International Journal of Computer Vision, 14, 5–24 (1995)

Visual Learning and Recognition of 3-D Objects from Appearance

HIROSHI MURASE
NTT Basic Research Laboratory, Atsugi-Shi, Kanagawa 243-01, Japan

SHREE K. NAYAR
Department of Computer Science, Columbia University, New York, N.Y. 10027

Received February 24, 1993; Revised January 16, 1994

Abstract. The problem of automatically learning object models for recognition and pose estimation is addressed. In contrast to the traditional approach, the recognition problem is formulated as one of matching appearance rather than shape. The appearance of an object in a two-dimensional image depends on its shape, reflectance properties, pose in the scene, and the illumination conditions. While shape and reflectance are intrinsic properties and constant for a rigid object, pose and illumination vary from scene to scene. A compact representation of object appearance is proposed that is parametrized by pose and illumination. For each object of interest, a large set of images is obtained by automatically varying pose and illumination. This image set is compressed to obtain a low-dimensional subspace, called the eigenspace, in which the object is represented as a manifold. Given an unknown input image, the recognition system projects the image to eigenspace. The object is recognized based on the manifold it lies on. The exact position of the projection on the manifold determines the object's pose in the image.

A variety of experiments are conducted using objects with complex appearance characteristics. The performance of the recognition and pose estimation algorithms is studied using over a thousand input images of sample objects. Sensitivity of recognition to the number of eigenspace dimensions and the number of learning samples is analyzed. For the objects used, appearance representation in eigenspaces with less than 20 dimensions produces accurate recognition results with an average pose estimation error of about 1.0 degree. A near real-time recognition system with 20 complex objects in the database has been developed. The paper is concluded with a discussion on various issues related to the proposed learning and recognition methodology.

1 Introduction

One of the primary goals of an intelligent vision system is to recognize objects in an image and compute their poses in the three-dimensional scene. Such a recognition system has wide applications ranging from visual inspection to autonomous navigation. For a vision system to be able to recognize objects, it must have models of the objects stored in its memory. In the past, vision research has emphasized the use of geometric (shape) models (Besl and Jain 1985; Chin and Dyer 1986) for recognition. In the case of manufactured objects, these models are sometimes available and are referred to as computer aided design (CAD) models. Most objects of interest, however, do not come with CAD models. Typically, a vision programmer is forced to select an appropriate representation for object geometry, design object models using the representation, and then manually input this information into the system. This procedure is cumbersome and impractical when dealing with large sets of objects, or objects with complex geometric properties. It is clear that recognition systems of the future must be capable of acquiring object models without human assistance. In other words, they must be able to automatically *learn* objects of interest.

Visual learning is clearly a well-developed and vital component of biological vision systems. If a

human is handed a three-dimensional object and asked to visually memorize it, he or she would rotate the object and study its appearance from different directions. While little is known about the exact representations and techniques used by the human mind, it is clear that the overall appearance of the object plays a critical role in its perception. Some recent psychophysical findings indicate that the human visual system represents objects by a set of two-dimensional views rather than a single object-centered three-dimensional model (Tarr and Pinker 1989; Edelman et al. 1989).

In contrast to biological systems, machine vision systems today have little or no learning capabilities. Only recently has visual learning for recognition emerged as a topic of research interest. The following is a brief sampling of recent results. Poggio and Girosi (Poggio and Girosi 1990) have analyzed the general problem of learning a function from a set of data points. They proposed a three-layered network, called the regularization network, that learns the mapping between an input space and an output space. This network was used by Poggio and Edelman (Poggio and Edelman 1990) to recognize three-dimensional stick figures from two-dimensional images. Subsequently, Edelman and Weinshall (Edelman and Weinshall 1991) demonstrated the use of a two-layered network for representing objects from multiple views using unsupervised Hebbian relaxation. Taking a different approach, Turk and Pentland (Turk and Pentland 1991) developed a face recognition system that uses principal component analysis to learn and recognize images of human faces. Ullman and Basri (Ullman and Basri 1991) showed that three views of an object can be used to represent its boundaries. The projection of the object's boundaries in other views can be expressed as a linear combination of the three model views provided the correspondence between points in all views is known. Using range images, Fan et al. (Fan et al. 1987) have developed a system that automatically generates surface descriptions of 3-D objects for recognition. In the context of assembly planning, Ikeuchi and Suehiro (Ikeuchi and Suehiro 1992) have proposed a system that learns assembly sequences from range images of a human operator in action and generates a program that enables a manipulator to perform the same task.

