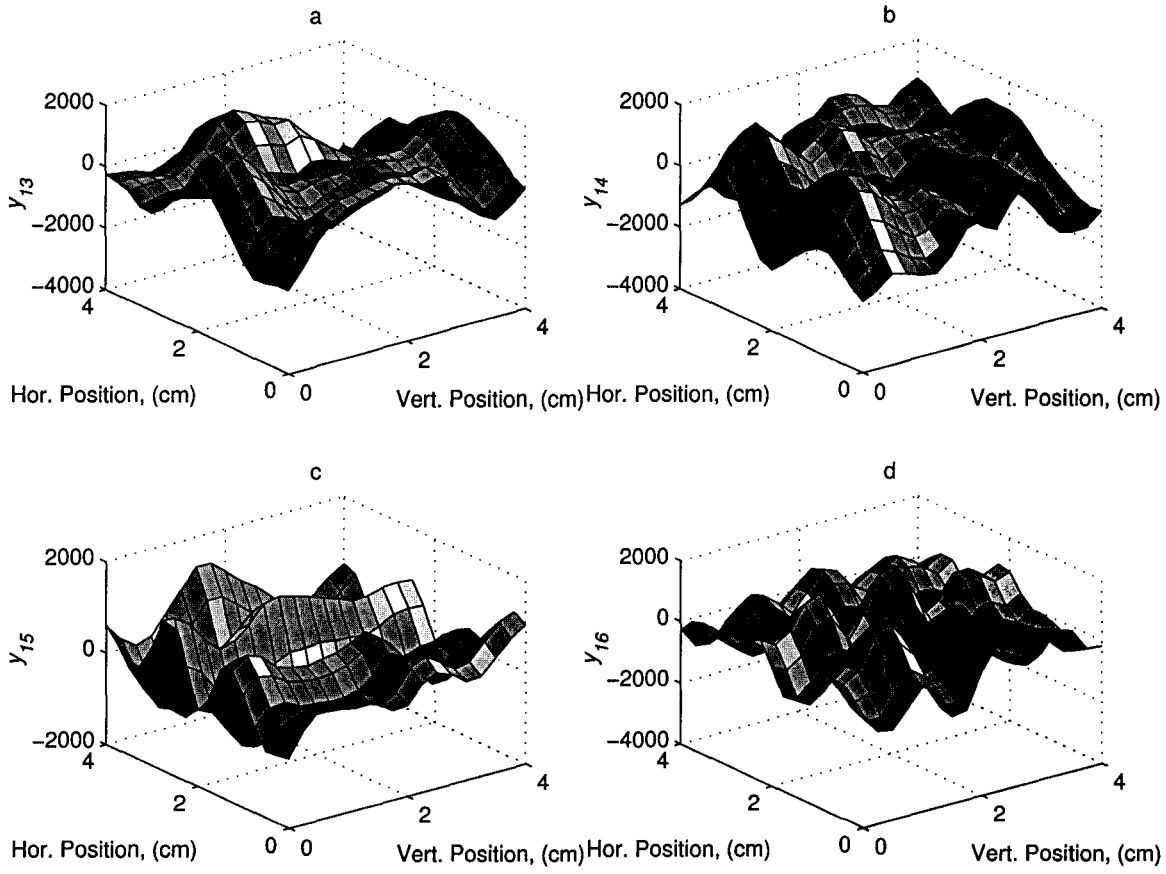

Figure 3.6: Projection Coefficient Value versus Position, (a) Thirteenth Eigenvector, (b) Fourteenth Eigenvector, (c) Fifteenth Eigenvector, (d) Sixteenth Eigenvector

to notice the movement of the visual features in the scene to different pixel locations within the four images. Looking at the position within the image of the hexagonal recess in the upper left image and comparing its position in the lower right image, the recess has moved approximately 1/2 the width of the image in both directions. In fact, this recess was measured to have moved 168 pixels in each direction across the movement range.

More formally for determining the accuracy possible with eigenspace positioning we introduce the measurement of the camera movement per pixel of image feature change D. In basic terms, D can be defined for each direction of camera movement

as the distance the camera moves between two images divided by the distance in pixels coordinates between the same image feature in both images. Or more simply, the amount of camera movement that is required to shift the entire image contents by one pixel in the specified direction (for simplicity camera perspective effects are ignored). For a simple case such as two dimensional translational movement of a camera, the change in location of image features over the movement range is easily discerned. Figure 3.7 shows an example of the distribution of the training images for the example in this chapter.

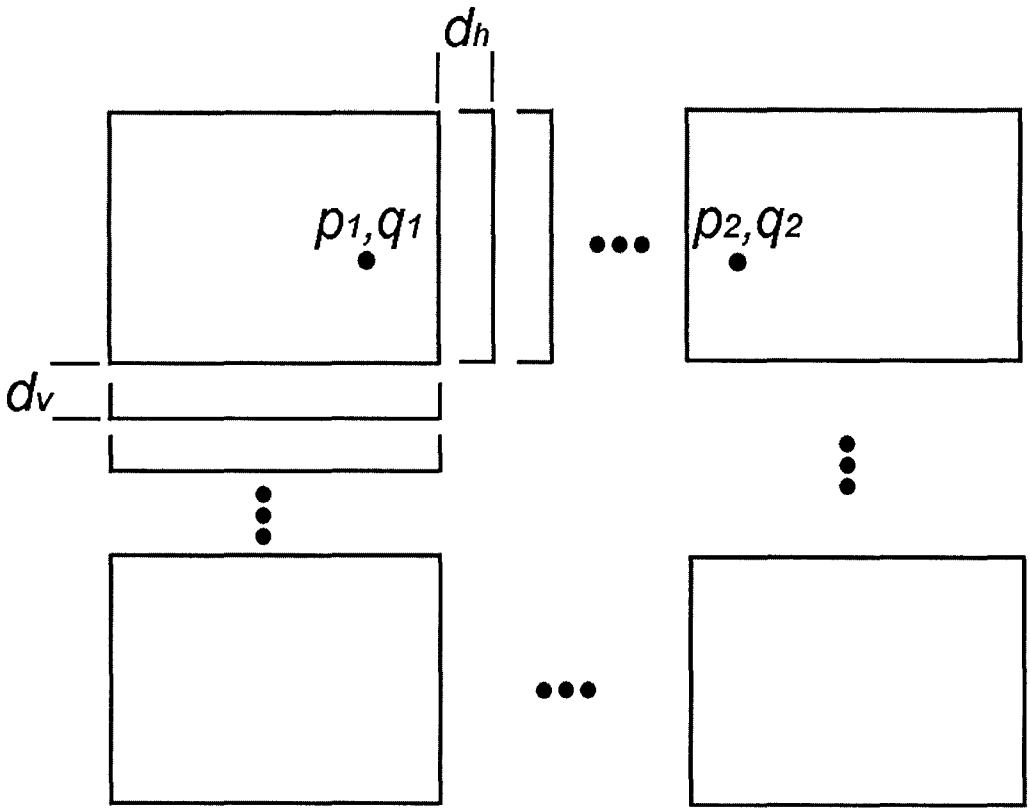

Figure 3.7: Example of Training Image Distribution

We define d_h as the distance between each training image in the direction of camera movement that produces horizontal pixel movement of the image features, similarly

d_v corresponds to the distance between training images for camera movement that produces vertical pixel movement. The location of the same image feature in two of the training images is defined as p_1, q_1 and p_2, q_2 where p_1 and p_2 refer to vertical pixel coordinates.

For horizontal camera movement, D_h for any two images is defined as:

$$D_h = \frac{d_h \cdot a}{|q_1 - q_2|} \tag{3.27}$$

where a is the number of d_h increments separating the images and $|q_1 - q_2|$ is the horizontal pixel distance between an image feature in both images. Similarly for vertical camera movement, D_v is defined by:

$$D_v = \frac{d_v \cdot b}{|p_1 - p_2|} \tag{3.28}$$

where b is the number of vertical increments between the images and $|p_1 - p_2|$ is the vertical pixel distance between an image feature in both images.

The D values give a bound on the maximum accuracy possible in a given direction of camera movement. Multiplying D by 2 raised to the power of the grayscale bits g of a pixel in the images defines the smallest camera movement SM that can be theoretically detected (by producing a change in the pixel values of the image) and thus an absolute upper bound on accuracy (assuming an infinite number of training images for forming the projection coefficients):

$$SM = D \cdot 2^g \tag{3.29}$$

The accuracy achievable for eigenspace positioning will be below that indicated in the above equation, since in practice several bits of g are lost due to noise and the fact a finite number of training images are used. Also images where the difference in contrast between image features is small may also effectively lower the SM value,

since the position of image features cannot be accurately detected. For example, if for an 8-bit grayscale image all the pixel values of the images were limited to values between 0 and 31 would effectively lower g to 5 ($2^5 = 32$).

As will be described in section 3.5.3 on the number of images for the projection coefficients, the SM value limits the accuracy since increasing the number of training images indefinitely will result in the situation where the projection coefficient values between adjacent images no longer change significantly due to the SM value. Further increases, as will be shown lead to almost no improvement in accuracy. Thus to further increase accuracy, either the D or g values must be increased.

The direction of camera movement is important for determining accuracy because D can vary with direction. For example, a camera moving forward (such that the contents of the scene appear to flow toward the observer) may typically have a much larger D compared to that corresponding to lateral movement. Consequently, accuracy in the forward direction will be poorer (as a large D value denotes a large amount of camera movement for each pixel an image feature moves within the image). The use of a fish-eye lens or an omnidirectional camera for eigenspace positioning complicates the calculation of SM values, since image features will change position at a different rate depending on their position with the image.

Note that this explanation relies on the fact that in the experiment in this chapter, that camera movement along one of the axes of the XY table produced almost exclusively a change in the horizontal image coordinates for the image features whereas in the other axis, instead predominately a change in vertical image coordinates for the features resulted. This was deliberately arranged to simplify the explanation of the overall concept. More correctly, for an axis of camera movement the distance in both image coordinates (p,q), $\sqrt{(p_1 - p_2)^2 + (q_1 - q_2)^2}$ should be considered, especially if the camera is aligned such that camera movement in each axis produces image feature change of similar magnitude in terms of both image coordinates. This idea of image

feature change is most important in regards to the number of images used to form the projection coefficients later in the chapter, but is also used in Chapter 6.

3.5 Adjustable Parameters

The nature of the visual subspace in terms of the relations between the number of the pixels that change and camera movement have a substantive effect on the accuracy that can be achieved including the absolute limits of accuracy that can be achieved. Obviously for visual subspaces where the image change for a given movement range is limited, a truly large amount of training images will not overcome this deficiency.

Nonetheless, for a given range of camera motion, the training parameters can have a significant effect on the positional accuracy achieved. Certainly the number of training images used to cover the movement range of the camera is important, both for deriving the eigenvectors, as well as for forming the set of projection coefficients. As well, since these are possibly a limited resource in terms of acquisition difficulty, it is important to know how many images will be required for adequate performance. The remainder of the chapter analyzes these factors experimentally to provide useful information for designing eigenspace positioning applications.

3.5.1 Number of Images for Calculating Eigenvectors

The number of training images used for calculating the eigenvectors for a visual subspace is a parameter that can cause significant effects on the accuracy achievable for eigenspace positioning problems. Intuitively, increasing the number of images used to form the eigenspace will increase the accuracy as the implicit covariance matrix formed by the training images will be a better estimate of the actual covariance matrix.

The number of images required for high positional accuracy is related to the

amount of image feature change over the movement range of the camera. Obviously a larger camera movement range will require a larger number of images to maintain the same image density.

When considering this density and its relation to accuracy it is important to consider the implicit covariance technique for estimating the eigenvectors. The implicit covariance matrix eigenvectors $\tilde{\mathbf{e}}_i$ are combined via the $\mathbf{X}$ matrix of the training images to find the approximation of the covariance eigenvectors $\mathbf{e}_i$:

$$\mathbf{e}_i = v_i^{-1/2}\mathbf{X}\tilde{\mathbf{e}}_i \tag{3.30}$$

Thus the eigenvectors themselves are linear combinations of the training images. The consequences of using the implicit covariance matrix and a limited number of images for 2D eigenspace positioning is especially felt when the density of the training images is low relative to the amount of image feature change. In situations where the difference in location of image features between adjacent training images is more than one pixel, the visualized eigenvectors produced by the linear combinations of the training images are not smooth. As an experimental example of this effect, Figure 3.8 shows the first visualized eigenvector $\mathbf{e}_1$ corresponding to 25, 81, 289 and 1089 training images equally spaced throughout the movement range of the camera. In these images, sharp, almost edge like grayscale changes are clearly visible, corresponding to the sharp edges contained within the training images. This effect is reduced with an increasing number of images.

The consequences of the sharp edges corresponding to eigenvectors produced by too few images is reduced positional accuracy as the resultant projection coefficient manifolds produced by these eigenvectors will not be smooth. If the visualized eigenvectors have sharp edges and areas of rapid change (such as that produced in the example with only 25 images) the interaction between these eigenvectors and the

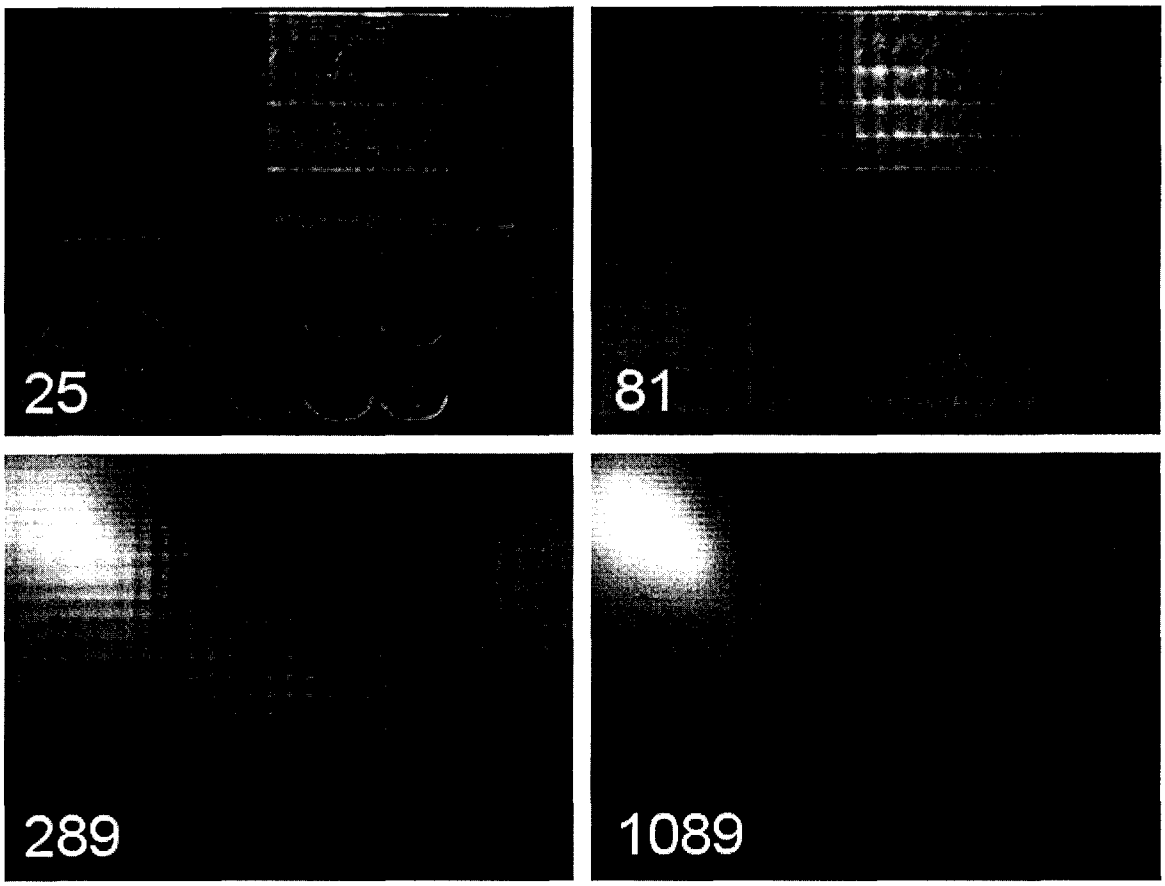

Figure 3.8: First Visualized Eigenvectors with Different Number of Training Images (Number of Images Shown in Figure)

high frequency portion of the images produced as the camera moves will produce projection coefficients that will themselves change rapidly and with reduced smoothness. Hence there will be more interpolation errors and consequently poor accuracy.

To demonstrate the effects on accuracy with the number of training images used for deriving the eigenvectors, Table 3.1 shows the absolute mean errors in both directions for 25, 81, 289 and 1089 images. Note 289 images were used to form the projection coefficients to isolate the effects of varying the number of images for forming the eigenvectors. As can be seen, the absolute mean errors decrease significantly with an increase in the number of training images. For the largest number of images, the

difference between the errors in both directions is also much improved.

Number of Training Images	Absolute Mean Error Vertical μm	Absolute Mean Error Horizontal μm
25	99.7	263
81	78.5	203
289	69.0	191
1089	42.5	54.0

Table 3.1: Absolute Mean Errors, Varied Number of Images for Eigenvectors

A consequence of this performance is that for an application where the number of pixels the image features change position over the movement range of the camera is large comparatively, achieving a high density of images requires an inordinate number of images which could be difficult to acquire.

Beyond a certain number of training images, the density of images will be such that the eigenvectors will not have any discontinuous artifacts and additional images will not result in an increased performance in terms of accuracy. Thus for applications where the image feature change is small because of a limited range of camera movement, the number of training images for forming the eigenvectors will be small in terms of requirements. This was the case with the experimental example in the next chapter, where the image features moved only approximately 15 pixel positions in both directions for the entire movement range compared to 160 for this chapter.

Clearly, for some instances, acquiring such large sets of training images may not be practical, either in terms of the time required for their acquisition or if there is difficulty in acquiring images with regards to the accuracy achievable during the training phase. The next section describes an eigenvector smoothing operation that can result in increased accuracy with a limited number of training images.

3.5.2 Eigenvector Filtering

The issue of performance with a limited number of training images is the poor results caused by the high frequency elements in the eigenvectors corresponding to the linear combination of the training images. As previously described, achieving smooth projection manifolds leads to higher accuracy as the interpolated values will be more accurate for the nearest neighbor search.

A solution to the problem of these eigenvectors is to perform a smoothing operation on the eigenvectors themselves. Such an operation will produce projection coefficient manifolds that are themselves smoother. Consider a region of the training images containing a sharp edge. As the camera moves, the position of the edge within the images moves. If this edge moves over a nonsmooth section of a visualized eigenvector, the projection coefficient produced will change rapidly and possibly discontinuously. Smoothing the eigenvectors will reduce the possibility and thus a smoother manifold of projection coefficients. The lack of sharp changes in value leads to more accurate interpolation and hence more accuracy. If the number of training image projections is similarly limited, a smoother manifold surface is easier to describe for interpolation purposes with a limited number of points compared to a manifold that changes rapidly with a small amount of camera movement which will require denser sampling for accurate interpolation.

A consequence of such a smoothing operation on the eigenvectors is that the altered eigenvectors will no longer satisfy the orthogonality condition of an orthonormal basis:

$$\mathbf{e}_i\mathbf{e}_j = 0, i \neq j \tag{3.31}$$

However, this is a requirement for accurate reconstruction of images from the visual subspace rather than the task of determining position and in any event, the basic structure and patterns of the eigenvectors are relatively unchanged by smoothing.

Potentially one could achieve a similar effect of smoothing the projection coefficient manifold by instead performing a preprocessing operation on the images and smoothing them instead. However the smoothing of the eigenvectors is an operation that can be performed off-line whereas smoothing the images would require additional run-time computation.

Several image processing techniques could be used to perform a smoothing operation on the eigenvectors, although the end results would be similar. For simplicity to demonstrate the principle, the operation used to provide smooth eigenvectors was a repeated application of averaging the individual eigenvector values (mean filter) over the eight neighbor adjacent values (assuming the eigenvectors are viewed as arranged in matrices corresponding to the image pixel arrangement (as in the visualized eigenvectors), rather than a long concatenated vector). Note that for these operations it is assumed that the eigenvectors are not in their row vector format but rather the image matrix form corresponding to the training images, the same as for the procedure that produces the visualized eigenvectors. Thus assuming $\mathbf{e}_i^{r,s}$ represents a value of the ith eigenvector at pixel location (r,s), the new value after the averaging operation will be defined as:

$$\mathbf{e}_i^{r,s} = (1/9) \sum_{j=r-1}^{r+1} \sum_{k=s-1}^{s+1} \mathbf{e}_i^{j,k} \tag{3.32}$$

A small window was used so that the amount of smoothing could be controlled by repeated applications of the mean filter. This also allows the demonstration of how an optimal level of eigenvector smoothness can be achieved where upon further filtering will not result in an accuracy improvement. Note that multiple applications of a

kernel convolution operation (such as a mean filter) can be equivalently performed with a single equivalent kernel operation.

To demonstrate the utility of performing the smoothing operation on the eigenvectors for improving accuracy, the example eigenspace accuracy for this chapter (Figure 3.1) was reevaluated using eigenvectors that had been smoothed a number of times. The eigenvectors were initially estimated using 289 training images; the same set of images was used to derive the interpolated projection coefficients. Table 3.2 shows the absolute mean errors (both vertical and horizontal) resulting from the test set using the filtered eigenvectors, with the number of mean filter operations varying from none to 40 times. As can be clearly seen, filtering results in a marked improvement in terms of the accuracy average over no smoothing, with the best performance for this example occurring with 30 filtering operations. No improvement is seen for filtering beyond 30 operations for this particular visual subspace. The amount of smoothing operations required to meet the optimal level is likely to depend on the relationship between the amount of pixel position change of the image features over the camera range and the number of training images over this range. Note that the absolute mean results for the 30 times filtering was actually better than using the unfiltered eigenvectors produced with 1089 training images (a four times increase in the number of training images). To illustrate the improvement in the error distribution, Figure 3.9(a) and 3.9(b) shows the histograms of the absolute error for the case of no filtering together with those representing 30 times filtering in Figure 3.9(c) and 3.9(d).

Note the improvement in terms of the worse case errors of the test set, which is of obvious importance for robotic applications. Figure 3.10 (a-f) shows the first eigenvector visualizations for 1, 5, 10, 20, 30 and 40 times filtered respectively. Compared to the unfiltered eigenvectors, there are no sharp changes over just a few pixels, but rather smooth areas of rapid change.

Thus in terms of the beneficial uses of smoothing the eigenvectors, the technique is

Number of Filtering Stages	Absolute Mean Error Vertical μm	Absolute Mean Error Horizontal μm
0	69.5	192
1	63.2	176
5	53.1	140
10	44.3	92.1
20	38.7	58.5
30	37.4	46.7
40	38.8	47.6

Table 3.2: Absolute Mean Errors, Varied Number of Images for Eigenvectors

useful in instances where the number of training images used to derive the eigenvectors is limited.

Potentially more advanced smoothing filters such as Gaussian filters could be used for the same effect. An area that would be worth investigating in the future would be to try and determine a relationship for optimal smoothing between the Gaussian spread parameter σ and the change in pixel location of image features between adjacent training images.

Note also for future reference that eigenvector smoothing was not used for the experiments in Chapters 4, 5 and 6. This was because for those examples the change in pixel location of image features between adjacent training images was considerably smaller than the example of this chapter and thus provided no benefit in terms of accuracy performance after smoothing was applied.

3.5.3 Coarse Search with Eigenvector Filtering

Since the absolute mean error was considerably reduced in the previous experiment with the mean filter applied 30 times, the same experiment with 30 times filtering was redone using the coarse search technique described in Chapter 2 to reduce the

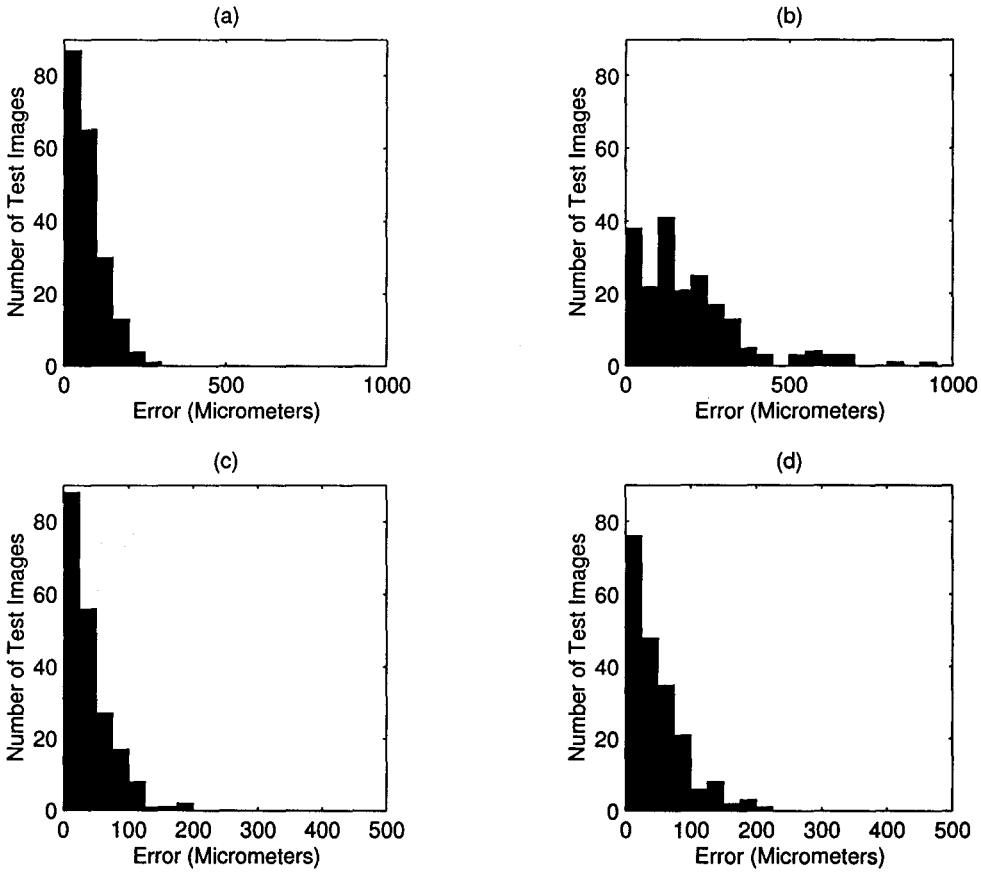

Figure 3.9: Histograms of Absolute Position Error, (a) Vertical, No Filtering, (b) Horizontal, No Filtering, (c) Vertical, Filtered, (d) Horizontal, Filtered

computation required for the nearest neighbor match phase of the projection coefficients. To reiterate, first a nearest neighbor search was performed on the projection coefficients corresponding to the 289 training images. Upon obtaining the nearest neighbor position for that set, a second nearest neighbor search was performed on a subset of the entire set of interpolated projection coefficients. This subset corresponded to the 131 by 131 interpolated positions surrounding the matched training image position (131 by 131 interpolated camera positions corresponds to 64 + 1 interpolated positions in each direction away from the training image position and thus incorporates the projection coefficients of adjacent training images). Obtaining the

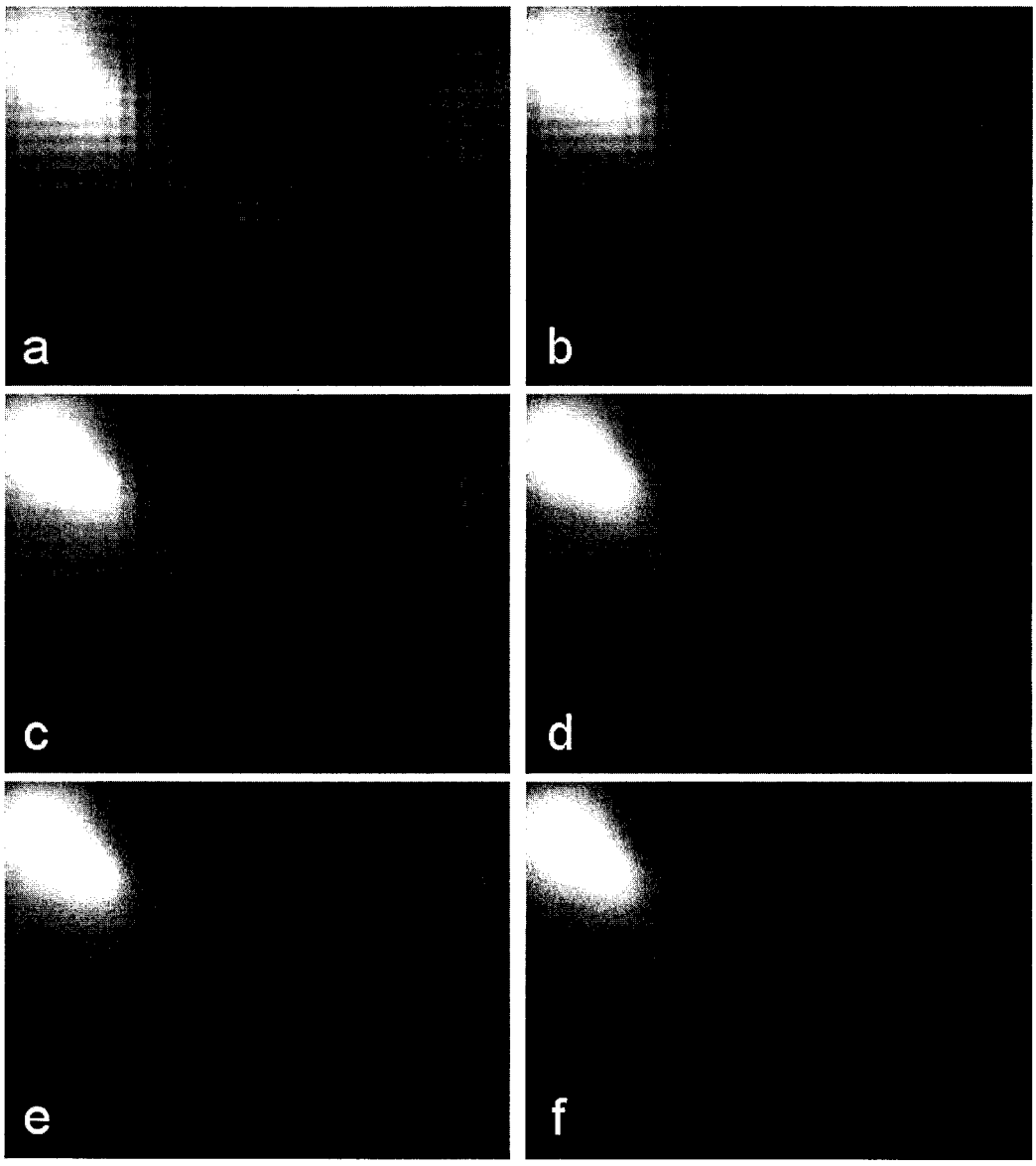

Figure 3.10: Smoothed First Visualized Eigenvectors, (a) Filtered Once, (b) Filtered Five Times, (c) Filtered Ten Times, (d) Filtered Twenty Times, (e) Filtered Thirty Times, (f) Filtered Forty Times

camera position of the second nearest neighbor search resulted in the same results as for the use of the entire set of interpolated projection coefficients. This same result was obtained with only 17450 comparisons of projection coefficients for the two nearest neighbors operations (289+131*131) compared to 1083681 (1041*1041) for the exhaustive search of the entire interpolated projection coefficients set (only 1.61 % of the exhaustive search). Thus for situations where the error is limited relative to the distance between adjacent training images, the coarse search approach can be used to reduce the search time considerably.

3.5.4 Number of Training Images for Projection Coefficients

The number of training images used for calculating the initial set of projection coefficients for interpolation also has a significant effect on accuracy. Compared to the number of images used to derive the eigenvectors, the number of training images used for the initial projection coefficients has a larger effect on the positional accuracy for both mean error and worst case error.

Note that while it seems odd to consider the number of training images used for forming the eigenvectors and those used for the projection coefficients separately, considering them separately lends insight into how accuracy functions with eigenspace positioning. Further the eigenvectors could potentially be replaced by alternate feature vectors such as discrete cosine transform basis vectors or generating the eigenvectors with a set of random images of the visual subspace rather than an equally spaced set.

Considering again the projection coefficient manifold plots for the experimental eigenspace in Figure 3.5 and 3.6, it is intuitive that using only a few training images (5 for example) in both directions of camera movement to produce the projection coefficients will result in large errors for the interpolated projection coefficients located between the actual projection coefficients. This effect will be exacerbated for the later

projection coefficient manifolds which exhibit high frequency characteristics and thus have a larger potential of interpolation error with a limited number of training images.

In terms of accuracy, there will be a point in terms of the number of training images below which accuracy will markedly decrease. This will be based on the number of training images required to capture the general shape of the first few projection coefficient manifolds.

More formally, the idea of the sampling theorem for properly reconstructing an analog signal from discrete samples [44] can be invoked. As shown in Figure 3.5, the projection coefficient manifold plots for the first four eigenvectors consist of low frequency waveforms across the movement range of the camera. For the first eigenvector, the frequency in each direction of camera movement is close to 1 Hz, and gradually increases for the next three eigenvectors. According to the sampling theorem, slightly more than double the highest frequency present is required to accurately reconstruct a signal for any desired time. This frequency is commonly known as the Nyquist frequency. Sampling below this frequency results in aliasing error whereby multiple frequencies can correspond to the discrete samples; thus interpolation performed on a reconstructed signal in this situation will likely be erroneous.

For the case of eigenspace positioning and the manifold of projection coefficients for each eigenvector, rather than time being the measure of distance between samples, it is camera position distance in each direction. Thus for proper interpolation the sampling (in this case the number of training images in each direction) should be more than double the underlying frequency of the manifolds, otherwise significant projection coefficient errors due to aliasing can occur. For eigenspace position accuracy, this will manifest in considerable error when the sampling rate is below twice the frequency of the manifolds for the first few eigenvectors, as will be seen experimentally. With the sampling frequency above twice the frequency of the projection coefficient manifolds of the first few eigenvectors (3-5), accuracy will be significantly

increased. Further increases in sampling as will be shown experimentally, leads to a slower increase in the rate of accuracy. This is likely due to the reduction of interpolation error via alias for the remaining eigenvectors, whose projection coefficient manifolds have higher frequencies and thus require higher sampling to avoid aliasing error. Note that the projection coefficient manifolds of the first few eigenvectors are not perfect low frequency sinusoids and also likely contain high frequency components, thus increased sampling will lead to better interpolation even above the predominant low frequency components.

This increase in accuracy will continue with increasing number of training images until the limits imposed by the amount of image feature change over the camera movement range are approached as described earlier in the chapter. As will be shown experimentally, the error will eventually only negligibly improve with increasing training images, in line with that predicted by the D and SM values.

An obvious drawback for increasing positional accuracy via a large number of training images are again the impracticalities of obtaining such a set, either with regards to time or positional accuracy. This problem is especially relevant for higher dimensional problems beyond simple 2D translational movement. For full six dimensions of movement (translational and rotational) for servoing relative to an object, covering a small range of camera motion can require enormous numbers of images. Chapter 6 discusses an approach to ameliorate the need for large number of training images for 3D translational positioning.

To illustrate the effects of varying the number of training images used for the projection coefficients, the number of training images used was varied from 25 (5 images by 5 images) to 4225 (65 images by 65 images). The eigenvectors used were those derived from 289 images and then filtered 30 times as described in the previous section to produce a high level of accuracy. Fifteen of those eigenvectors were used for the nearest neighbor search; the training images were interpolated linearly to provide

1041 interpolated projections in both directions. Table 3.3 shows the absolute mean errors in both the vertical and horizontal directions for the random position image set and the errors as a percentage of the distance between training images. Figure 3.11 shows a semilog plot of the same values.

Number of Training Images	Absolute Mean Error Vertical μm	Absolute Mean Error Horizontal μm	Absolute Mean Error Vertical Percentage of Training Image Distance	Absolute Mean Error Horizontal Percentage of Training Image Distance
25	464	486	4.64	4.86
81	144	110	2.88	2.20
289	37.4	46.6	1.49	1.86
1089	24.8	28.1	1.98	2.25
4225	23.1	22.4	3.70	3.58

Table 3.3: Absolute Mean Errors, Varied Number of Images for Projections

These results confirm our earlier conjecture where the mean errors are initially quite high with the smallest training set since there are only 5 sampling points in each direction and decline rapidly in both directions when the set is quadrupled (more accurately, increasing m by a factor of 2 according to a number of images defined by $2m+1$ by $2m+1$). Further quadrupling the set improves the accuracy, but at a much slower rate. The last quadrupling barely increases the mean accuracy. Thus increasing the number of images from 1089 to 4255 has a muted positive effect. This would indicate the limits of accuracy imposed by the camera movement per pixel of image feature change are being approached. For this experiment, D_h and D_v are both approximately 238 μm per pixel, however the absolute mean errors are approximately 23 μm in each direction. By equation $SM = D \cdot g$, if SM is set as the mean error in the vertical direction, g is approximately 3.32 bits, which is in line

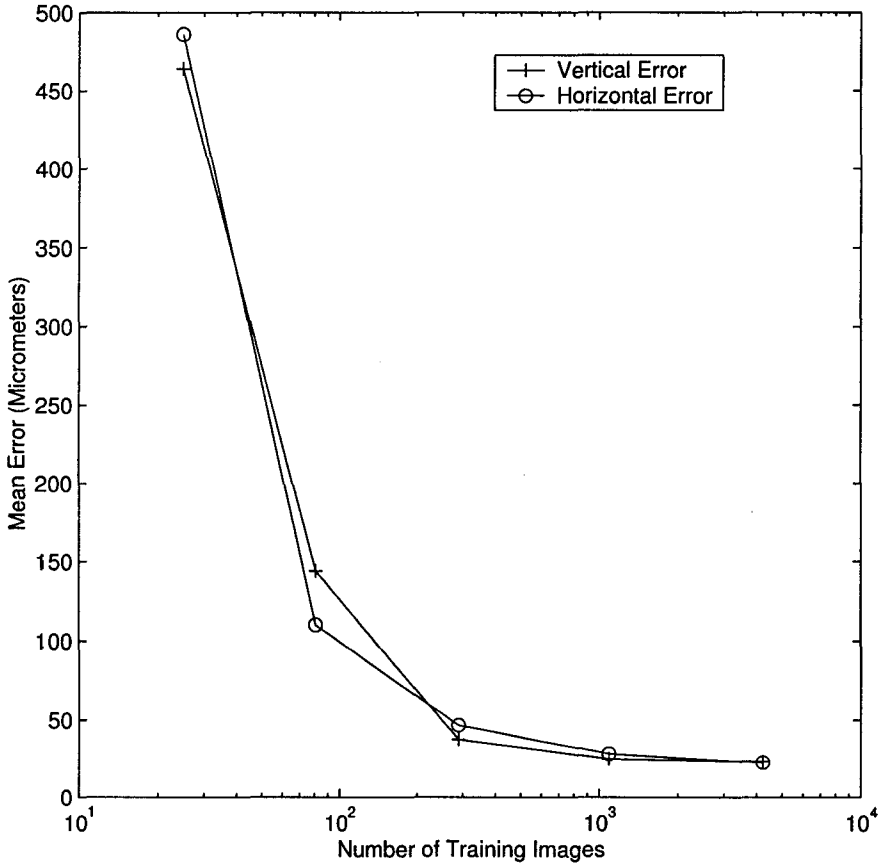

Figure 3.11: Absolute Mean Errors versus Number of Images for Projection Coefficients

with our prediction since the *SM* value is an absolute theoretical upper limit. To further increase the accuracy substantially with more images would require altering the camera optics to achieve more image feature change with the same movement or the use of a larger CCD or a CCD with a larger dynamic range.

To illustrate the effects of increasing the training images on the distribution of the error, Figure 3.12(a) and 3.12(b) shows the histograms of the absolute error for the 25 image set and Figure 3.12(c) and 3.12(d) the 1089 image set. For the smallest number of images, the distribution of the error is large and spread out. Conversely for the 1089 image results, the errors are much reduced, with the worst case errors on the order of 100 μm.

To summarize, as the number of training images satisfies the Nyquist criterion for the first several eigenvector manifolds, error improves rapidly from below this point. Further increases in training images leads to steady increases in accuracy until the limits predicted by the camera movement per pixel of image feature change guidelines are approached, whereby the increase in accuracy slows considerably.

3.5.5 Number of Eigenvectors

The number of eigenvectors used for the nearest neighbor search certainly has an effect on the accuracy achievable by eigenspace positioning. However it is important to remember its role in the nearest neighbor search algorithm and consequently in eigenspace positioning. Each eigenvector $\mathbf{e}_i$ produces a manifold of projection coefficients corresponding to the positions of the camera across the movement range. In the nearest neighbor search phase, values produced by each eigenvector are compared to the stored interpolated values and the closest Euclidean distance is used. Ideally, as previously mentioned, the interpolated projection coefficients will be as close as possible to the actual values to allow accurate recognition. In theory, position could

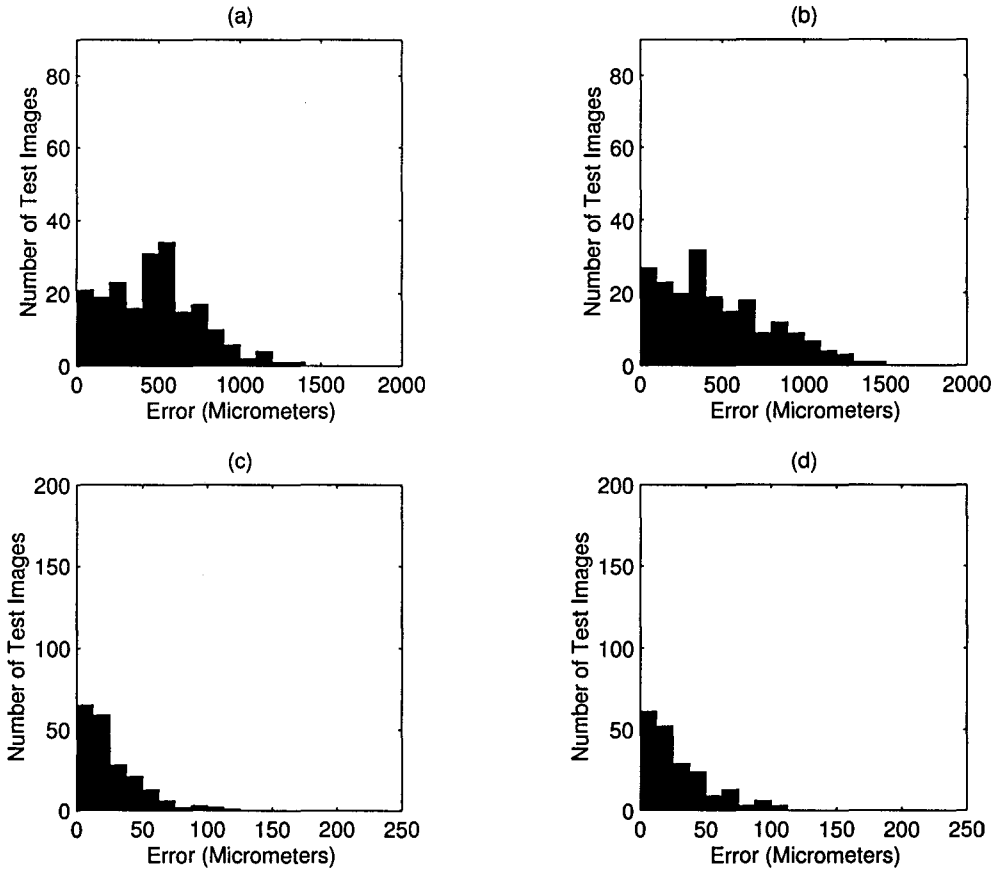

Figure 3.12: Histograms of Absolute Position Error, (a) Vertical Error, 25 Training Images, (b) Horizontal Error, 25 Training Images, (c) Vertical Error, 1089 Training Images, (d) Horizontal Error, 1089 Training Images

be determined accurately for only a few eigenvectors as long as their individual manifolds of projection coefficients were smooth as well as unambiguous (separate camera positions do not have the same approximate value of projection coefficients).