This paper presents a technique for automatically learning three-dimensional objects from their appearance in two-dimensional images. The appearance of an object is the combined effect of its shape, reflectance properties, pose in the scene, and the illumination conditions. Recognizing objects from brightness images is therefore more a problem of *appearance matching* rather than shape matching. This observation lies at the core of our work. While shape and reflectance are *intrinsic properties* that do not change for any rigid object, pose and illumination vary from scene to scene. We approach the visual learning problem as one of acquiring a compact model of the object's appearance under different poses and illumination directions. The object is "shown" to the image sensor in several orientations and illumination directions. This can be accomplished using, for example, two robot manipulators; one to rotate the object while the other varies the illumination direction. The result is a very large set of object images. These images could either be used directly or after being processed to enhance object characteristics. Since all images in the set are of the same object, consecutive images are correlated to a large degree. The problem then is to compress the large image set to a low-dimensional representation of object appearance.

A well-known *image compression* or coding technique is based on principal component analysis. Also known as the Karhunen-Loeve transform (Oja 1983; Fakunaga 1990), this method computes the eigenvectors of an image set. The eigenvectors form an orthogonal basis for representing individual images in the set. Though a large number of eigenvectors may be required for very accurate reconstruction of an object image, only a few eigenvectors are generally sufficient to capture the significant appearance characteristics of an object. These eigenvectors constitute the dimensions of what we refer to as the *eigenspace* of the image set. From the perspective of machine vision, the eigenspace has a very attractive property. It is optimal in a *correlation* sense: If any two images from the set are projected to eigenspace, the distance

between the corresponding points in eigenspace is a measure of the similarity of the images in the l^2 *norm*. In machine vision, the Karhunen-Loeve method has been applied primarily to two problems; handwritten character recognition (Murase et al. 1981) and human face recognition (Sirovich and Kirby 1987; Turk and Pentland 1991). These applications lie within the domain of pattern classification and do not address the problem of learning complete parametrized models of objects.

In this paper, we develop a continuous and compact representation of object appearance that is parametrized by the variables, namely, object pose and illumination. This new representation is referred to as the *parametric eigenspace*. First, an image set of the object is obtained by varying pose and illumination in small increments. The image set is then normalized in brightness and scale to achieve invariance to image magnification and illumination intensity. The eigenspace for the image set is constructed by computing the most prominent eigenvectors of the set. Next, all object images (learning samples) are projected to eigenspace to obtain a set of points. These points lie on a *manifold* that is parametrized by pose and illumination. In our implementation, the manifold is computed from the discrete points using cubic spline interpolation. It is important to note that this parametric representation of an object is obtained without prior knowledge of the object's shape and reflectance properties. It is generated using just a sample of the object.

Each object is represented as a parametric manifold in two different eigenspaces; the universal eigenspace and the object's own eigenspace. The *universal eigenspace* is computed using image sets of all objects of interest to the recognition system, and the *object eigenspace* is computed using only images of an object. The universal eigenspace is best suited for discriminating between objects, whereas the object eigenspace is better for pose estimation. Recognition and pose estimation can be summarized as follows. Given an image consisting of an object of interest, we assume that the object is not occluded by other objects and can be segmented from the remaining scene. The segmented image region is normalized in scale and brightness, such that it has the same size and brightness range as the images used in the learning stage. This normalized image is first projected to universal eigenspace to identify the object. After the object is recognized, the image is projected to the object's eigenspace and the location of the projection on the object's parametrized manifold determines its pose in the scene. Two different techniques have been tested for determining the closest manifold point, one is based on binary search and other uses an input-output mapping network.