Experimentally, using only a few eigenvectors (three, four or five) generally results in accuracy substantially lower than using ten or more. However, since the eigenvectors themselves are based on optimal reconstruction rather than position determination, adding a few more eigenvectors usually results in gradually increased

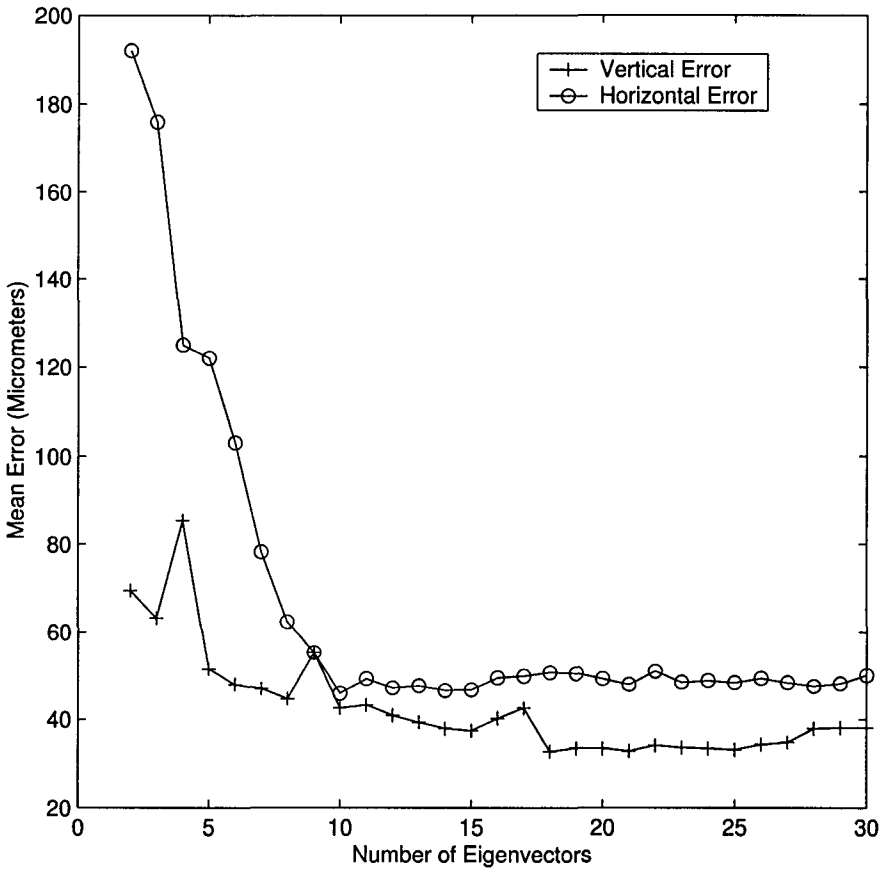

Figure 3.13: Absolute Mean Errors versus Number of Eigenvectors

accuracy. This is more than likely due to the fact that some of the eigenvectors beyond the first few incorporate reconstruction information that produces a manifold that is useful for discriminating position in certain areas of the movement range of the camera.

As an example, Figure 3.13 shows the mean error in both directions for the main experiment in this chapter with the number of eigenvectors varied from 2 to 30. The eigenvectors produced by filtering those corresponding to the 289 training image set 30 times were used for higher accuracy. The same set of 289 images were used to produce the projection coefficients and consequently the interpolated projection coefficients.

For this experiment, with only a few eigenvectors the absolute mean error was poor in both directions, but improved quickly with additional eigenvectors. Beyond ten eigenvectors, the improvement in each direction was either minor or nonexistent.

The reason the absolute mean errors improve little beyond the first few eigenvectors can be explained in two ways. First, the projection coefficients of the later eigenvectors in most cases do not vary in value as compared to that of the first few eigenvectors. This is true because the eigenvalues corresponding to the eigenvectors define the variance of the individual projection coefficients over the visual subspace as mentioned in the previous chapter:

$$\lambda_i = E[(\mathbf{x}^T\mathbf{e}_i - E[\mathbf{x}^T\mathbf{e}_i])^2], i = 1, ..., k \tag{3.33}$$

Figure 3.14 shows a plot of the eigenvalues corresponding to the visual subspace. Thus the projection coefficients of the later eigenvectors with relatively small eigenvalues will not be much of a factor for the Euclidean distance measure compared to those of the first eigenvector which will dominate the Euclidean distance measure. Conceivably for less compact visual subspaces than this one where the eigenvalues are more uniformly distributed, this would not be as large a factor.

The other reason is the nature of the projection coefficient manifolds produced by the later eigenvectors and their relationship to producing accuracy. The later eigenvectors are inevitably high frequency versions of the early eigenvectors because of the optimization for image reconstruction. Thus the manifolds produced also display high frequency characteristics. Figure 3.5 earlier in the chapter showed the manifold projection coefficients of the first four eigenvectors and Figure 3.6 which corresponds to those of the the 13th through 16th eigenvectors. Obviously the manifold of the 16th eigenvector compared to the first is much higher in frequency and more importantly, with a limited number of training projection coefficients over the movement range of

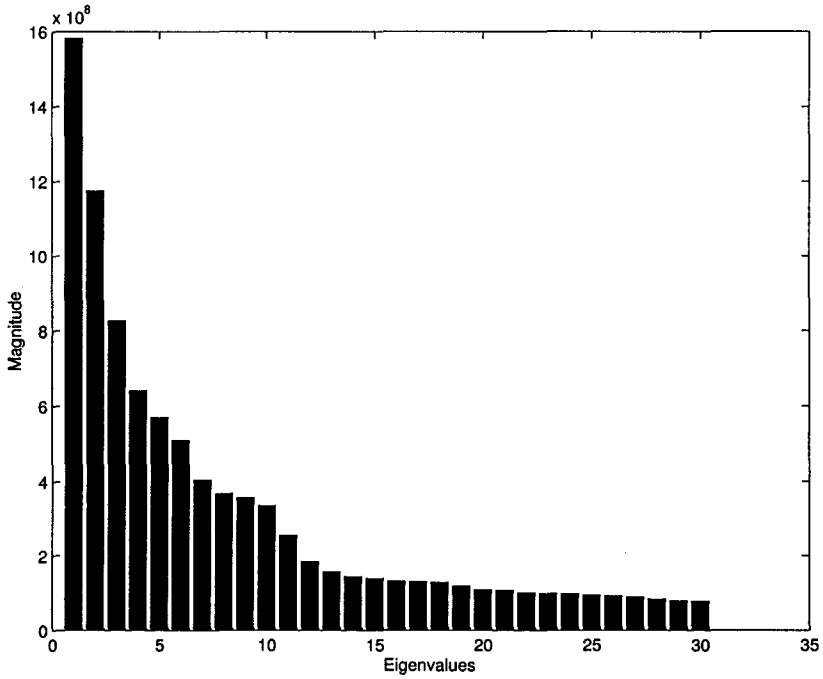

Figure 3.14: Magnitude of Eigenvalues

the camera as discussed in the previous section, the interpolations produced could be affected by aliasing error as opposed to the interpolated projection coefficients of the first eigenvector. As stated many times previously, if the interpolated projection coefficient values are inaccurate, positional accuracy will suffer. Potentially this effect could be partially ameliorated by using an increased number of actual projection coefficients to increase the sampling frequency above the Nyquist criterion for the later eigenvectors.

Thus from these results for this instance ten eigenvectors would be appropriate, however adding more does improve performance to a small degree. Considering that visual subspaces could exist that are less visually compact, a useful guide would be to error on the side of caution and use fifteen to twenty eigenvectors as the minimal standard.

3.5.6 Spline Interpolation

Another parameter available for alteration in eigenspace positioning is the type of interpolation performed. Considering the idea that accuracy can be improved by ensuring that the interpolated values are as close as possible to the actual values, the type of interpolation certainly deserves some consideration.

Linear interpolation is obviously the simplest form of interpolation and has been used in the previous experiments in this chapter. Murase and Nayar in their pioneering work on eigenspace positioning suggest using cubic spline interpolation as more suitable rather than linear interpolation. Thus for a manifold consisting of the projection coefficients for one eigenvector for a camera or object moving over a specified range, the smooth manifold should be better approximated in the intermediate positions with spline interpolation than with linear and we would expect positional accuracy to improve.

One consideration for the effect of spline interpolation on positional accuracy is the factors considered previously with regards to the number of images used to form the projection coefficients. As was seen in that section, the shapes of the manifolds formed by the projection coefficients for each eigenvector versus camera position were of increasing frequency. Consequently if the number of training image samples in each direction was too low with regards to these frequencies interpolation error could occur due to aliasing and thus increased positional error resulted. For spline interpolation the same problems can occur with erroneous interpolation if the training image sampling rate is too low, especially for the later eigenvectors with high underlying frequencies and could limit the benefits of spline interpolation over linear interpolation.

To quantitatively illustrate the effects of varying the type of interpolation, the main experiment was performed with altered interpolation. The filtered eigenvectors (30 times) from 289 training images were used together with the projection coefficients

of the 289 training images. The standard 15 eigenvectors filtered 30 times were used. Straight linear interpolation was used via Matlab as well as the standard piecewise cubic spline interpolation function provided by Matlab. The absolute mean errors for the linear interpolation were 37.4 μm in the vertical direction and 46.7 μm in the horizontal direction. The absolute mean errors for the cubic spline interpolation were 33.6 μm in the vertical direction and 42.6 μm in the horizontal direction. Figure 3.15 shows the histogram of the absolute errors for both the linear and cubic spline interpolation. The absolute mean errors are reduced approximately by 10 % in both directions for spline interpolation from the linear results. However examining the histograms of the absolute error, the worst case error in the horizontal direction is slightly increased for the spline interpolation, but reduced in the vertical direction.

Thus for this example spline interpolation provides improved performance, but compared to the differences that can be achieved with varying the number of training images for both forming the eigenvectors and the projection coefficients the difference is relatively minor. Nonetheless a good topic for future investigation could include a significantly more exhaustive study of the effects of different types of spline interpolation for maximizing accuracy performance once the other factors are sufficiently addressed. This would be especially valid for situations where a large number of training images are available to allow the splines to capture the manifold shapes without the possibility of aliasing error.

3.6 Chapter Summary

This chapter illustrated the main factors that affect eigenspace positioning accuracy. The relationship between the amount of image feature change over the movement range of a camera and the resultant accuracy possible was demonstrated. This

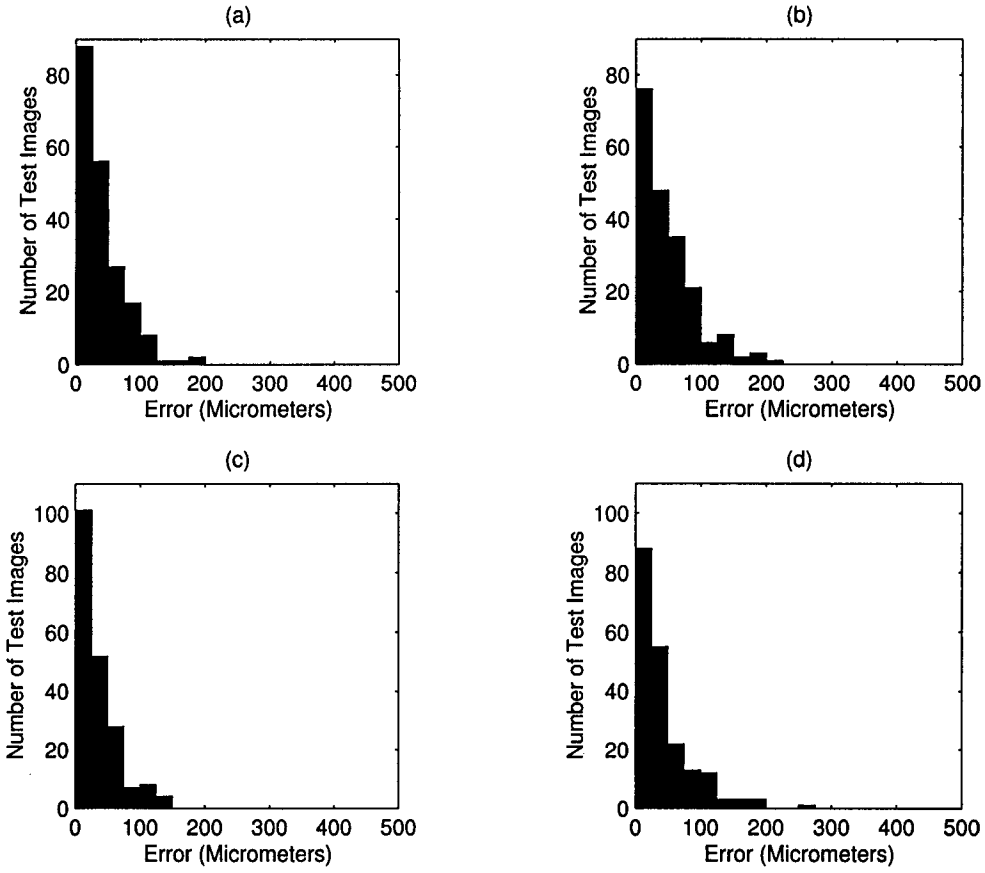

Figure 3.15: Histograms of Position Error, (a) Vertical, Linear Interpolation, (b) Horizontal, Linear Interpolation, (c) Vertical, Cubic Spline Interpolation, (d) Horizontal, Cubic Spline Interpolation

has particular implications for 3 dimensional eigenspace positioning and is used extensively in Chapter 6 for performing 3 dimensional positioning with multiple cameras.

In terms of varying the eigenspace parameters, their effect can be substantial, but is ultimately limited by the amount of image feature change over the movement range of the camera. Increasing the number of images used for the projection coefficients is obviously of importance for positional accuracy; initially increasing it rapidly improves performance as the interpolated projection coefficients more closely resemble that of the actual manifold via the avoidance of aliasing error until the limits of pixel

change start to become apparent.

For the number of images used to form the eigenvectors the effect of increasing the number of images was similar; initially increasing the number of images led to a rapid improvement while eventually a limit was reached. The eigenvector smoothing operation was shown to be of use for increasing the accuracy with a limited number of training images by removing artifacts from the implicit covariance matrix produced eigenvectors.

The effect on accuracy for the number of eigenvectors used in the nearest neighbor search was such that once the projection manifolds were of high frequency compared to the density of the sampling of the projections, little improvement was gained. Using spline interpolation rather than linear interpolation resulted in a small increase in accuracy compared to varying the number of training images.

Chapter 4

Occlusion Robust Eigenspace Positioning

4.1 Occlusion and Eigenspace Positioning Methods in the Presence of Occlusion

Occlusion presents difficulty for all eigenspace methods in computer vision, including their use for positioning applications. Possible occlusion scenarios for positioning are easily imagined. For visual servoing applications such as moving a part, a robot gripper could be a potential occlusion. For mobile robot navigation, possible occlusions include people moving into the view of a camera or also scene changes over time, such as moving a chair or a desk. Thus occlusion situations could be quite common and robust techniques are necessary for implementing eigenspace positioning in actual applications.

Thus this chapter is based around occlusion and eigenspace methods. The first portion discusses how occlusions affect positional accuracy and what factors affect the degree to which occlusion negatively impedes accuracy. This is followed by a

discussion of global versus local information as a possible means of solving the occlusion problem for eigenspace positioning. Two methods incorporating the use of local information are then described: first separate eigenspaces which treat each image image section as a separate eigenspace to isolate occlusions, followed by an improved method, subsectioning and recombination that isolates occlusions and avoids ambiguity induced errors that can occur with the separate eigenspaces method. Note that these techniques have been previously described by Quick and Capson [42][43] prior to the writing of this thesis.

The next section introduces an experimental example used throughout the next two chapters to facilitate the understanding of occlusion and eigenspace methods.

4.2 Experimental Example

To demonstrate the effects of occlusion on eigenspace positioning accuracy and methods to ameliorate the negative effect of occlusion, a simple example simulating an industrial assembly visual servoing problem was chosen. Two metal parts were placed close together, with one fixed and the position of the other controlled via an XY table. The task was to determine the planar position of the XY table controlled part over a small range to simulate proper placement for the task of mating the two parts. In this case the camera was fixed over the two parts, thus rather than a camera positioning problem (as in the previous chapter) it is instead an object positioning problem. The focus of these experiments was to perform accurate final positioning, thus the range of the movement of the part was confined to a 4 mm by 4 mm plane.

Figure 4.1 shows the four images corresponding to the corner positions of the movable object. The four images represent the images from the camera at the four extreme corner positions. The camera lens was located approximately 20 cm from the surface of the objects. Over the 4 mm by 4 mm movement range 289 training

images were acquired; thus each image was separated by 0.25 mm in both directions. These images were used to both derive the eigenspace and to produce a set of projection coefficients. These projection coefficients were linearly interpolated to provide 24 positions between training images in each direction for a total of 401 by 401 linearly interpolated projection coefficients (spaced 10 μm apart). Unless explicitly mentioned, 15 eigenvectors were used to produce the projection coefficients for the experiments in this section.

The image feature change resulting from the movement range of the object was restricted in these experiments due to the small distance the object movement was limited to. In each direction, the image feature change over the movement range of the part was limited to 20 pixels in each direction. No eigenvector filtering was performed for the experimental results in this chapter as with the small amount of image feature change in this example, the eigenvectors were sufficiently smooth with 289 training images and thus filtering did not improve performance.

As in the previous chapters, a testing set of 100 random images equally spaced throughout the movement range of the part were acquired. Absolute error for each image was reported as the difference between the correct position and the returned position, separated into two directions. Note that for these experiments, the motion of the part that goes from the top of the image towards the bottom will be referred to as the vertical error and the error in the other direction will be referred to as the horizontal error. Figure 4.2(a) shows the histogram of the absolute error in the vertical direction for the testing set; Figure 4.2(b) shows the histogram of the absolute error in the horizontal direction (for the rest of the chapter, the vertical error will always be on the left). The absolute mean error in the vertical direction was 10.2 μm (4.02 % of the distance between training images in this direction) and in the horizontal direction it was 21.3 μm (8.52 % of the distance between training images). Note that for these particular experiments, an error of zero is possible as the interpolated

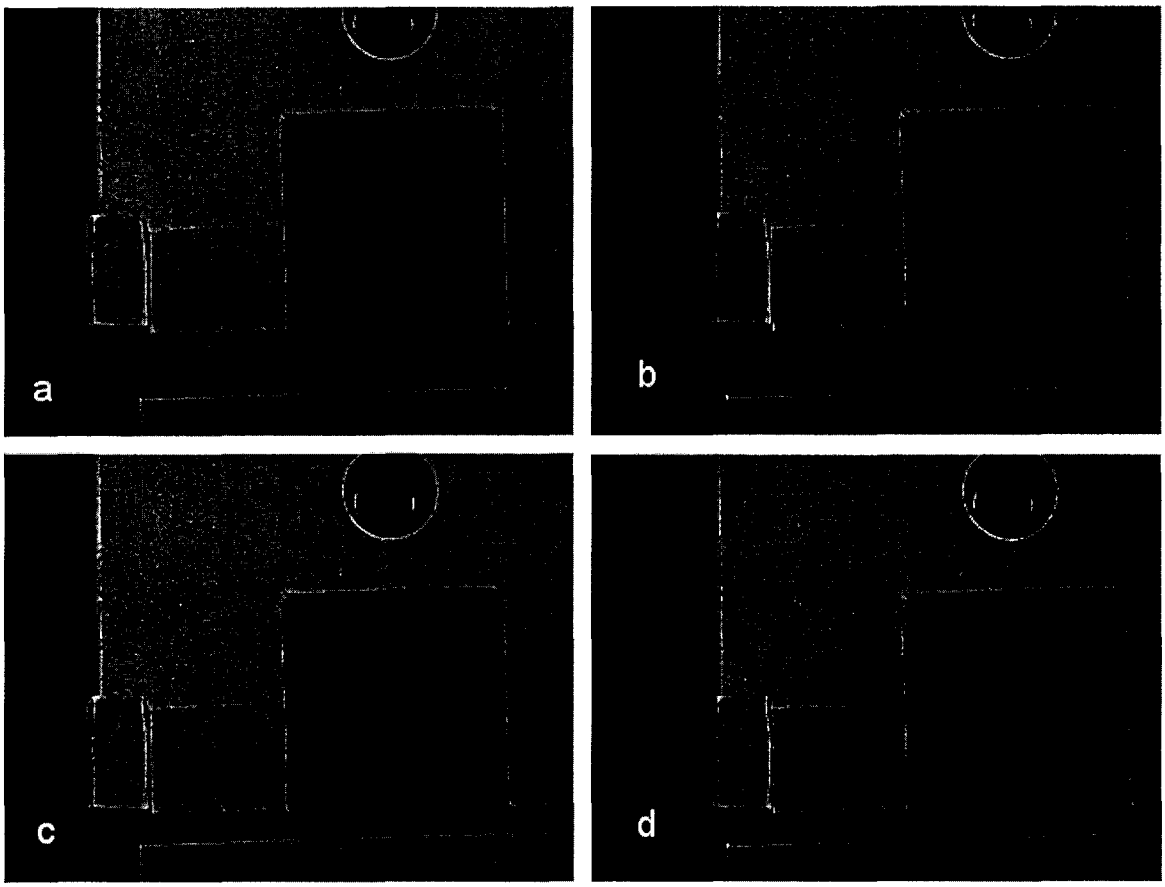

Figure 4.1: Images from Extremes of Camera Motion Range

projection coefficients positions coincide with the positions of the random images (due to the XY table being limited to steps of 10 μm), so the actual absolute mean error would be somewhat higher. The absolute mean errors and histograms illustrate that eigenspace methods can be used to achieve accuracy levels for precision assembly tasks without need of elaborate algorithms or complex camera arrangements.

Figure 4.3 shows the first four visualized eigenvector for the visual subspace. Note that for this visual subspace, since the number of pixels that actually changes is quite limited, it is relatively easy to see the relations between the eigenvectors and the visual characteristics of the moving metal object. Namely the edges of the object with the background are the emphasized features. Considering that the eigenvectors

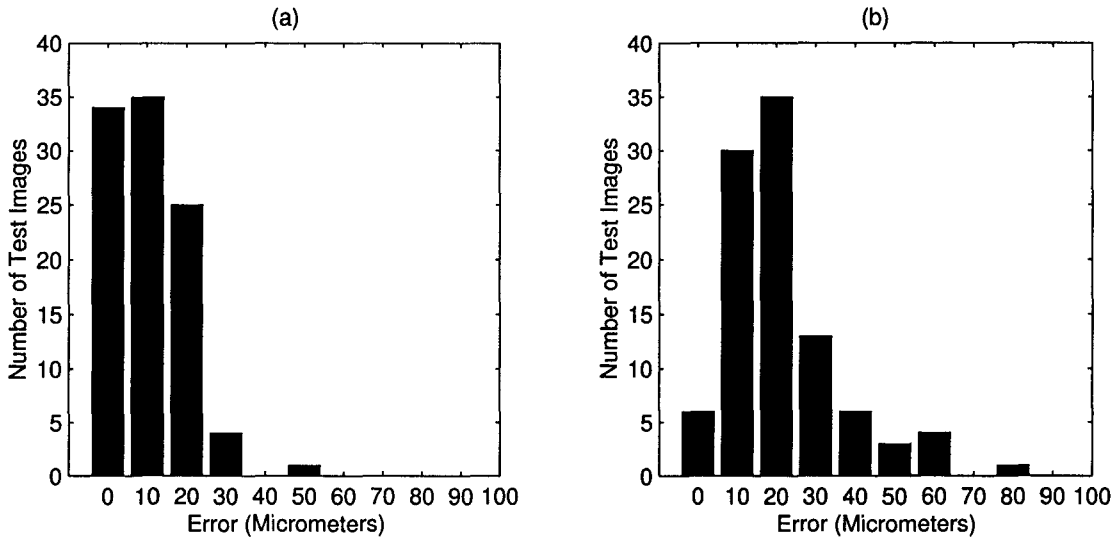

Figure 4.2: Histograms of Absolute Position Error, (a) Vertical, (b) Horizontal

minimize the reconstruction error this is not surprising for these experiments, since most of the variance of the pixels within the training images occurs in these border areas and thus would be the focus of the reconstruction, as opposed to the large areas where the appearance does not vary despite the position of the metal object. This experimental example will be used extensively in this chapter to demonstrate the effects of occlusion on accuracy and to test methods for finding position accurately despite occlusion.

4.3 Occlusion Effects on Positional Accuracy

Occlusions and scene changes to the visual subspace that is used for training can have negative effects on the accuracy achieved by eigenspace methods for determining position. It should be noted that the effects of an occlusion are somewhat unpredictable in terms of the degree to which accuracy is effected, depending on the visual subspace and the occlusion. For a particular occlusion several factors can increase the impairment of positional accuracy, however the exact quantitative effects can only

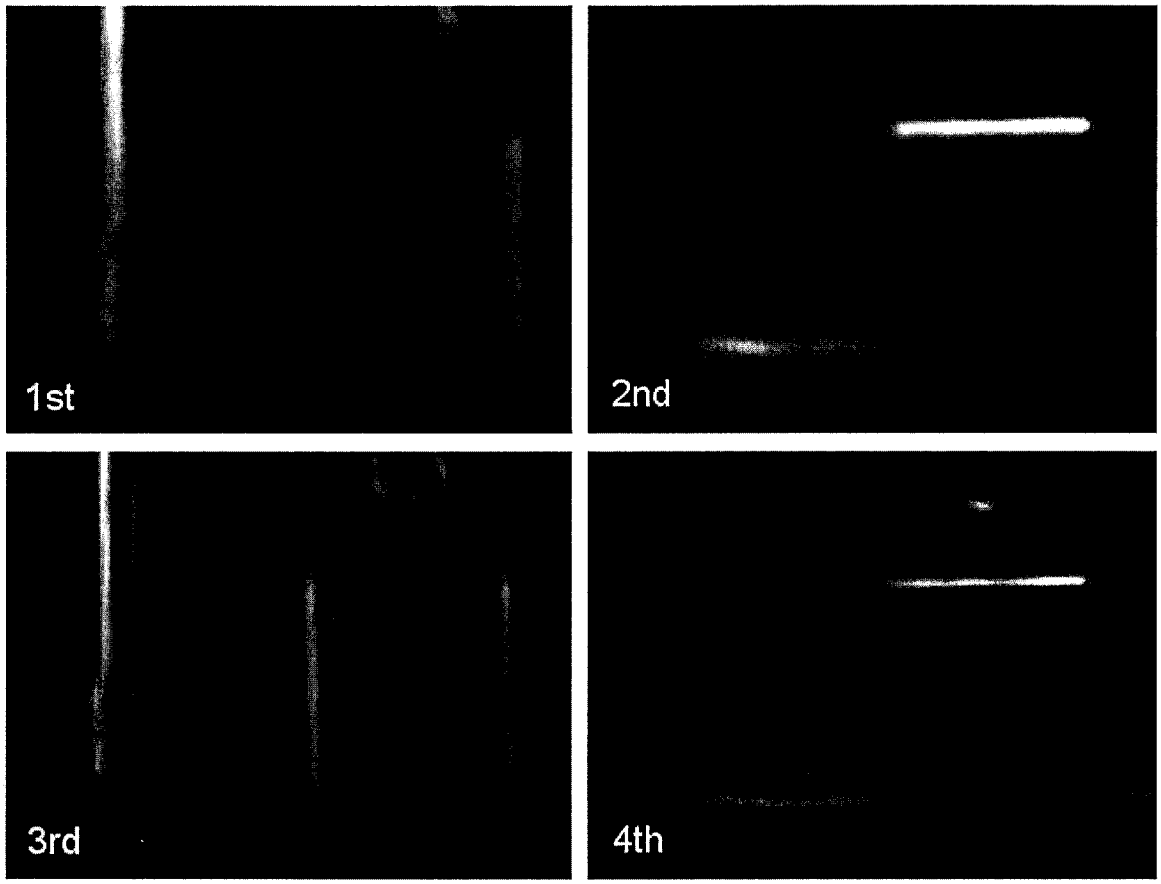

Figure 4.3: First Four Eigenvectors Visualized

be discovered through actual testing of images from random images throughout the subspace.

The factors affecting the severity of the negative effect of an occlusion include the size of the occlusion (in proportion to the size of the image), the magnitude of the difference between the appearance of the occlusion and that of the original background as well as in some cases the location of the occlusion within the image.

When considering occlusions and the effect on accuracy of the aforementioned factors, it is important to remember the nature of the basic eigenspace method. The basic eigenspace method is global in the sense that the eigenvectors $\mathbf{e}_i$ used to form the projection coefficients $\mathbf{y}$ respond to the entire image $\mathbf{x}$. Thus an occlusion located

in any local section a of the image can potentially affect the global projection coefficients formed from the inner product of each eigenvector and the image. Note that considered in this way, the basic eigenspace method stores global image information in the low dimensional projection coefficients.

Consider an unoccluded image $\mathbf{x}$ and an occluded image $\mathbf{x}_{occluded}$ from the same camera or object position. Each will have projection coefficients $\mathbf{y}$ and $\mathbf{y}_{occluded}$ associated with them via the eigenvectors of the visual subspace according to equation 2.17. Due to the occlusion, $\mathbf{y}$ and $\mathbf{y}_{occluded}$ will inevitably be different and this difference can manifest itself in positional errors in the nearest neighbor search for position (equation 2.19).

Since how the projection coefficients are altered from the original unoccluded case is somewhat random, it is quite likely $\mathbf{y}_{occluded}$ will be closest to the stored projection coefficients corresponding to an incorrect position, rather than the correct one. Larger occlusions, as we will show experimentally, will in general lead to larger differences between the occluded and unoccluded projection coefficients and thus potentially larger errors.

An important issue to consider for how occlusion effects eigenspace positioning methods, is not so much how the mean positional error is effected, but the potential for much larger errors. With the changes that can occur to the projection coefficients for an occluded image, there is always the possibility that during the nearest neighbor search, projection coefficients corresponding to a position far from the actual position could be the closest match leading to a catastrophically large error. As shall be seen from our experiments, occlusions usually do not alter the absolute mean error to an extreme extent, however there is a probability of much larger worst case errors that is not present with the unoccluded case. Obviously for applications such as using the positioning methods for manufacturing applications, such large errors cannot be tolerated. Thus any solution to the occlusion problem should not only reduce the

mean error to similar levels of the unoccluded case, but also reduce the possibility of these large errors.

The following sections provide experimental evidence to illustrate the general effects on accuracy for several different occlusion factors, namely size, degree of difference from the background and the location of the occlusion within an image as well as the fact that the exact effects on positional accuracy are unpredictable.

4.3.1 Occlusion Size

For an occluded image, as the proportion the occlusion occupies of the image increases, the projection coefficients will diverge increasingly with those of the unoccluded images. Thus as more and more pixels differ from the original background pixels, the difference between the unoccluded projection coefficients and the occluded projection coefficients increases. To illustrate this factor experimentally, we embed artificial occlusions into the visual subspace corresponding to the moving metallic part described earlier in the chapter. For simplicity and for ease of comparison, the choice for the artificial occlusion was zero intensity squares located at the center of the images (as illustrated in Figure 4.4).

For testing purposes, the original random test set was embedded with different size squares, to produce image sets ranging from squares of size 10 pixels per side to squares of 100 pixels per side. Figure 4.5 shows the absolute mean error values for each direction of object movement for the various sizes of the squares.

Interestingly, the error does indeed increase with increasing occlusion size in both directions, but not with any clear relationships. Indeed, initially in the vertical direction there is almost no difference between the unoccluded absolute mean error 10.2 μm, and the occluded error. This illustrates the concept, that although generally error increases with increasing occlusion size it can remain dependent on other factors

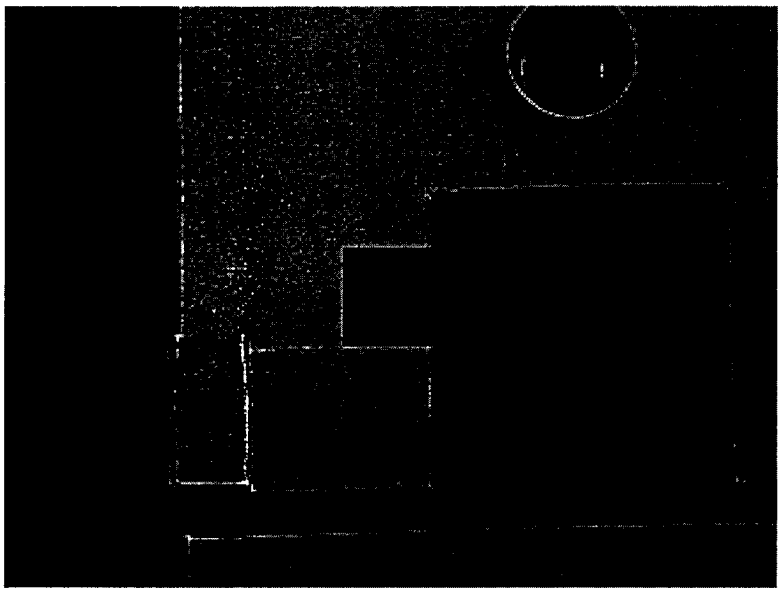

Figure 4.4: Example of Artificial Occlusion (30 Pixel Square)

such as the location of the occlusion and the appearance of the occlusion to determine to which extent the mean error is affected. For this particular experiment, the differentiation between the absolute mean error in each direction of object movement can be attributed to the fact that at the central location of the squares, the smaller squares fall only on a vertical edge of the part, thus are much more likely to alter error in one direction (horizontally) as opposed to the other. An important idea also is that for small occlusions proportionately ($< 1\%$), there is almost no difference in terms of position error. Thus proportionally small occlusions can be disregarded as an impediment to position accuracy.

As an example of the possible effects of occlusion on the distribution of the positional error, Figure 4.6 shows the histogram of the absolute error for the 50 pixel square occlusion test set. Note the increased worst case error in the vertical direction compared to the unoccluded case in Figure 4.2.

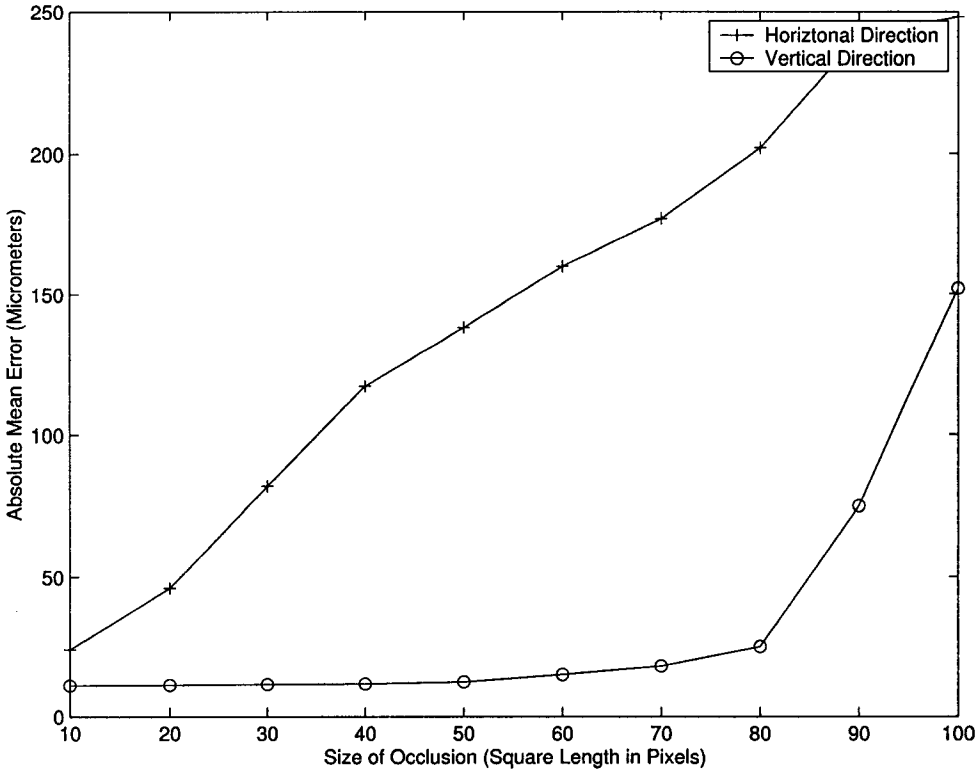

Figure 4.5: Occlusion Size Versus Absolute Mean Error

4.3.2 Appearance of Occlusion versus the Background

Another major factor in how occlusion affects positional accuracy is the difference between the values of the occluded pixels and their original unoccluded values. As the difference increases, the difference between the unoccluded and occluded projection coefficients should increase as well, leading to a decrease in accuracy.

To illustrate this concept experimentally, two more test sets were created by embedding squares with maximal intensity (pixel value of 255) as well as embedding squares of half intensity (pixel value 127) in the center of the images. For comparison purposes, the size of the squares for the two sets was chosen as 50 pixels per side.

Figure 4.7 shows the error histogram for the maximal intensity occlusion, while Figure 4.8 shows the absolute error histogram for the half intensity occlusion. The

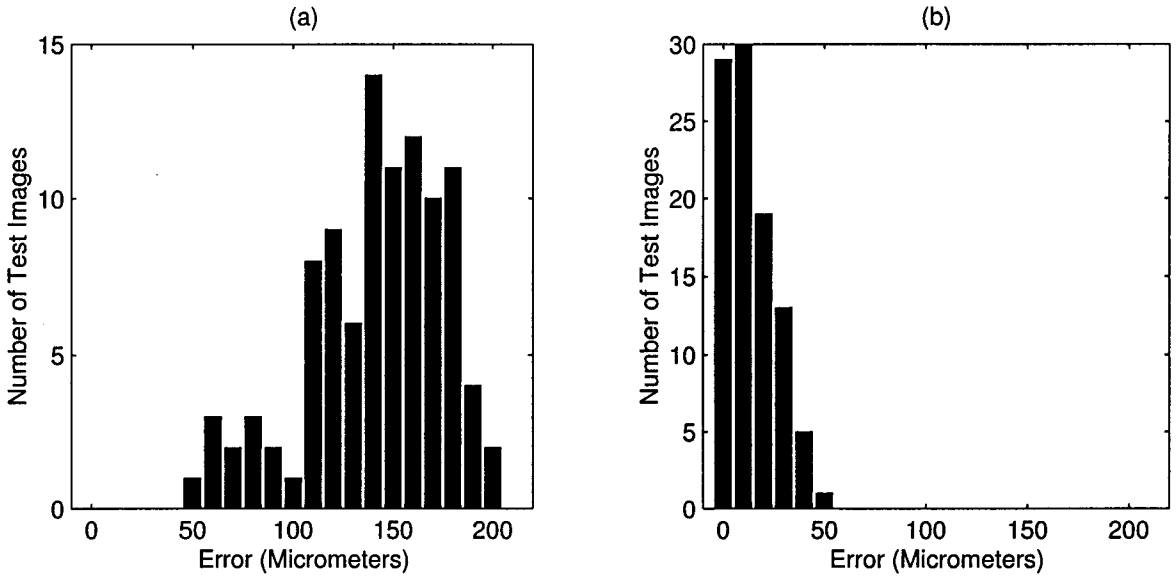

Figure 4.6: Histogram of Absolute Positional Error, Fifty Pixel Square Occlusion, (a) Vertical, (b) Horizontal

maximal intensity occlusion positional errors are higher, both for the mean as well as the worst case errors.

In terms of pixel value, the half intensity occlusion pixels are closer in value to the original pixel values than those of the maximal intensity occlusion, resulting in the lower errors. The half intensity pixel values are also closer in value to the original pixels than the zero intensity occlusions used for the results in Figure 4.6. Comparing Figure 4.6 and Figure 4.8, the half-intensity error is less than the full intensity error.

Thus when considering the effect for a given occlusion, besides size, the possibilities of the effect of differing appearances must also be considered. Conceivably for a given size of occlusion, with the right appearance an occlusion could have a large effect.

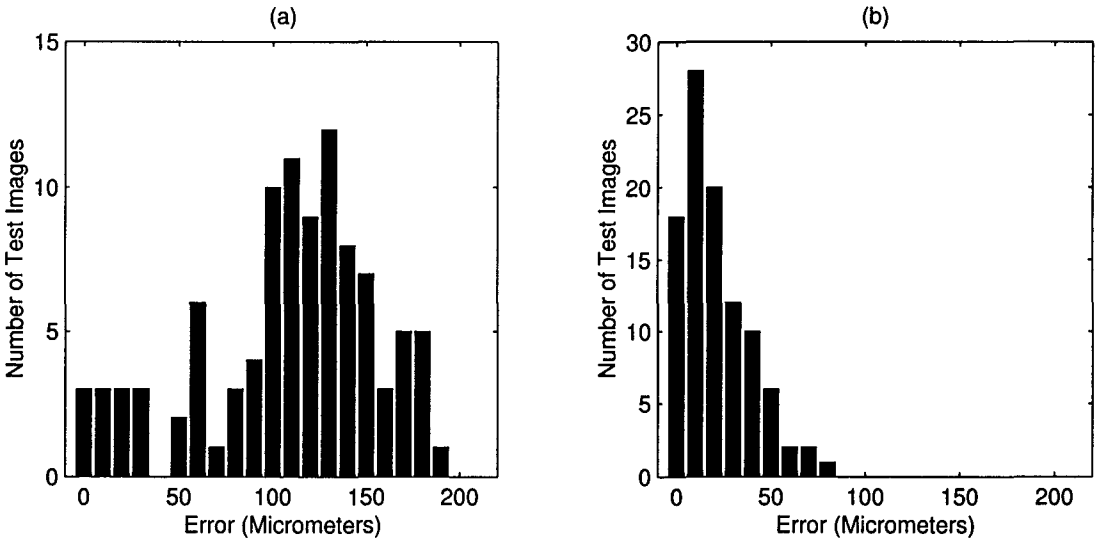

Figure 4.7: Histogram of Absolute Position Error, Maximal Intensity Occlusion (50 Pixels), (a) Vertical, (b) Horizontal

4.3.3 Location of Occlusion

The location of the occlusion can affect the change on the projection coefficients, as certain regions of the image can be more sensitive to change due to the characteristics of the visual subspace. For instance, for the visual subspace derived from a mobile robot moving throughout a large room with many different features, it is likely the eigenvectors will be responsive throughout the entire image. However for servoing in a small range relative to a metallic part with few features, the eigenvectors will only respond to those areas where the features move as in the example for this chapter. Thus if an occlusion occurs in a featureless area, the projection coefficients will be minimally effected, whereas if it occurs in the featured area, the projection coefficients could be radically changed compared to the same occlusion in the mobile robot subspace.

As an example of such phenomenon, rather than the center, an alternative location was chosen to embed an artificial occlusion of zero intensity squares with 50 pixels per

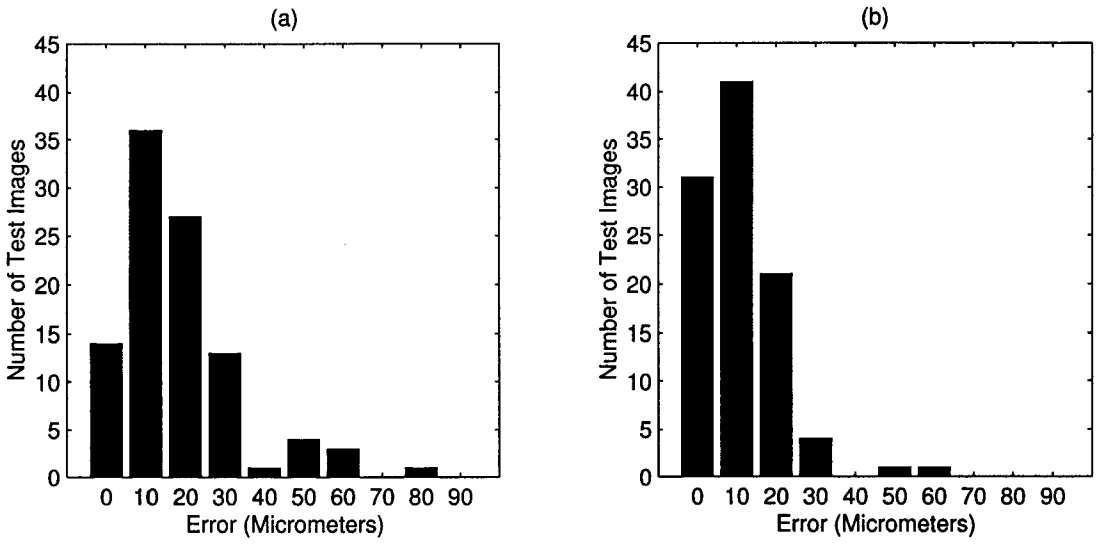

Figure 4.8: Histogram of Absolute Position Error, Half Intensity Occlusion (50 Pixels), (a) Vertical, (b) Horizontal

square size as illustrated in Figure 4.9. Note that this location has both horizontal and vertical edges of the moving metallic object located within the occlusion, rather than just a vertical edge for the original occlusion location.

Figure 4.10 shows the histogram of the absolute error for the 50 pixel square in the new location. Interestingly, the error in the horizontal direction (Figure 4.10(b)) is significantly increased compared to the results for the original centered location of the occlusion.

In general, for applications such as providing navigation data to a mobile robot where the visual subspace is such that the visual features are spread relatively evenly throughout the image, the location of the occlusion is not a major factor. Conversely, for visual servoing tasks over a small range, it is much more likely that the position of the occlusion is a factor.

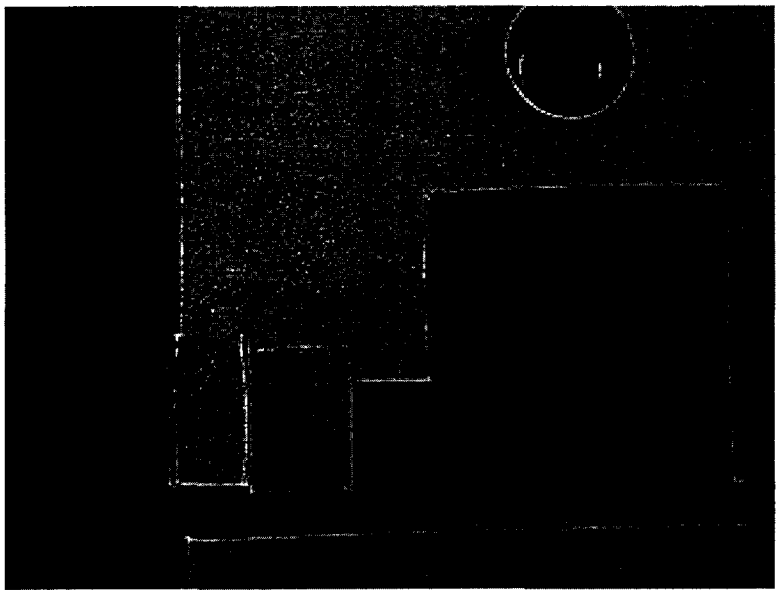

Figure 4.9: Image with New Location of Occlusion

4.3.4 Local versus Global Information

In general, basic eigenspace techniques for positioning are not robust to occlusions due to the fact the stored projection coefficients represent global information with regards to the images, in the sense that no local information corresponding to specific portions of the image is preserved. This is a useful abstraction, in the sense that no work needs to be done to look for specific local features to perform pose analysis.

However, occlusions that will be encountered in applications are generally local in nature, in the sense that they will consist of connected regions spaced throughout the image. For a metallic part being servoed, a robotic gripper grasping a section of the part could be considered a locally confined occlusion. For a mobile robot, a person walking in front of the camera is another local occlusion. In both these examples, only a portion of the input image is changed from the original unoccluded images acquired during the training phase for the eigenspaces, yet due to the global nature of the basic eigenspace method, the projection coefficients will be changed, with the

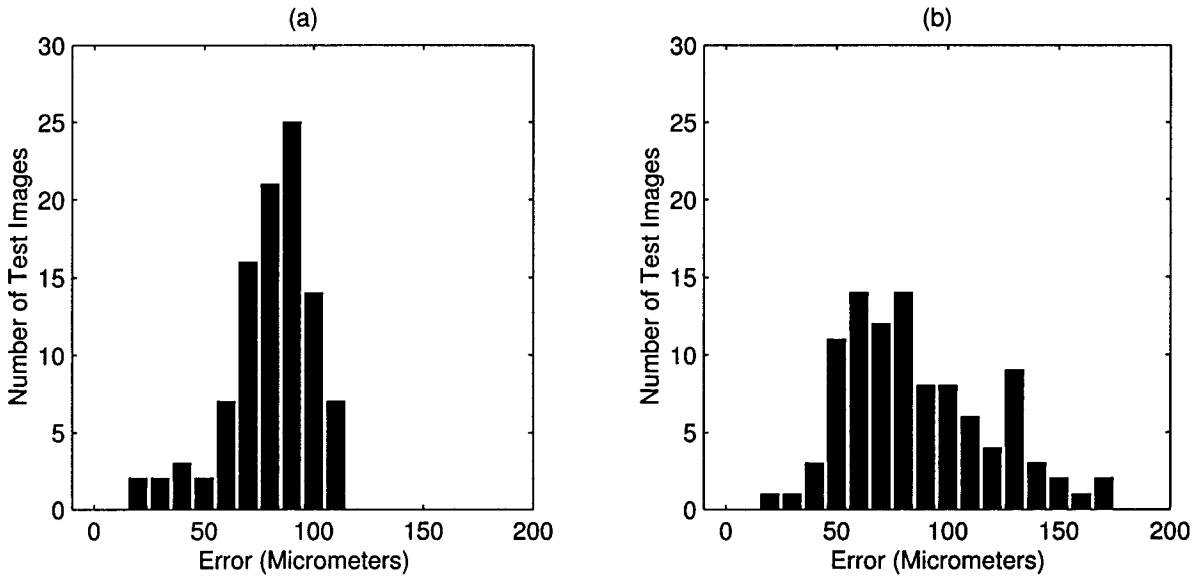

Figure 4.10: Histogram of Absolute Position Error, New Location of Occlusion, (a) Vertical, (b) Horizontal

possibility of positional error.

As a way to make eigenspace positioning systems more robust to occlusions, we propose methods of storing additional local information, such that a local occlusion confined to one area of an image will not negatively affect the performance of position estimation. We propose dividing the input image into separate subsections and storing projection coefficient information for each local section, rather than only storing global projection coefficients. Figure 4.11 shows an example subdivision of an input image for the metallic object visual subspace. These methods depend on the visual subspace having enough visual features, such that the local information is redundant enough in terms of features to perform position determination accurately.

Potentially the shape of the image subsections could be considered arbitrary. However since occlusions are generally unpredictable in terms of size and morphology, the shape of the subsections should be chosen such that they are reasonably compact

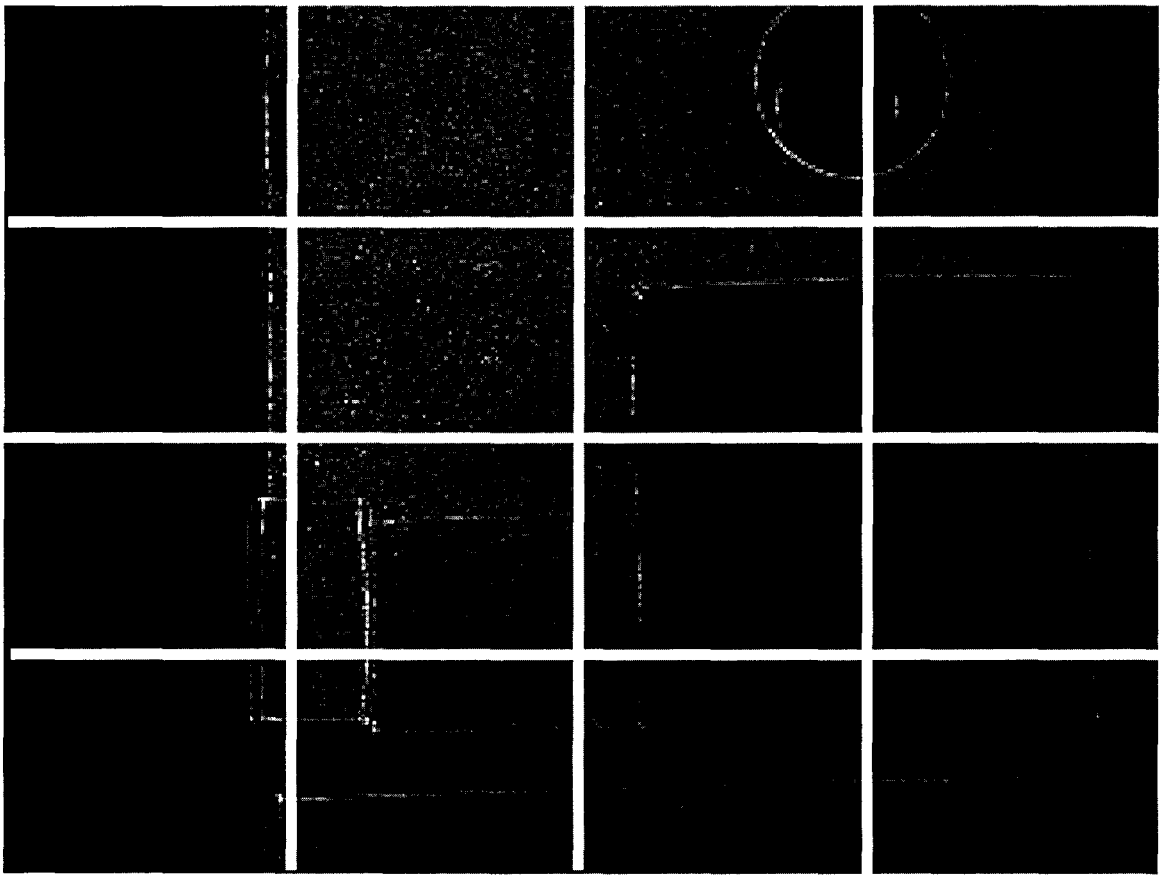

Figure 4.11: Subdivision of Images

about the centroid. Since the aim is to have as many local sections as possible without occlusions, having long strips as the section shape could result in an elongated occlusion in the opposite direction of the subsections occupying a large number of subsections. For our work, the shapes of the subsections were chosen to be rectangles with the same aspect ratio as the camera.

By storing local information for each section, occlusions will only affect the projection coefficient information for those sections in which they occur. The remaining sections with their unaltered local information can then be used to determine the position of the camera accurately.

Determining which sections are occluded is not a trivial problem and is discussed

in depth in the following chapter. A basic method to detect occlusions in a section is to compare the eigenspace reconstruction of the section with that of the original image. Images with sufficiently large occlusions will in general have a larger reconstruction error, allowing a threshold to be developed for separating occluded and unoccluded image sections.

Assuming an adequate method to differentiate between occluded and unoccluded sections, we propose two methods that can be used to utilize local information to perform accurate position determination in the presence of occlusions. These are:

- Separate Eigenspaces
- Subsectioning and Recombination

Separate eigenspaces is the simplest method of utilizing local information as a redundant technique for finding position, but suffers with problems with ambiguity. Subsectioning and recombination is more complex but is more redundant to ambiguity.

4.3.5 Separate Eigenspaces

The separate eigenspaces method is the simplest conceptually for handling occlusions using local information, but has some significant drawbacks. In essence, the images are divided into sections to preserve local information. Each section is subsequently considered independent of the others. Thus each image section is treated as if it was an individual image. Separate eigenspace analysis is then performed for each image section j with a separate set of eigenvectors for each:

$$\mathbf{E}^j = [\mathbf{e}_1^j \dots \mathbf{e}_k^j] \tag{4.34}$$

Similarly for each section j, a separate set of projection coefficients is stored:

$$\mathbf{Y}^j = [\mathbf{y}_1^j \dots \mathbf{y}_m^j] \tag{4.35}$$

This is the local information for each section. We can then proceed to perform a separate nearest neighbor search for each section to find the stored projection coefficients closest to those of the section j of the new image $\mathbf{y}_{new}$:

$$\text{Nearest Neighbor } nn^j = arg\min \|\mathbf{y}_{new}^j - \mathbf{y}_i^j\|, \forall i = 1, \dots, m \tag{4.36}$$

This allows the determination of an independent camera or object position for each section. Occluded sections will have suspect information that should obviously be avoided. The occlusion detection techniques described in the next chapter can be applied to evaluate whether an individual section is occluded or not. The remaining unoccluded sections can then be used to determine the overall position by averaging their respective positions.

This technique depends on the assumption that in most visual subspaces there is significant redundancy across the entire image in terms of salient features such that the position determined by each individual section is similar in accuracy to that of the global image. Whether that is true or not depends on the visual characteristics of the subspace and the number of image sections. Experimentally, it has been found that often times the accuracy for a subsection can be poor compared to the performance for an eigenspace based upon the entire image.

The main factor involved that decreases performance for individual sections is image ambiguity. In general, image ambiguity can occur even when using the basic eigenspace method for determining the position of a camera or object with the entire

image. Essentially image ambiguity occurs when the appearance of images over a range of camera or object motion is similar (beyond the normal correlation similarity that occurs for a visual subspace as the camera or object is moved). For such situations, the projection coefficients over this range of camera movement will also be similar, ensuring that for the nearest neighbor matching process significant error can be introduced since an image taken from within this range could be matched with virtually any position within the similar range. Note that for eigenspaces that encompass multidimensional movement, ambiguity could be present in terms of determining position for one direction, but not for the others. Ambiguity depends on a lack of salient features that change in appearance as the position of the camera is moved. For global images, a mobile robot equipped with a camera moving down a long hallway with smooth featureless walls would be an example of a subspace that had significant ambiguity. In such cases there is little that can be done with that information, since the subspace is inherently ambiguous.

For separate eigenspaces, ambiguity is an even larger problem. Even when the entire image contains visual features such that there is no ambiguity across the entire images, individual image sections can still be ambiguous. Thus for a mobile robot operating within a laboratory, a view of a wall could be unambiguous as the robot moves, however one image subsection could contain a section of blank wall over the robot movement. Similarly for a camera servoing accurately relative to a metallic part, some image sections could be ambiguous where there are no representative features.

Increasing the number of image sections enhances the possibility that ambiguous sections will occur, since it is more likely that for smaller image sections, that a featureless area will occur. Thus for the separate eigenspaces technique, it is important that the number of sections be such that ambiguity is not likely to occur for a given subspace. In our experiments, smaller sections can result in extremely large errors.

It should also be noted that ambiguity is such a problem because it is difficult to know when a section will present ambiguous information without rigorous testing of the range of motion of the camera.

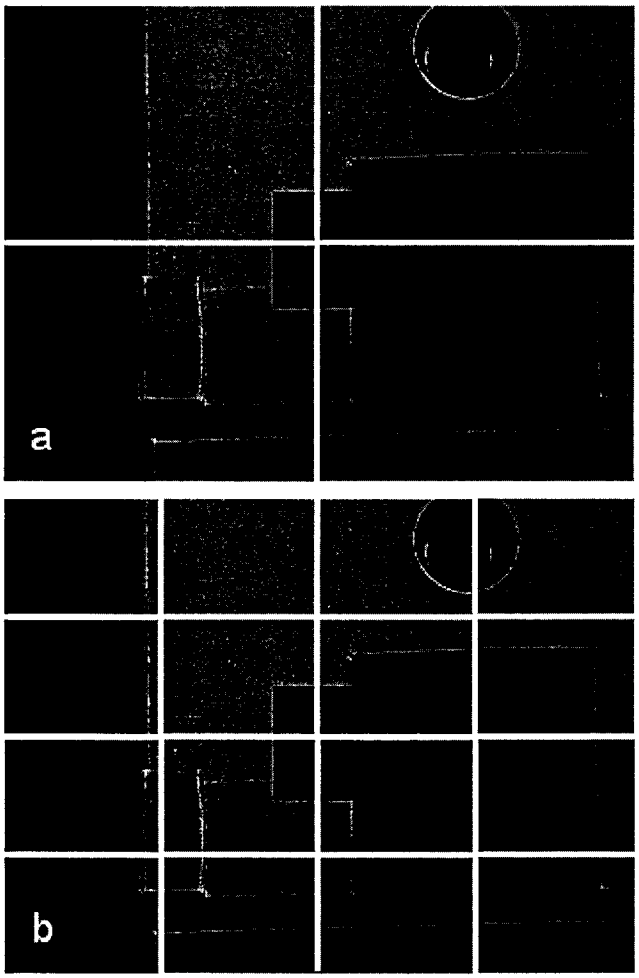

Figure 4.12: Occluded Image (a) Four Divisions, (b) Sixteen Divisions

An argument for using a large number of sections is to maximize the amount of unoccluded information from an image to be included in the position determination phase. Using more sections will ensure a given occlusion will not remove a large portion of the image from consideration. Figure 4.12 shows an image from the metallic object visual subspace with an artificial occlusion located at the center overlaid with

two different section divisions. Using only four sections unfortunately means that all four sections are occluded, ensuring 100 % of the image will not be included, thus no accurate positioning can be performed. Conversely, using sixteen sections ensures that 75 % of the image sections can be used for occlusion measurement. Thus an argument for using more image subsections is that using too few can mean that a well placed occlusion could disallow too many sections from being included in the position measurement phase.

4.3.6 Experimental Results

To illustrate the use of the separate eigenspaces technique and its limitations with regard to image ambiguity, several experiments were performed with the metallic object visual subspace. The standard setup was used as described in earlier in the chapter, with regard to the number of training images and projection coefficients, used for each section.

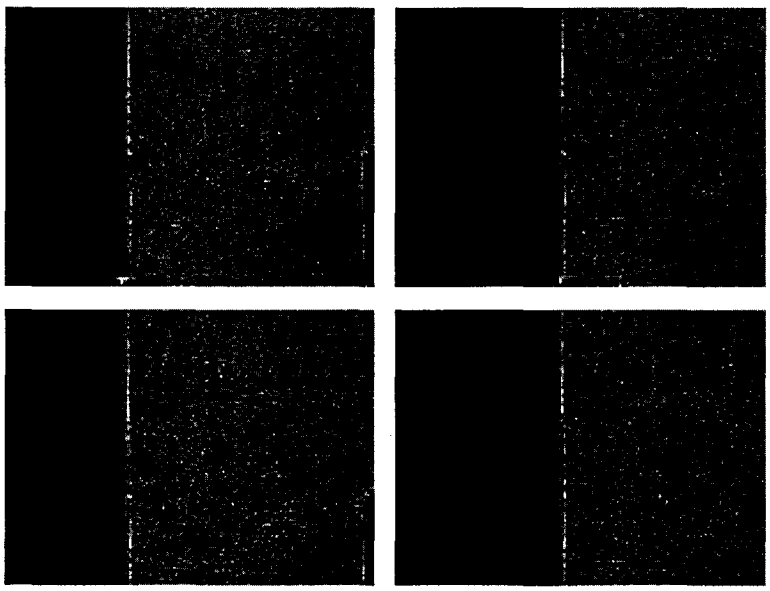

Figure 4.13: Corner Images of Camera Motion, Upper Left Section

For the first experiment, the image was divided into four different sections. For the visual subspace corresponding to the upper left quarter, Figure 4.13 shows the four images corresponding to the corner positions of the moving part. Note that for this particular section that it is clear that there is ambiguity with regard to the part moving vertically in the image, as the changes in the images are minor. Conversely, moving the object horizontally is clearly unambiguous as the dark background is reduced in size.

This ambiguity in the one direction is borne out by the results for determining position using the eigenspace for just this section. The same 100 random image set used to evaluate positional accuracy used throughout this chapter was used to form a set of images corresponding to the upper left half of the original images. The absolute mean error in the vertical direction for the moving part was 275.9 μm; in the horizontal direction it was only 21.9 μm. Thus the ambiguousness of the visual subspace in the one direction manifests itself into a larger absolute mean error in the vertical direction compared to the entire image (10.2 μm). In the other direction where there was no such ambiguity, the absolute mean error was similar to the entire image result (21.3 μm). Thus for the entire image where there was no ambiguity, for this section by itself with its own eigenspace there was significant ambiguity.

Figure 4.14 shows the images corresponding to the corner positions of the metallic object for the upper right quarter section. From these images it is clear that unlike the previous section, there are visual features present that change sufficiently with the object moving that one would expect to have good accuracy in both directions, similar to that for the entire image.

For this section, the accuracy achievable for the random set of images is much improved over the previous section. The absolute mean error for the vertical movement of the part was 15.7 μm. In the other direction it was 8.7 μm. This compares favorably to the mean results achieved with the entire images.

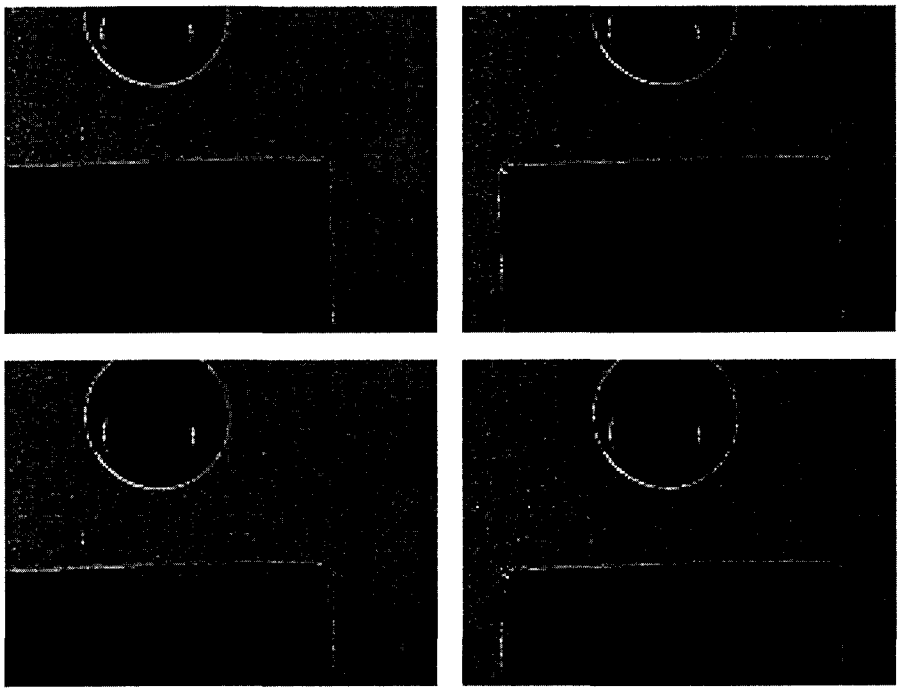

Figure 4.14: Corner Images of Camera Motion, Upper Right Section

Figure 4.15 shows the images corresponding to the corner positions of the metallic object for the lower left quarter section. From these corner images it can be seen that this section is unambiguous in both directions, as the portion of the metallic part in the images is a significant visual feature and varies significantly with movement in both directions. For the sections of the random image set corresponding to this section, the absolute mean error in the vertical direction for the part was found to be 23.2 μm. In the other direction it was found to be 27.6 μm.

Figure 4.16 shows the corner images corresponding to the lower right section. Although the one edge of the object disappears from view in two of the corner images, the other edge of the moving object is continuously in view for the entire movement range of the object and is an unambiguous feature in both directions. In the vertical direction of the object movement, the absolute mean error for the random set was 19.3 μm. In the side to side movement, the mean error was 11.8 μm. Thus for this

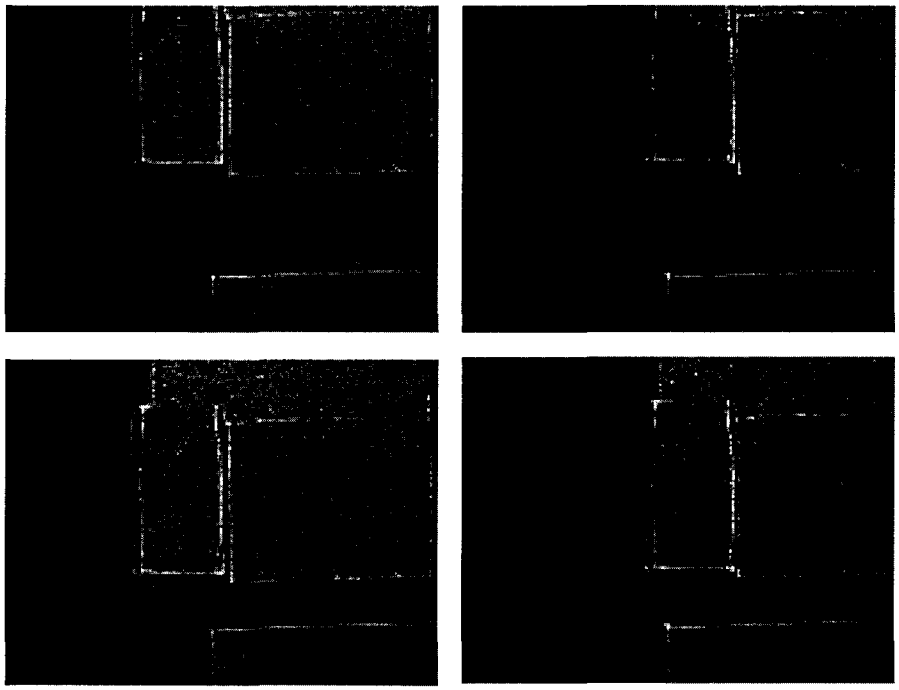

Figure 4.15: Corner Images of Camera Motion, Lower Left Section

section the absolute mean errors were actually better than that of the entire image.

Thus for this particular visual subspace, three of the sections were such that there was no ambiguity to negatively impact on the positional accuracy achievable and thus theoretically could be used in the separate eigenspaces technique for finding position despite occlusions. However the fact that one of the four sections performs poorly for determining accuracy in one direction due to considerable ambiguity over the visual subspace means that for the separate eigenspaces method, any use of that section will result in impaired accuracy. Thus the ambiguity present locally that is not present globally will impact on the separate eigenspaces performance.

The use of only four sections may not be very effective for occlusions because an occlusion located at the center will occlude all four sections. Increasing the number of sections when using separate eigenspaces also increases the chance of ambiguity induced error.

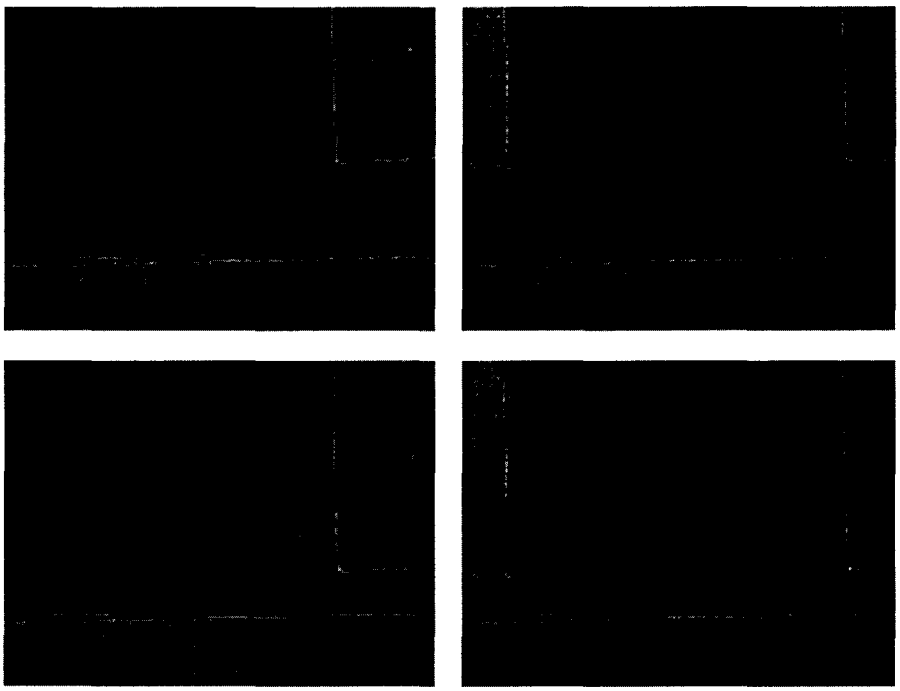

Figure 4.16: Corner Images of Camera Motion, Lower Right Section

To illustrate this phenomenon, the previous experiments with the metallic object visual subspace was performed, except using sixteen sections instead of four. Again each section was treated separately with its own eigenspace calculated and separate projection coefficients calculated and separate nearest neighbor searches.

Table 4.1 shows the absolute mean errors in both directions for all sixteen individual sections. Note they are numbered 1 to 16, with the top left section corresponding to 1 and then numbered left to right, top to bottom.

As can be seen from the various results for each of the sections, there is significant variability between the results in both directions of movement. There are a few sections where the absolute mean positional errors are similar to that of the entire image, namely sections 4 and 7. For the remaining sections however, in either one or both of the measured directions there are elevated absolute mean errors. Compared to the results for the separate eigenspaces with only four sections, the results have

Section	Absolute Mean Error Vertically μm	Absolute Mean Error Horizontally μm
1	235.7	22.0
2	819.5	81.5
3	34.9	36.2
4	90.6	56.7
5	461.1	20.2
6	448.7	160.5
7	8.1	30.7
8	8.5	51.6
9	105.0	30.8
10	576.3	28.0
11	776.3	12.6
12	768.7	33.0
13	26.6	86.6
14	14.3	404.2
15	82.9	107.6
16	18.2	394.9

Table 4.1: Absolute Mean Errors by Section

become significantly worse for the smaller sections.

Thus for separate eigenspaces, increasing the number of sections to achieve a reasonable number for handling occlusions at different locations within the image can lead to instances of ambiguity and poor results.

As a technique for providing robustness to occlusion, separate eigenspaces is hampered by the ambiguity issue.

4.4 Subsectioning and Recombination

The subsectioning and recombination technique is a more advanced use of local information designed specifically to address the problems with ambiguity that can occur with separate eigenspaces. The goal is to combine the benefits of storing local

information for robustly handling occlusion together with the original global projection coefficients which are less likely to be affected by image ambiguity.

Thus the task is to combine the local information of each image section into a technique that still retains global information and its inherent resistance to ambiguity. The subsectioning and recombination method is similar to the separate eigenspaces method in that local information is stored for each image section. Instead of calculating separate eigenvectors for each image section, however the subsectioning and recombination uses the eigenvectors calculated for the entire image using the basic eigenspace method to produce the local information. Thus the eigenvectors for each image section consist of the portions of the global eigenvectors corresponding to that section of the image (assuming there are p sections in total):

$$\mathbf{e}_i \rightarrow \mathbf{e}_i^1, \mathbf{e}_i^2, \ldots, \mathbf{e}_i^p \tag{4.37}$$

$$\mathbf{E}_i \rightarrow \mathbf{E}^1, \mathbf{E}^2, \ldots, \mathbf{E}^p \tag{4.38}$$

Using these eigenvectors, we calculate a set of local projection coefficients for each image section j of each image i:

$$\mathbf{y}_i^j = (\mathbf{E}^j)^T \mathbf{x}_i^j \tag{4.39}$$

Instead of searching each of these sets of projection coefficients separately as in separate eigenspaces, the subsectioning and recombination method forms a modified global set of projection coefficients from the local sections to perform the nearest neighbor search. Similar to the separate eigenspaces method, for a new image, each image section is evaluated for occlusions using one of the occlusion detection methods

described in the next chapter. Those image sections that are deemed to be unoccluded have their set of projection coefficients included in the global set. The global set of projection coefficients is formed by adding together the unoccluded local sets of projection coefficients for each interpolated camera position. Thus if an image section l is deemed occluded, the modified global projection coefficients for an interpolated camera position are formed by:

$$\mathbf{y}^{modified} = \mathbf{y}^1 + \mathbf{y}^2 + ...\mathbf{y}^{l-1} + \mathbf{y}^{l+1} + ... + \mathbf{y}^p \tag{4.40}$$

where p is the number of image sections. Note that if no sections are occluded, the modified global projection coefficients are equivalent to those of the basic global eigenspace method $\mathbf{y}$.

By this technique, occluded sections make no contribution to the nearest neighbor position determination phase. However because the modified set of global projection coefficients still consists of contributions from all areas of the image, we retain the global information of the basic eigenspace method and thus we avoid the ambiguity errors that can result from the separate eigenspaces method when searching a small subsection.

The basic idea behind the subsectioning and recombination method is that there is significant redundancy between the entire images and the global projection coefficients corresponding to them. Consider that the task of determining position from eigenspaces is significantly different from the task of reconstructing the image with minimal error. For subsectioning and recombination, the conjecture is that for many eigenspace positioning problems, removing several sections will still produce a set of combined local projection coefficients that allow position to be determined at a similar accuracy as that of the entire global projection coefficients.

As an example of this property, consider again the first four visualized eigenvectors

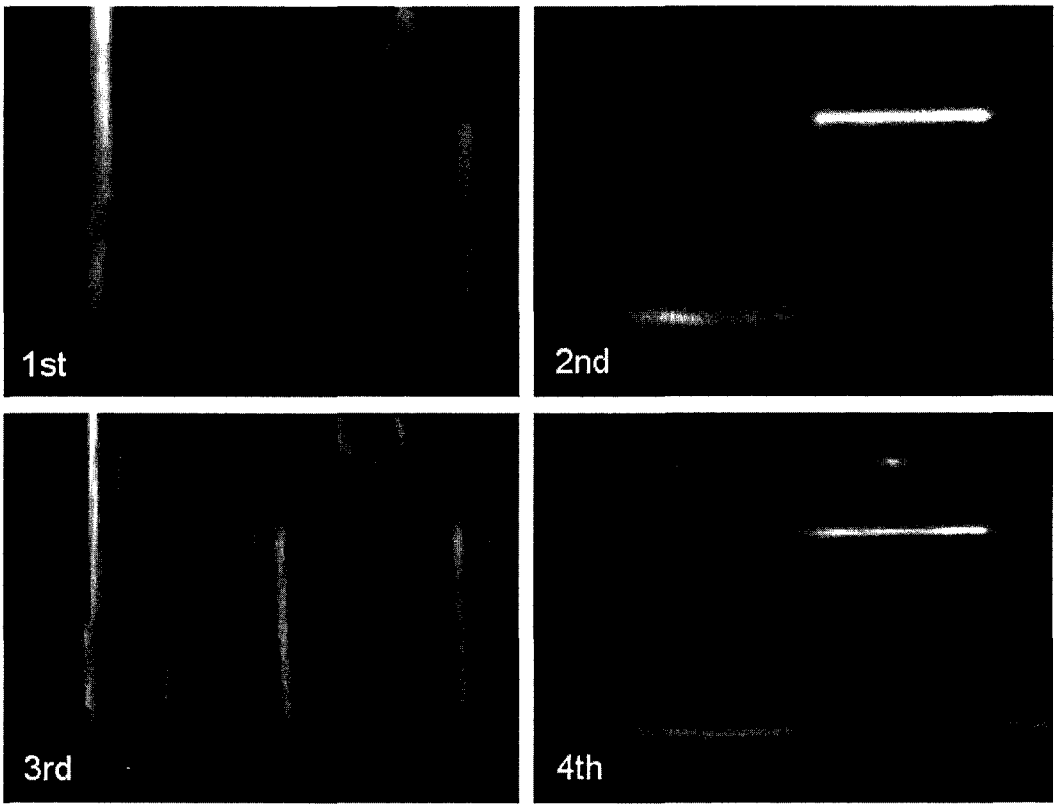

Figure 4.17: First Four Eigenvectors Visualized

for the metallic object example as shown in Figure 4.17. Since these eigenvectors are relatively simple structurally, the relationships between the eigenvectors and the projection coefficients can be relatively easily deduced.

To further illustrate this point, compare the plots of individual projection coefficient components y_i versus position for the global projection coefficients versus the subsectioning and recombination projections $y_i^{modified}$ consisting of the left half of the images. Figure 4.18 shows the plots for the first four eigenvectors for the global projections, while Figure 4.19 shows the plots for the left half. In comparison, the projection coefficients of the entire images individually have a larger range in the value of the projection coefficients produced, however the overall shapes are quite similar. More importantly, however, it can be seen that for the plots consisting of

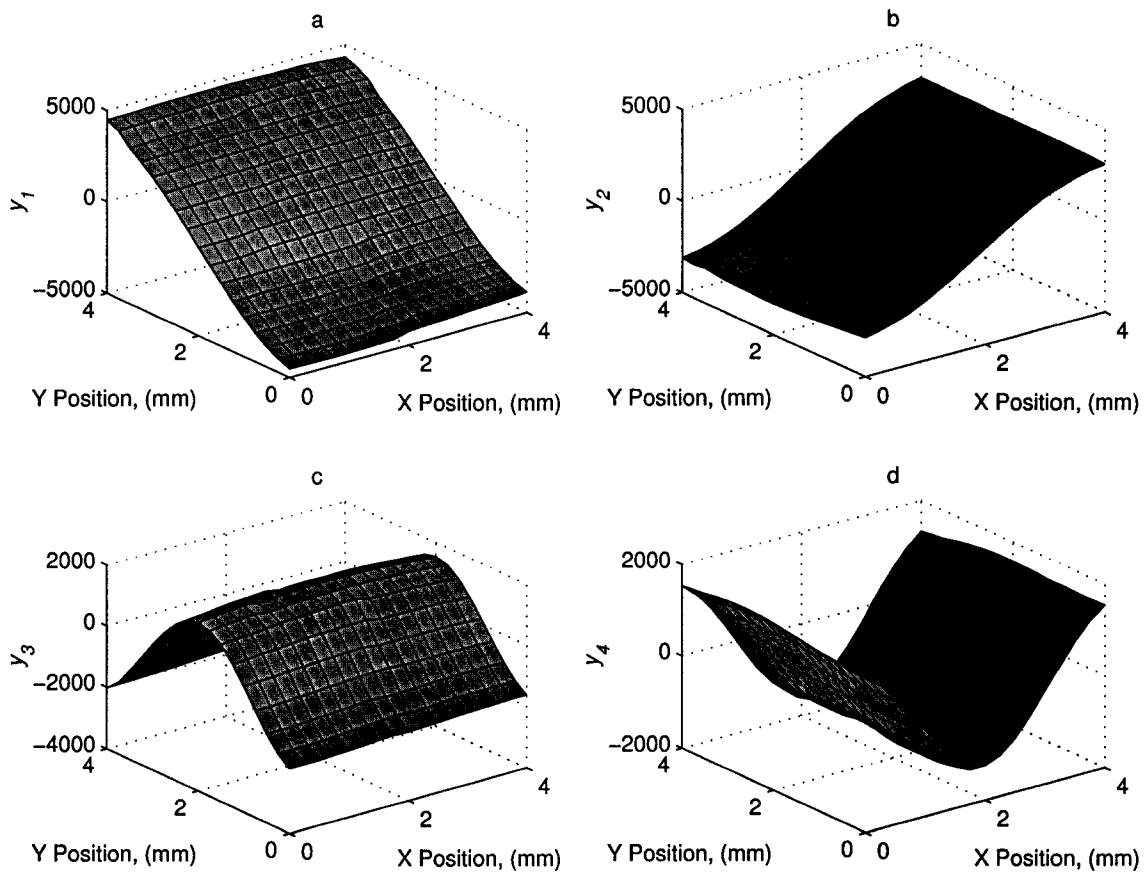

Figure 4.18: Projection Coefficient Value versus Position (Global),(a) First Eigenvector, (b) Second Eigenvector, (c) Third Eigenvector, (d) Fourth Eigenvector

only the left half of the images, that position can be unambiguously derived. As the position is altered anywhere within the range of motion, the projection coefficient values are unique.

4.4.1 Number of Sections

An important consideration for the implementation of subsectioning and recombination for an application is the number of sections to use. As opposed to separate eigenspaces, smaller sections have no negative ambiguity effects. A benefit for having a large number of sections is the previously mentioned fact that for a given occlusion,

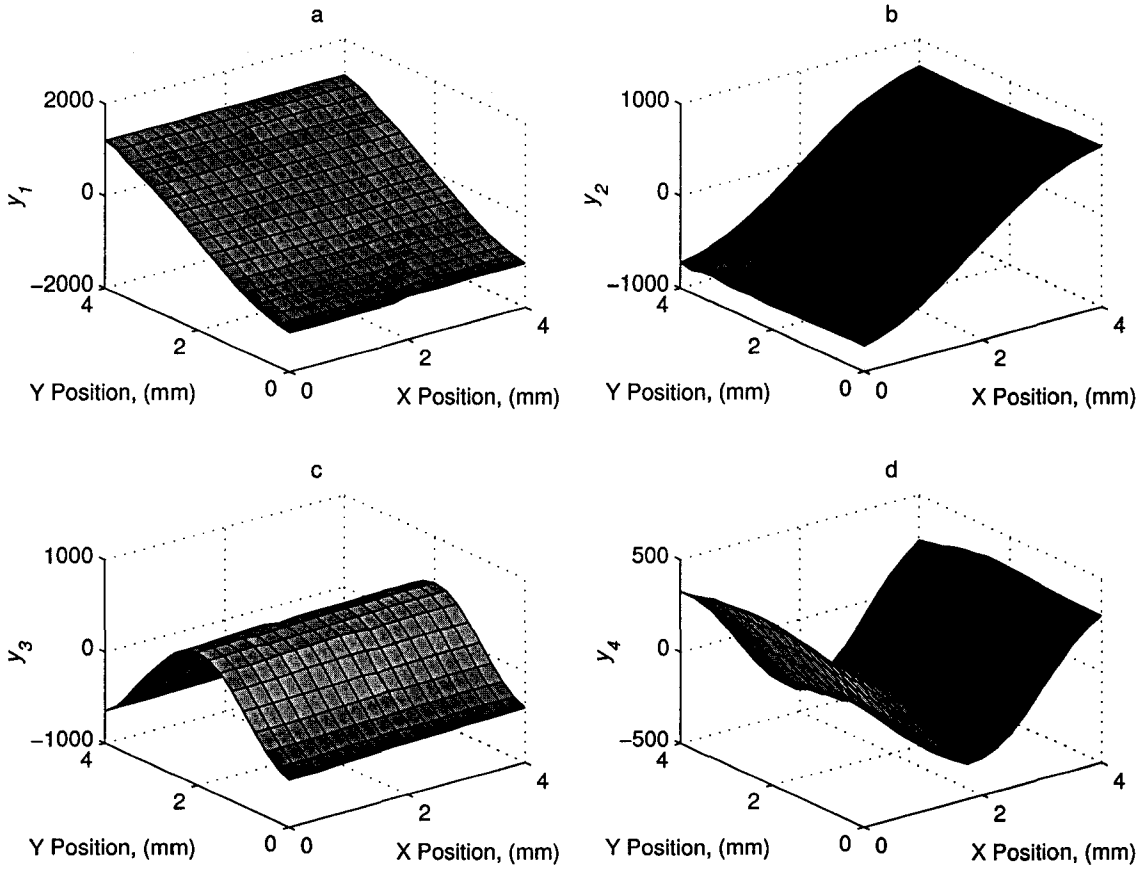

Figure 4.19: Projection Coefficient Value versus Position (Combined Local Projection Coefficient of Left Half of Images), (a) First Eigenvector, (b) Second Eigenvector, (c) Third Eigenvector, (d) Fourth Eigenvector

a smaller proportion of the image will be removed as occluded.

A drawback for using a large number of sections is the increased search time associated with forming the modified global projection coefficient set dynamically. For the basic eigenspace method with m stored projection coefficients to search and k eigenvectors, the nearest neighbor search comprises mk subtractions corresponding to the difference between the input projection coefficients $\mathbf{y}_{new}$ and each of the stored projection coefficients stored in $\mathbf{Y}$, plus m comparisons between the Euclidean distance of the resultant vector and the smallest Euclidean distance thus far encountered.

In comparison, for p sections, the subsectioning and recombination technique for the nearest neighbor stage additionally requires for each of the stored projection coefficients searched, *(p-1)k* additions to determine the total stored projection coefficient term. Since the extra *(p-1)k* additions must be performed for each of the m projection coefficients, $(p\text{-}1)mk + mk + m$ addition/subtraction operations will be required in total for subsectioning and recombination compared to just $mk + m$ for the basic eigenspace technique. Thus the number of addition/subtraction operations will increase approximately linearly with increasing p. The subsectioning and recombination method precludes the use of the binary search algorithm of Nene and Nayar [38] described in Chapter Two to reduce the computation time of the nearest neighbor procedure for eigenspace camera positioning. This algorithm requires preprocessing of the projections coefficients set, which is not available for subsectioning and recombination since it is formed dynamically.

However the coarse search approach described in Chapter 2 and demonstrated in Chapter 3 could be used with subsectioning and recombination whereby a nearest neighbor search of the training image projection coefficients is performed followed by a constrained search of the interpolated projection coefficients based on the matching training image position to reduce the computation time required. No preprocessing is required of this method save maintaining a separate set corresponding to the training image projection coefficients for each section.

In terms of practicality and the experiments performed in this chapter, sixteen sections was chosen as a balance between capturing enough unoccluded information and computational speed. Since eigenspace methods are inherently fast computationally, multiplying this search time by a factor of approximately sixteen would not adversely affect the use of the method in practical terms.

4.4.2 Ambiguity and Subsectioning and Recombination

Ambiguity issues certainly need to be considered for subsectioning and recombination. If ambiguity is a problem for the basic eigenspace technique for the entire image and thus the visual subspace is inherently ambiguous, subsectioning and recombination will not solve the problem and not much can be done with that data to achieve accurate positioning. Instead different camera views where the scene contents are not ambiguous may be required. More importantly, subsectioning and recombination can suffer from ambiguity issues relating to occluded sections that are removed.

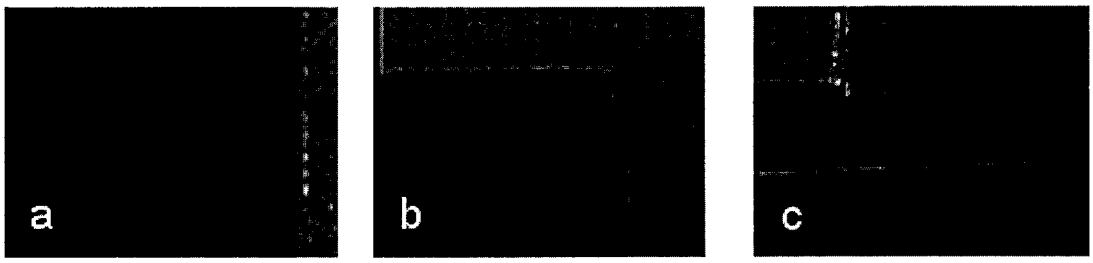

Figure 4.20: Image Sections with Varying Ambiguity, (a) First Section, (b) Eighth Section, (c) Fourteenth Section

Since the eigenvectors are optimized with respect to the entire image using PCA, areas of the image with larger amounts of variance contribute more to the formation of the PCA eigenvectors. Thus, for an image that overall is not ambiguous, removing a large number of sections due to occlusion could introduce ambiguity if the remaining image sections lack salient features. As an example consider the three images in Figure 4.20 which consist of three different sections from the metallic object example. Clearly in Figure 4.20(a), the edge of the object present allows direction to be determined horizontally, but images will be very similar for the object's movement vertically. Conversely in Figure 4.20(b) the object has perpendicular edges present that will prevent ambiguity in both directions. Figure 4.20(c) shows an edge that is perpendicular to that of Figure 4.20(a), allowing accurate position determination

in the other direction. Thus for subsectioning and recombination, combining the sections from Figure 4.20(a) and 4.20(c) will remove ambiguity in both directions.