Object learning requires the acquisition of large image sets and the computationally intensive process of finding eigenvectors. However, learning is typically done off-line and hence can afford to be relatively slow. In contrast, recognition and pose estimation are often subject to severe time constraints, and the proposed approach offers a very simple and computationally efficient solution. Extensive experimentation has been conducted to demonstrate the robustness of the parametric eigenspace representation. We conclude with a discussion on the merits and limitations of the proposed learning and recognition technique.

2 Visual Learning of Objects

In this section we describe the learning of object models using the parametric eigenspace representation. First, we discuss the acquisition of object image sets. Eigenspaces are computed using the image sets and each object is represented as a parametric manifold. Throughout this section, we will use a sample object to illustrate the learning process.

2.1 Normalized Image Sets

While constructing image sets we need to ensure that all images of the object are of the same size. Each digitized image is first segmented (using a threshold) into an object region and a background region. The background is assigned zero brightness value and the object region is re-sampled such that the larger of its two dimensions fits a pre-selected image size. We now have a scale normalized image. This

image is written as a vector $\widehat{\mathbf{x}}$ by reading pixel brightness values in a raster scan manner:

$$\widehat{\mathbf{x}} = [\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_N]^{\mathrm{T}} \tag{1}$$

This vector represents an unprocessed brightness image. Alternately, processed images such as smoothed images, first derivatives, Laplacian, power spectrum of the brightness image, or any weighted combination of such images may be used in place of the brightness image. The image type is selected based on its ability to capture distinct appearance characteristics of the objects of interest. Here, for the purpose of developing the learning method we use raw brightness images, bearing in mind that the same method is directly applicable to any image type.

Unlike an object's shape and reflectance properties, its pose and illumination are expected to vary from scene to scene. If the illumination conditions of the environment are constant, appearance is affected only by object pose. Here, we assume that the object is illuminated by the environment's ambient lighting as well as one additional distant light source whose direction may vary. Thus, all possible appearances of the object can be captured by varying its pose and the light source direction with respect to the viewing direction of the sensor. We denote each image as $\widehat{\mathbf{x}}_{r,l}^{(p)}$ where r is the rotation or pose parameter, l is the illumination direction, and p is the object number. The complete image set obtained for an object is referred to as the *object image set*:

$$\{\widehat{\mathbf{x}}_{1,1}^{(p)}, \ldots, \widehat{\mathbf{x}}_{R,1}^{(p)}, \widehat{\mathbf{x}}_{1,2}^{(p)}, \ldots, \widehat{\mathbf{x}}_{R,L}^{(p)}\} \tag{2}$$

Here, R and L are the total number of discrete poses and illumination directions, respectively. If a total of P objects are to be learned by the recognition system, we can define the *universal image set* as the union of all object image sets:

$$\begin{aligned}&\{\widehat{\mathbf{x}}_{1,1}^{(1)}, \ldots, \widehat{\mathbf{x}}_{R,1}^{(1)}, \widehat{\mathbf{x}}_{1,2}^{(1)}, \ldots, \widehat{\mathbf{x}}_{R,L}^{(1)}, \\ &\widehat{\mathbf{x}}_{1,1}^{(2)}, \ldots, \widehat{\mathbf{x}}_{R,1}^{(2)}, \widehat{\mathbf{x}}_{1,2}^{(2)}, \ldots, \widehat{\mathbf{x}}_{R,L}^{(2)}, \\ &\quad\vdots \\ &\widehat{\mathbf{x}}_{1,1}^{(P)}, \ldots, \widehat{\mathbf{x}}_{R,1}^{(P)}, \widehat{\mathbf{x}}_{1,2}^{(P)}, \ldots, \widehat{\mathbf{x}}_{R,L}^{(P)}\}\end{aligned} \tag{3}$$

It is assumed that the imaging sensor used for learning and recognition has a linear response, i.e. image brightness is proportional to scene radiance. It is also desirable that our recognition system be unaffected by variations in the intensity of illumination or the aperture of the imaging system. This can be achieved by normalizing each image, such that, the total energy contained in the image is unity, i.e. $\|\mathbf{x}\| = 1$. This brightness normalization transforms each measured image $\widehat{\mathbf{x}}$ to a normalized image $\mathbf{x}$:

$$\mathbf{x} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}} \tag{4}$$

where:

$$x_n = \frac{1}{A}\widehat{x}_n, \qquad A = \sqrt{\sum_{n=1}^{N} \widehat{x}_n^2} \tag{5}$$

The above described normalizations with respect to scale and brightness give us normalized object image sets and a normalized universal set. In the following discussion, we will simply refer to these as the object and universal sets.