4.5 Subsectioning and Recombination Experiments

To illustrate experimentally the usefulness of subsectioning and recombination for dealing with occlusions, two test sets were acquired of the metallic objects example with different occlusions added. Note that these test set positions corresponded to the original unoccluded random test set.

For both experiments, the occlusion detection methods from the next chapter were used to decide for each image which sections to be included. For these experiments, the occlusions were such that the performance of the occlusion detection algorithm was perfect in terms of detecting occluded sections. For this chapter we are more interested with the possible performance than with the optimal choice of occluded sections.

For the first test, an occlusion was added that consisted of a metal block added beneath the metallic parts. This is shown in Figure 4.21, which shows one of the example test images. Note that in the images, the occlusion maintains its position, while the test object moves throughout the set. While the occlusion does not take up an overly large proportion of the entire image, some of the areas indicated by the visualized eigenvectors of the global eigenspace as being highly important for determining position (namely the border of the moving part with the background) are occluded which should lead to impairment of the positional accuracy possible with the global projections.

Figure 4.22(a) and 4.22(b) show the histograms of the absolute position error for the occluded test set in both directions, for the case of using the basic eigenspace method with the global projections. In the vertical direction it was 99.4 μm and 61.8

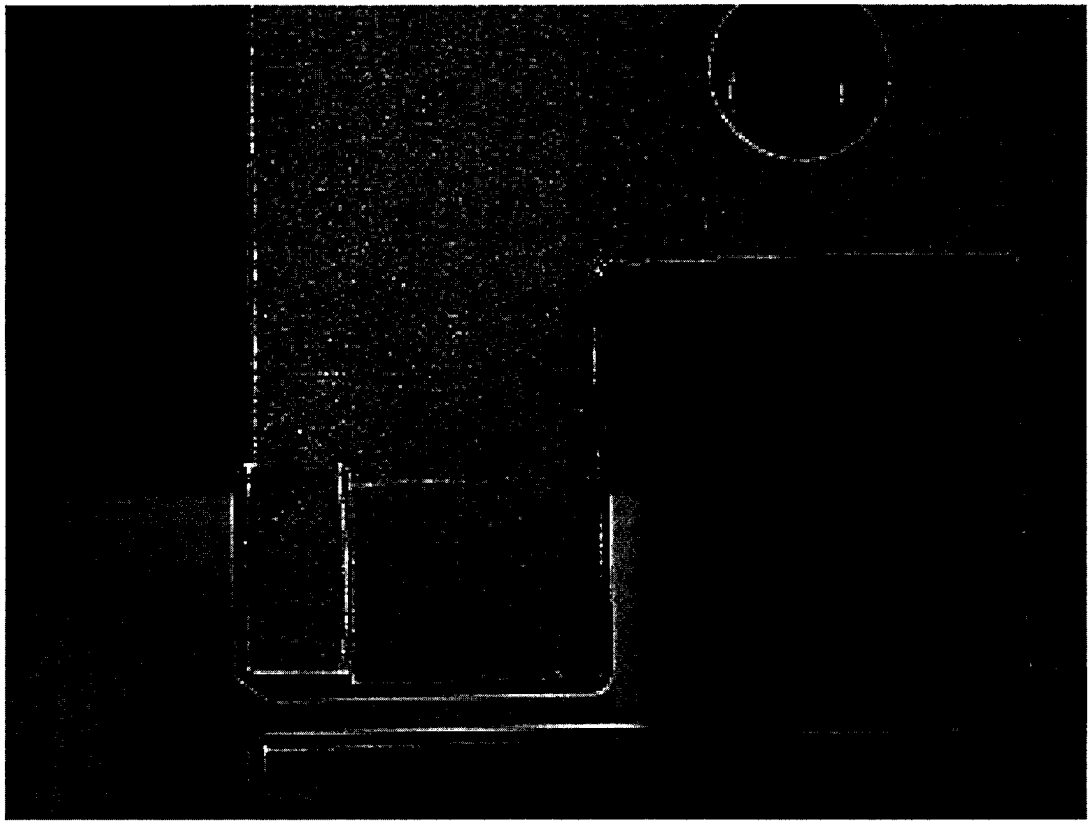

Figure 4.21: Background Occlusion Image

μm in the horizontal direction (39.8 % of the distance between training images in the vertical direction; 24.7 % of the distance between training images in the horizontal direction). It is important to note that not only are the absolute mean errors higher in both directions, compared to the unoccluded case, but that the variance of the error is larger too. Consequently, the worst case errors are higher. This is indicative of the unpredictable effects of occlusion when combined with the nearest neighbor search. Obviously for robotics and assembly applications the presence of such errors could cause difficulty.

Figure 4.22(c) and 4.22(d) show the histograms of the absolute position error for the occluded test set in both directions, this time implementing the subsectioning

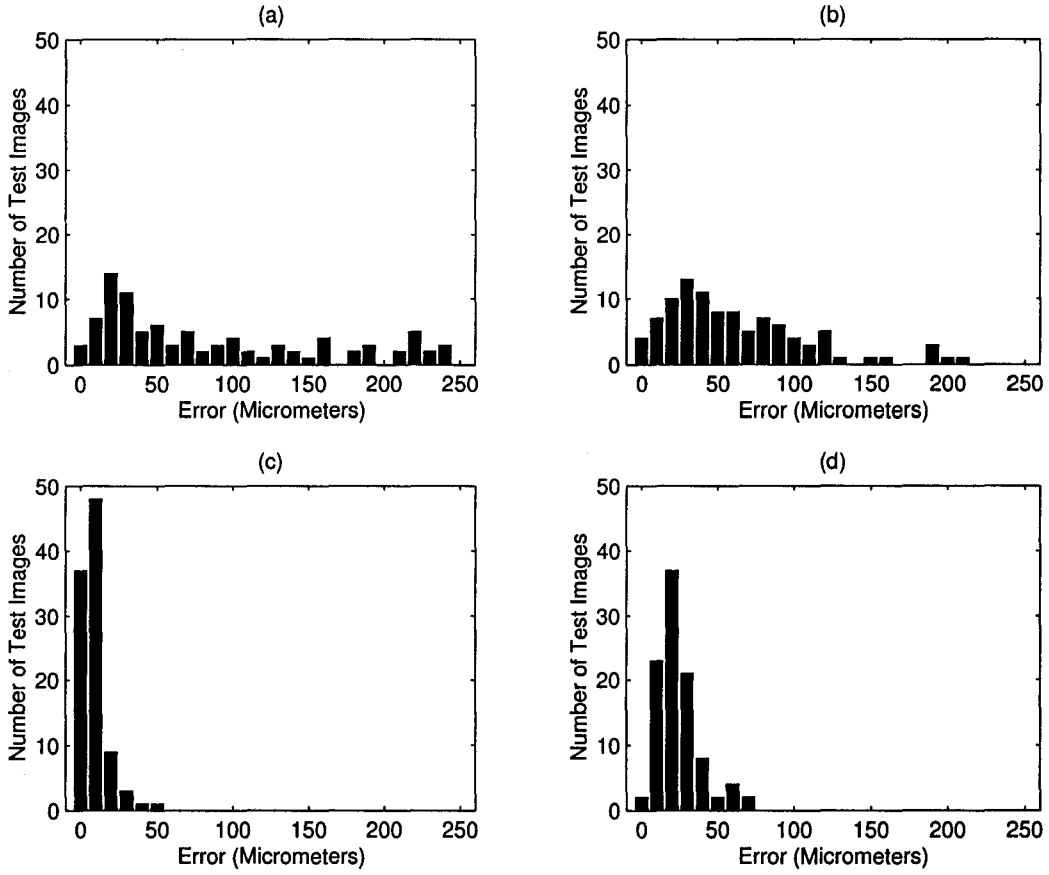

Figure 4.22: Histogram of Absolute Position Error, Background Occlusion, (a) Vertical without Correction, (b) Horizontal without Correction, (c) Vertical with Subsectioning and Recombination, (d) Horizontal with Subsectioning and Recombination

and recombination method using the sixteen section division. In the vertical direction it was 9.5 μm and 24.3 μm in the horizontal direction (3.80 % of the distance between training images in the vertical direction; 9.72 % of the distance between training images in the horizontal direction). Note that the absolute mean errors in the vertical and horizontal directions, were similar to the unoccluded case using the global projection coefficients and a major improvement over the occluded global projection coefficients results. More importantly, the instances of relatively large errors ($> 50\mu m$) is greatly reduced, both in number as well as the worst case possibility.

Thus for robotics and assembly applications, subsectioning and recombination can provide improved performance over the basic eigenspace method in the presence of occlusion.

Figure 4.23 shows an example image of the second occlusion experiment. As can be seen, the wrench added to the images is a significant occlusion in terms of the proportion of the overall image as well as the difference in appearance from the original unoccluded images.

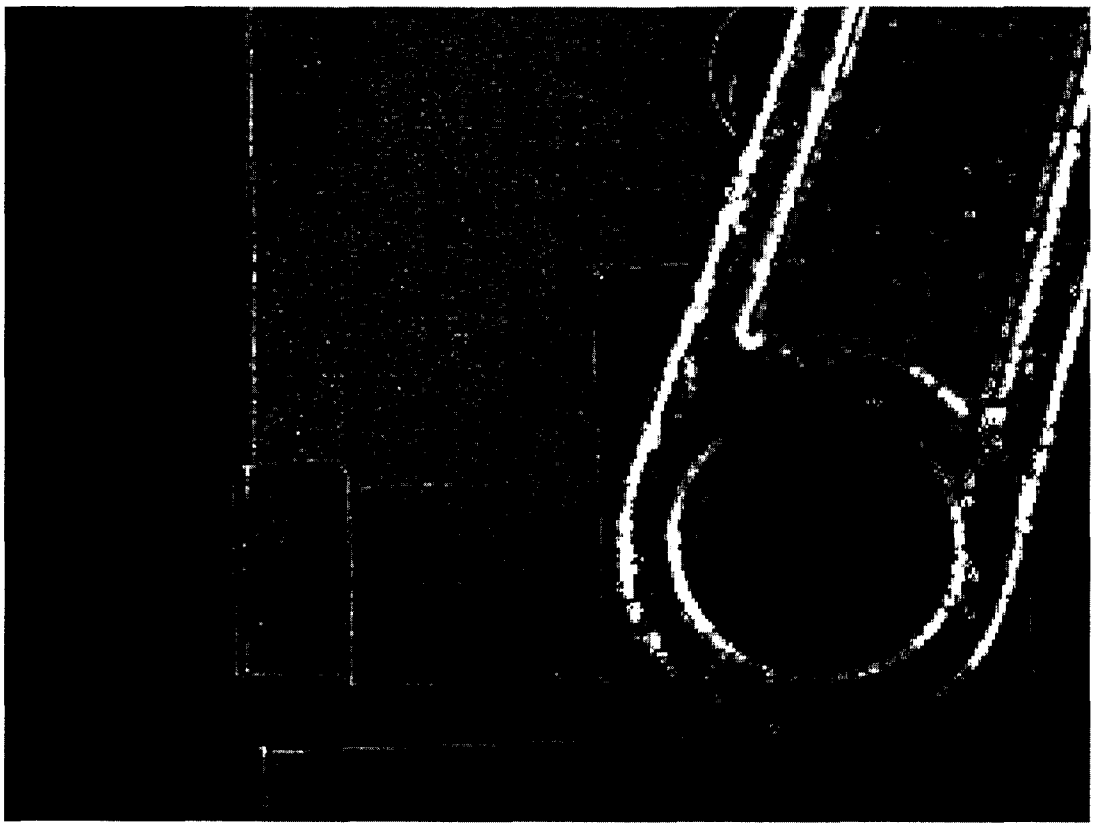

Figure 4.23: Wrench Occlusion Image

The error results for the wrench occlusion using the basic eigenspace method were higher than the first occlusion experiment, with absolute mean positional errors of 70.9 μm in the vertical direction and 150 μm in the horizontal direction (28.4 % of the distance between training images in the vertical direction and 60.0 % for the

horizontal direction) and significantly higher than the unoccluded results (10.2 μm and 21.3 μm respectively). The histograms of the absolute errors in both directions are shown in Figure 4.24(a) and Figure 4.24(b). Note the presence of some extremely large errors of position, as well as their sizeable number. These large errors occurring is typical of such a large occlusion covering of the entire image.

Figure 4.24(c) and Figure 4.24(d) show the histograms of the absolute error in both directions with subsectioning and recombination implemented. The absolute mean errors in both directions were 19.7 μm and 30.3 μm in the vertical and horizontal directions respectively (7.89 % of the distance between training images in the vertical direction and 12.2 % for the horizontal direction), which are an improvement over the uncorrected occluded results. Again the largest improvement is in the reduction of the worst case errors. Thus subsectioning and recombination can result in significant improvements in absolute mean error and the size of the worst case errors even with significant occlusions. Even with only half the sections, performance can be close to the unoccluded case, provided that the remaining sections contain information that leads to an unambiguous manifold over the visual subspace. This contrasts to the methods of robustly estimating the projection coefficients with Leonardis and Bischof's method [23] to deal with occlusion that breaks down with occlusion occupying 50 % of the image and up.

4.5.1 Performance with Severe Occlusion

As shown previously with the real occlusion experiments, subsectioning and recombination can produce accuracy similar to the unoccluded cases even with significant proportional occlusion (50 % for the wrench test set). An interesting question is the performance for even higher degrees of occlusion, that for which other techniques of occlusion correction (such as Leonardis' [23] alternate projection estimation technique) begin to break down. From our discussion of the theory behind the method, it

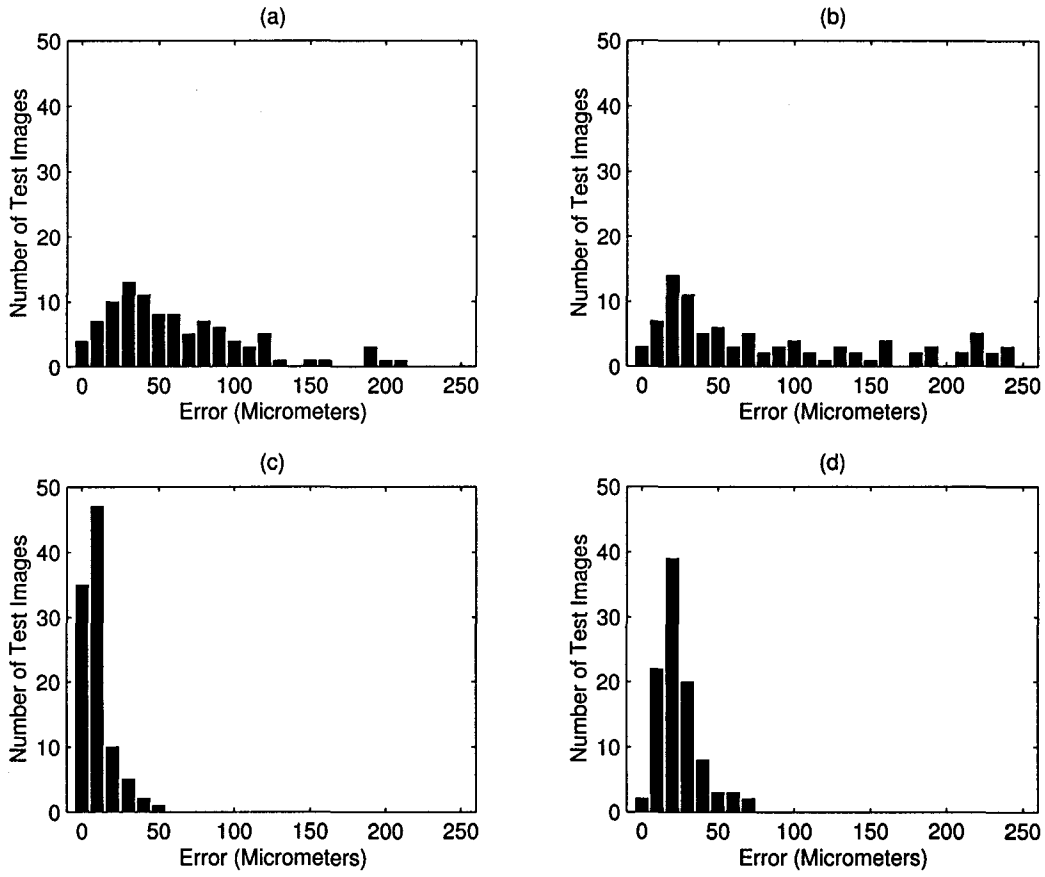

Figure 4.24: Histogram of Absolute Position Error, Wrench Occlusion, (a) Vertical without Correction, (b) Horizontal without Correction, (c) Vertical with Subsectioning and Recombination, (d) Horizontal with Subsectioning and Recombination

would appear that the method can still be successful with high degrees of occlusion if the manifold over the movement range produced by the combined local projection coefficients is not ambiguous. Although obviously the possibility of ambiguity is dependent on the particular visual subspace, it is important to remember that with the subsectioning and recombination method, disparate (in terms of spatial location) unoccluded sections increase the possibility of there being no ambiguity as opposed to separate eigenspaces.

To investigate the success of subsectioning and recombination with a high degree

of occlusion experimentally, performance with only four sections was chosen as an indicator. Rather than using images with an occlusion in the correct number of sections, for these experiments, the unoccluded sections were chosen randomly instead of using an occlusion detection algorithm. This was done because the aim was not the evaluation of the occlusion detection per se (which is covered extensively in the next chapter), but merely the performance of subsectioning and recombination when there is only a limited number of sections.

The experiment was performed four times, each with a different random selection of the section to be included as unoccluded. Note that the original unoccluded random test set of 100 images was used to provide the unoccluded sections for evaluation.

For the first experiment, sections 4, 7, 10 and 13 were chosen. Figure 4.25 shows the histograms of the absolute errors for the subsectioning and recombination with these sections. The absolute mean error in the vertical direction was 25.8 μm; in the other direction it was similar at 25.2 μm. This is quite close to the absolute mean errors for the entire image with the basic eigenspace method, although for horizontal direction the histogram shows instances of higher errors. Thus for this choice of sections, the performance was satisfactory considering the few sections used.

For the second experiment, sections 1, 4, 7 and 10 were chosen. Figure 4.26 shows the histograms of the absolute errors for the subsectioning and recombination with these sections. The absolute mean error in the vertical direction was 88.5 μm; in the other direction it was 20.8 μm. Compared to the global unoccluded image performance, in the vertical direction the absolute mean error was higher (88.5 μm versus 10.2 μm).

For the third experiment, sections 6, 7, 9 and 10 were chosen randomly. Figure 4.27 shows the histograms of the absolute errors for the subsectioning and recombination with these sections. The absolute mean error for the random set of images in the vertical direction was 20.4 μm. In the horizontal direction it was only 16.7 μm. This

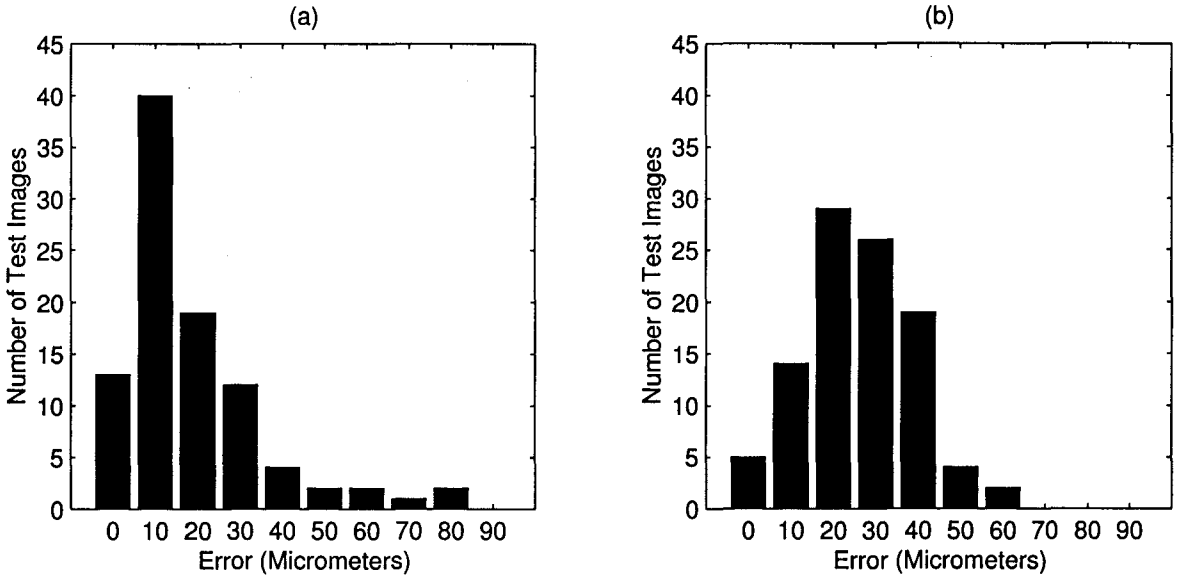

Figure 4.25: Histogram of Absolute Position Error, First Severe Occlusion (a) Vertical, (b) Horizontal

compares favorably to the global unoccluded performance, although for the horizontal direction there were instances of higher worst case error.

For the fourth of the trials with four random sections, sections 2, 5, 8 and 16 were chosen. Figure 4.28 shows the histograms of the absolute errors for the subsectioning and recombination with these sections. The absolute mean error in the vertical was 12.4 μm and 39.4 μm in the horizontal direction of the object's movement. Thus from the absolute mean error and the histogram, the vertical error was less than that of the global unoccluded results, but for the horizontal direction, the absolute mean was larger.

From these limited examples, it would appear that subsectioning and recombination can still give reasonable performance with a small number of sections located in different parts of the image. Compared to the results of the separate eigenspace method with sixteen sections, several sections spaced throughout the image can reduce the ambiguity error that is associated with separate eigenspaces. It should be

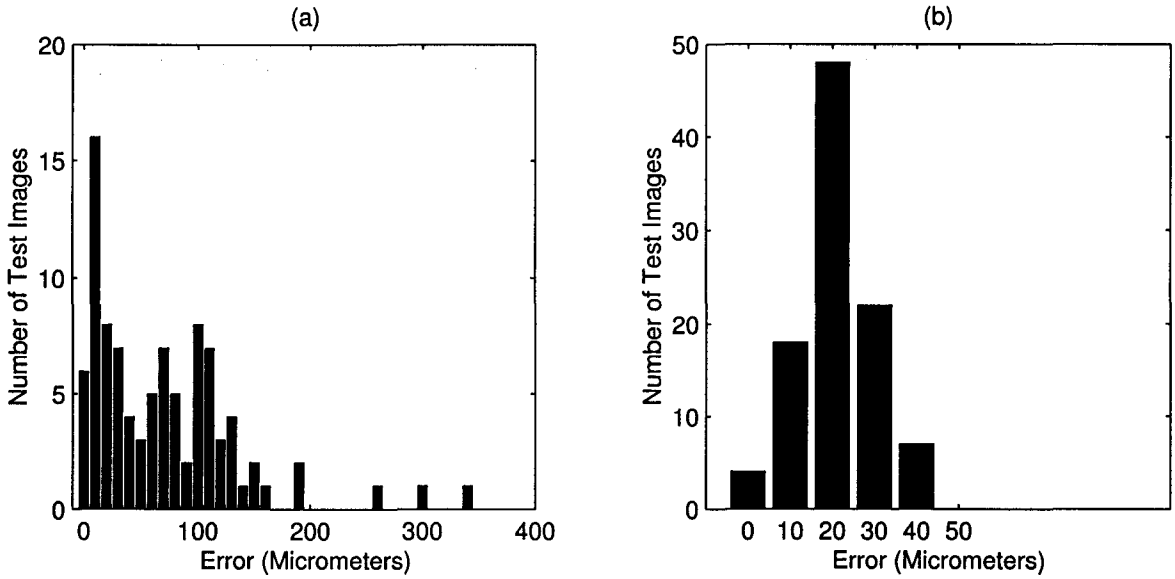

Figure 4.26: Histogram of Absolute Position Error, Second Severe Occlusion (a) Vertical, (b) Horizontal

noted however that with only a few sections (depending on the visual subspace, four in this example) that ambiguous situations are still certainly possible and thus poor results could occur with the right combinations of occlusion locations.

It should also be noted that in terms of the number of sections occluded (12), that this constitutes proportionately a high degree of occlusion for the entire image. Twelve out of sixteen sections considered occluded would for the basic eigenspace technique result in significant error, especially considering that for the earlier section on the effects of occlusion, the errors associated with occlusions occupying 10 % of the image. Thus compared to the results using the basic eigenspace positioning technique with the entire image with large occlusions, the subsectioning and recombination technique can provide improved results.

For the unlikely case of only one section being left unoccluded, then the subsectioning and recombination method is likely to give reduced performance, especially compared to that of one section using the separate eigenspaces method since that

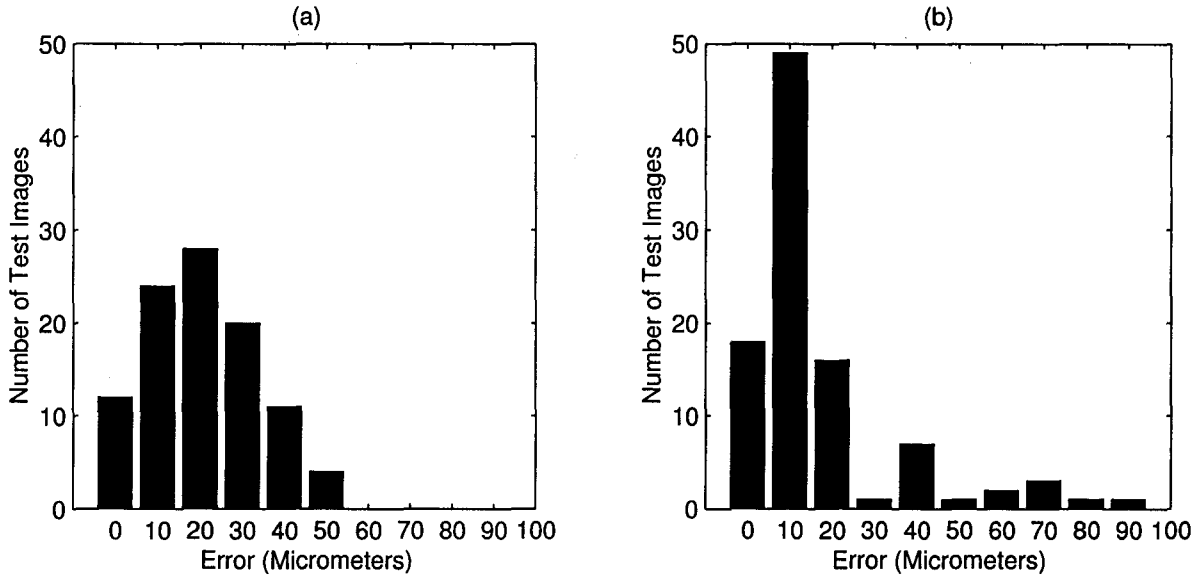

Figure 4.27: Histogram of Absolute Position Error, Third Severe Occlusion (a) Vertical, (b) Horizontal

method will be in a sense optimized to the case of only one section remaining. To illustrate the wide range of performance possible with only one section, Table 4.2 shows the mean errors for all sixteen sections taken independently using subsectioning and recombination.

As can be seen from these results, the performance of subsectioning and recombination with only one section out of sixteen for this visual subspace is highly variable across the subsections. For almost all sections the error is significantly higher than the global unoccluded results. Thus in instances of severe occlusion, it is quite conceivable that subsectioning and recombination will not produce useable positioning information.

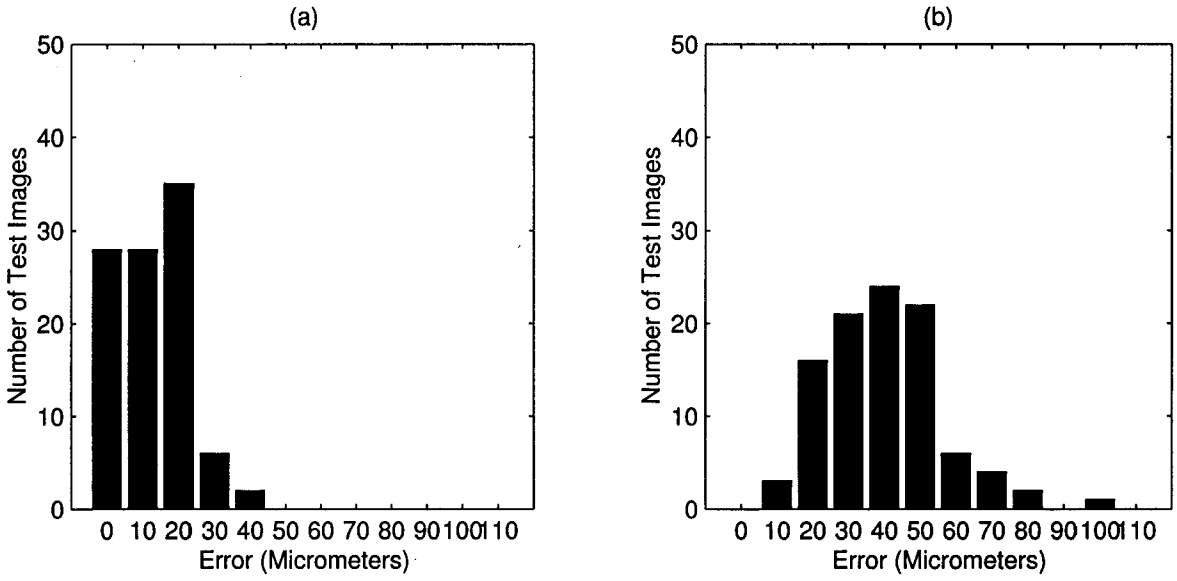

Figure 4.28: Histogram of Absolute Position Error, Fourth Severe Occlusion (a) Vertical, (b) Horizontal

4.6 Chapter Summary

In this chapter, occlusions were shown to introduce significant errors to the basic eigenspace technique due to the global nature of the projections. Error increased with proportional occlusion size and larger differences in appearance from the original background. The concept of storing local information that is unaffected by occlusions in other regions of the image was introduced first by the separate eigenspaces technique as a method to determine position accurately despite occlusion. However, separate eigenspaces suffered from ambiguity related error. With smaller sections the chance of this error increased, and along with it the possibility of catastrophic errors.

To circumvent the ambiguity the subsectioning and recombination technique was introduced as a means of combining the local information of separate eigenspaces to accommodate occlusions with the resistance to ambiguity of the entire image. In experimentation, subsectioning and recombination proved to achieve performance

Section	Absolute Mean Error Vertically μm	Absolute Mean Error Horizontally μm	Section	Absolute Mean Error Vertically μm	Absolute Mean Error Horizontally μm
1	309.8	20.2	9	89.4	52.9
2	212.9	1162.7	10	552.1	427.2
3	37.0	38.0	11	1004	21.3
4	92.7	71.6	12	705.2	43.4
5	568.1	24.0	13	33.6	100.3
6	216.1	210.5	14	29.6	797.7
7	17.6	53.1	15	75.1	123.8
8	19.6	59.7	16	14.8	429.3

Table 4.2: Absolute Mean Errors by Section Number

comparable to unoccluded images despite significant occlusion. Even with level of occlusion approaching 75 %, the increase in absolute mean error was only minor compared to the basic global eigenspace technique.

Chapter 5

Occlusion Detection

5.1 Occlusion Detection for Subsectioning and Recombination

In the previous chapter, it was assumed that occluded sections could be detected for subsectioning and recombination. This is not a trivial task and this chapter presents two approaches to detect occluded sections.

Eigenspace reconstruction error is shown to be an effective measure for differentiating between occluded and unoccluded sections. A threshold based on Gaussian statistics was derived to balance between rejecting unoccluded sections as occluded and vice versa. The measure is demonstrated successfully on the experiment described in the previous chapter.

Manifold distance was also evaluated as a measure to differentiate between occluded and unoccluded sections. It also proved to be effective at this task.

5.2 Eigenspace Reconstruction Metric for Detecting Occlusions

The eigenspace reconstruction method for detecting occlusions in visual subspaces corresponding to object or camera movement is based on the method of Turk and Pentland [45] to determine whether an image contains a face. The task was to construct a visual subspace consisting of all human faces (ideally) and evaluate whether an image belonged to this visual subspace. Alternatively, we construct visual subspaces corresponding to a prescribed object or camera movement and determine if an image belongs to this visual subspace without any changes (occlusions).

In the earlier description of the eigenspace method, visual subspaces for a range of camera or object movement were defined. It is then assumed that input images will always be drawn from within the subspace. As previously described in Chapter 2, the eigenvectors $\mathbf{e}_i$ of the visual subspace are such that they provide the best mean squared reconstruction error for those images in the subspace. For a image vector $\mathbf{x}$, the eigenspace reconstruction $\mathbf{x}'$ using k eigenvectors is defined by:

$$\mathbf{x}' = \sum_{i=1}^{k} (\mathbf{x}^T \mathbf{e}_i)\mathbf{e}_i \tag{5.41}$$

The reconstruction error ξ_r for $\mathbf{x}$ is then defined as the Euclidean difference between the reconstruction $\mathbf{x}'$ and the original image vector $\mathbf{x}$:

$$\xi_r = \| \mathbf{x}' - \mathbf{x} \| \tag{5.42}$$

Potentially other distance measures such as the Manhattan distance $|\mathbf{x}' - \mathbf{x}|$ could be used, but in our experiments the Euclidean measure performed adequately. Again,

for image vectors within the range of motion of the camera, thus within the subspace, the eigenvectors corresponding to the covariance matrix of the visual subspace minimize the mean squared reconstruction error.

We can use this preciseness of the eigenvectors for occlusion detection. Significantly occluded images will no longer fall within the visual subspace, thus the reconstruction errors of occluded images will be significantly higher than those of unoccluded images. As an example of the differences in image reconstruction between occluded and unoccluded images, the metallic object visual subspace from the previous chapter was used. One of the random testing images was chosen and reconstructed with 30 eigenvectors. These eigenvectors were derived from 289 training images equally spaced throughout the movement range of the object. Figure 5.1(a) shows the original image, 5.1(c) the reconstruction. To illustrate the effects of occlusion on reconstruction, a zero intensity square (80 pixels per side) was embedded at the center of the image. Figure 5.1(b) shows this artificially occluded image and 5.1(d) the reconstruction. Note that after producing $\mathbf{x}'$ via the reconstruction process, the eigenspace average vector was added to $\mathbf{x}'$ before converting the image vector back to an image for visualization purposes.

As can be seen from the reconstruction images, the unoccluded image and its reconstruction are visually very similar as would be expected for such a compact visual subspace. Conversely, the occluded image and its reconstruction differ significantly, not only in the immediate area of the occlusion, but also in the other regions of the image. Thus the eigenvector projection coefficients altered by occlusion cause additional error in the unoccluded area of the image as well, increasing the overall reconstruction error. This additive error is useful from the perspective of thresholding the reconstruction error for detecting occlusions.

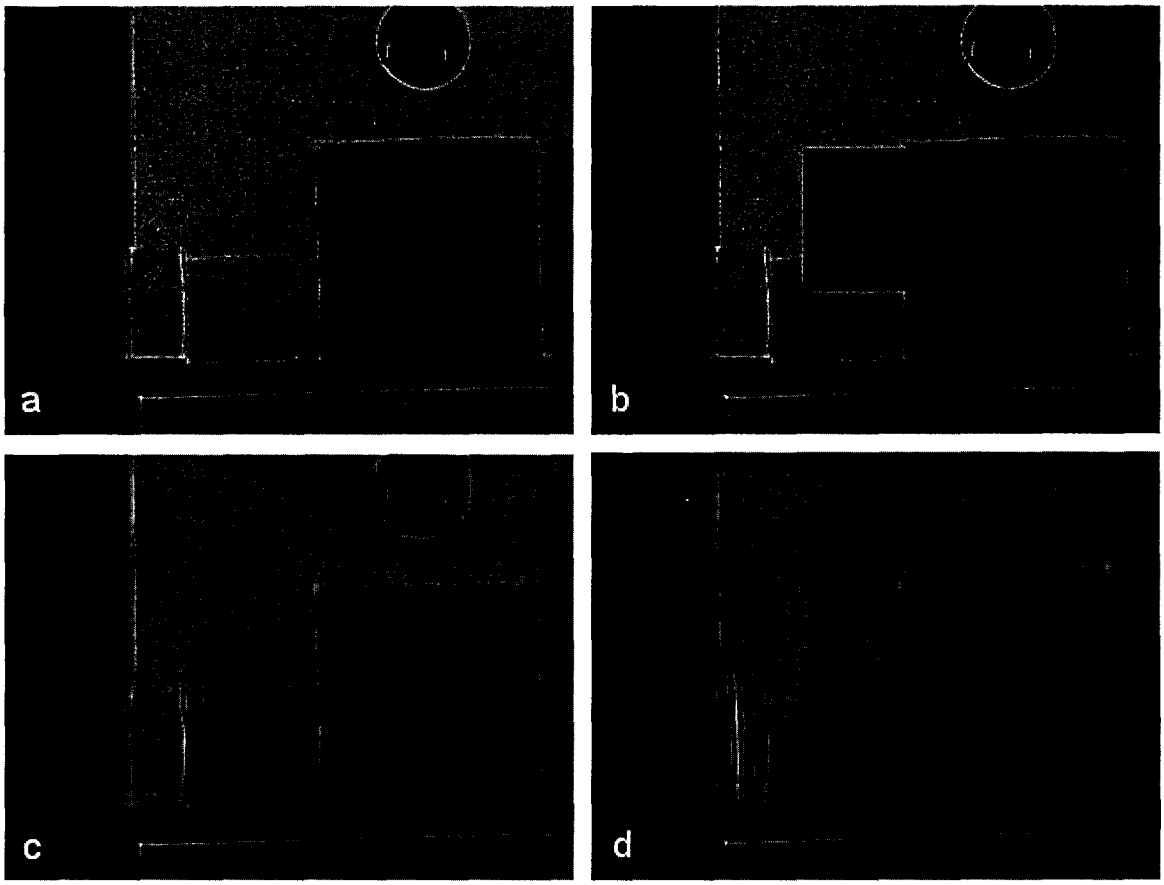

Figure 5.1: Original and Reconstructed Images, (a) Original Unoccluded, (b) Occluded Unoccluded, (c) Original Reconstructed, (d) Occluded Unoccluded

5.2.1 Reconstruction Error and Thresholding

To show the ability of the ξ_r metric as a method of detecting occlusions via a threshold, the 100 image random position set of the metallic object visual subspace was used. Figure 5.2(a) shows a histogram of the reconstruction errors of this set (with no occlusion) using 30 eigenvectors for reconstruction. For comparison, an occluded set was constructed consisting of the unoccluded set embedded with an artificial occlusion of a zero intensity square (of eighty pixels per side), at the center of the images. Figure 5.2(b) shows the histogram of the ξ_r values of this set with the same parameters as the unoccluded set.

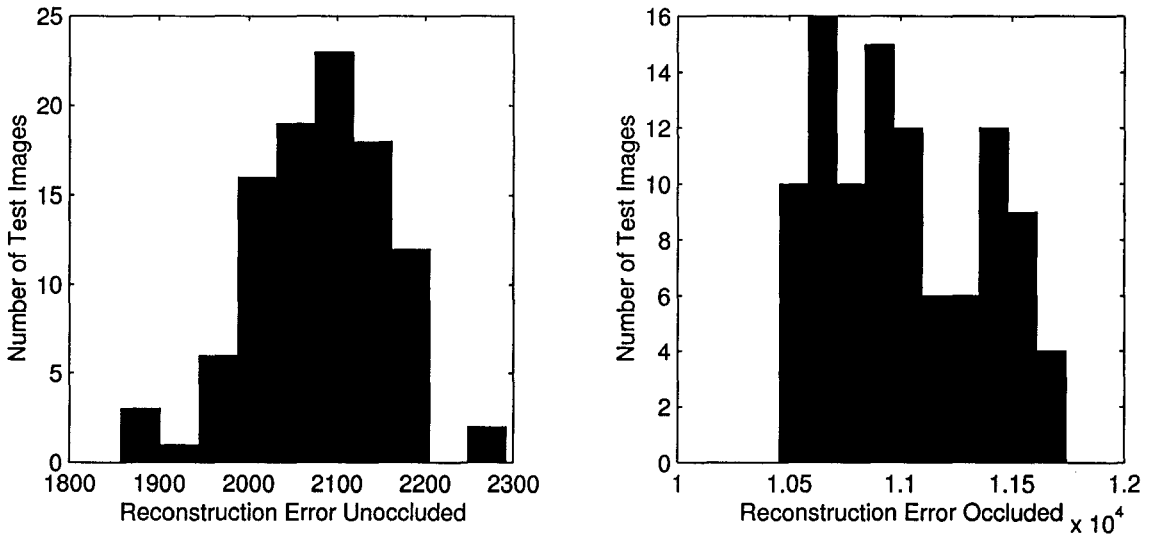

Figure 5.2: Histogram of Reconstruction Error, (a) Unoccluded, (b) Occluded

As can be seen from the histograms, the unoccluded image reconstruction errors are significantly less than that of the occluded images. The mean of the unoccluded errors is approximately 1/5 of the occluded case and more importantly there is no overlap between the two distributions, thus a threshold for detecting occlusions can be easily set (note that the area occupied by the occlusions was 8.3 % of the entire image).

An important factor to note however for using occlusion detection as a front end for the subsectioning and recombination technique is the relationship between overall occlusion size and the error induced in the position measurement for the basic eigenspace method. From the previous chapter, for the entire image, a proportionally small occlusion (<2%) usually does not have much of an effect on the accuracy and would have difficulty being detected as part of the entire image. However dividing the image into sixteen separate sections will allow the occlusion to be detected more easily since it will occupy proportionately a much larger area of an individual section. Thus for subsectioning and recombination, the small section size increases the ability

to detect occlusions.

5.3 Factors Affecting Eigenspace Reconstruction Occlusion Detection

The important controllable factors that affect the occlusion detection technique for eigenspace reconstruction, are the number of eigenvectors used for the reconstruction as well as the number of images used to estimate those eigenvectors. These two factors are important for performing the thresholding operation to determine if an occlusion is present, since increasing these factors causes a large spread between the reconstruction error of an occluded image and the reconstruction error of an unoccluded image.

5.3.1 Number of Eigenvectors For Reconstruction

Increasing the number of eigenvectors used for reconstruction for the unoccluded images will result in a lowered ξ_r as the added eigenvectors will more accurately capture the information of the visual subspace. Conversely, for occluded images, adding eigenvectors will generally not result in increased accuracy as the eigenvector coefficients for reconstruction altered by the occlusion will continue to interfere with proper reconstruction.

The vast majority of the information for reconstruction is contained in the first few eigenvectors. As an example, Figure 5.3 shows the plot of the eigenvalues for the visual subspace corresponding to the moving metallic object of Figure 4.1 in the previous chapter. As can be clearly seen, the vast majority of the information is captured in the first few vectors. Thus increasing the number of eigenvectors beyond the first few should not result in a significant gain in differentiability between occluded and unoccluded images.

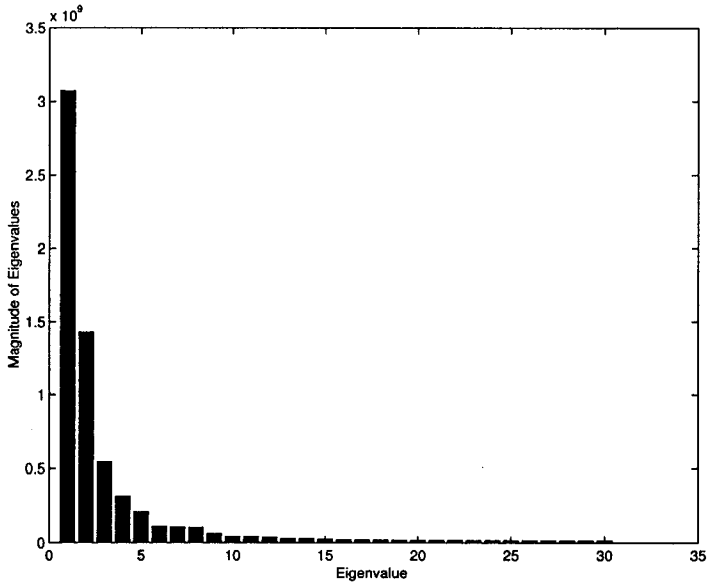

Figure 5.3: Eigenvalue Magnitudes

To illustrate the effects of varying the number of eigenvectors for occlusion detection experimentally, the unoccluded test images for the metallic object as well as the occluded test images with the zero intensity squares were subjected to eigenspace reconstruction with differing numbers of eigenvectors (for this experiment, the squares were only 50 pixels per side). Example images from these test sets are shown in Figure 5.4.

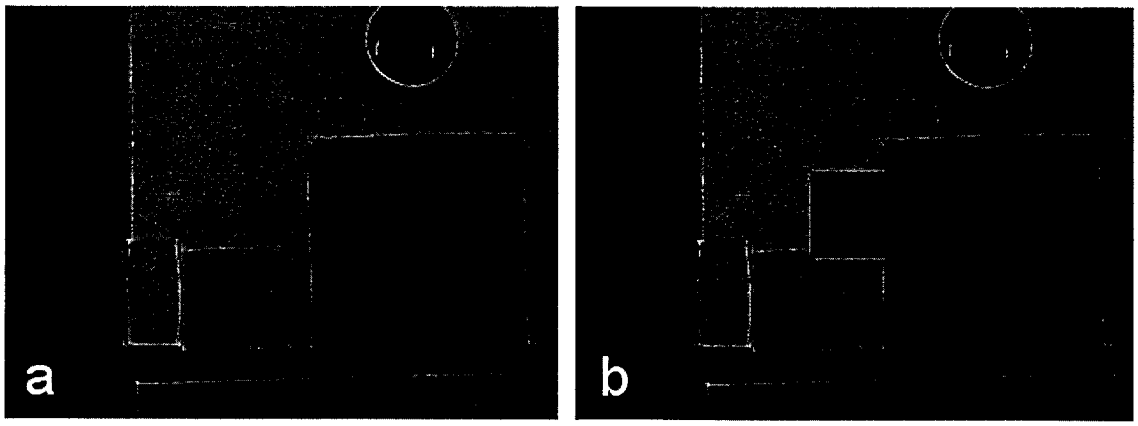

Figure 5.4: Example Images, (a) Unoccluded, (b) Occluded, 50 Pixel Squares

Figure 5.5 and 5.6 shows the histogram of ξ_r for both sets of test images with 5, 10, 20 and 30 eigenvectors. For the unoccluded histograms, consisting of 5.5(a), 5.5(b), 5.5(c) and 5.5(d) for 5, 10, 20 and 30 eigenvectors respectively it can be seen that as the number of eigenvectors is increased, the overall histogram distribution shifts in the direction towards zero reconstruction error. Adding additional vectors does continue to result in decreasing overall error. Conversely increasing the number of eigenvectors for the occluded images (Figure 5.6(a), 5.6(b), 5.6(c) and 5.6(d) for 5, 10, 20 and 30 eigenvectors respectively) has a smaller effect in comparison. This can be explained by the fact that despite adding additional eigenvectors, the projection coefficients produced will still be erroneous due to the occlusion. Note that for all the results there was significant separation between the histograms with no overlap for any of the samples.

In terms of differentiating between occluded and unoccluded images, it should be noted that while increasing the number of eigenvectors results in decreased reconstruction error, this change is small compared to the overall difference between the unoccluded histograms and the occluded histograms. Considering that for this example, the occlusion only occupied 3.26 % of the overall image area, using a large number of eigenvectors (30) for this instance was not necessary. However, if the aim is to be able to detect proportionately smaller occlusions, increasing the number of eigenvectors will increase the ability to properly discriminate between occluded and unoccluded images.

One other consideration to take into account for the number of eigenvectors to use in reconstruction is the amount of computation time required. From equation 5.41 increasing k increases the computation time for calculating the reconstructed image linearly. For the experiments performed later in this chapter for the use of eigenspace reconstruction with subsectioning and recombination, with 30 eigenvectors and images of 320 by 240 pixels, approximately 4,760,000 floating point operations were required.

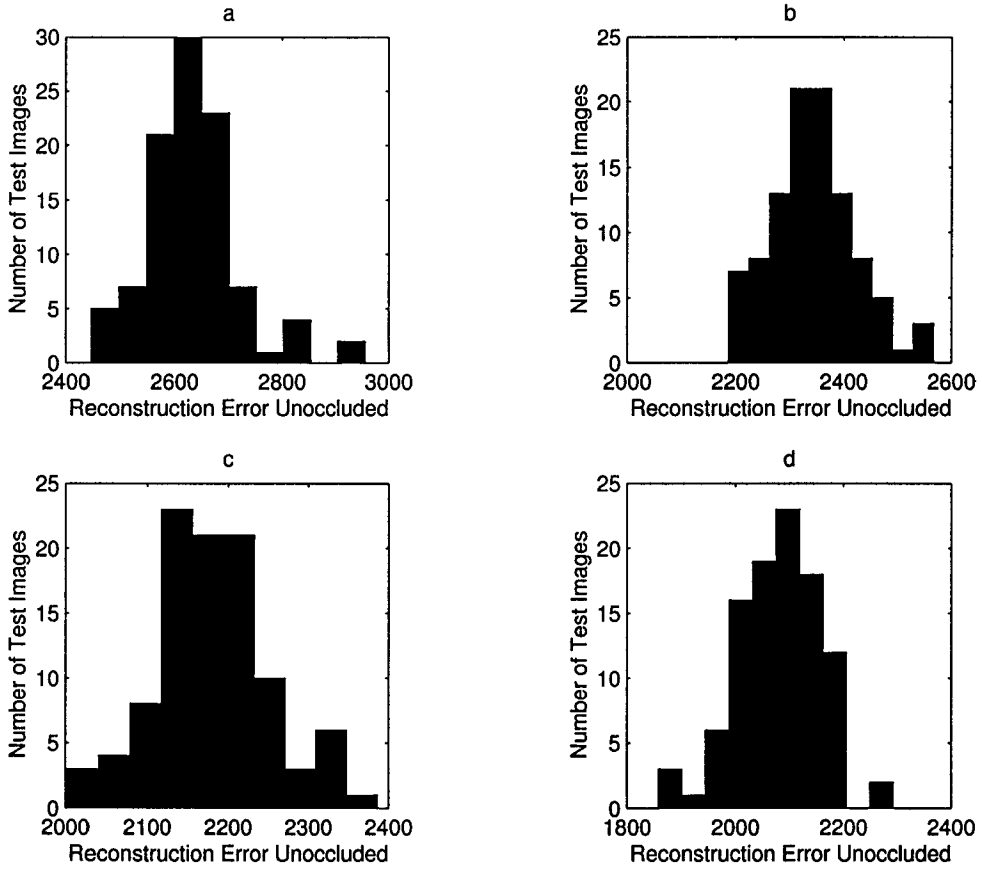

Figure 5.5: Histograms of Reconstruction Error, Unoccluded, (a) Five Eigenvectors, (b) Ten Eigenvectors, (c) Twenty Eigenvectors, (d) Thirty Eigenvectors

With fifteen eigenvectors only approximately 2,450,000 floating point operations were required.

5.3.2 Number of Images Used For Training

The other controllable factor related to the accuracy of eigenspace reconstruction is the number of images used to estimate the eigenvectors. Increasing the number of training images will result in a more accurate estimation of the actual covariance matrix pertaining to the particular visual subspace and reduces the incidence of unsmooth eigenvectors as discussed in Chapter 3. Consequently the eigenvectors

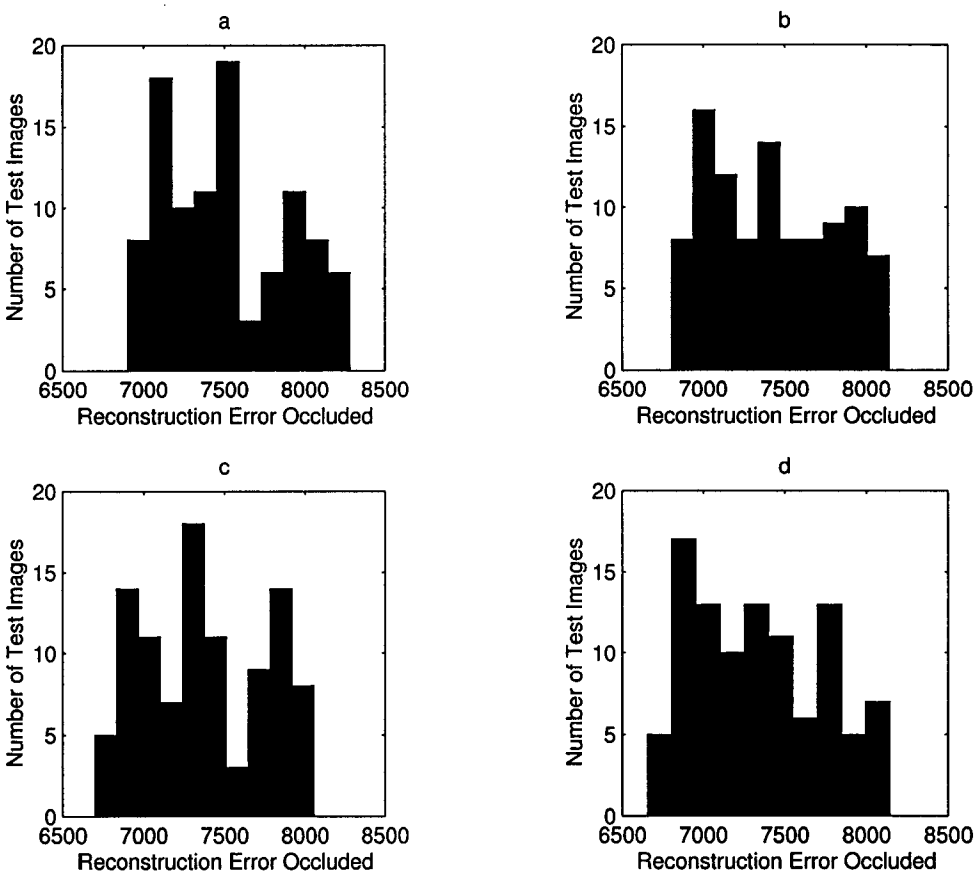

Figure 5.6: Histograms of Reconstruction Error, Occluded, (a) Five Eigenvectors, (b) Ten Eigenvectors, (c) Twenty Eigenvectors, (d) Thirty Eigenvectors

corresponding to the use of more training images should result in a decreased reconstruction error comparatively.

To illustrate this concept experimentally, the random set of unoccluded images and the random set embedded with zero intensity squares (50 pixels per side) were used again (example images shown in Figure 5.4). Figure 5.7 and 5.8 shows the histograms of the reconstruction error for these sets using eigenvectors derived from 25, 81 and 289 training images. Twenty eigenvectors were used for each histogram to produce the ξ_τ.

For the unoccluded results with 25, 81 and 289 training images corresponding to

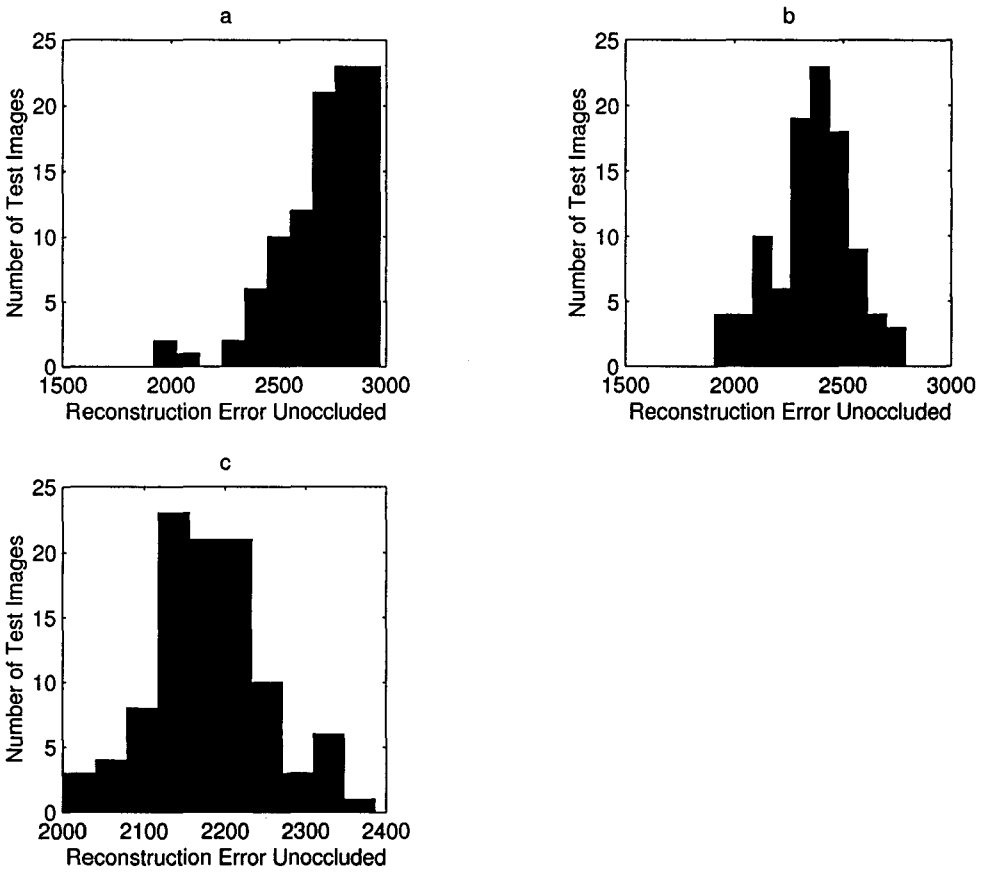

Figure 5.7: Histograms of Reconstruction Error with Varied Number of Training Images, Unoccluded, (a) 25 Training Images, (b) 81 Training Images, (c) 289 Training Images

Figure 5.7(a), 5.7(b) and 5.7(c), the histogram of the ξ_r improves in the sense that as the number of training images is increased, the overall distribution shifts towards zero reconstruction error. The mean ξ_r values were 2700, 2395 and 2182 for 25, 81 and 289 images respectively, indicating that as the number of training images is increased, the improvement from adding more images becomes less significant.

The occluded histograms corresponding to Figure 5.8(a), 5.8(b) and 5.8(c) for 25, 81 and 289 training images respectively conversely changed little in their overall distribution in terms of reconstruction error. Thus increasing the training images

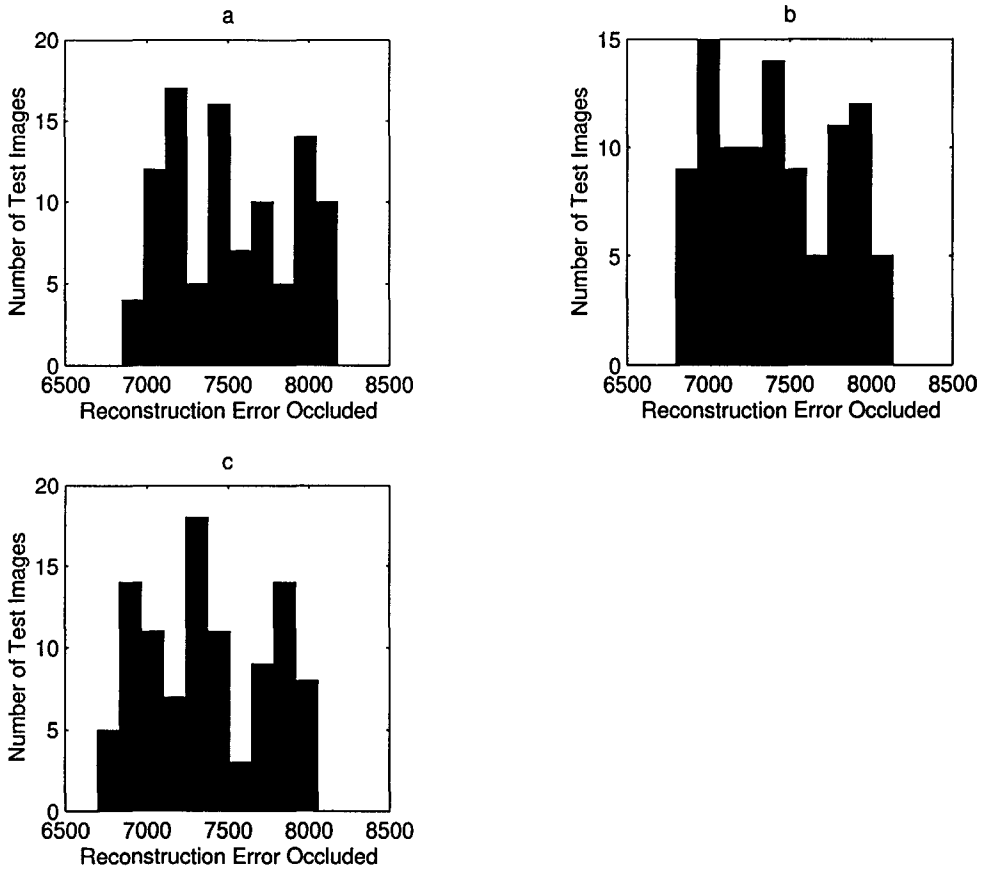

Figure 5.8: Histograms of Reconstruction Error with Varied Number of Training Images, Occluded, (a) 25 Training Images, (b) 81 Training Images, (c) 289 Training Images

used results in increased ability to discriminate for occlusion detection.

In terms of the number of training images to use, unlike with the number of eigenvectors, increasing the number of training images results in only additional off-line computation. Thus since the number of training images will not effect the online computation times for performing occlusion detection, for implementation of subsectioning and recombination the largest set of training images available should be used, likely that used to calculate the projection coefficients for determining position. Obtaining a large number of training images for a given application could potentially be

more of a practical difficulty than a computational one.

5.3.3 Accurate Thresholding for Occlusion Detection

The criteria of whether an image or an image section should be marked as occluded depends on the application occlusion detection is to be used for. We mainly consider occlusion detection for the task of determining whether an image section is occluded for subsectioning and recombination

For subsectioning and recombination, the obvious aim is to include those sections that contain no occlusions while excluding those that contain any occlusions. Setting an adequate threshold seeks to find an adequate balance between the two.

The question of including unoccluded sections is especially important for subsectioning and recombination. Falsely rejecting a section as occluded when it is not will often make little difference for subsectioning and recombination in the situation where the vast majority of the sections are unoccluded. Conversely, for the situation where the image is highly occluded with only a small number of unoccluded sections, not including an unoccluded section could potentially have a pronounced negative effect. As seen from the results of the previous chapter with subsectioning and recombination, error can reach catastrophic levels with only a few sections in the case of image ambiguity across the visual subspace. Since every extra section potentially could remove ambiguity, including every unoccluded section is of paramount importance. Thus the threshold should be set to ensure that in almost every circumstance unoccluded sections are included.

The question of when to reject a section as being occluded for subsectioning and recombination should be recast as whether to reject a section because the occlusion is large enough and/or is different in appearance enough to have a non-negligible effect on positional accuracy. Occlusions of small sizes proportionately (<2%) have

generally a negligible effect overall on positional accuracy. Thus the balance of detecting occlusions should be set to include all unoccluded sections if occluded sections accepted will have almost no effect on accuracy.

To illustrate the ability of eigenspace reconstruction to detect small occlusions, the metallic object example was again used. As previously, the unoccluded random set was embedded with zero intensity squares in the center of the images as artificial occlusions. The size of the occlusions was chosen specifically to be quite small in comparison to the reconstruction error of the unoccluded set to illustrate the limits for detecting occlusions unquestionably. Figure 5.9 shows the histogram of the ξ_r for the case of no occlusion, as well as squares of 10 pixels per side, 20 pixels per side and 30 pixels per side. The 30 eigenvectors used for the reconstruction were derived from 289 training images.

From these results, it can be seen that thresholding for occlusion detection will be difficult to properly differentiate between the unoccluded results from the square occlusions with 10 and 20 pixel side lengths. However a threshold can be set to separate the 30 pixel occlusion from the unoccluded images. From our results of the previous chapter, the 30 pixel square occlusion has only a minor negative effect on the accuracy, thus being properly able to detect it is encouraging. The 30 pixel occlusion only takes up approximately 1.1 % of the area of the overall image.

5.3.4 Deriving a Threshold

In terms of setting the threshold for differentiating between occluded and unoccluded images there are several approaches that are possible. From the previous section, for use in subsectioning and recombination the aim is to set a threshold high enough to virtually rule out rejecting unoccluded images and otherwise keeping it as low as possible.

Through experimentation with random image sets for different visual subspaces,

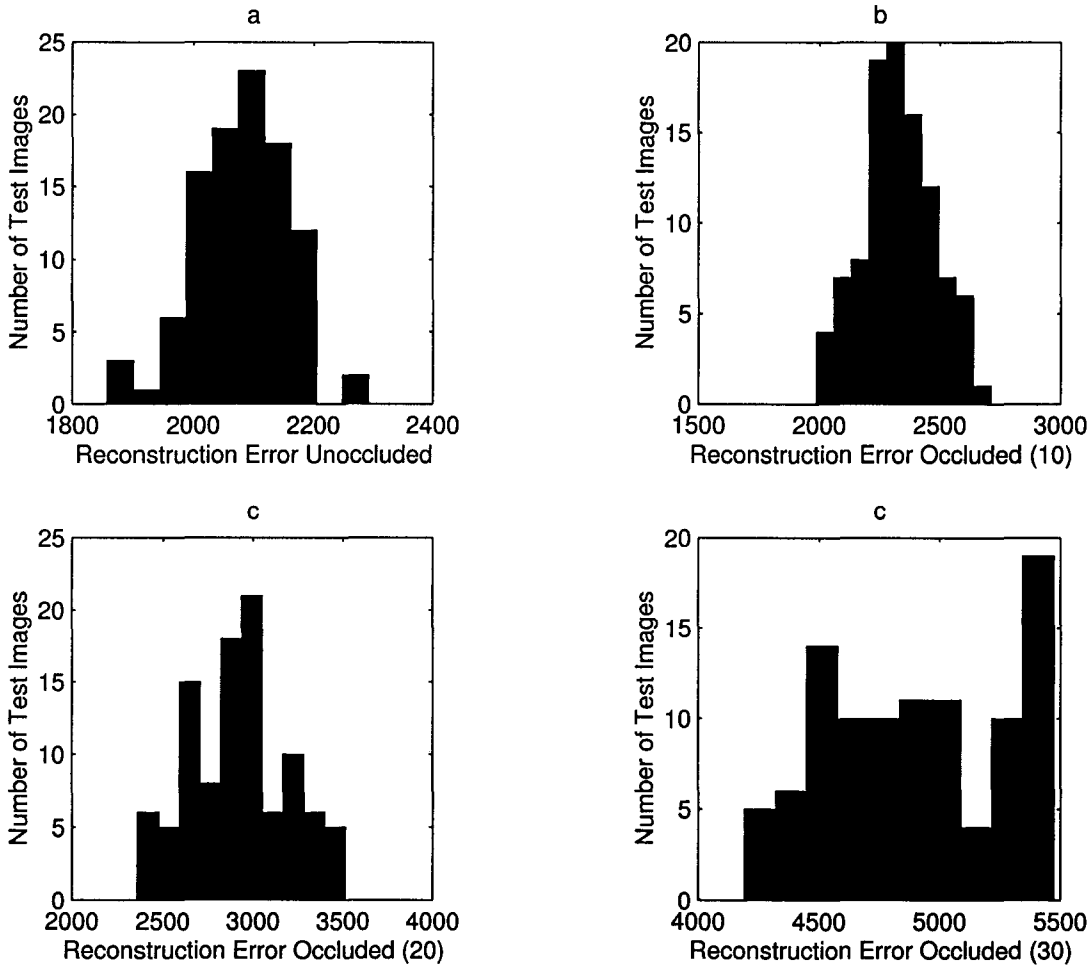

Figure 5.9: Histograms of Reconstruction Error, (a) Unoccluded, (b) Occluded (10 Pixel Square), (c) Occluded (20 Pixel Square), (d) Occluded (30 Pixel Square)

it was observed that the probability distributions for reconstruction error had similarities to that of a Gaussian distribution, but certainly were not perfect Gaussian distributions. It is likely the distribution has Gaussian characteristics due to the additive nature of the eigenvector reconstruction measure across many pixels. As an example, Figure 5.10 shows the histograms of the reconstruction error for four image sections of the metallic object visual subspace. The images were divided into the standard sixteen sections and the four sections chosen corresponded to the top four

sections. Each image section had their own eigenvectors derived from the 289 training images; the histograms were obtained from a set of 100 random images with 30 eigenvectors. As can be seen, the distributions in appearance have Gaussian characteristics particularly that of Figure 5.10(b). No outliers are present in any of the distributions either. To avoid rejecting unoccluded images falsely, a threshold of three standard deviations above the mean was chosen, which as will be shown experimentally, also allows occlusions of a reasonable size (5 % of the image) to be correctly detected for subsectioning and recombination.

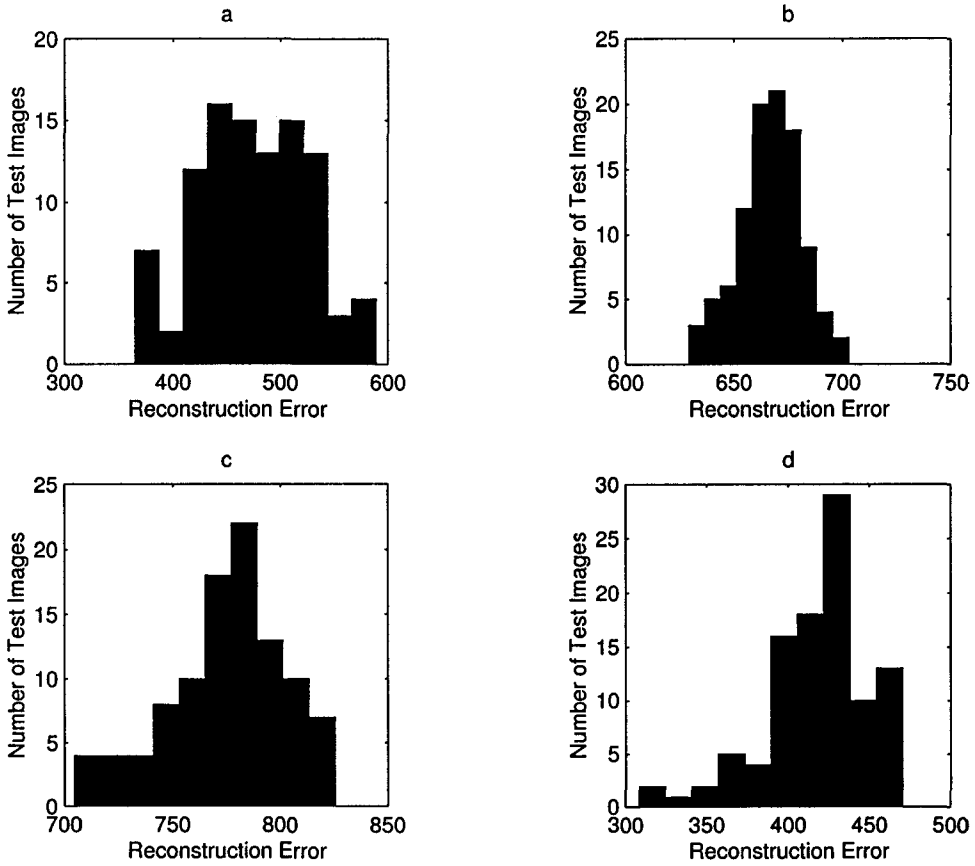

Figure 5.10: Histogram of Reconstruction Error, Individual Sections, (a) First Section, (b) Second Section, (c) Third Section, (d) Fourth Section

Depending on the choice on the number of sections, detecting an occlusion of a

given size will be less difficult for the individual sections compared to the entire image since the occlusion occupies proportionately more of the image. For our standard choice of 16 image sections for subsectioning and recombination, being able to detect occlusions of 5 % of the original image allows the detection of occlusions in the individual sections occupying only 0.3125 % of the original image (assuming there is no difference in the ability to detect occlusions proportionally between the entire image and an individual section). From our experimentation with occlusion size and the effects on accuracy, occlusions large enough to negatively impact accuracy for the global image are easily detected in the much smaller sections.

For proper implementation, each section should have its own eigenspace and hence its own set of eigenvectors for reconstruction. These will be more precise for the individual section and can be performed off-line.

5.3.5 Choice of Images for Estimating Thresholds

It is important to use a random set of images from within the movement range of a camera or object to collect the necessary statistics for determining a threshold instead of the training image set used to estimate the eigenvectors used for reconstruction. Using only the training images will give artificially low ξ_r values due to their direct use in estimating the eigenvectors. Conversely, random images will occur in positions intermediate of the training images and consequently their ξ_r values will be higher.

To illustrate this concept, Figure 5.11(a) shows the ξ_r histogram for the original 289 training images used to form the eigenvectors. Conversely the ξ_r histogram of the set of 100 images spaced randomly throughout the metallic object's movement range, is shown in Figure 5.11(b). As can be seen, the reconstruction error statistics corresponding to the training images used to form the eigenvectors is significantly lower than those corresponding to random images. Using the training images for calculating a threshold for occlusion detection could result in a threshold that rejects

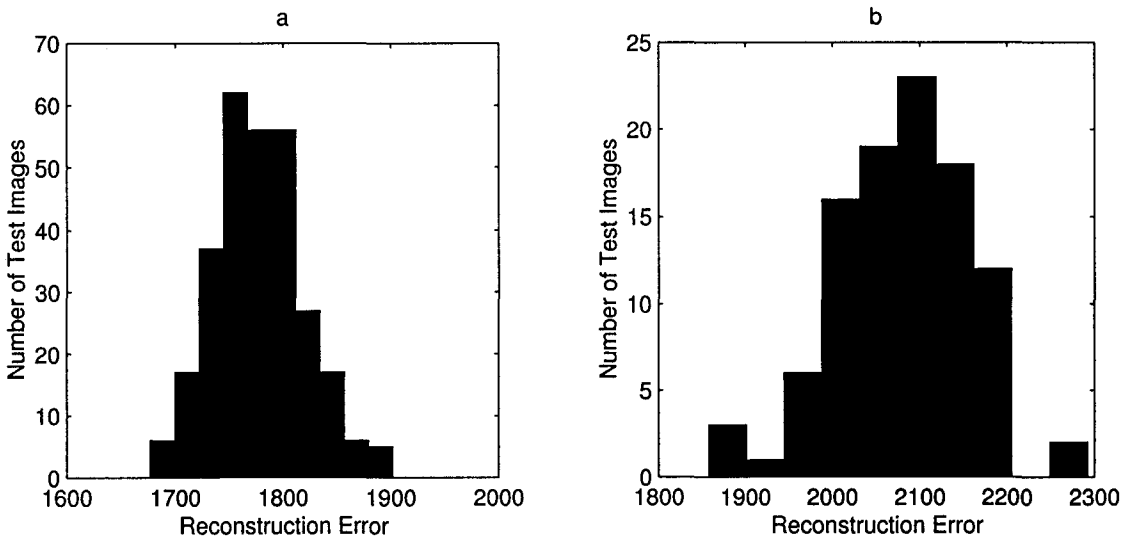

Figure 5.11: Histogram of Reconstruction Error, (a) Training Set Images, (b) Random Set

randomly distributed images as occluded.

5.3.6 Experiments with Implementation of Occlusion Detection for Subsectioning and Recombination

To illustrate the use of eigenspace reconstruction based occlusion detection for subsectioning and recombination we use the set of random images with the wrench occlusion with the metallic object subspace. Figure 5.12 shows an example image from this set together with the image divided into 16 sections. Note that the position of the wrench occlusion was fixed, while the metal object was moved below it, thus in all the test images the occlusion occupied approximately the same pixels. With sixteen sections, the occlusion was present in the right half of the sections, varying from covering a relatively small portion of the section to the entire section. The occluded sections in the image are labelled from (i) to $(viii)$ for future reference.

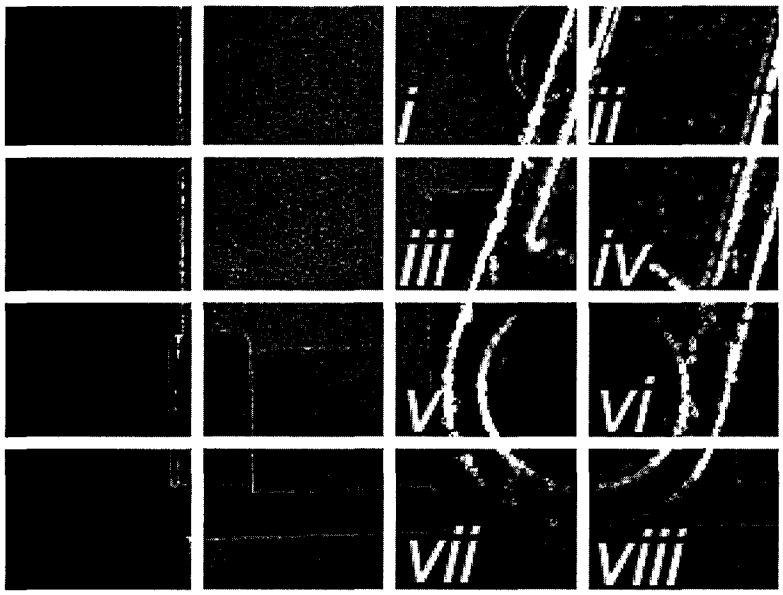

Figure 5.12: Wrench Occlusion Image

To show the effectiveness of eigenspace reconstruction to detect the occluded sections for this example, the histograms of the reconstruction error were calculated for each of the occluded sections for the random set of occluded images. The individual ξ_r values were calculated using the standard 30 eigenvectors with the eigenvectors derived for each image section. As a comparison for each section, the histogram for each section corresponding to the original unoccluded random set was calculated as well.

Figure 5.13 shows the histograms for both the unoccluded (5.13(a)) and occluded (5.13(b)) tests set for the section labelled (*i*). It can be clearly seen that there is significant separation between the two histogram distributions, thus for this section setting an occlusion detection threshold is not particularly difficult as long as all the unoccluded sections are retained. Similarly for the rest of the histograms, shown in Figure 5.14 through Figure 5.20 for sections *ii* through *viii* respectively, there is significant separation between the unoccluded and occluded distributions. Thus for these instances the occluded sections should be completely detected by the 3 SD

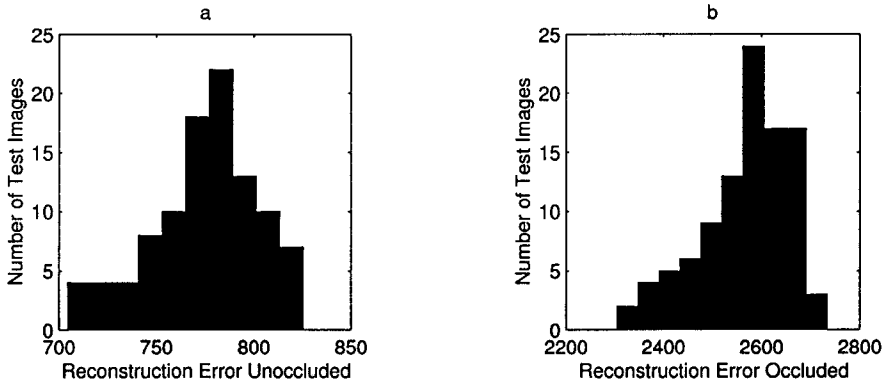

Figure 5.13: Histograms of Reconstruction Error, Section *i*, (a) Unoccluded, (b) Occluded

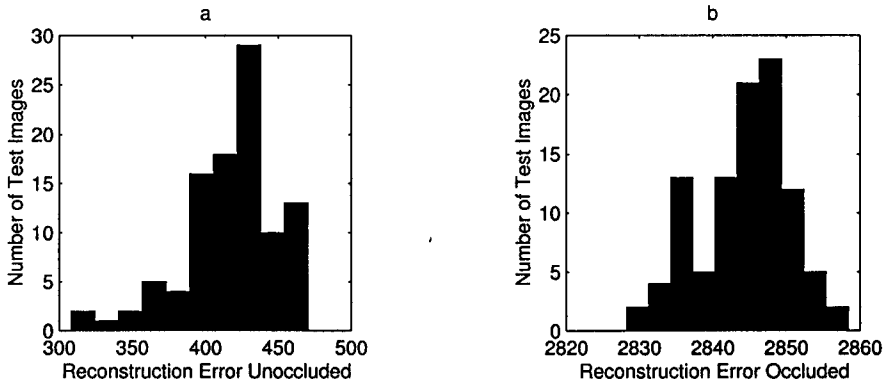

Figure 5.14: Histograms of Reconstruction Error, Section *ii*, (a) Unoccluded, (b) Occluded

threshold.

To demonstrate that our threshold works effectively for this example, all the sections for each image in the wrench occlusion set were classified according to their ξ_r. For each section, the standard deviation and the mean were calculated from the unoccluded random image set and a threshold set as described earlier at three standard deviations above the mean. Table 5.1 shows for each unoccluded section of the wrench occluded images, the preset threshold, the mean, the standard deviation, and the highest and the lowest ξ_r. Similarly, Table 5.2 shows the results for the occluded sections of the images. For this table, the sections were labelled starting with 1 for

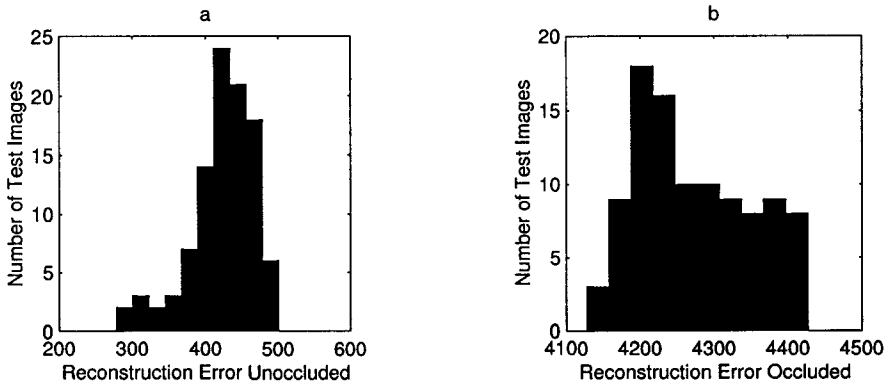

Figure 5.15: Histograms of Reconstruction Error, Section *iii*, (a) Unoccluded, (b) Occluded

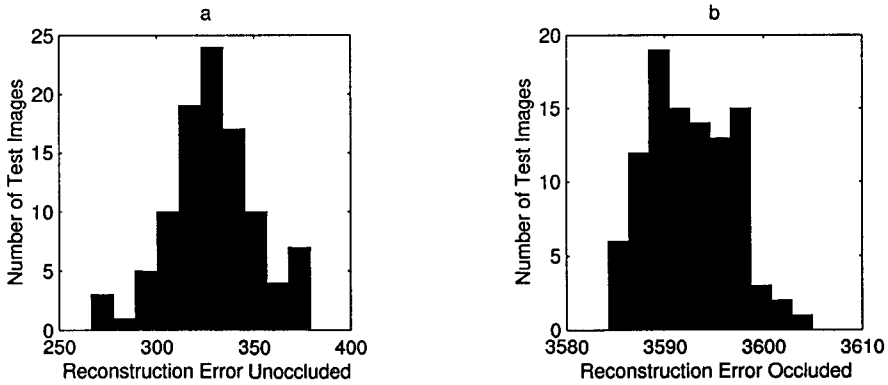

Figure 5.16: Histograms of Reconstruction Error, Section *iv*, (a) Unoccluded, (b) Occluded

the top-left section, proceeding left to right, top to bottom.

As can be seen from the tables, for the unoccluded sections, none were rejected as occluded as all the highest ξ_r values were below the threshold. Conversely, for the occluded sections, all were successfully rejected as occluded because they were all over the threshold. Note that they were all significantly (> 2 times above) over the thresholds when the means and standard deviations (thus the overall distributions) are taken into account.

Thus for this example we were able to achieve 100 % accuracy in terms of classifying the sections as occluded or not. For this example fortunately, the occlusions in

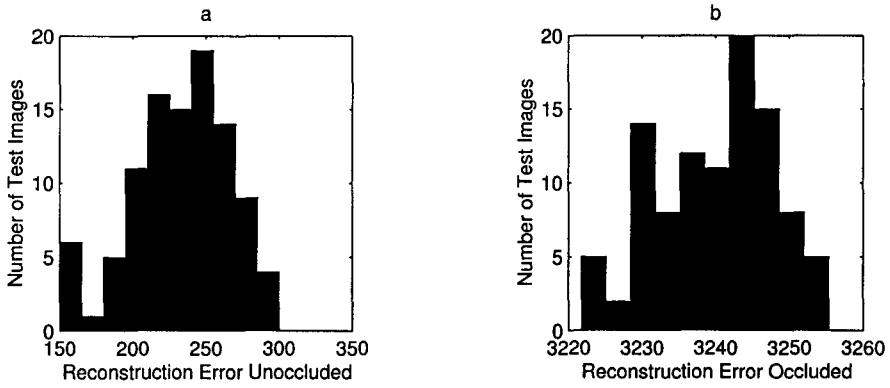

Figure 5.17: Histograms of Reconstruction Error, Section *v*, (a) Unoccluded, (b) Occluded

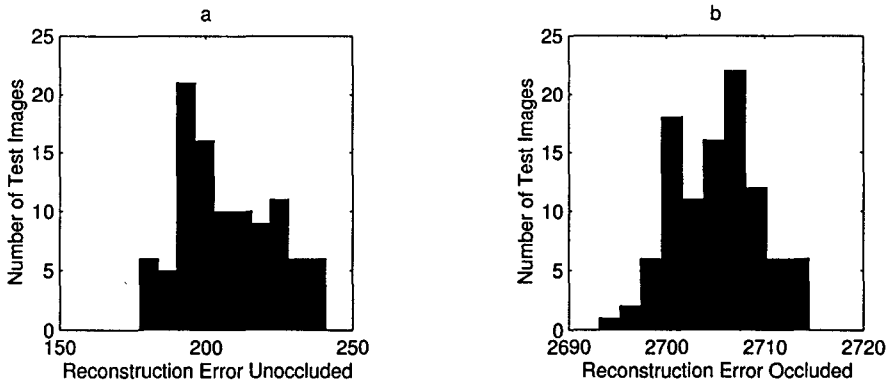

Figure 5.18: Histograms of Reconstruction Error, Section *vi*, (a) Unoccluded, (b) Occluded

the sections generally made up an appreciable portion of the sections, thus they were relatively easier to detect. For sections with slightly less occlusion coverage, there would be more of a chance to improperly classify the occlusions. These occlusions as a percentage of the entire image would be small.

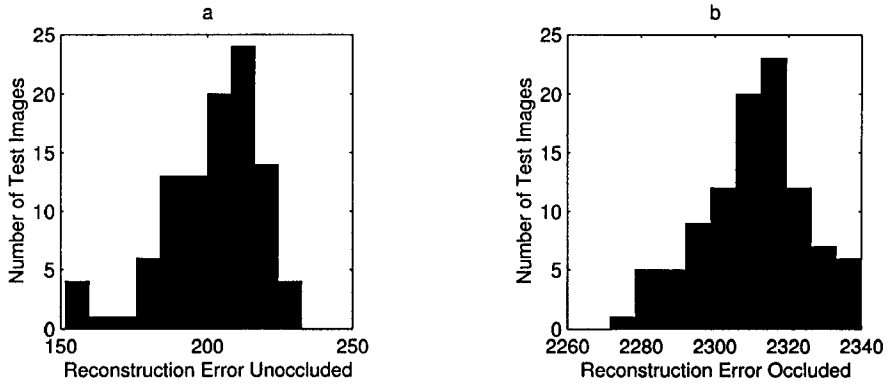

Figure 5.19: Histograms of Reconstruction Error, Section *vii*, (a) Unoccluded, (b) Occluded

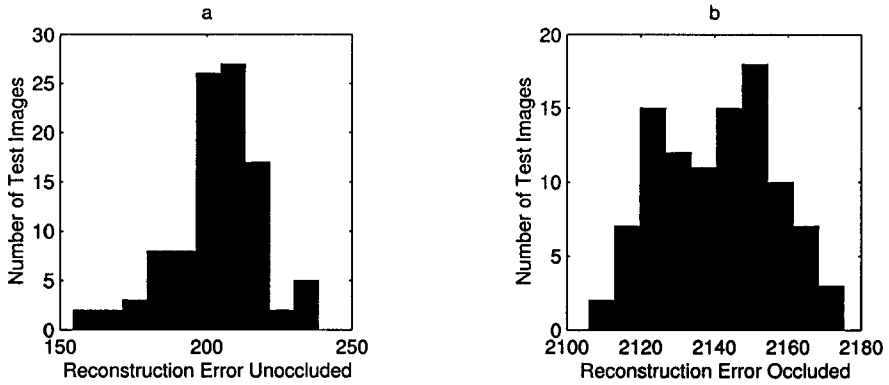

Figure 5.20: Histograms of Reconstruction Error, Section *viii*, (a) Unoccluded, (b) Occluded

5.4 Manifold Distance Measure for Occlusion Detection

An alternative measure to eigenspace reconstruction for detecting occlusions in visual subspaces is the Manifold Distance (MD) measure. This technique was first suggested by Nayar et. al. [37] as a solution for detecting manufacturing errors for printed circuit boards. For their experiment, a camera was moved over the circuit board in a path to completely cover the board with images. At specified intervals,

Section	Mean (Pixels)	Standard Deviation (Pixels^2)	Threshold (Pixels)	Highest ξ_r (Pixels)	Lowest ξ_r (Pixels)
1	475	50.4	626	589	364
2	667	14.5	711	702	629
5	521	49.8	670	606	409
6	705	18.6	761	732	653
9	477	53.3	637	578	369
10	609	54.0	771	713	487
13	278	36.8	388	361	184
14	408	48.3	553	512	300

Table 5.1: Reconstruction Error Results by Section (Unoccluded)

the MD was checked, and manifold distances beyond a threshold were taken to be occluded.