In practice, image sets can be obtained in several ways. If the geometrical model and reflectance properties of an object are known, its images for different pose and illumination directions can be synthesized using well-known rendering algorithms. Here, we do not assume that object geometry and reflectance are given. Instead, we assume that we have a sample of each object that can be used for learning. One approach then is to use two robot manipulators; one grasps the object and shows it to the sensor in different poses while the other has a light source mounted on its end-effector and is used to vary illumination direction.

In our experiments, we have used the setup shown in Figure 1. The object is placed on a motorized turntable and its pose is varied about a single axis, namely, the axis of rotation of the turntable. The turntable position is controlled via software and can be varied with an accuracy of 0.1 degrees. Most objects have a finite number of stable configurations when placed on a planar surface. For such objects, the turntable is adequate as it can be used to vary pose for each of the object's stable configurations. Illumination direction is varied using the robot manipulator seen in Figure 1.

Figure 2 shows half the image set obtained by rotating the object in Figure 1 through 90

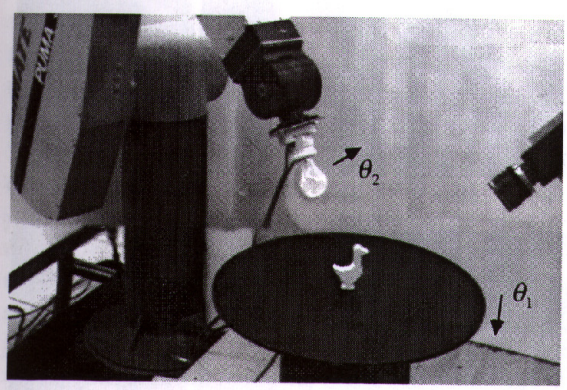


Fig. 1. Setup used to automatically acquire image sets. The object is placed on a motorized turntable.

discrete poses (4 degrees apart). Each image is 128×128 pixels in size and is normalized in scale and brightness as previously described. A similar image set is obtained for each illumination direction. In our experiments, we have used a total of 5 light source directions. Each object image set therefore has a total of 450 images. Note that these images can be discarded after the object's appearance representation is computed.

2.2 *Computing Eigenspaces*

Our first step is to compress the large image sets into low-dimensional subspaces that capture the gross appearance characteristics of objects. A suitable compression technique is based on the Karhunen-Loeve expansion (Fukunaga 1990). We compute two types of subspaces; the universal eigenspace that is obtained from the universal image set, and object eigenspaces computed from individual object image sets.

To compute the universal eigenspace, the average $\mathbf{c}$ of all images in the set is subtracted from each image. This ensures that the eigenvector with the largest eigenvalue represents the dimension in eigenspace in which the variance of images is maximum in the correlation sense. In other words, it is the most important dimension of the eigenspace. A new image set is obtained by subtracting $\mathbf{c}$ from each image:

$$\mathbf{X} \triangleq \{\mathbf{x}_{1,1}^{(1)} - \mathbf{c}, \mathbf{x}_{2,1}^{(1)} - \mathbf{c}, \ldots, \mathbf{x}_{R,1}^{(1)} - \mathbf{c}, \ldots, \mathbf{x}_{RL}^{(P)} - \mathbf{c}\} \tag{6}$$

The image matrix $\mathbf{X}$ is $N \times M$, where $M = RLP$ is the total number of images in the universal set, and N is the number of pixels in each image. Next, we define the *covariance matrix*:

$$\mathbf{Q} \triangleq \mathbf{X}\mathbf{X}^{\mathrm{T}} \tag{7}$$

This matrix is $N \times N$, clearly a very large matrix since a large number of pixels constitute