In contrast to the eigenspace reconstruction measure, which is effectively independent of the basic eigenspace positioning technique, MD requires the standard nearest neighbor search to determine location to be performed. Rather than using the position of the nearest neighbor match for a set of projection coefficients $\mathbf{Y}$, defined by:

$$\text{Nearest Neighbor } nn = arg\min \|\mathbf{y}_{new} - \mathbf{y}_i\|, \forall i = 1, \ldots, m \tag{5.43}$$

The MD measure depends upon the difference between the nearest neighbor match projection coefficients $\mathbf{y}_{nn}$ and the projection coefficients of the new image $\mathbf{y}_{new}$:

$$\text{MD} = \mathbf{y}_{new} - \mathbf{y}_{nn} \tag{5.44}$$

Section ξ_r	Mean (Pixels)	Standard Deviation (Pixels^2)	Threshold (Pixels)	Highest ξ_r (Pixels)	Lowest ξ_r (Pixels)
3	775	27.44	857	2733	2305
4	417	32.3	514	2858	2830
7	422	45.4	558	4426	4131
8	328	22.8	396	3584	3504
11	233	34.3	336	3255	3211
12	207	16.1	255	2712	2693
15	202	16.5	252	2339	2283
16	203	15.2	249	2168	2105

Table 5.2: Reconstruction Error Results by Section (Occluded)

For unoccluded images, the MD distance return should be relatively small, provided that the set of interpolated projection coefficients used to determine it generally results in low positional error. For occluded images however the MD should be larger comparatively. Occlusions will change the values of the projection coefficients corresponding to the occluded image. If the correct position projection coefficients are still matched, it is unlikely that the MD difference will stay the same or be smaller with the occluded projection coefficients due to the unpredictable effect of occlusions on each of the projection coefficients coefficients. Similarly, if another erroneous position is matched, it is unlikely that the MD distance returned will be small relatively, it will just happen to be the closest.

As with eigenspace reconstruction, larger occlusions will generally result in larger differences in the projection coefficients of the image, producing correspondingly larger MD distances. Again, this will allow for subsectioning and recombination the ability to find occlusions large enough to cause problems with positional accuracy for the entire image.

An important difference between eigenspace reconstruction and manifold distance

to consider is the computation time required for each procedure. As previously mentioned, eigenspace reconstruction depends on the size of the image in pixels and the number of eigenvectors used for reconstruction. Essentially for each eigenvector used, an inner product between the eigenvector and the image is required, followed by a scaling of the eigenvector by that inner product and finally an addition of the scaled eigenvector to the sums of the other scaled eigenvectors.

Conversely, for the manifold distance measure, the computation time depends not only on the number of pixels in the image and the number of eigenvectors, but also the size of the set of stored projection coefficients. The total time is equivalent to that required to determine position with the basic eigenspace technique. To determine the projection coefficients of the new image for searching, an inner product between each eigenvector and the image is required. This is followed by a nearest neighbor search with all the stored projection coefficients. As will be shown experimentally, increasing the number of projection coefficients by interpolation over the training image projection coefficients gives better occlusion discerning ability. Thus, depending on the number of projection coefficients and the computation time resulting from the nearest neighbor search, determining the manifold distance measure could be computationally demanding compared to using eigenspace reconstruction. Nene and Nayar's binary search method [38] could be used to reduce this computation time or alternatively the coarse search technique mentioned in Chapter 2.

For use in subsectioning and recombination compared with an entire image, the issue of computation time between manifold distance and eigenspace reconstruction is further biased in favour of eigenspace reconstruction. Because for eigenspace reconstruction, the computation time is related only to the number of pixels and eigenvectors used; dividing the image into separate sections will result in almost the same total computation time as the entire image. The only addition will be more comparison operations with the thresholds corresponding to each section. For MD, each

section will require an additional search operation compared to only one for the entire image which will result in a considerable computation increase for a large number of sections. Thus for subsectioning and recombination with a large (ie. 16) number of sections, the manifold distance measure could be computationally costly depending on the particular parameters. For the experiments in this chapter, eigenspace reconstruction with 30 eigenvectors required approximately 4,760,000 floating point operations. For subsectioning and recombination with sixteen sections and fifteen eigenvectors, approximately 95,600,000 floating point operations were required with an exhaustive search of the interpolated projection coefficients. Assuming that the nearest neighbor search could be reduced to only 5% of the exhaustive search via a coarse search of the training image projection coefficients followed by a constrained search of the interpolated projection coefficients in the matched training image vicinity, approximately 22,200,000 floating point operations would be required.

5.4.1 Example of MD Changing with Occlusion

To illustrate the concept of MD increasing with occluded images and thus the suitability as an occlusion detection measure, the metallic object visual subspace was used as an example. MD distance data was acquired using the 100 image random set, combined with different sized artificial occlusions. The occlusions consisted of zero intensity squares embedded in the center of the images. For the projection coefficients used to derive the MD results, they consisted of the 289 training image projection coefficients spaced equally throughout the part's movement range linearly interpolated to provide an additional 24 positions between the original projection coefficients in both directions for a total of 160801 projection coefficients.

Figure 5.21 shows the histograms of the MD for four different image sets. Figure 5.21(a) corresponds to images with no occlusion, Figure 5.21(b) corresponded to images with squares of 40 pixels per side embedded, Figure 5.21(c) corresponded to

images with squares of 50 pixels per side embedded and Figure 5.21(d) corresponding to squares with 60 pixels per side.

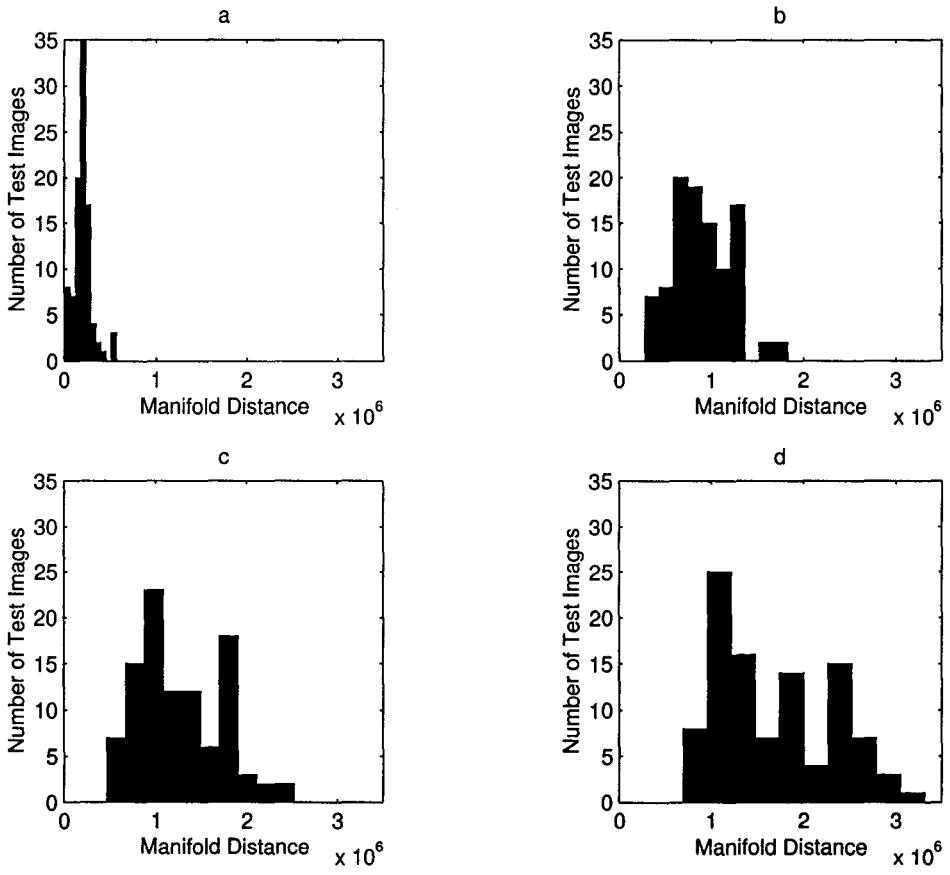

Figure 5.21: Histograms of MD, (a) No Occlusion, (b) 40 Pixel Square Occlusion, (c) 50 Pixel Square Occlusion, (d) 60 Pixel Square Occlusion

As can be seen from the example histograms, the MD measure on average increases with increasing occlusion size. However it should also be noted that for the purposes of differentiating between unoccluded and occluded images there is overlap between the distributions for the case of the smaller occlusion sets. Only the results from the largest occlusion histogram (sixty pixels per side, occupying 4.7% of the overall image size) are totally separated from that of the zero occlusion set.

Compared to the eigenspace reconstruction results from the previous section, the

MD measure is poorer in differentiating between images with small occlusions and those with no occlusion at all. This hinders the determination of an appropriate threshold for detecting occlusions for subsectioning and recombination because a threshold set to guarantee the inclusion of all the unoccluded images would accept occlusions of a size large enough to be determined to have a negative impact on positional accuracy.

An additional parameter to consider for the effectiveness of MD for differentiating between occluded and unoccluded images is the number of projection coefficients used to search for the nearest neighbor. For unoccluded images, increasing the number of projection coefficients to provide intermediate positions to match should decrease the MD as images located nearer to the new intermediate positions than the original positions should produce lower values. For occluded images, additional intermediate positions could provide an improvement if the occlusion is very small, however sizeable occlusions should push the projection coefficients away from the entire region of the correct projection coefficients. As mentioned previously however, a drawback to increasing the number of projection coefficients is the implied increase in the search time required.

To illustrate the average lowered manifold distance measure for unoccluded images with an increased number of projection coefficients, the previous experiments with the metallic object visual subspace were repeated for the random unoccluded images. The parameters were kept the same, except that instead of using the interpolated set of projection coefficients (numbering 160801 individual projection coefficients), only the original set of projection coefficients corresponding to the 289 training image were used. Figure 5.22 shows the histogram of the manifold distance for both the non-interpolated set of projection coefficients as well as the interpolated projection coefficients.

Comparing the two histogram distributions for both the non-interpolated case

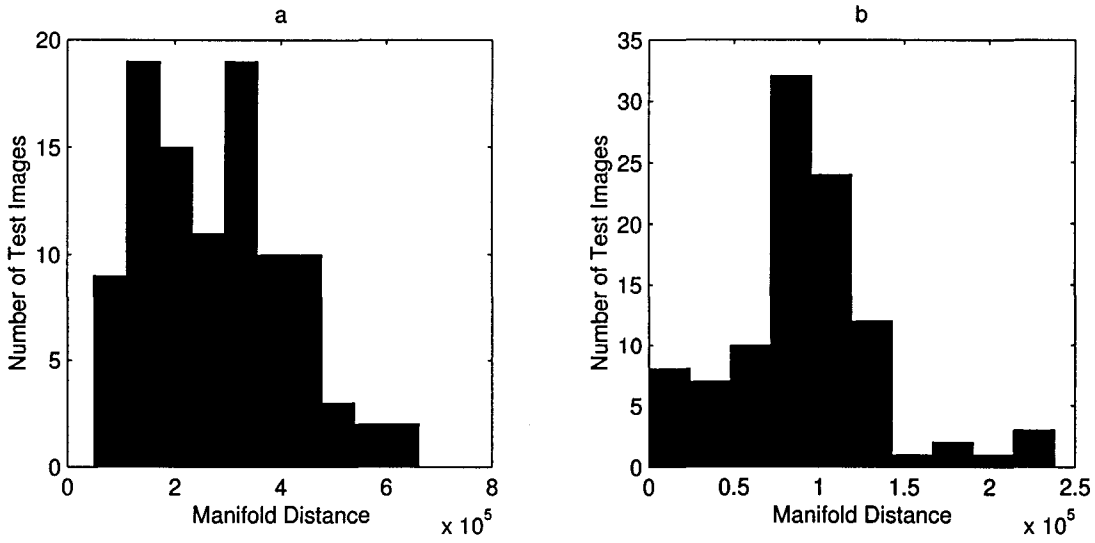

Figure 5.22: Histograms of MD, (a) No Interpolation, (b) Interpolated Projection Coefficients

and the interpolated case, it is quite clear from these experimental results that the additional intermediate projection coefficients on average reduce the manifold distance measure. As well, the worst case instances for the interpolated projection coefficients are less than half those of the non-interpolated projection coefficients. Thus for the purpose of using manifold distance for detecting occlusions, increasing the number of projection coefficients can increase performance in the sense that for unoccluded images the measure is significantly reduced, allowing lower thresholds for occlusion to be set that still includes almost all unoccluded images. It should also be noted that for this experiment that rather than increasing the number of projection coefficients via more training images, they were only interpolated projection coefficients, but there was still a noticeable improvement. If instead of interpolation, additional training images were used instead, we would expect to see an even larger improvement, as the training image projection coefficients would have no error (except for noise or inaccurate positioning) compared to the interpolation projection coefficients.

For this experiment in terms of computation time, using the additional projection

coefficients necessitates a search of 160801 projection coefficients versus only 289 for the training images.

5.4.2 MD Thresholding

To determine a threshold for occlusion detections using MD as a measure (specifically for use in the subsection and recombination algorithm) we proceeded with the same methodology as for eigenspace reconstruction. Namely that the criteria to choose the threshold was based upon almost guaranteeing the inclusion of unoccluded images in the classification of unoccluded.

To consider the typical properties of the histograms of the Manifold Distance measure for subsectioning and recombination, Figure 5.24 shows the histograms for a subset of the sections for the metallic object experiment with sixteen sections. Figure 5.23(a) shows the MD histogram for the top left section, Figure 5.23(b), Figure 5.23(c) and Figure 5.23(d) show the MD histograms for the remaining sections corresponding to the top row moving left to right. Note that the parameters used to calculate the manifold distances were the same as the previous histograms for the entire image, save that the input images were smaller.

Viewing the histograms for the four sections for the MD measure, it appears that compared to the eigenspace reconstruction histograms for the same sections that they are not as similar to a Gaussian distribution and somewhat unpredictable in terms of not being very similar amongst themselves. This difference is expected as manifold distance is not the product of many individual sums (as eigenspace reconstruction is as the sum of the errors of many pixels) but a one off measurement. Additionally it is likely that the MD value will be less for images where the camera or object is close to the original training image projection coefficients, as the interpolated projection coefficient values often will be slightly different then the actual value.

Although the histograms shown are less likely to be Gaussian compared to the

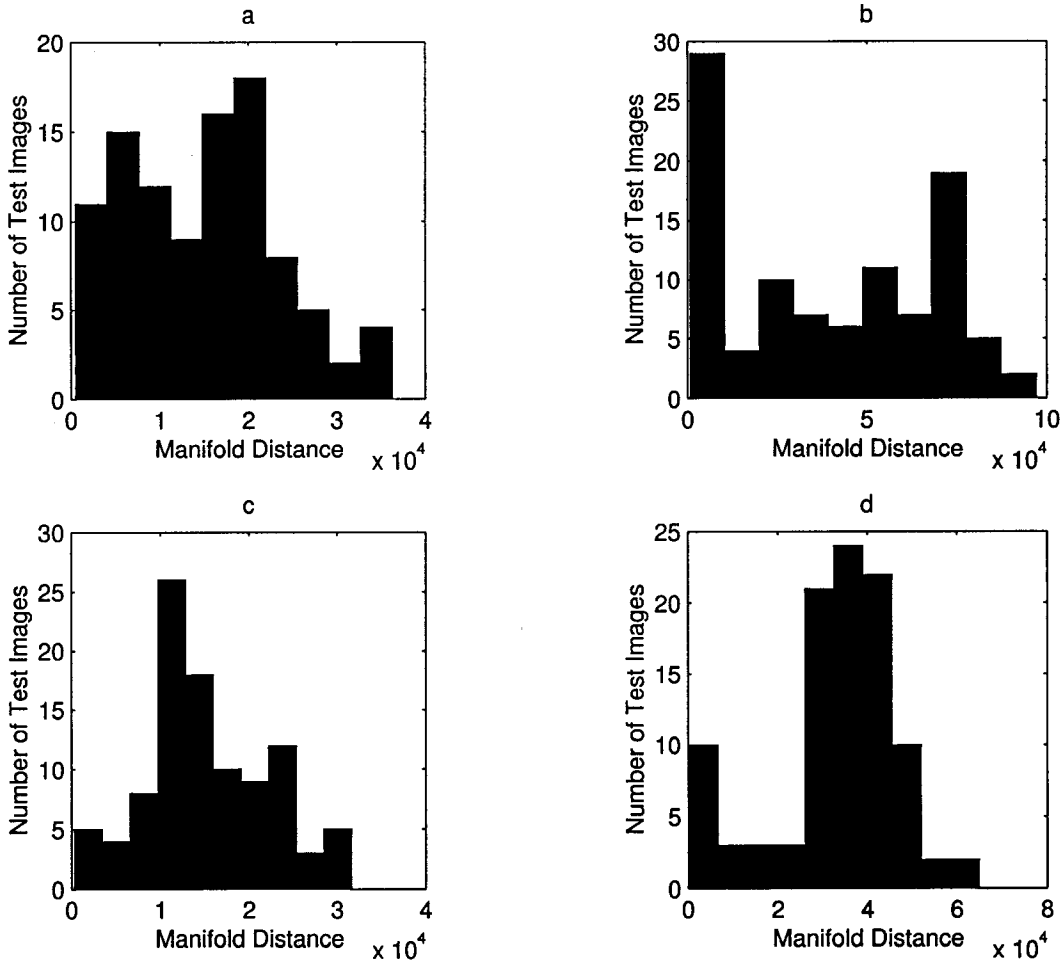

Figure 5.23: Histograms of MD, Individual Sections (a) First Section, (b) Second Section, (c) Third Section, (d) Fourth Section

eigenspace reconstruction measure, a threshold based on the standard deviation (and thus second order statistics) can still be used effectively. For the purpose of occlusion detection for subsectioning and recombination using MD, we chose a threshold based again on not excluding unoccluded image sections. To achieve this aim we chose a threshold for occlusion detection of 3 standard deviations above the mean, based upon a set of 100 random images throughout the visual subspace. Note that using more random images would increase the accuracy of the statistical parameters used

in calculating the threshold.

To demonstrate the suitability of this threshold, we once again used the occluded set of 100 random images corresponding to the wrench occlusion in front of the metallic part from the previous chapter. Note that the sections corresponding to the left half of the image were unoccluded with this example and that the right half was occluded. The mean, standard deviation and the thresholds were calculated for each section from the set of random unoccluded images. Table 5.3 compares these values with the highest and lowest MD values for the unoccluded sections of the wrench occlusion images. Table 5.4 shows the results for the occluded sections of the wrench occlusion images. Note the sections are numbered 1 to 16 based upon starting from the top, moving left to right and then top to bottom.

Table 5.3 reports the mean, standard deviation, the threshold set based on the mean and standard deviation, the highest MD value and the lowest MD value for each unoccluded section. Similarly Table 5.4 shows the same results for the occluded sections of the images. Note the sections are numbered 1 to 16 based upon starting top to bottom, moving left to right.

From the table data, it is easy to see that the occluded sections were easily detectable except for one section (namely 16); the lowest values for the occluded sections were well above the set thresholds save the one section. However for several of the unoccluded sections, the highest value for the random image set exceeded the threshold, notably for sections 9, 10 and 13. For two of the sections, increasing the threshold to four standard deviations above the mean would include all the images for two of the sections; increasing the threshold to five standard deviations above the mean would include all the images for all the sections.

Thus for the wrench occlusion, eigenspace reconstruction proved to be superior in terms of including all unoccluded sections. In terms of rejecting occluded sections, both methods were similar in that most of the occluded sections would rejected easily

except for one section for the MD measure.

Section	Mean (Pixels)	Standard Deviation (Pixels^2)	Threshold (Pixels)	Highest MD Error (Pixels)	Lowest MD Error (Pixels)
1	14974	8775	41299	36274	505
2	39120	29378	127344	97068	729
5	14494	7747	37735	30656	516
6	39341	14600	83141	68002	89143
9	15482	8087	39743	46957	272
10	34475	17467	86876	96833	1108
13	9070	8967	35971	50690	27067
14	15002	8099	39299	32099	755

Table 5.3: MD Results by Section (Unoccluded)

5.5 Eigenspace Reconstruction Versus Manifold Distance

From the experimental results, it is clear that both eigenspace reconstruction as well as manifold distance can be used effectively as a measure for occlusion detection. However it would appear that eigenspace reconstruction is a better overall approach. In terms of performance and the ability to discriminate between occluded and unoccluded image sections, only eigenspace reconstruction was able to differentiate properly between all the sections for the wrench occlusion example. Additionally in terms of time, eigenspace reconstruction is superior because the expensive search operation is not required. For occlusion detection as a front end for subsectioning and recombination, manifold distance is especially costly, with a separate search required for each section. Conversely, eigenspace reconstruction computation times are similar for an entire image versus separate sections.

Section	Mean (Pixels)	Standard Deviation (Pixels^2)	Threshold (Pixels)	Highest MD Error (Pixels)	Lowest MD Error (Pixels)
3	15296	7067	36497	2009300	408890
4	32690	13960	57176	1962100	183800
7	24199	10428	55483	2456382	168770
8	19423	3298	29317	1898100	1862500
11	19869	12335	56931	5365300	4713209
12	13337	11231	47030	772880	721890
15	28557	15852	76113	419870	178300
16	19510	13618	60364	819540	60254

Table 5.4: MD Results by Section (Occluded)

Potentially these occlusion detection techniques could also be used for other applications. For example, for a pan-tilt mounted camera used for surveillance, occlusion detection could be used to detect change in individual sections. This could also be used to constrain a face detection search to only changed sections.

5.6 Chapter Summary

Eigenspace reconstruction error was proposed as a measure for finding occlusions for subsectioning and recombination. The eigenspace reconstruction error was shown to be significantly larger for occluded images. For individual sections, smaller occlusions could be detected for subsectioning and recombination. Using a threshold of 3 standard deviations above the mean for a random set of images resulted in 100 % accuracy for the experimental example. MD was also demonstrated to be effective at detecting occlusions, but slightly more likely to misclassify than eigenspace reconstruction.

Chapter 6

Multiple Cameras and Higher Dimensions

This chapter focuses on the use of multiple cameras for enhancing accuracy with eigenspace positioning. A difficulty with using only single cameras occurs with determining position involving movement along the optical axis of the camera, due to the small amounts of image feature change. Consequently, poor performance along the optical axis results. Alternatively aligned multiple cameras can be used to circumvent the problem by providing additional information. Several techniques are suggested for performing the fusion of image information between multiple cameras and their relative effectiveness and superiority over a single camera is demonstrated via a simple example.

The remaining portion of the chapter focuses on higher dimensional eigenspace positioning, specifically 3D translational movement. The multiple camera techniques are demonstrated to be effective for improving performance in terms of accuracy over a single camera. Additionally, a technique for using less training images (the main implementation difficulty for higher dimensional eigenspaces) while maintaining accuracy is demonstrated.

6.1 Discrepancy in Accuracy in Different Directions

A visual subspace for a camera moving relative to a scene or a specific object can consist of three translational directions as well as three rotational directions. In the previous chapter the experiments dealt specifically with two dimensional translational movement of a metallic object relative to a fixed camera. Accuracy in both directions of movement of the object was similar for both the mean and the histogram distribution of the errors. However, one notable factor for this visual subspace was that the movement of the object was such that no movement occurred along the optical axis of the camera. Similarly in the third chapter, there was no camera movement along the optical axis.

Conceivably many potential applications of eigenspace positioning could rely upon translational movement along the optical axis of the camera, whether it be a camera moving relative to a fixed environment or an object moving relative to a fixed camera. As an example, the problem of determining the position of a mobile robot in a room via a camera fixed on the robot with orientation similar to the eyes of a human being would require determining the position along the optical axis of the camera as the robot moves forward. Similarly, determining the position of a robot held part in three dimensions in front of a fixed camera will require determining position with movement along the optical axis of the camera. Unfortunately, it will often be the case that determining position along the optical axis of the camera will result in significantly lower accuracy compared to translational movement in directions perpendicular to the optical axis.

The reason behind this phenomenon is the fact that the amount of image feature change (in the sense as described in the third chapter) for a camera moving along its optical axis will in most instances be significantly less than the image feature change

for movements in the other directions. Depending on the focal length of the lens, often the width and/or length of the scene area covered by the image will be less than that of the distance between the camera lens and the surface of the contents of the scene. If a camera is moved ahead towards the contents of the scene by a small amount c (maintaining the orientation of the camera), the distance between the lens and the scene contents will not be proportionally changed much, resulting in an almost identical image.

If instead the camera is moved perpendicular to the optical axis of the camera by the same amount c, the scene contents at the center of the image produce is shifted by c, resulting in a potentially large image feature change comparatively. Thus the contents of the images change position less with camera movement along the optical axis than in the other directions.

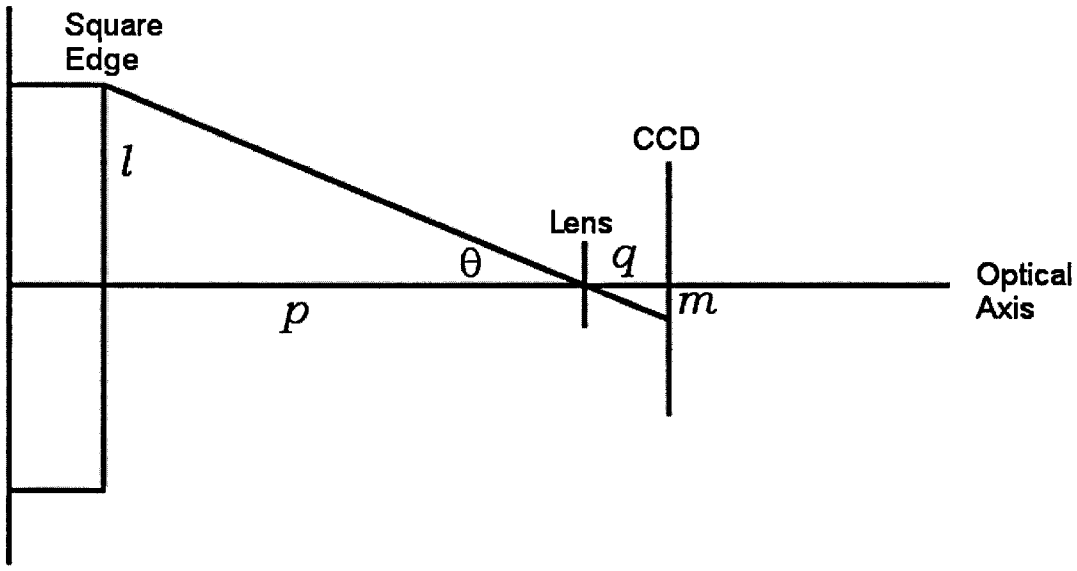

Figure 6.1: Image Feature Change Example

Figure 6.1 shows a sample situation for illustrating the dichotomy between moving forward along the optical axis and moving laterally in terms of the amount of image feature change. It consists of a camera facing a featureless white wall, save a black square in the center of the image The original distance from the camera lens to the square is p, and the distance from the lens to the CCD is then q. The edge of the square is l units from the optical axis and thus appears m units from the center of the CCD as shown in the diagram and is defined by the following formulas (assuming a pinhole representation of the camera model for simplicity):

$$\Theta = \arctan(l/p) \tag{6.45}$$

$$m = q \tan \Theta \tag{6.46}$$

Moving the camera either along or perpendicular to the optical axis will result in changes to where the edge of the square falls on the CCD (m) and thus the overall appearance of the image. Moving the camera forward c units along the optical axis changes the distance from the lens to the square to p-c, altering the position of m on the CCD because Θ changes to:

$$\Theta = \arctan(l/(p-c)) \tag{6.47}$$

Similarly moving the camera up by c units perpendicular to the optical axis also changes m because Θ changes again:

$$\Theta = \arctan((l-c)/p) \tag{6.48}$$

Consider the situation for some plausible numbers: $l = 200$ mm, $p = 5$ m, $q = 10$ mm, $c = 100$ mm. Originally the position of the square edge on the CCD m is 0.4 mm. Moving the camera forward by d along the optical axis, the value of m is changed to 0.4082 mm. Moving the camera perpendicular to the optical axis, the value of m changes to 0.2 mm (thus a change of position of the edge of the square on the CCD of 0.2 mm versus 0.0082 mm).

Thus moving forward, the change in m and thus the appearance of the square in the image will be significantly less than moving perpendicularly. Thus the overall amount of image feature change in the image will be significantly less.

Because images are only two dimensional, with three dimensional camera movement, the relationship between the image feature change and camera movement along the optical axis of the camera is not as easily described as with the convenient camera alignment of Chapter 3. For the above square on the wall example, as the camera moves along the optical axis towards the wall, the location of image features (such as the corner of the square) will change in both dimensions of the images. A way to approximately quantify this image feature change in the forward direction is to use D_f, the camera movement in the forward direction per pixel of image feature change produced by:

$$D_f = \frac{d_f \cdot a}{\sqrt{(p_1 - p_2)^2 + (q_1 - q_2)^2}} \tag{6.49}$$

where d_f is the distance between training images in the forward direction and a is the number of d_f increments between the two training images and p_1, q_1 represent the coordinates of an image feature in one of the training images and p_2, q_2 represent the coordinates for the other training image (refer back to Figure 3.7).

Note that unlike for movement perpendicular to the optical axis, this value will vary depending on the pixel location of the measured image feature within the training images and will be highest in the corners of the image. A corner located at the center

of the image will not change in terms of pixel coordinates with movement along the optical axis. For consistency, measuring D_f in this chapter was performed by selecting an image feature in the corner of the training images, thus the average value of D_f across the image will be higher than the estimated D_f value (since greater image feature change results in smaller D_f). Thus we only use the D_f value as a guideline towards eventual performance. It will be shown experimentally that the relatively lower amounts of image feature change along the optical axis of camera movement as indicated by D_f compared to the D value for other directions of camera movement coincides with significantly lowered accuracy in camera movement along the optical axis compared to the other directions.

Another factor with regard to accuracy to consider is the effect on the eigenvectors used for determining position. Since the eigenvectors for a visual subspace are formed with regards to minimizing the mean reconstruction error, it is likely that the first few eigenvectors will be more responsible for minimizing the reconstruction error with regards to the greater visual change in the horizontal direction. Thus using a limited number of eigenvectors could possibly not encompass those responsible for describing the changes in the visual subspace for forward movement and correspondingly responsible for allowing the determination of forward position.

As a simple visual example, consider the three images shown in Figure 6.2. They were obtained by mounting a camera on the XY table within our laboratory, aligned such that the optical axis of the camera was perpendicular to the wall. Figure 6.2(a) shows the image obtained with the original position of the XY table. Figure 6.2(b) shows the image that results from moving the camera laterally 20 cm on the XY table. Figure 6.2(c) shows the image that results from instead moving the camera 20 cm forward.

Clearly the image features change more with lateral movement, confirming the earlier discussion. For this example, the lateral camera movement per pixel of image

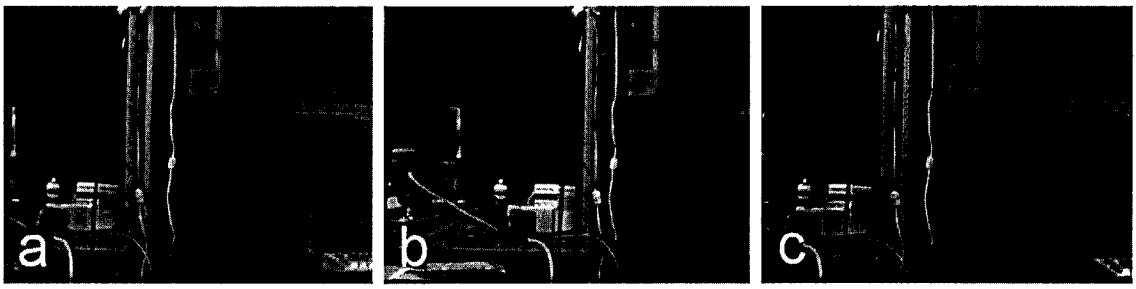

Figure 6.2: Illustration of Pixel Change, (a) Original Position, (b) Image after Lateral Movement, (c) Image after Forward Movement

feature change D_l will be defined by:

$$D_l = \frac{d_l \cdot a}{|q_1 - q_2|} \tag{6.50}$$

where d_l is the distance between adjacent training images in the direction of lateral camera movement, a is the number of increments between two training images and q_1 represents the horizontal pixel coordinate of an image feature in the first training image, and q_2 for the second training image. Note that this exploits for this example image feature change that is almost exclusively in the horizontal pixel coordinate for lateral camera movement.

For this example, D_l equals 2.1 mm per pixel of image feature change, compared to an estimated D_f value of 13.2 mm per pixel of image feature change (calculated with a corner of the keyboard in the lower right regions of the training images). Because of this difference we would expect a significant difference in accuracy. To illustrate this, 289 equally spaced training images (17 by 17) were acquired with the camera moved through a 20 cm by 20 cm square via the XY table. The axis of the XY table movement was aligned such that one direction was parallel to the optical axis of the camera, and the other perpendicular. Figure 6.3 shows a selection of four images from the training set corresponding to the four corner positions of the XY table. The scene contents are approximately 4 meters from the lens of the

camera, with no objects in the immediate foreground. The 289 images were used to form both the eigenvectors $\mathbf{E}$ and the set of projection coefficients $\mathbf{Y}$ as described in Chapter 2. A linearly interpolated set of projection coefficients was produced consisting of 801 by 801 positions spaced equally throughout the movement range (0.25 mm between each position). To test the accuracy of the positioning, a set of 100 images was obtained spaced randomly throughout the range of motion of the camera. The projection coefficients were produced with fifteen eigenvectors as more eigenvectors produced negligible increases in accuracy. As in the previous chapters, the error in both directions was determined by comparing the actual known camera position of the random image in each direction with that found by the nearest neighbor search algorithm.

Figure 6.4(a) shows the histogram of the absolute error of the random image set in the forward direction; Figure 6.4(b) shows the histogram in the lateral direction. The absolute mean position error in the forward direction was 6.25 mm (50 % of the distance between training images) and 0.40 mm in the lateral direction (3.2 % of the distance between training images). As can be seen from these results, accuracy in the forward direction was considerably worse than that in the lateral direction, both in terms of mean absolute error (more than ten times larger) and the distribution of the error. Thus a large differential in the estimates for camera movement per pixel of image feature change in the two directions coincides with a large differential in the absolute mean errors between the two directions.

Figure 6.5 shows the first eight visualized eigenvectors corresponding to the visual subspace. Notice that comparing the eigenvectors with the visual features of the scene, that for these first eight eigenvectors, the focus is clearly on those features dealing with lateral movement.

Thus for applications such as determining the position of a robot relative to a room or determining the position of an object relative to the camera, the use of only

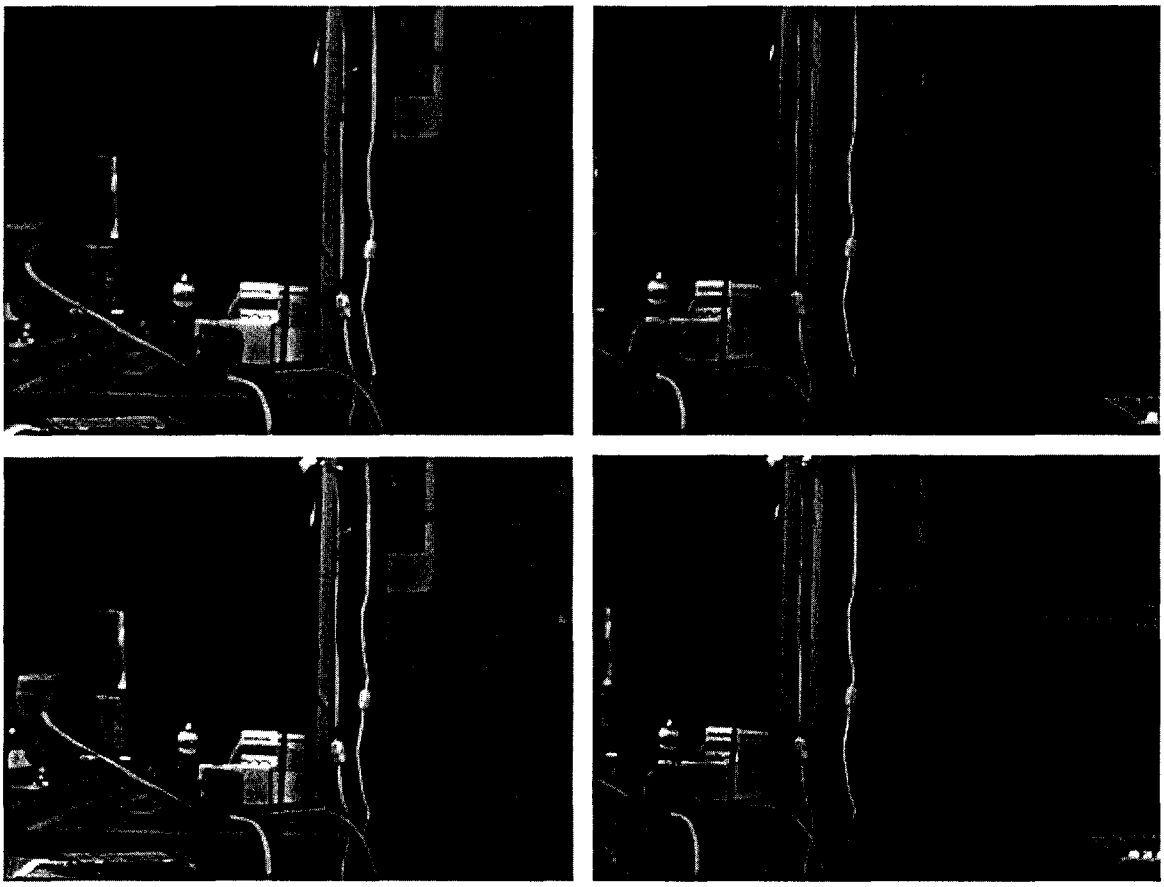

Figure 6.3: Images Illustrating Range of Camera Motion

one camera for determining position along the direction of the optical axis of the camera can cause difficulties in terms of the elevated error along the optical axis due to the difference in image feature change between the directions of camera movement.

6.2 Increasing Accuracy with Multiple Cameras

From the previous section it was shown that in certain circumstances, eigenspace positioning can have unequal rates of error for different directions of camera movement (and correspondingly for object movement as well) based upon the large differentials in image feature change.

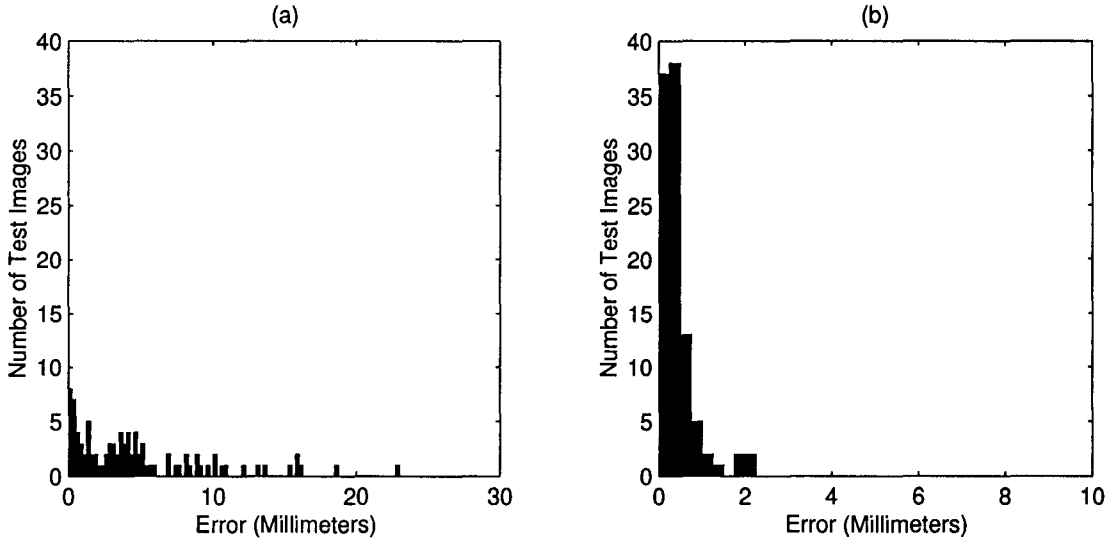

Figure 6.4: Histograms of Absolute Position Error, (a) Forward Direction, (b) Lateral Direction

For potential applications of eigenspace positioning, such as the mating of parts via a robotic end-effector, similar accuracy will be required in each direction of movement.

Since the previous experiments showed that determining position for movement parallel to the optical axis of the camera results in poor accuracy, an obvious solution would be the use of multiple cameras aligned in different directions, appropriate for the range of motion of the given application to compensate for the lack of image feature change in the direction of the optical axis.

For the previous experiment of determining position of the camera relative to the room in two translational directions (similar to that of a small scale mobile robot), the use of another camera could be used to reduce error.

To increase accuracy in the movement direction corresponding to the optical axis of the first camera, a second camera could be placed such that its optical axis is within the plane of the two dimensional camera movement, but perpendicular to that of the optical axis of the first camera (as shown in Figure 6.6). Thus for the second camera, movement along the optical axis of the first camera will produce a large amount of

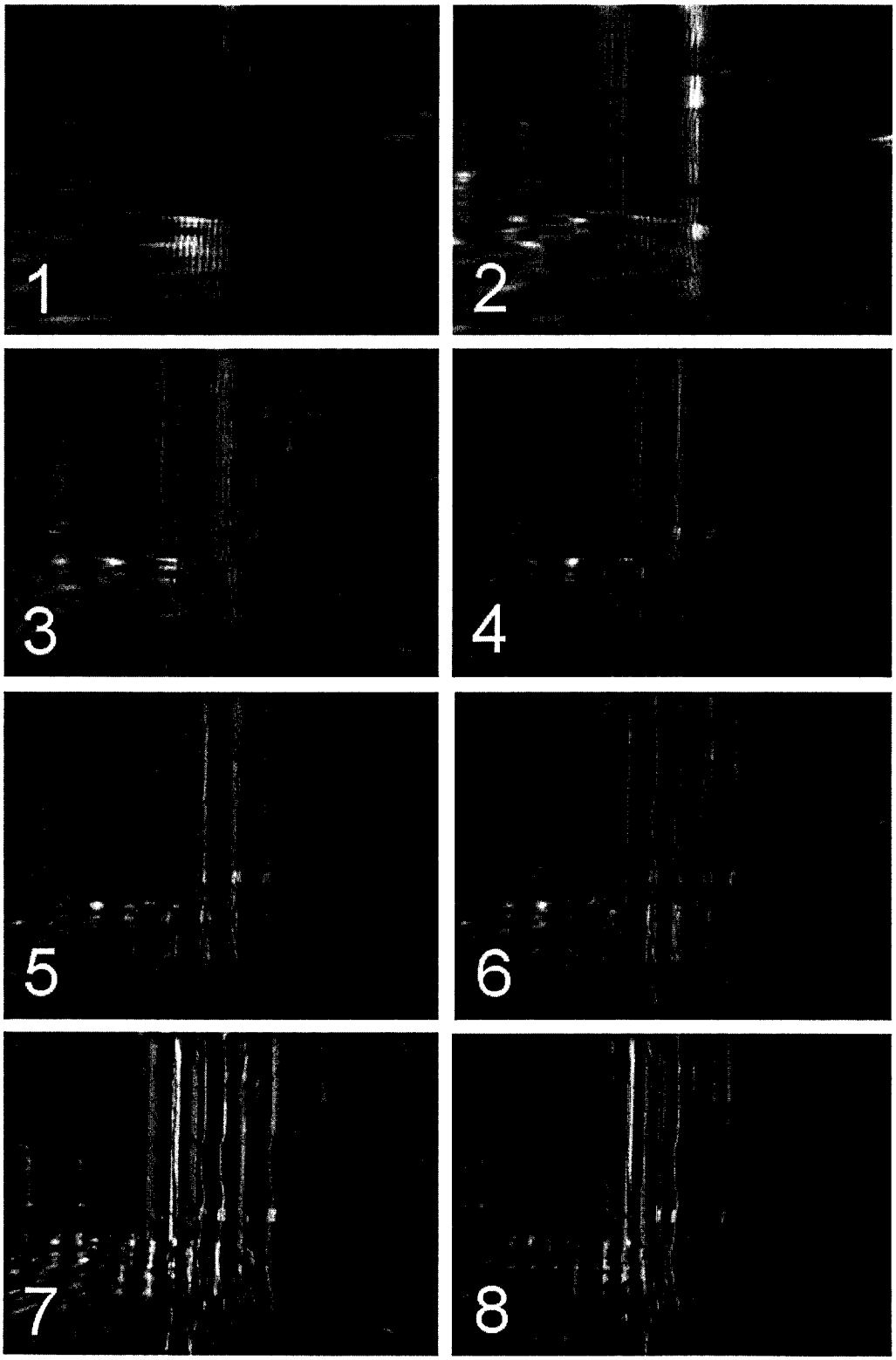

Figure 6.5: First Eight Visualized Eigenvectors

image feature change in comparison to that of the first camera. This should allow position in this direction to be identified accurately with the second camera.

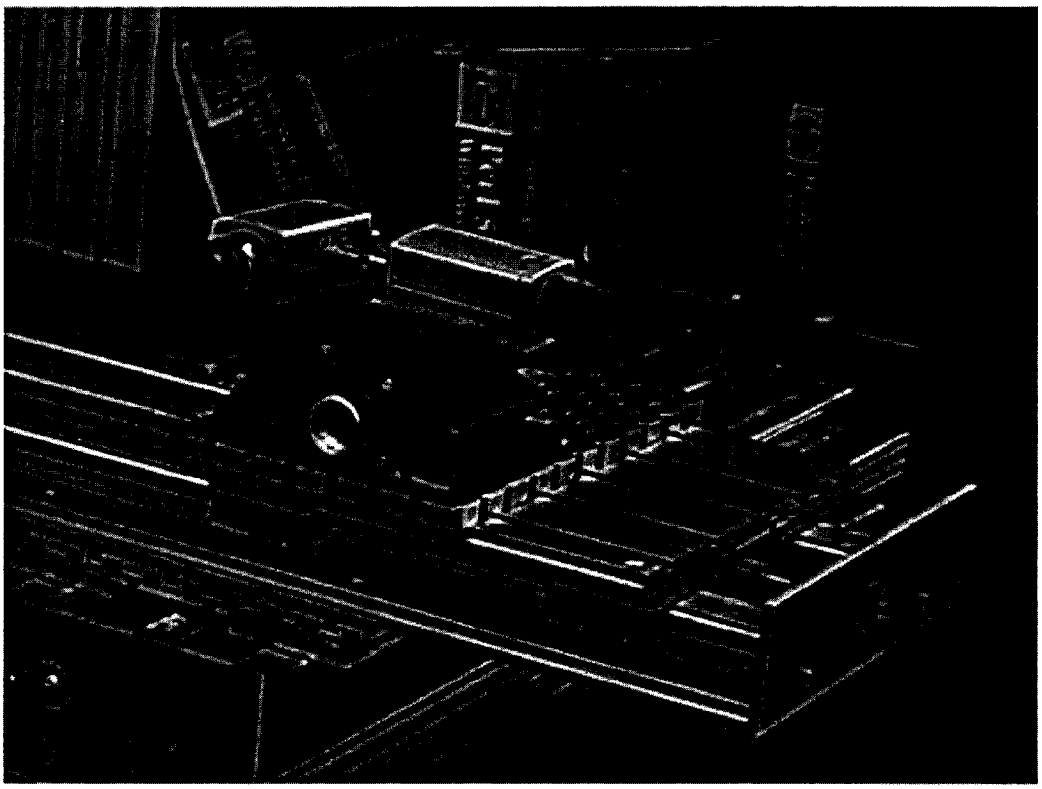

Figure 6.6: Dual Camera Setup

To illustrate this approach, for the first experiment in this chapter, a second camera was also used, aligned 90 degrees from the first, mounted also on the XY table. Training and random images were obtained at the same positions as the first camera.

Figure 6.7 shows the four corner images from the second camera that correspond to the corners of the translational movement range of the camera. The D_l and D_f values (calculated with a corner of the small black square in the training images) were calculated as 2.56 mm per pixel of image feature change and 15.8 mm per pixel of image feature change respectively. Thus as with the first camera, there was a

significant difference between the image feature change in the forward (along the optical axis) and lateral directions.

The distance between the contents of the scene and the second camera was approximately 5 m. An eigenspace was derived in the same manner as the first camera, using the 289 training images spread over the 20 cm by 20 cm movement range. Again the projection coefficients corresponding to the original training images were linearly interpolated to produce a set of 801 by 801 projection coefficients. Figure 6.8(a) shows the histogram of the absolute error in the forward direction of the second camera for the 100 random images; Figure 6.8(b) shows the histogram of the absolute error for the lateral direction. The absolute mean error in the forward direction was 1.75 mm (14 % of the distance between training images); in the lateral direction it was 0.17 mm (1.36 % of the distance between training images). Compared to the results of the first camera, the histograms and the errors were somewhat different as is usually the case with eigenspace methods however the same pattern arose with regards to the dichotomy between the positional error in the forward and lateral directions as would be expected with differing D_l and D_f values.

6.2.1 Decoupled Cameras

The simplest manner for utilizing the additional image information from multiple cameras to increase accuracy is to specify one camera responsible for a direction(s) to which it is most suited towards for high accuracy. The other cameras would subsequently be responsible for the remaining impaired directions of camera or object movement. This technique is termed *decoupled cameras*, since for each direction, the other cameras have no direct input on the position determination along it. This can be accomplished via the nearest neighbor search, whereby the closest projection coefficients are returned. The forward and lateral position for these projection coefficients

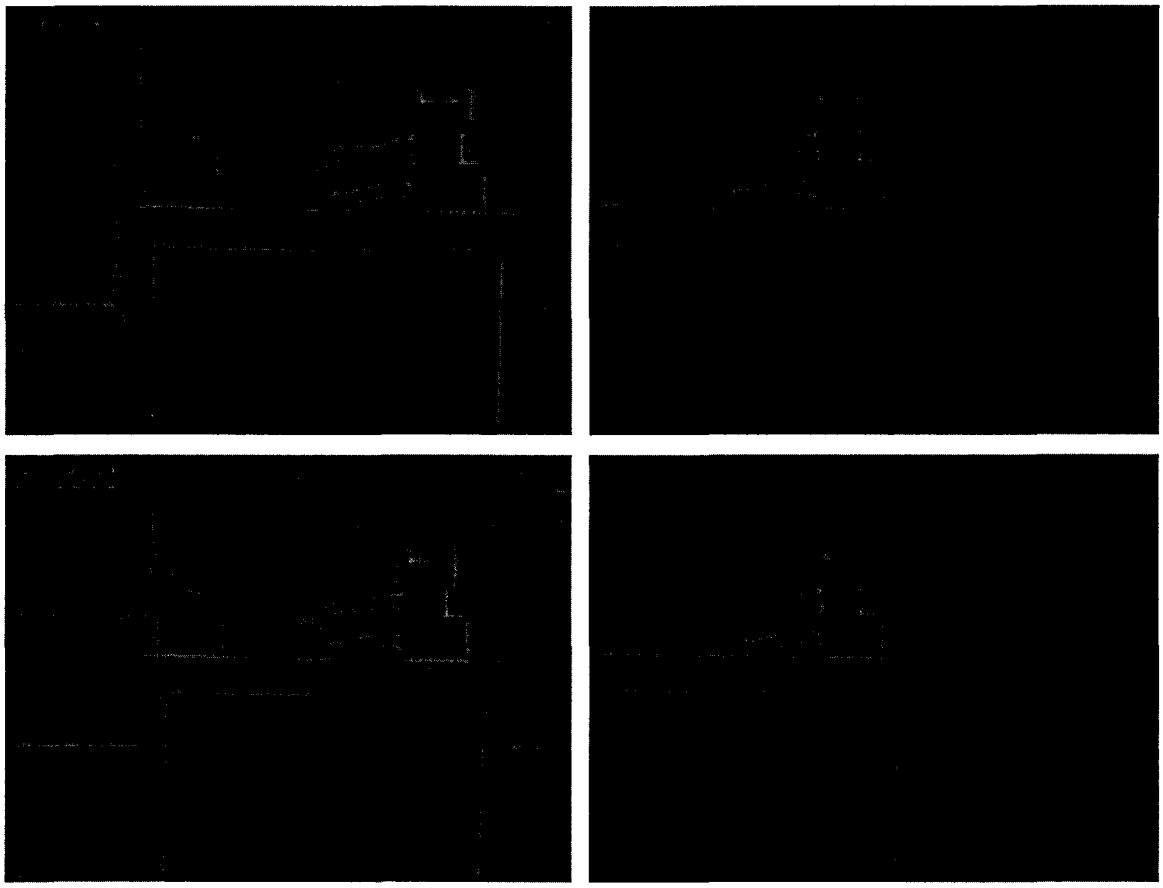

Figure 6.7: Images Illustrating Range of Camera Motion, Second Camera

are known separately and thus only one of them used. One drawback for this approach is that for each application, the designer must carefully choose the alignment of the cameras and specify which is responsible for which direction. This goes against the normal ethos of eigenspace positioning whereby the designer avoids the use of heuristics and simply lets the system perform all the learning necessary for position determination.

For the previous experiments with two cameras facing in perpendicular directions, it is easy to choose each camera being responsible for one direction, namely the lateral direction for each camera. Figure 6.9 shows the histograms of the absolute error with the combined lateral movement information for both cameras, with 6.9(a) being the

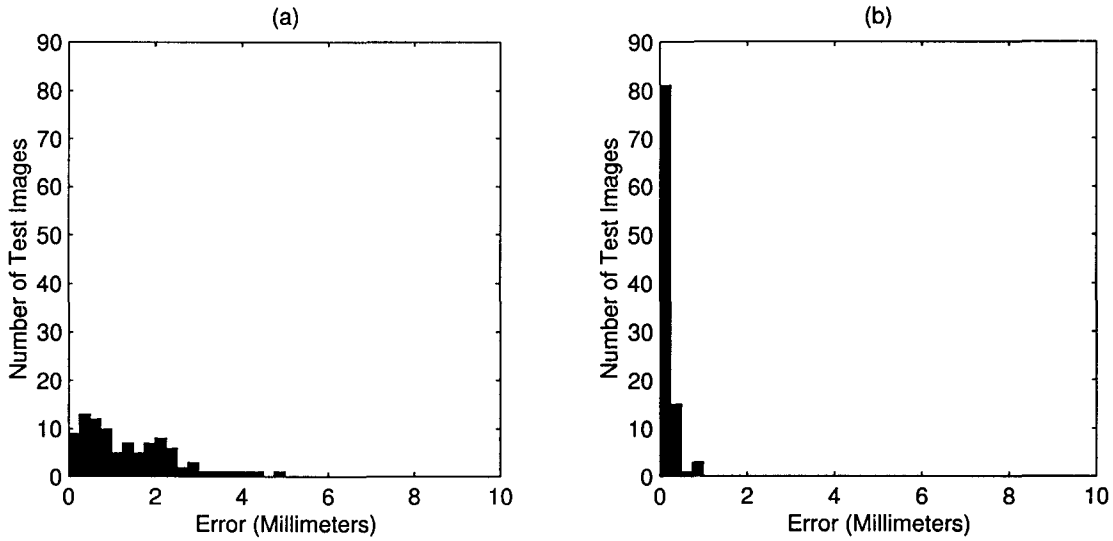

Figure 6.8: Histograms of Absolute Position Error, Second Camera, (a) Forward Direction, (b) Lateral Direction

lateral direction of the first camera and 6.9(b) being the lateral direction of the second camera. The absolute mean error for the lateral direction of the first camera was 0.40 mm (3.2 % of the distance between training images); for the lateral direction of the second camera it was 0.17 mm (1.36 % of the distance between training images). Compared to the results of both cameras separately, the overall accuracy across the entire movement range is much improved.

Thus for many applications, especially involving movement in the same direction as the optical axis of the camera, the use of decoupled cameras can improve the accuracy achievable by arranging a camera to have a comparatively large amount of image feature change in the appropriate direction where for another camera the amount of image feature change is less. One advantage of decoupled cameras is the fact that the method is easily implemented in parallel, with multiple computers controlling one camera each. Since the information from each camera is independent of the others, each computer could process each image and then report back via a network only the position in the direction(s) that it is responsible for.

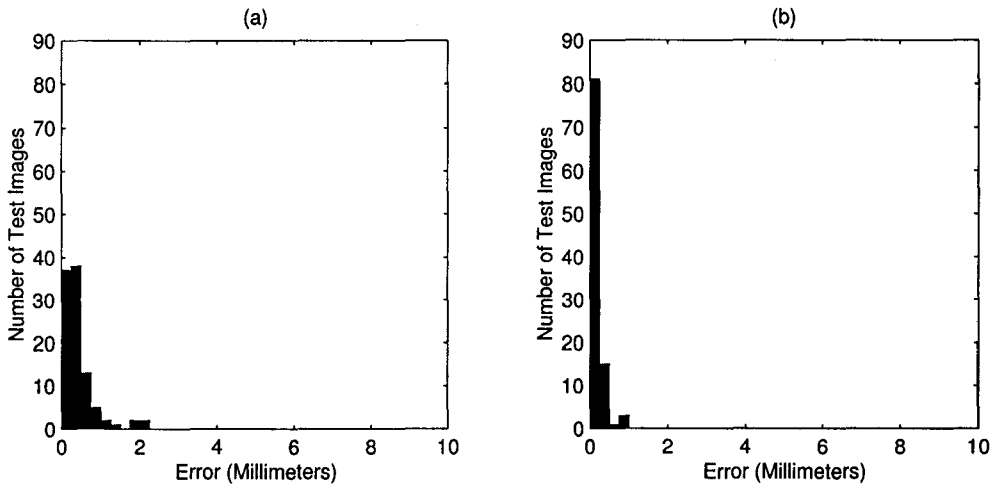

Figure 6.9: Histograms of Absolute Position Error, Decoupled Cameras Lateral Direction, (a) First Camera, (b) Second Camera

6.2.2 Combined Image Information: Fused Images

A difficulty mentioned previously with the decoupled cameras approach to using multiple cameras is the need to arrange the cameras with regards to the directions and then specify which camera is responsible for determining position in a certain direction. This adds additional complication compared to the simplicity normally afforded by appearance based methods over model based computer vision methods based on heuristically chosen geometric features.

Ideally one would rather methods that could combine the image information to provide increased accuracy in different directions using multiple cameras without requiring the specification of responsibility of a direction for each camera. This would in a sense be a sensor fusion operation between the data from the different cameras.

A simple method for accomplishing this is to fuse the images x_{i_a} and x_{i_b} from both (or potentially more) cameras (termed a and b) into a single super image $x_{i_{a+b}}$ by concatenating the image vectors:

$$x_{i_{a+b}} = \begin{bmatrix} x_{i_a} \\ x_{i_b} \end{bmatrix} \tag{6.51}$$

These concatenated images would then be used to determine position in all directions with a single eigenspace. Although the correlation between the pixels of combined images will likely not be as large as for the individual images, the eigenvectors that result should be such that position in both directions can be determined accurately.

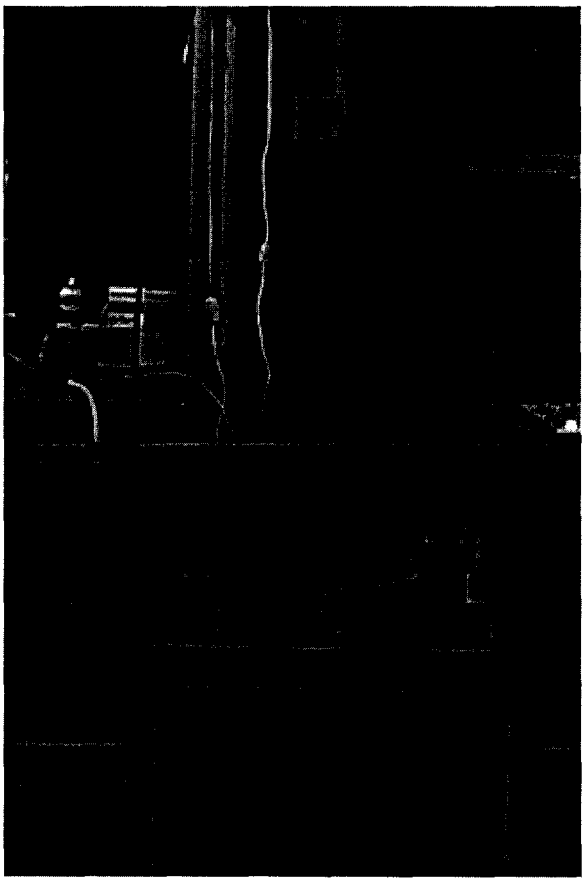

Figure 6.10: Sample Fused Image

To experiment with this procedure, the two previous sets of images corresponding

to the two cameras mounted on the XY table in the laboratory were fused into a single image, for both the training images as well as the random set of images. Figure 6.10 shows a sample of the fused images. Thus where previously for the single camera images, camera movement resulted in different amounts of image feature change with different camera directions, here at least a section of the image had a comparatively high amount of image feature change for each direction.

The standard training set size of 289 images was used to both determine the eigenvectors as well as the projection coefficients as in previous experiments. The projection coefficients were interpolated to provide 801 positions in each direction, as in the previous experiments.

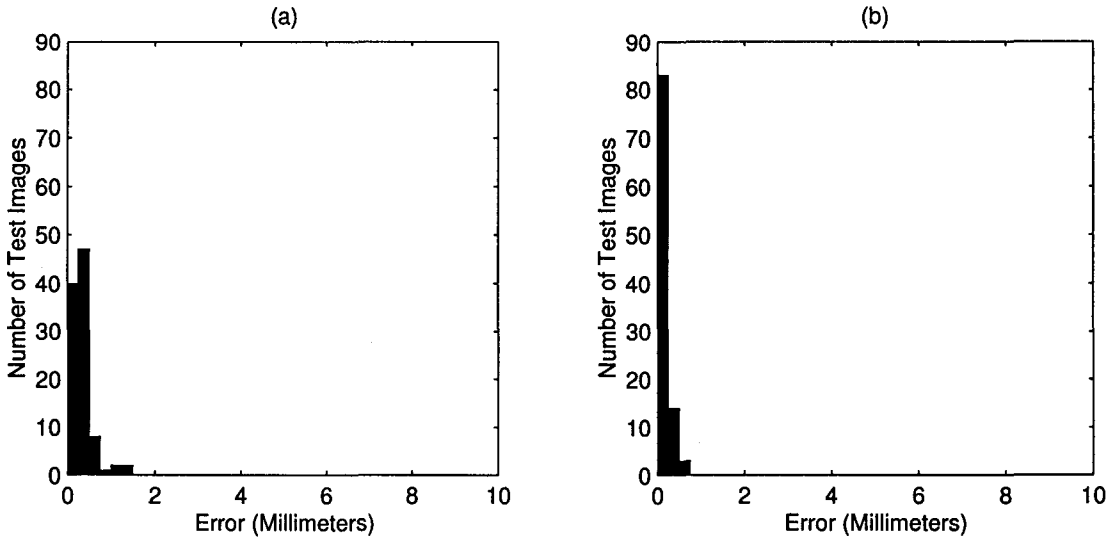

Figure 6.11: Histograms of Absolute Position Error, Fused Images, (a) Lateral Direction First Camera, (b) Lateral Direction Second Camera

The same number of eigenvectors (fifteen) used for the single images was used in the nearest neighbor search. Figure 6.11(a) shows the histogram of the absolute error corresponding to the lateral direction of the first camera; Figure 6.11(b) shows the histogram of the absolute error corresponding to the lateral direction of the second

camera (forward direction of the first). The absolute mean error in the lateral direction of the first camera was 0.31 mm and 0.15 mm in the lateral direction of the second camera (2.48 % and 1.20 % of the distance between training images respectively). Compared to previous results using decoupled cameras by themselves, the results were similar but slightly superior to the decoupled cameras results.

To illustrate partially how the data is combined with the fused images, Figures 6.12 and 6.13 show the first eight of the eigenvectors corresponding to the concatenated eigenspace (produced as described in Chapter 3). The visualized eigenvectors have several interesting aspects. For the first few eigenvectors, either the top or bottom section is dominant with features that appear to respond to lateral movement. In each such eigenvector, the opposite section has features that appear to correspond to forward movement. Thus for the total eigenvector the two halves reinforce each other in terms of the projection coefficient response as the forward movement of one section is the lateral movement of the opposite. It should be remembered that the eigenvectors are derived based on minimizing the reconstruction error for the entire fused images.

In terms of overall accuracy, the fused images give excellent performance (compared to that of a single camera) considering the fact that no designation of direction is required as with decoupled cameras. In fact it would appear as opposed to decoupled cameras, with the as low or lower absolute mean errors that an added robustness is achieved. This could possibly be due to more accurate interpolation as the fused images could produce a smoother projection coefficient manifold in both directions of camera movement, thus leading to higher accuracy. Thus it would appear that the additional information (via the increased image feature change) provided by the second camera is exploited usefully with the fused image approach. Potentially such approaches could prove useful for increasing the accuracy of eigenspace based face recognition with multiple cameras in situations such as airports where dedicated

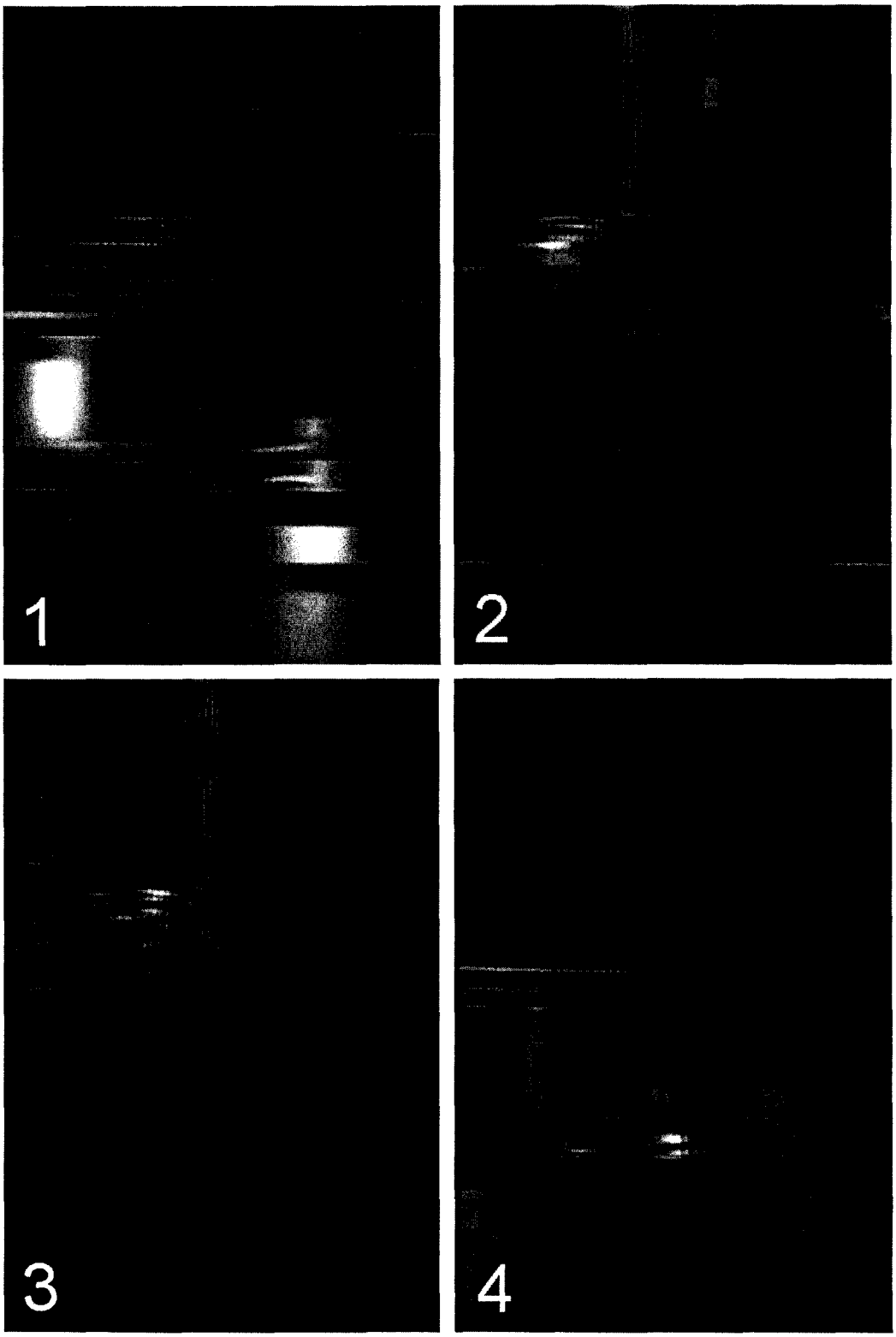

Figure 6.12: First Through Fourth Visualized Eigenvectors, Fused Images

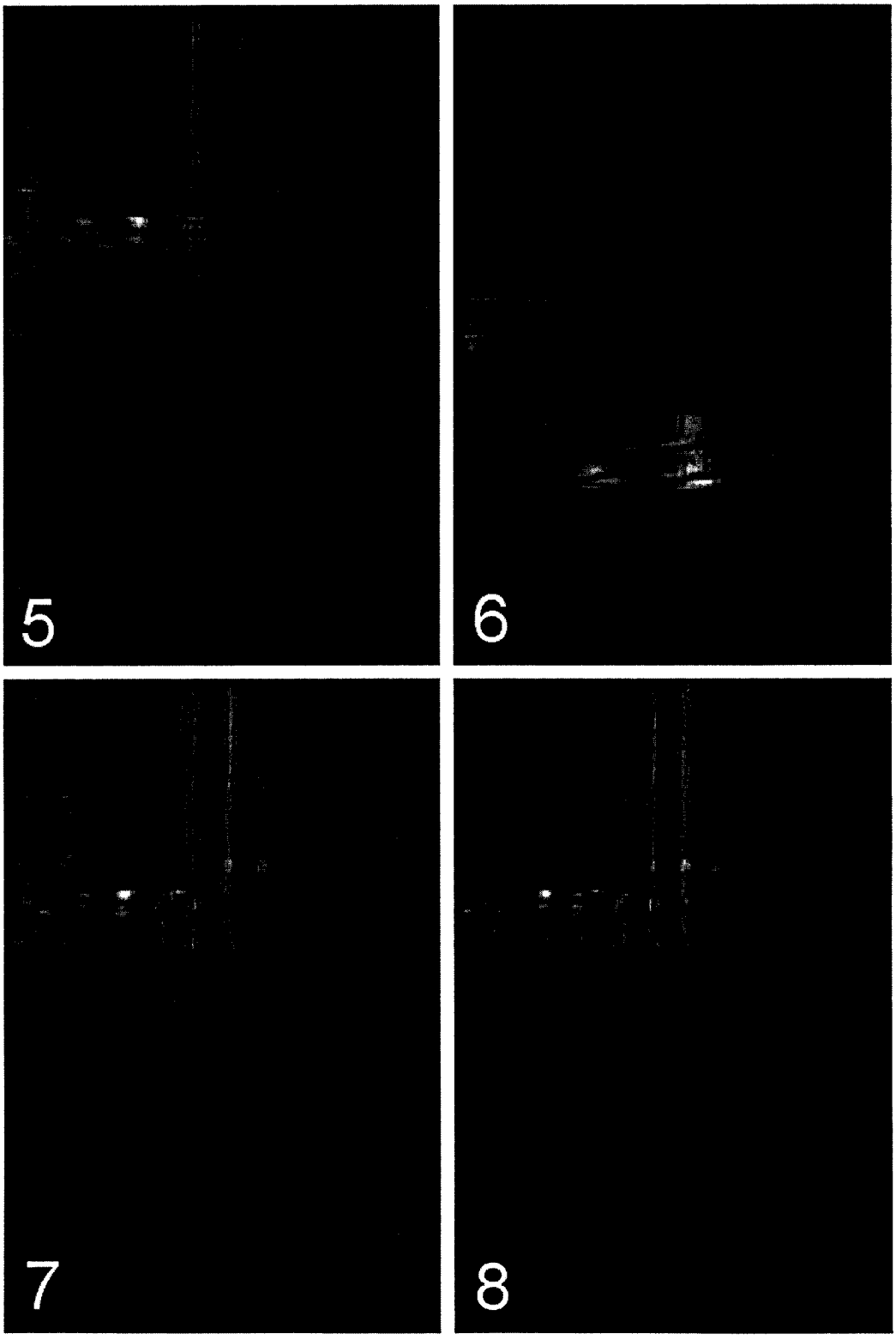

Figure 6.13: Fifth Through Eighth Visualized Eigenvectors, Fused Images

multiple camera setups could be arranged.

In a sense the fused images approach has similarities to the use of omnidirectional cameras for determining the position of mobile robots [8][17]. For omnidirectional cameras, the entire image contains regions that respond differently in terms of image feature change to different directions of movement. However for omnidirectional cameras, the images are continuous. Our fused images approach shows that even noncontinuous images from different cameras can improve accuracy with proper alignment. Similarly, as will be demonstrated in Section 6.3, multiple cameras can be trained on the same object with different alignments to produce fused images with higher accuracy. The key is recognizing situations where performance for one camera will be impaired in terms of a lower rate of image feature change.

The fused images approach also lends itself well to integration with subsectioning and recombination. Potentially multiple aligned cameras could provide additional redundancy against ambiguity for subsectioning and recombination, where one view of a part was occluded in one camera, but not in another.

One negative situation with fused images that can occur is a difference in the image statistical characteristics between the two cameras. Since the eigenspace formed is based upon the minimization of reconstruction error, if one set of images displays less variance across the set in terms of individual pixel values, potentially most of the eigenvectors will respond to properly reconstructing the other region of the image. An example situation would be one set of images from one camera with varying regions of highest and lowest intensity paired with another set that had most of its image contents within a small intensity band. In this case the images corresponding to the small intensity band would contribute little to position determination through the lack of eigenvectors responding to these images. A possible solution would be to normalize the images as a preprocessing step.

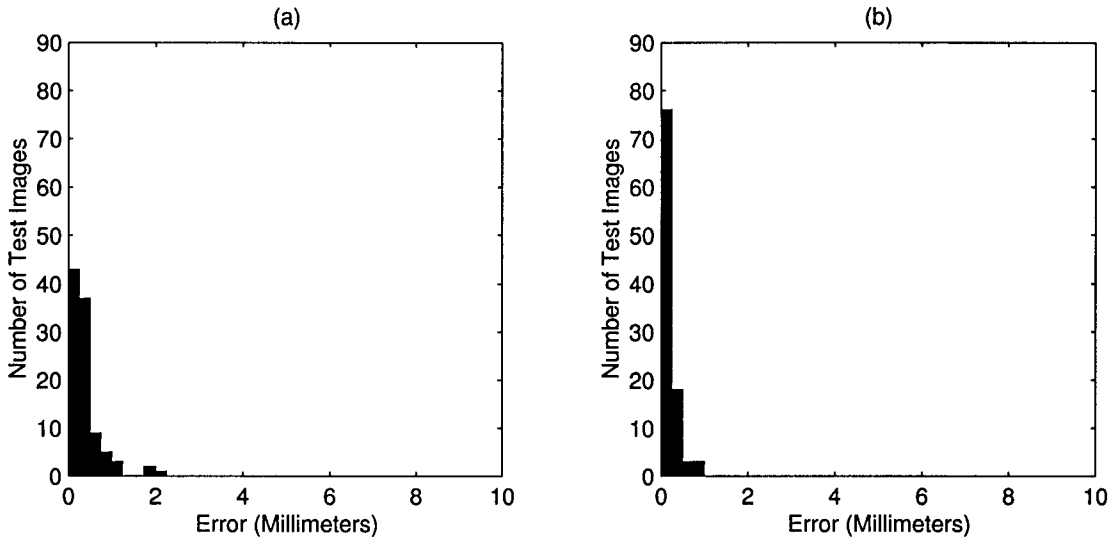

Figure 6.14: Histograms of Absolute Position Error, Fused Projections, (a) Lateral Direction First Camera, (b) Lateral Direction Second Camera

6.2.3 Fused Projection Coefficients

Other approaches to simply fusing the images are available for combining multiple image information in the eigenspace approach. One intuitive method would be to maintain separate eigenspaces $\mathbf{E}_a$ and $\mathbf{E}_b$ for each camera and fuse the projection coefficients from each camera together. Thus for each position i a set of projection coefficients $y_{i_{ab}}$ would be produced by combining the projection coefficients for position i from camera a (y_{i_a}) and camera b (y_{i_a}) via:

$$y_{i_{a+b}} = \begin{bmatrix} y_{i_a} \\ y_{i_b} \end{bmatrix} \tag{6.52}$$

Subsequently, the nearest neighbor search would be performed with this combined set of projection coefficients. The benefit of this approach as opposed to just fusing the images is that the projection coefficients corresponding to each set of images will be optimized for each set of images alone. When the combined projection coefficients

are evaluated during the nearest neighbor search, the Euclidean distance will depend on both sets of independently derived projection coefficients. As with the fused images approach, the fused projection coefficients do not require specifying a camera direction as with the decoupled camera approach.

To illustrate the effectiveness of fusing the projection coefficients as compared to the images, the previous positioning experiments with the two cameras were repeated, except that the separately derived projection coefficients from the two sets of images were concatenated into one set of projection coefficients. Fifteen projection coefficients from each eigenspace were used. All the other parameters were kept the same.

Figure 6.14(a) shows the histogram of the absolute positional error corresponding to the lateral direction of the first camera and Figure 6.14(b) shows the histogram of the absolute error for the lateral direction of movement for the second camera. The absolute mean error was 0.35 mm in the lateral direction of the first camera and 0.19 mm in the lateral direction of the second camera (2.80 % and 1.52 % of the distance between training images respectively). Interestingly, in comparison with the fused images approach, the fused projection coefficients results were slightly higher (0.31 mm and 0.15 mm) in each direction. A possible explanation for the difference could be due to the nearest neighbor calculation, whereby the projection coefficients for one camera that are more accurate in a particular direction are combined with those of the other camera which are more likely to be erroneous. The projection coefficients from the other camera could be considered to add noise for that direction and thus slightly alter the overall Euclidean nearest neighbor value produced.

Nonetheless, the fused projection coefficients approach performs well in comparison to only a single camera in both directions and similar to the decoupled cameras approach without the need for aligning the cameras specifically and choosing a direction for each camera.

6.3 Higher Dimensions and Multiple Cameras

Thus far, the examples presented have been constrained to two dimensional translational movement. Higher dimensional eigenspace solutions have not been directly considered, but it is easy to see their usefulness. In manufacturing, moving an end-effector into position relative to an object is likely to require movement not constrained to 2D translational movement. Higher dimensional eigenspace problems have their own unique difficulties. The previous multiple camera techniques with the additional information they utilize are well suited for higher dimensional eigenspace problems, particularly 3D translational movement relative to an object.

6.3.1 3D Translational Positioning using Multiple Cameras

To illustrate the benefits of the use of multiple cameras for 3D translational positioning relative to an object, two cameras were set up over a metal part. Each camera was mounted on an XYZ table which consisted of the XY table from the previous experiments with an extra stage.

The alignment of the cameras for this example was specifically not chosen to be 90 degrees offset from each other for practical reasons. One camera was positioned looking directly down at the object, thus from the previous sections of this chapter it should provide high accuracy for the two directions of movement corresponding to the plane perpendicular to the optical axis of the camera.

The other camera was aligned at a position close to the 90 degrees offset from the first camera, however it was in fact only offset at 65 degrees from the optical axis of the first camera. This alignment is illustrated via a simple diagram in Figure 6.15, showing that the the optical axes of both cameras are separated by 65 degrees. This positioning was chosen because it would allow the second camera to approach a large relatively flat surface and measure vertical position, with a sizeable amount of image

feature change, whereas a camera offset 90 degrees from the vertical, even if moved to almost touching the surface would capture little of the surface in the image and thus not be practical.

This 65 degree offset from the vertical however is enough such that the image feature change for the vertical movement (corresponding to the optical axis of the first camera) of the XYZ table is significantly increased, while decreased in the XYZ table direction closest to the optical axis of the camera. In the remaining XYZ table direction the image feature change is similar between cameras. Thus for the second camera, positioning in the vertical direction of the XYZ table (termed the z direction herein) should be significantly improved, while poorer in another direction.

The movement range of the cameras was chosen to be a 2 cm by 2 cm by 2 cm cube in the vicinity of the metal object. Images of size 320 by 240 pixels were acquired with both cameras over 4913 positions spaced equally throughout the movement range (17 by 17 by 17 images). Additionally, images were acquired from both cameras at 200 different random locations over the three dimensions to form a testing set.

Figure 6.16 shows eight images acquired from the first camera over the movement range consisting of the eight extreme corner positions possible. Figures 6.16(a) through 6.16(d) show the corner images with the camera zoomed out along the optical axis; Figure 6.16(e) through 6.16(h) show the corner images with the camera zoomed in along the optical axis. As can be clearly seen, the difference in terms of image feature change is much less over the movement in the direction of the optical axis compared to the other two directions. Figure 6.17 shows the eight images acquired from the second camera. Figures 6.17(a) through 6.17(d) show the corner images with the second camera zoomed out along the optical axis, Figures 6.17(e) through 6.17(h) show the corner images with the second camera zoomed in along the optical axis. Note that for the second camera, the movement shown is not truly along the optical axis for the zooming in and out pictures since the second camera was only

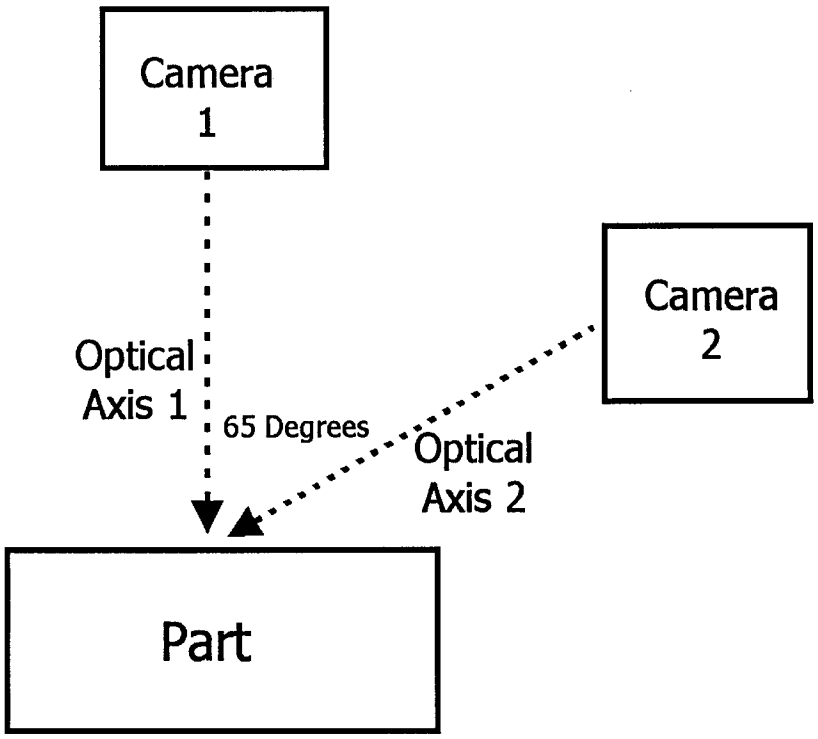

Figure 6.15: Illustration of Camera Alignment

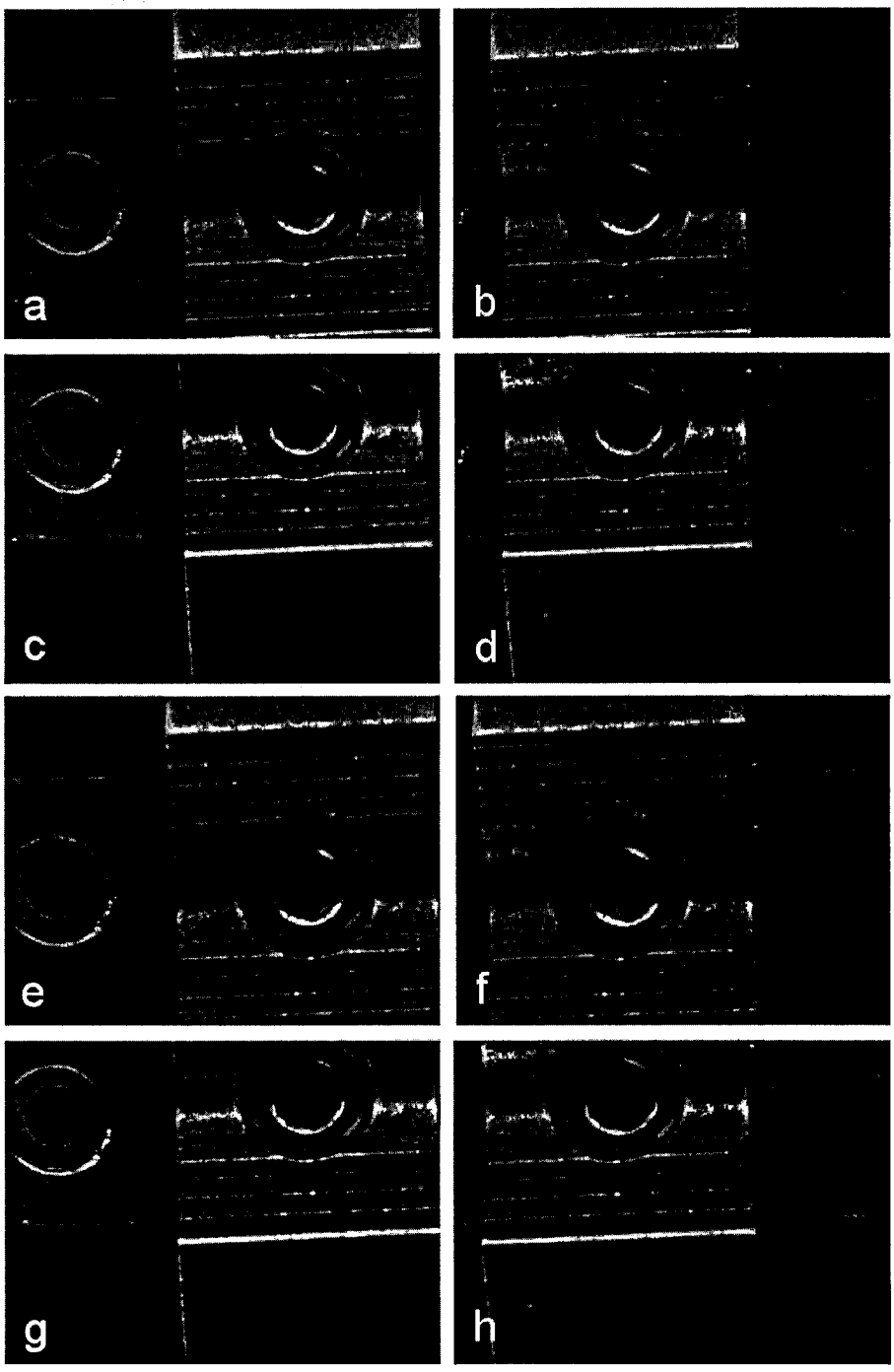

Figure 6.16: Extreme Images of Camera Motion, First Camera, (a) Zoomed Out Upper Left, (b) Zoomed Out Upper Right, (c) Zoomed Out Lower Left, (d) Zoomed Out Lower Right, (e) Zoomed In Upper Left, (f) Zoomed In Upper Right, (g) Zoomed In Lower Left, (h) Zoomed In Lower Right

offset 65 degrees from the optical axis of the first camera.

For nomenclature purposes, the x-direction error refers to that corresponding to the camera movement via the XYZ producing motion with the image contents moving left to right for the first camera in Figure 6.16. Similarly the y-direction error refers to that corresponding to the camera movement producing motion with the image contents moving up and down. Finally, the z-direction error refers to that corresponding to the camera motion along the optical axis, with the images zooming in and out.

For explanatory purposes, the camera movement per pixel of image feature change values D_x, D_y, and D_z were calculated for both cameras for the x, y and z directions of movement via:

$$D_x = \frac{d_x \cdot a}{\sqrt{(p_1 - p_2)^2 + (q_1 - q_2)^2}} \tag{6.53}$$

where d_x is the distance increment between adjacent training images in the x direction, a is the number of increments between the two training images used for the calculation and p_1, q_1 and p_2, q_2 represent the image pixel coordinates of the same image feature in the two training images. Note the training images chosen for the calculation consisted of those at the corners of the camera movement range cube and images features near the corners of the images were used.

For the first camera, D_x was equal to 0.22 mm per pixel of image feature change, D_y was equal to 0.20 mm per pixel of image feature change and D_z (the optical axis direction of the camera) was equal to 1.28 mm per pixel of image feature change. Thus D_z was considerably higher than D_x and D_y meaning the image feature change was lower in this direction.

For the second camera, D_x was equal to 0.66 mm per pixel of image feature change, D_y was equal to 0.13 mm per pixel of image feature change and D_z was equal to 0.18 mm per pixel of image feature change. Thus D_x was highest which makes sense since the optical axis of the second camera was closest to this direction of camera

movement. Similarly D_y was lower than D_z because the second camera was only offset 65 degrees from the optical axis of the first camera not 90 degrees.

To demonstrate the lower accuracy achievable using only a single camera, the first camera was used alone. The eigenspace was formed using a subset of the original set of training images consisting of 729 images (9 by 9 by 9 images) spaced equally throughout the movement range. The entire set was not used due to the large number of images requiring heavy use of virtual memory to derive the eigenvectors.

The entire set of 4913 training images was used to produce a set of initial projection coefficients. This set was interpolated to provide 7 positions between each training image position in each direction for additional accuracy. To speed up the search time, a coarse search as described in Chapter 2 was used. The original 4913 training projection coefficients were searched first. Based on this match, a constrained subset of the interpolated projection coefficients was searched. This subset consisted of 25% of the surrounding interpolated projection coefficients of the original match in each dimension. This reduced the number of interpolated sections to be searched by a factor of 64 (1/4*1/4*1/4) with no loss of accuracy. For each test image, the new search required approximately 0.6 seconds to perform.

Additionally, once the nearest neighbor match was found for the interpolated set, an additional 26 camera positions surrounding the nearest neighbor position had new interpolated projections generated for them. These positions were midway between the nearest neighbor position match and the surrounding interpolated positions. Subsequently these projection coefficients plus the original matched projection coefficients from the interpolated set had a new nearest neighbor search performed. Since 3D positioning puts large search demands in terms of the number of points, this allows additional possible accuracy to be achieved without hugely increasing the number of positions to be searched initially. More importantly for this implementation it

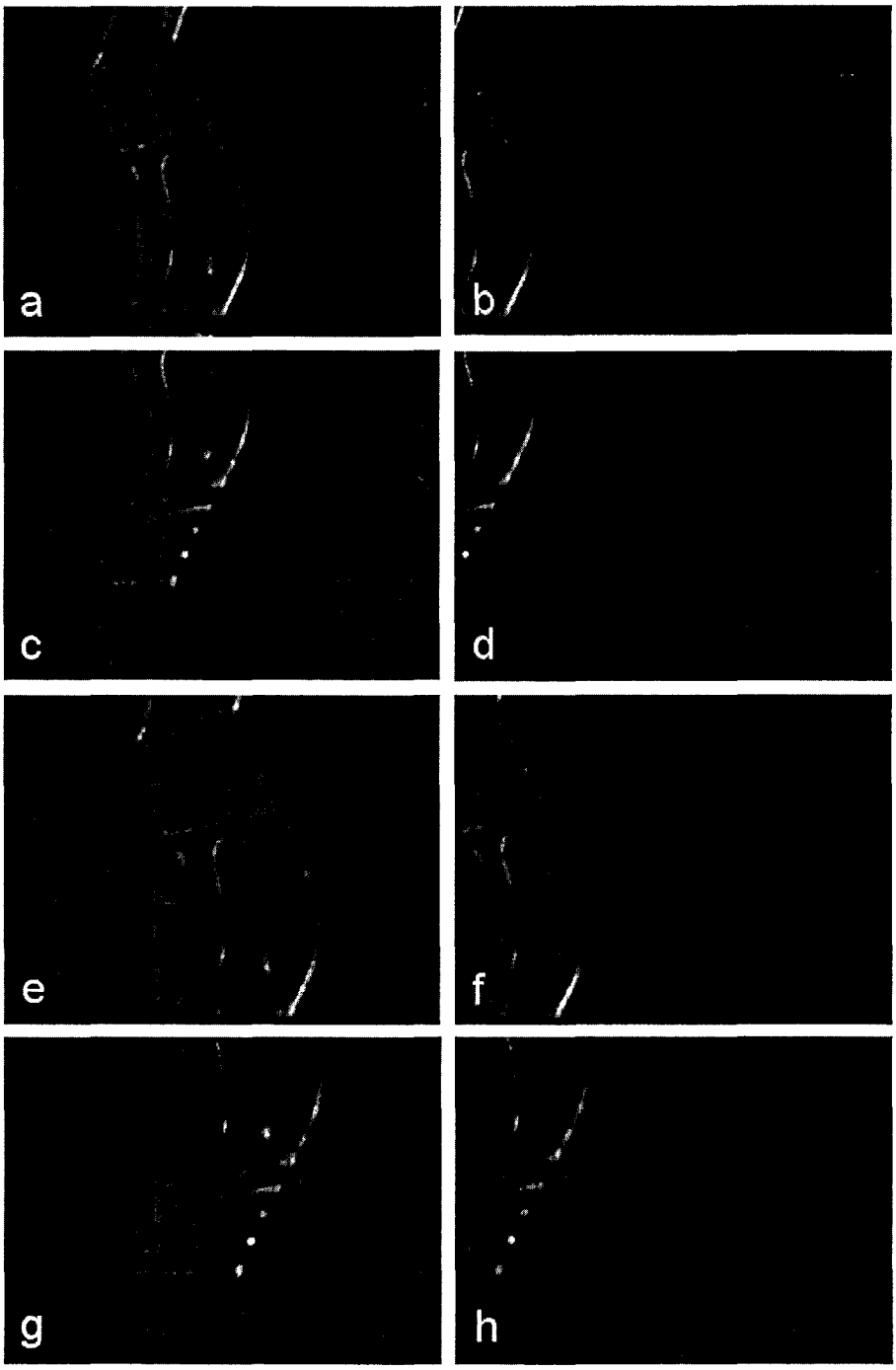

Figure 6.17: Extreme Images of Camera Motion, Second Camera, (a) Zoomed Out Upper Left, (b) Zoomed Out Upper Right, (c) Zoomed Out Lower Left, (d) Zoomed Out Lower Right, (e) Zoomed In Upper Left, (f) Zoomed In Upper Right, (g) Zoomed In Lower Left, (h) Zoomed In Lower Right

reduced the amount of memory dedicated to the stored interpolated projection coefficients. Potentially this procedure could be used recursively on only the training images projection coefficients, however without an initial relatively dense search of the manifold, this proved to be poor for 3D translational positioning.

Figure 6.18 shows the histograms of the absolute error for all three directions for the first camera, with 6.18(a) corresponding to the x-direction, 6.18(b) corresponding to the y-direction and 6.18(c) corresponding to the z-direction. The absolute mean error results were 0.128 mm in the x-direction, 0.118 mm in the y-direction and 2.36 mm in the z-direction (10.2 %, 9.44 %, and 188 % of the distance between training images respectively). As expected, both the absolute mean error and the error histograms in the two directions perpendicular to the optical axis of the first camera were relatively low. With the large amount of image feature change associated with the movement in these directions it is clear that the decoupled cameras approach could be successfully used for these directions. Conversely the absolute mean error and histogram for the z-direction along the optical axis were larger, almost twenty times worse than the other directions. From the images illustrating the movement range of the camera in Figure 6.16, and the calculated D_x, D_y, D_z values (0.22, 0.20, and 1.28 mm per pixel respectively), it is clear that there is much less image feature change associated with this direction of movement comparatively. Consequently the resultant performance suffers comparatively. These results reinforce the difficulty of obtaining 3D translational positioning via only one camera.

Similarly, the same eigenspace procedure was applied with the second camera completely independently, as per the decoupled cameras approach. The same subset of images was used to derive the eigenvectors; the entire set of training images was used to derive the projection coefficients and interpolated to the same degree. For reference purposes, the same direction of camera movement will be used as for the first camera, however due to the alignment of the second camera, image feature change

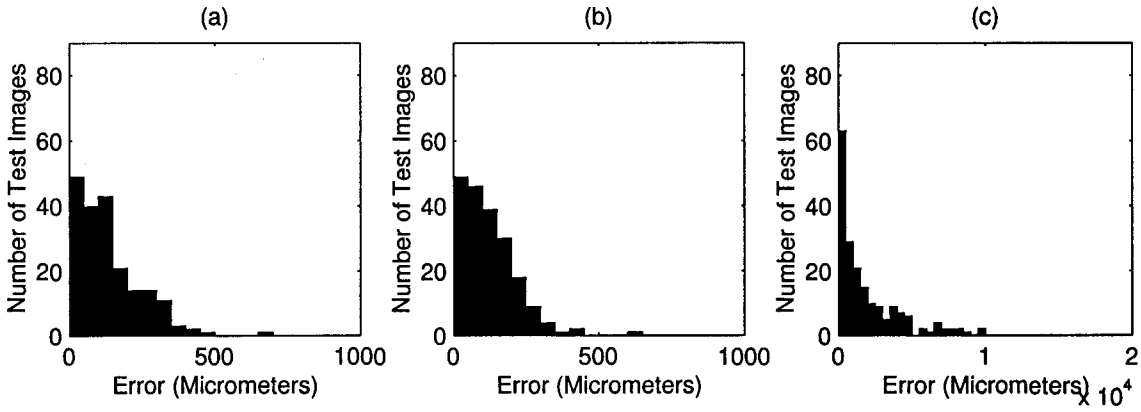

Figure 6.18: Histograms of Absolute Position Error, First Camera, (a) x-direction, (b) y-direction, (c) z-direction

was reduced significantly in the x-direction and increased in the z-direction as shown by the estimated D_x, D_y and D_z values (0.66, 0.13, 0.18 mm per pixel respectively).

Figure 6.19(a) shows the histograms of the absolute error for the x-direction of camera movement, 6.19(b) the y-direction and 19(c) for the z-direction for the second camera. The absolute mean error results were 0.777 mm in the x-direction, 0.0923 mm in the y-direction and 0.313 mm in the z-direction (62.2 %, 7.34 %, and 25.0 % of the distance between training images respectively).

Thus for the second camera, the mean error and histogram in the x-direction was significantly worse than for the first camera, which is not surprising considering that in the different alignment D_x was higher comparatively to D_y and D_z and thus the image feature change lower. In the y-direction, the mean error and histogram was slightly better than for the first camera; likely due to the low value of D_y compared to D_x and D_z. However in the z-direction, both the mean error and the histogram of the error were improved for the second camera over the results of the first camera again likely due to the fact that D_z for this camera was lower compared to D_x and D_y meaning there was higher image feature change.

Combining the results of the two cameras via the decoupled cameras approach with

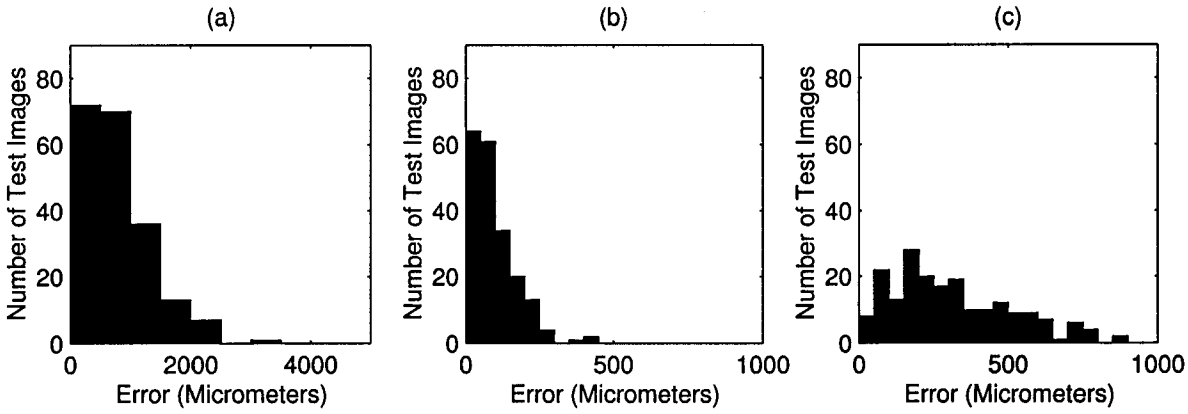

Figure 6.19: Histograms of Absolute Position Error, Second Camera, (a) x-direction, (b) y-direction, (c) z-direction

x and y from the first camera and z from the second camera provides a significant improvement over using only one camera. Thus 3D translational positioning can be improved with the use of two cameras via the decoupled cameras approach.

6.3.2 Image Fusion for 3D Translational Positioning

Potentially the fused image approach could also be used to try and obtain additional accuracy by combining the information from both cameras. Again, a benefit for using fused images over decoupled cameras is that there is no requirement to choose which camera is responsible for which direction of movement. For this experimental example, the fused images should show improved performance for the optical axis direction of the first camera due to the additional information from the second camera in terms of move image feature change in this direction.

Thus for the 3D translational experiments, the training image sets from both cameras were fused together to produce one set of images consisting of the images from the first camera on the bottom and those corresponding to the second camera on the top, as in the example image shown in Figure 6.20. Likewise, the random image sets from both cameras were combined into one. The standard eigenspace procedure

was then performed on these combined images as with the decoupled camera results; 729 training images were used to derive the combined eigenvectors. The full set of 4913 combined training images were used to derive the training projection coefficients. These were interpolated to provide seven additional positions in each direction. For the random images, the same search procedure as with the individual cameras was performed.

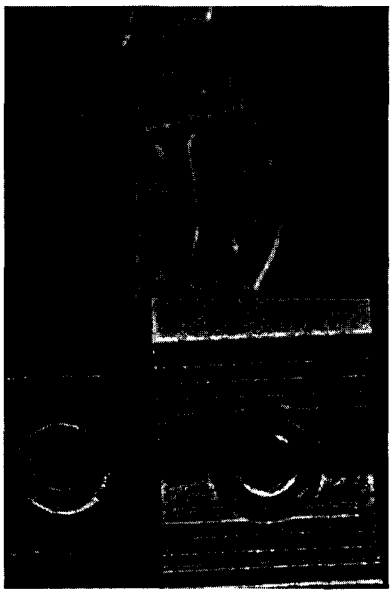

Figure 6.20: Example of Fused Image

Using the same alignment of directions for error as the first camera, Figure 6.21(a) shows the histogram of the absolute error in the x-direction, Figure 6.21(b) shows the histogram of absolute the error in the y-direction and Figure 6.21(c) shows the histogram of the absolute error in the z-direction. The absolute mean errors were 0.113 mm in the x-direction, 0.126 mm in the y-direction and 0.237 mm in the z-direction (9.04 %, 10.1%, and 18.9 % of the distance between training images respectively).

Compared to the first camera alone the z-direction error is much lower, but also significantly lower than for the second camera by itself (0.237 mm versus 0.313 mm). Thus the combined images provide additional accuracy in this direction above and

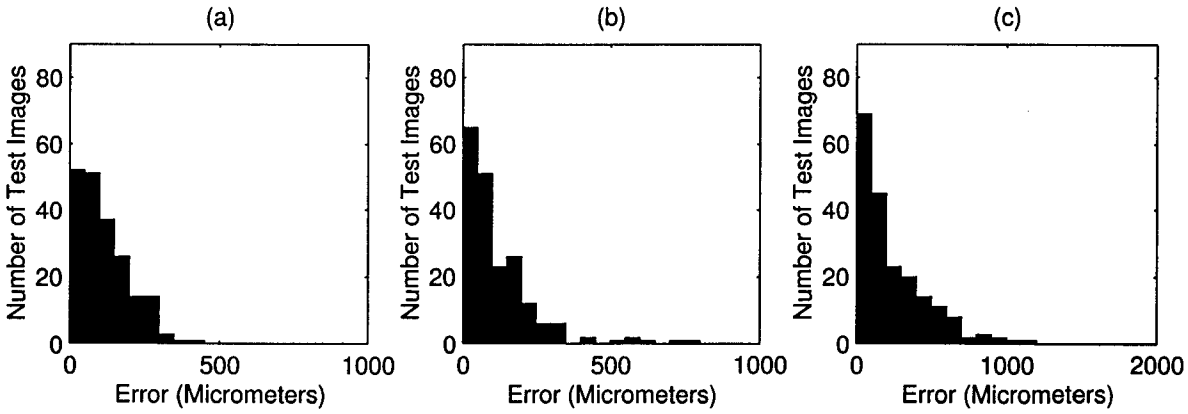

Figure 6.21: Histograms of Absolute Position Error, Fused Images, (a) x-direction, (b) y-direction, (c) z-direction

beyond the individual cameras.

The error in the other directions was similar to that of the first camera alone. Compared to the second camera, the error in the x-direction was superior (0.113 mm versus 0.660 mm), again as would be expected due to the presence over the entire fused images of a high level of image feature change in this direction. Figure 6.22 shows the first four visualized eigenvectors to illustrate the types of feature vectors formed with the combined images. It can be clearly seen that both image regions contribute to the responses of the eigenvectors shown.

Thus from these results, it is shown that the performance of 3D translation positioning can be significantly enhanced through the use of multiple cameras, especially using the fused images approach. Conceivably the performance in the z-direction or optical axis of the first camera could be enhanced by the use of a third offset camera, positioned similarly to the second but such that their optical axes were perpendicular. This would enhance the visual information available pertaining to the movement of the first camera along its optical axis and allow accuracy along this axis to be brought closer to the level of the other two directions. Potentially, fusing only the images of the two offset cameras could produce feature vectors that are especially responsive to

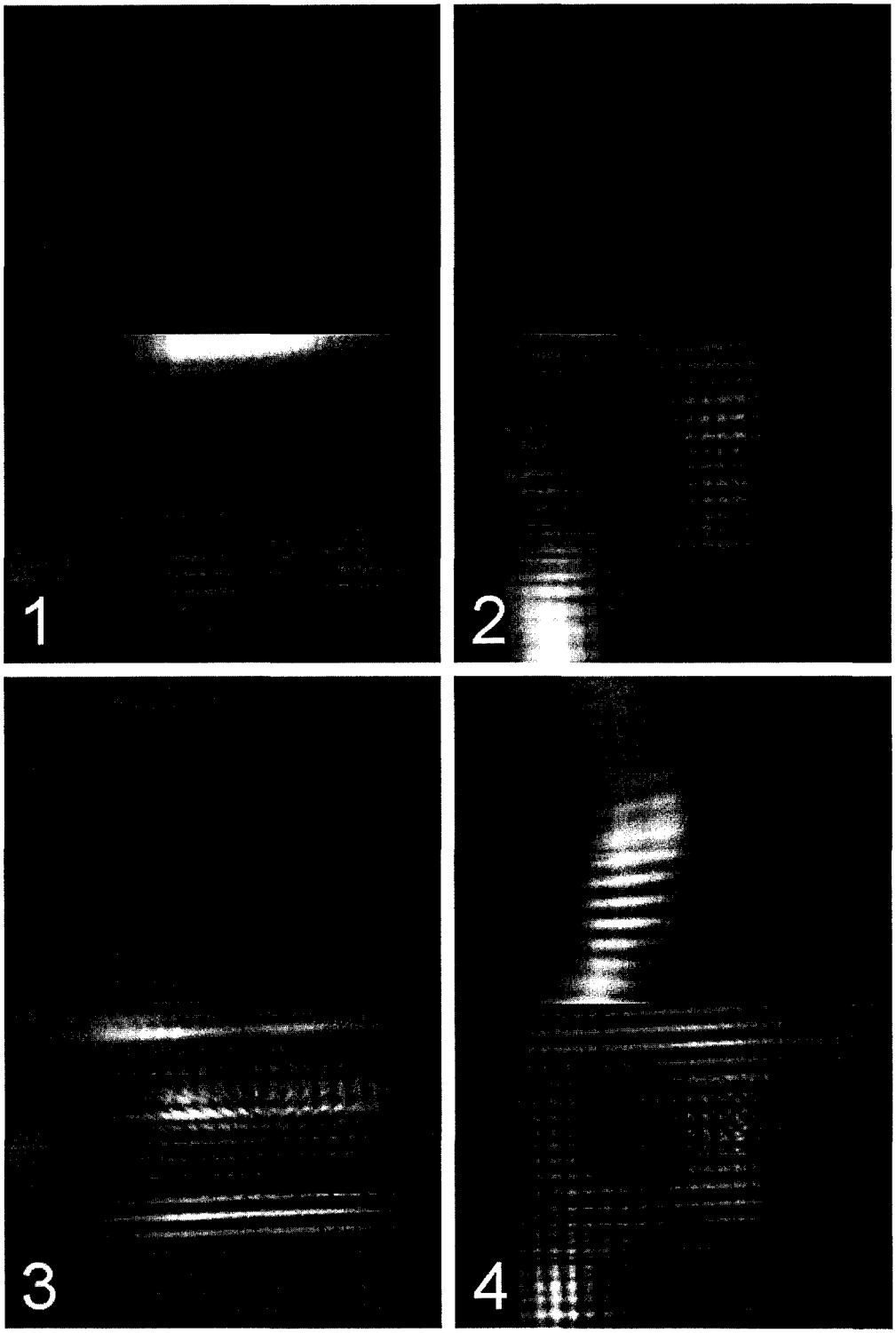

Figure 6.22: Combined Camera Visualized Eigenvectors, First Through Fourth

the z-direction of movement (relative to the first camera). Ideally one could imagine a specially constructed sensor, consisting of three cameras aligned as in these experiments. The sensor could then be mounted on the end-effector of a robot to allow precise visual servoing with eigenspace positioning in relation to a part.

An issue for 3D positioning and higher dimensional positioning ignored to this point, is the large number of images required to obtain reasonable accuracy. As shown in the previous example, each camera required 4913 images for the training projection coefficients to acquire only a relatively small number (17) of images in each direction. The next section describes methods to reduce the requirements for the number of images when using the decoupled cameras technique.

6.4 Use of Partial-Invariant Properties of Multiple cameras

An important issue for performing higher dimensional eigenspace positioning and perhaps the prime reason why eigenspace positioning has not to date been exploited for industrial manufacturing applications and remains primarily an academic novelty is the requirements for huge sets of training images. Considering the large number of images required for the previous experiments with 3D translational movement, adding a small amount of rotation movement in even one direction is coming close to the limits of practical feasibility. Adding in multiple cameras only exacerbates the problem.

The number of training images required to provide a dense sampling in each direction (>10000) can be impractical due to the difficulties in acquiring them, including time, as well as increased search time and massive storage requirements including keeping a large number of interpolated projection coefficients in main memory for

fast access. Thus ways of reducing the number of images required is an important research goal.

A possibility for reducing the number of images required for accurate higher dimensional eigenspace positioning is the exploitation of different amounts of image feature change with regards to the direction of movement of a camera. As shown previously, for 3D translational camera movement the image feature change in the direction of the optical axis for one camera is significantly less than in the other two directions. Thus while the images are not invariant to camera movement in this direction, they certainly change significantly less than the other two directions of translational movement. We term the images as *partially-invariant* with regards to the optical axis.

This partial-invariant property can be combined with the decoupled cameras approach. Since the images change in appearance comparatively less along the optical axis, they can be represented at a lower level of sampling in this direction and still allow the position in the other two directions of movement to be ascertained accurately. For multiple camera configurations with decoupled cameras, the sampling rate could be lowered for each camera in their respective optical axis direction.

The partial-invariant procedure is then as follows: instead of obtaining an equal distribution of training images across all directions, acquire a lower number in the direction of movement along the optical axis of the camera. Perform the standard eigenspace analysis with these reduced training images. Produce a set of projection coefficients corresponding to the training images. At this stage interpolation can then be performed in two ways. First the training projection coefficients can be interpolated directly with equal number of interpolated projection coefficients between training images in each direction. Alternatively, the training images projection coefficients can first be interpolated along the optical axis to produce a set with an equal number of positions in each direction and then subsequently interpolated equally in

each direction. This gives a set of projection coefficients similar to that if the number of training images was equal in each direction. This has no advantage, except for facilitating the technique for use with the fused projection coefficients technique and allowing comparison for finding position along the optical axis with the full set of training images. We used this approach in our experiments for consistency with the previous experiments. Position in each direction can then be determined as per the normal nearest neighbor search.

To demonstrate the feasibility of this technique the data set for the first camera of the 3D translation experiment was used. As an illustration of the lowered image feature change, Figure 6.22(a) shows an image with the camera at a corner of the movement range. Figure 6.23(b) shows an image for moving the camera laterally while Figure 6.23(c) shows an image produced by moving along the optical axis of the camera.

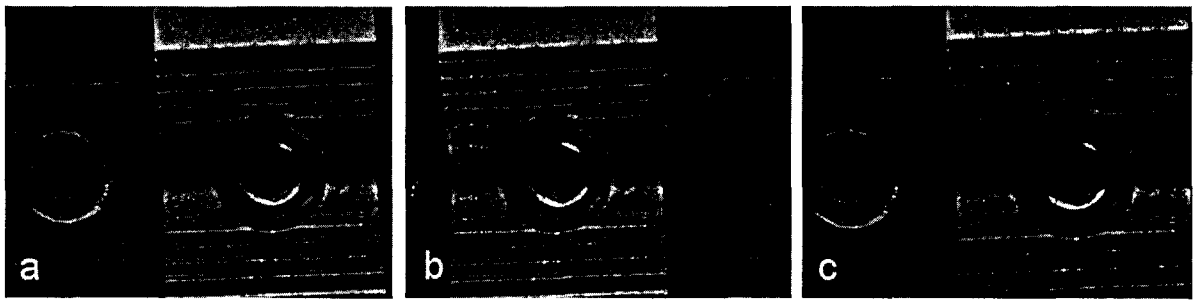

Figure 6.23: Image Change Comparison, (a) Original Position, (b) After Lateral Camera Movement, (c) After Zoom Camera Movement

Note that the camera movement per pixel of image feature change D_x, D_y and D_z were estimated to be 0.22, 0.20 and 1.28 mm per pixel, such that image feature change was approximately six times less along the optical axis direction of camera movement. Since the original set of training images consisted of 4913 images corresponding to 17 by 17 by 17 camera positions, three layers of 17 by 17 images perpendicular to the optical axis were used since $17/3 \simeq 6$, is close to the ratio of the image feature

change between the optical axis direction of camera movement and the remaining two directions. The layers consisted of the top, middle and bottom layers with regards to the optical axis.

This set of 867 images was used to derive the eigenvectors. The projection coefficients corresponding to the 867 images were then calculated using these eigenvectors. These projection coefficients were first interpolated in the optical axis direction to obtain the 4913 positions corresponding to the original training image set and then these were subsequently interpolated to provide 7 positions in each direction between each of the projection coefficients. Position in each direction was determined via the nearest neighbor search described in section 6.3.1.

Figure 6.24(a) shows the histogram of the absolute error in the x-direction (as previously defined), Figure 6.24(b) shows the error in the y-direction and Figure 6.24(c) shows the error in the z-direction. The absolute mean errors were 0.0956 mm, 0.113 mm and 3.34 mm in the x, y and z-directions respectively. Recalling that for the full set of training images, the absolute mean errors were 0.128 mm, 0.118 mm and 2.36 mm, it is clear that although the performance is significantly worse in the z-direction as would be expected, the results with the partial-invariant technique were similar to the full set of training images and in fact slightly improved.

Considering that only 17.6 percent of the full training set was used, this was a remarkable performance, illustrating that the partial-invariant technique exploiting the differences in image feature change can be quite useful for reducing the number of images required. Conceivably, the decoupled cameras approach could be used to obtain a reduced set of training images for the second camera to obtain higher accuracy in the optical axis direction of the first camera. This would require an additional 867 camera positions or 35.3 percent of the original 4913 positions, which is still a significant reduction in the number of separate camera positions. Alternatively, the fused projection coefficients approach could also be used if the projection coefficients

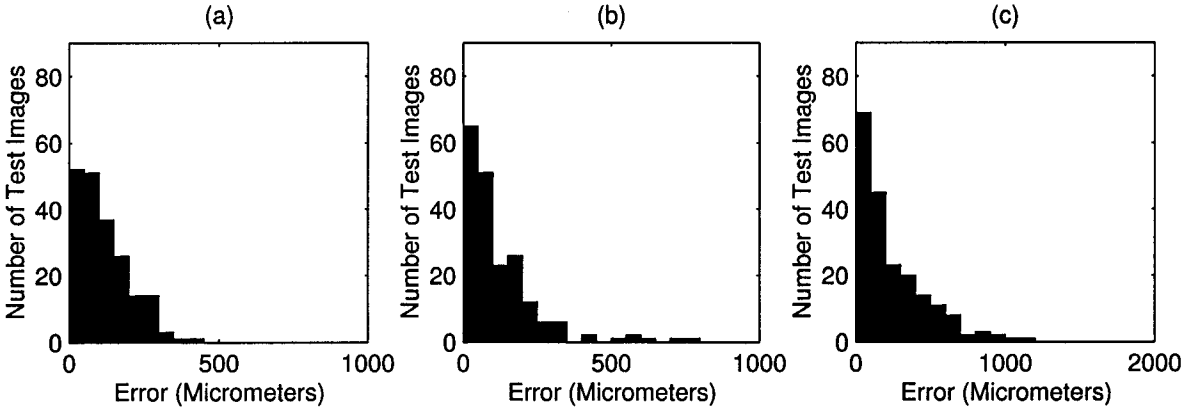

Figure 6.24: Histograms of Absolute Position Error, Pseudo-Invariant Technique, (a) x-direction, (b) y-direction, (c) z-direction

are interpolated to provide equal numbers in each direction. A future area of investigation could include how to best optimize accuracy with regards to the locations of the training images.

Potentially this approach could be expanded to incorporate rotational information as well. Rotating a camera around the optical axis will produce a large amount of image feature change for an image of an object. With the camera aimed at the center of an object, rotation in the other two directions will produce considerably less image feature change. With a three camera setup, each of the three rotational dimensions of movement could be covered by one camera where the rotational movement would be around the optical axis and the image feature change is greatest. Then for each camera, a set of images could be collected where the sampling was higher in the optical axis rotation direction and lower in the other two directions similar to the 3D translation case.

Further pursuit of these training image reduction techniques could eventually allow full six dimensional eigenspace positioning to be performed relative to an object over a small range in each direction. This could be used for final positioning for the end-effector of a robot over a limited range of positions relative to a larger part where

insertion or a similar task is to be performed. Ideally, a highly accurate XYZ table combining with a 3 dimensional rotational stage could be used to acquire the training images relative to the part with a three camera sensor. The sensor could then could be mounted on the end-effector of a robot and the high accuracy from the XYZ table and rotational stages could be transferred to the robot via the eigenspace technique. However considerable work remains in the future to achieve this goal.

6.5 Chapter Summary

Multiple offset cameras were introduced as a technique to obtain additional visual information to achieve similar accuracy in all directions. The decoupled cameras technique was then demonstrated via a simple example of combining information for multiple cameras; each camera determined position in the directions it was most suited towards. Fusing the images of multiple cameras under a single eigenspace was also shown to be effective at increasing accuracy, outperforming the decoupled camera approach without the direction needing to be specified. Fusing the projection coefficients of multiple cameras each with their own eigenspace was also shown to be effective.

The multiple camera techniques were applied to the task of 3D translational positioning. The decoupled cameras technique improved accuracy over one camera. Using the fused images approach gave even better performance.

Finally, the concept of partial-invariance was introduced for the task of reducing the number of training images required for higher dimensional eigenspace positioning. Lower sampling along the direction of the optical axis of the camera proved to provide similar levels of accuracy in the other directions as a full sampling in all three directions.

Chapter 7

Conclusions

7.1 Eigenspace and Accuracy

The concept of image feature change was introduced as a technique for explaining eigenspace positioning performance. The amount of image feature change over the movement range of the camera or object as determining the limit of accuracy was discussed. Amongst the controllable parameters available, the number of training images used to form the projection coefficients had the largest effect on the accuracy. Initially increasing this number had a large effect, however as the number approached the limits of the image feature change, almost no improvement in accuracy was shown. The number of training images used for forming the eigenvectors also had an effect on accuracy, but not to the same extent. For situations where the number of training images was low compared to the image feature change, filtering the eigenvectors to smooth out artifacts from the implicit covariance matrix was shown to be effective at improving mean accuracy. For the number of eigenvectors to use, it was shown that the later eigenvectors had projection coefficient manifolds that were of higher frequency in nature, consequently the number of training projection coefficients could

be too low for proper sampling, resulting in inaccurate interpolation and little discriminant value.

7.2 Occlusion and Occlusion Detection

The main work of the thesis was developing a technique to make eigenspace positioning robust to occlusion. The effects of occlusion on positional accuracy were demonstrated, showing that for proportionally small occlusions (or those little different in appearance from the original pixels) positional accuracy suffered little. Comparatively, as occlusion became proportionally larger (or substantially different from the original pixels) the effect on the positional accuracy could become catastrophic due to the large distance from the original manifold of the highly occluded eigenspace projection coefficients.

Overcoming occlusion was implemented through the use of local information within the image as opposed to the global information of the standard eigenspace coefficients which were vulnerable to error with occlusion in any part of an image. Dividing the image into specific sections allowed (providing occlusion detection could be performed relatively accurately) the preservation of information that could be used to determine position for unoccluded sections. It was shown that naively proceeding with each section as functioning independently from each other, with its own eigenspace and projection coefficients and thus determining a separate position for each section was not an adequate solution. This was due to the fact that with small individual sections, ambiguous situations could arise wherein images over ranges of camera movement would be similar even for where the entire image was unambiguous. This lead to potentially large errors for the individual sections and consequently poor accuracy overall.

Thus rather than sacrificing the global information present across the entire image

that gives resistance to such locally ambiguous situations, subsectioning and recombination was suggested as a technique to incorporate local information as a hedge against occlusion while maintaining global information for resistance to ambiguity. Subsectioning and recombination was based on the observation that the problem of position determination was different than that of eigenspace reconstruction in that the requirement is for the eigenspace projection coefficients to vary across the movement range of the camera or object enough to allow position to be accurately determined. Thus the global eigenvectors were still used, however for each section the subtotal of the projection coefficients was stored. Summing these projection coefficients for every section gave the standard global projection coefficients, however removing several sections still gave a set of modified global projection coefficients that could be adequately used to determine position. Thus at the additional cost of storing the local projection coefficients and a few additions for each search term to find the new global projection coefficients, occluded sections could be removed and position still determined. It was then demonstrated that even with substantial amounts of occlusion, performance was often similar to that with no occlusion and in any case superior to the uncorrected basic technique.

For occlusion detection, it was shown that thresholding eigenspace reconstruction was an effective technique for evaluating whether an image from a constrained visual subspace contained an occlusion or not. The technique became much more effective as the occlusion became larger as a proportion of the image as well as for occlusions differing significantly from the original pixels at their location. One strength of the eigenspace reconstruction approach was that for images with occlusions, not only were the reconstructed pixel values different in the location of the occlusion, but also in other regions of the image, since the altered eigenspace coefficients resulted in slightly different combinations of the basis vectors across the entire image, increasing the sensitivity to occlusion. The ability to detect small occlusions in an image improved

through dividing the images into sections and then deriving separate eigenspaces for each. This was important for subsectioning and recombination because it allowed the detection of any occlusion large and/or different in appearance enough to impact on the positional accuracy and thus allow subsectioning and recombination to be a viable method for overcoming occlusion. Using the distance from the eigenspace manifold as a thresholding technique to detect occlusion also showed good performance, however compared to eigenspace reconstruction there was a slightly higher chance of incorrect classification.

7.3 Multiple Cameras and Higher Dimensions

It was demonstrated that for instances of problems where part of the movement of the camera was along its optical axis, there was a significant difference between the error in that direction and the perpendicular directions. This was due to the much smaller image feature change along the optical axis. The use of multiple cameras offset from one another's optical axis was suggested as a technique to increase the quality of the information available and achieve similar performance in the different directions. Three techniques for combining this information were demonstrated. Decoupled cameras, whereby each camera was responsible for movement in only one direction was shown to be effective at increasing positional accuracy. Fused images, whereby the images of individual cameras were fused into one super-image with its own eigenspace was shown to also increase accuracy over a single camera. Fused projection coefficients, where each camera had its own eigenspace and the projections combined also showed good performance.

The use of multiple cameras was then applied to the problem of three dimensional translational positioning. The decoupled camera technique was shown to be effective at improving the determination of 3D position compared to only a single camera. The

fused images approach gave even better results. Also demonstrated was the concept of partial-invariance for higher dimensional problems, whereby sampling less images over directions where the images changed little relatively, still allowed satisfactory accuracy over the other directions.

7.4 Future Directions

For camera and object positioning, standard PCA has been the usual choice for feature vectors although it is well known that PCA is optimized for reconstruction error rather than for determining position. An interesting avenue of future research would be to explore alternative feature vectors that were more suited to the task of finding accurate position information.

The performance of subsectioning and recombination using the standard global eigenvectors was effective, however it could still be improved. In addition to the global eigenvectors, extra feature vectors could also be included. These would be produced by suppressing to zero large regions of the input images. Consequently these eigenvectors would be concentrated on the remaining regions and determining position with only this information. With several of these eigenvectors to cover all the image, this would provide extra local information that would provide extra accuracy with a high degree of occlusion.

An interesting possible use for subsectioning and recombination would be its application to face recognition, namely situations where parts of the face are subject to occlusion. This could have advantages over previous approaches where separate parts of the face were treated independently and thus gestalt information of the entire face was neglected.

Occlusion detection via eigenspace methods has possible uses beyond that of a

front-end for subsectioning and recombination. Potentially, it could be used for security purposes, whereby a camera was moved via a pan and tilt unit to cover a larger range of territory. Building up an eigenspace for individual image sections for this visual subspace, would allow changes in the scene to be detected as the camera sweeps over the preset range. As well with individual image sections, the location of scene changes (i.e. possible intruders) within the image could be determined. Such a setup could also be used as a front end to a face detection algorithm over a large area. Only changed areas would be searched for the presence of faces.

In terms of multiple cameras, it would appear that their use would be well suited to higher dimensional problems. Certainly solving higher dimensional problems would open up possibilities in applying eigenspace positioning to manufacturing problems. In real applications it is unlikely that movement will be confined to only planar translations, as rotations and three dimensions are the norm. Thus it is imperative to develop eigenspace based techniques that can handle the higher dimensions.

The partial-invariant property with regards to cameras aligned in specific directions and the amount of image feature change across different directions of movement has definite potential for accommodating the main issue with higher dimensions. Namely that for full six degree of freedom movement, the traditional eigenspace approach would require an inordinate amount of training images. Potentially using the partial-invariant property and decoupled cameras could allow a full range of motion to be accommodated, provided that in some directions the movement was limited. Thus for moving a part to an eventual mating position, small amounts of rotation movement could be compensated with the appropriate alignment of cameras.

Potentially, one use for higher dimensional eigenspace approaches would be the use of a highly accurate XYZ position system combined with a three dimensional rotational stage to acquire a highly accurate set of training images about an object for positioning with a multiple camera sensor. Deriving an accurate eigenspace for

this range of motion would allow the sensor to be placed on the end-effector of a robot and then allow compensation for inaccurate robot positioning with the knowledge of the highly accurate position of the sensor in relation to the object.

More work is required on illumination invariance for eigenspace positioning. Early work has been done mainly for object recognition, but to perform accurate positioning for manufacturing, techniques that are tolerant to reasonable levels of varied lighting are required. Possibly with a dedicated multiple camera sensor, a strong light source could be incorporated to keep object pixel values constant as in the work of Nayar et. al. [37]. Alternatively, another possibility would be using sensors beyond the visual spectrum with their own source of appropriate energy provided so that other visible light sources do not interfere with the pixel values.

Bibliography

[1] H. Bischoff and A. Leonardis. Robust Recognition of Scaled Eigenimages Through a Hierarchical Approach. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 664–670, 1998.

[2] H. Bischoff, A. Leonardis, and F. Pezzei. A Robust Subspace Classifier. *Proceedings of the International Conference on Pattern Recognition*, 1:114–116, 1998.

[3] H. Bischoff, H. Wildnauer, and A. Leonardis. Illumination Invariant Eigenspaces. *Proceedings of the International Conference on Computer Vision*, 1:233–238, 2001.

[4] M. Black and A. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *Proceedings of the European Conference on Computer Vision*, pages 666–666, 1996.

[5] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz. A Comparison of Probablistic, Possiblistic and Evidence Theoretic Fusion Schemes for Active Object Recognition. *Computing*, 4(62):293–319, 1999.

[6] R. Campbell and P. Flynn. Eigenshape for 3D Object Recognition in Range Data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2:505–510, 1999.

[7] P. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice/Hall, Inc., 1982.

[8] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-Based Navigation and Environmental Representation with an Omnidirectional Camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898, 2000.

[9] A. Goldstein, C. Harmon, and A. Lisk. Identification of Human Facess. *Proceedings of the IEEE*, 59:748–760, 1971.

[10] G. Golub and C. Van Loan. *Matrix Computations.* John Hopkins University Press, 1996.

[11] D. Huttenlocher, R. Lilien, and C. Olson. View-Base Recognition Using an Eigenspace Approximation to the Hausdorff Measure. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 21(9):951–955, 1999.

[12] M. Jagersand. Image Based View Synthesis of Articulated Agents. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1047–1053, 1997.

[13] M. Jagersand. Hierarchical uncalibrated predictive display for a 16 DOF Utah/MIT hand. *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 1:124–129, 1999.

[14] M. Jagersand. Image Based Predictive Display for Tele-Manipulation. *Proceedings of IEEE International Conference on Robotics and Automation*, 1:550–556, 1999.

[15] A. Jepson, D. Fleet, and T. El-Maraghi. Robust Online Appearance Models for Visual Tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1:415–422, 2001.

[16] M Jogan and A Leonardis. Robust Localization of Spinning Images. *IEEE Workshop on Omnidirectional Vision*, pages 37–44, 2000.

[17] M. Jogan and A. Leonardis. Robust Localization Using Panoramic View Based Recognition. *Proceedings of the International Conference on Pattern Recognition*, 4:136–139, 2000.

[18] I. Joliffe. *Principal Component Analysis.* Springer-Verlag, Inc., 1986.

[19] Takeo Kanade. Picture processing system by computer complex and recognition of human faces. In *doctoral dissertation, Kyoto University.* November 1973.

[20] M. Kirby and L. Sirovich. Procedure for the Characterization of Human Faces. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[21] J. Krumm. Eigenfeatures for Planar Pose Measurement of Partially Occluded Objects. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 55–60, 1996.

[22] A. Leonardis and H. Bischof. Dealing with Occlusions in the Eigenspace Approach. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–458, 1996.

[23] A. Leonardis and H. Bischof. Robust Recognition Using Eigenimages. *Computer Vision and Image Understanding*, 12(1):99–118, 2000.

[24] M. Loeve. *Probability Theory.* Van Nostrand, 1955.

[25] A. Martinez. Recognition of Partially Occluded and/or Imprecisely Localized Faces Using a Probabilistic Approach. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 712–717, 2000.

[26] A. Martinez. Recognizing Imprecisely Localized, Partially Occluded, and Expression Variant Faces from a Single Sample per Class. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 14(6):748–763, 2002.

[27] 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-Part B: Cybernetics*, 31(5):669–682, 2001.

[28] G. Matheron. *Random Sets and Integral Geometry.* John Wiley, 1975.

[29] B. Moghaddam, C. Nastar, and A. Pentland. Bayesian Face Recognition using Deformable Intensity Surfaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–645, 1996.

[30] B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Detection. *Proceedings of the International Conference on Computer Vision*, pages 786–793, 1995.

[31] H. Murakami and B. Kumar. Primary Images From a Set of Images. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 5(4):39–50, 1982.

[32] H. Murase and S. Nayar. Learning and Recognition of 3D Objects from Appearance. *Proceedings of the IEEE Workshop on Qualitative Vision*, pages 511–515, 1993.

[33] H. Murase and S. Nayar. Visual Learning and Recognition of 3D Objects from Appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[34] S. Nayar and H. Murase. Dimensionality of Illumination in Appearance Matching. *Proceedings of IEEE International Conference on Robotics and Automation*, 2:1326–1332, 1996.

[35] S. Nayar, H. Murase, and S. Nene. Learning, Positioning and Tracking Visual Appearance. *Proceedings of IEEE International Conference on Robotics and Automation*, 4:3237–3244, 1994.

[36] S. Nayar, S. Nene, and H. Murase. Real-Time 100 Object Recognition System. *Proceedings of ARPA Image Understanding Workshop*, 3:2321–2325, 1996.

[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] S. Nene and S. Nayar. Closest Point Search in High Dimensions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–865, 1996.

[39] K. Ohba, Y. Sato, and K. Ikeuchi. Appearance-based Visual Learning and Object Recognition with Illumination Invariance. *Machine Vision Applications*, 12(4):189–196, 2000.

[40] E. Oja. *Subspace Methods of Pattern Recognition.* Research Studies Press,, 1983.

[41] A. Pentland, B. Moghaddam, and T. Starner. View-based and Modular Eigenspaces for Face Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 84–91, 1994.

[42] P. Quick and D. Capson. Combating Occlusion and Scene Changes for Camera Position Determination. *Proceedings of the IAPR Workshop on Machine Vision Applications*, 2000.

[43] P. Quick and D. Capson. Subspace Position Measurement in the Presence of Occlusion. *Pattern Recognition Letters*, 23(14):1721–1733, 2002.

[44] C. E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37:10–21, 1949.

[45] M. Turk and A. Pentland. Eigenspaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):586–591, 1991.

[46] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omni-directional Vision for Robot Navigation. *IEEE Workshop on Omnidirectional Vision*, pages 21–28, 2000.