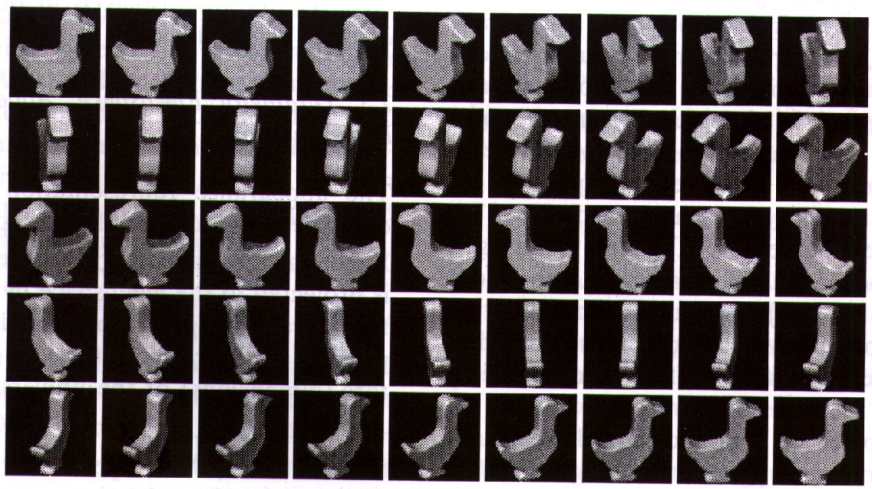


Fig. 2. Image set obtained by rotating the object in Figure 1 about a single axis. These images are scale and brightness normalized.

an image. The eigenvectors $\mathbf{e}_i$ and the corresponding eigenvalues λ_i of $\mathbf{Q}$ are determined by solving the well-known eigenstructure decomposition problem:

$$\lambda_i \mathbf{e}_i = \mathbf{Q}\mathbf{e}_i \tag{8}$$

Though all N eigenvectors are needed to represent images exactly, only a small number ($k \ll N$) is generally sufficient for capturing the primary appearance characteristics of the objects. These k eigenvectors correspond to the largest k eigenvalues of $\mathbf{Q}$ and constitute the universal eigenspace. How many dimensions should the eigenspace have in order for us to represent images with adequate accuracy? One approach is to select k such that the first k eigenvectors of $\mathbf{Q}$ capture important appearance variations in the image set, that is:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \geq T_1 \tag{9}$$

where the threshold T_1 is close to, but less than, unity. Note that our covariance matrix is positive definite and thus the eigenvalues are also positive. For the objects used in our experiments, eigenspaces with 20 or less dimensions ($k \leq 20$) are found to be quite adequate. Since the universal eigenspace is computed using images of all objects, it is *tuned* to discriminate between images of different objects.

Computing the eigenvectors of a large matrix such as $\mathbf{Q}$ can prove computationally intensive. Practical solutions to this problem have been investigated in the areas of image compression and pattern recognition. A few efficient algorithms are described in Appendix A. In our experiments, we have used the STA algorithm (see Appendix A). The result is a set of eigenvalues $\{\lambda_i \mid i = 1, 2, \ldots, k\}$ where $\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k\}$, and a corresponding set of eigenvectors $\{\mathbf{e}_i \mid i = 1, 2, \ldots, k\}$. Note that each eigenvector is of size N, i.e. the size of an image.

Once an object has been recognized, we are interested in finding its pose in the image. The accuracy of pose estimation depends on the ability of the recognition system to discriminate between different images of the same object. Hence, pose estimation is best done in an eigenspace that is tuned to the appearance of a single object. To this end, we compute an object eigenspace from each of the object image sets. The procedure for computing object eigenspaces is identical to that used for the universal eigenspace. In this case, the average $\mathbf{c}^{(p)}$ of all images of object p is subtracted from each of the object's images. The resulting images are used to compute the covariance matrix $\mathbf{Q}^{(p)}$. The eigenvectors $\{\mathbf{e}_i^{(p)} \mid i = 1, 2, \ldots, k\}$ of $\mathbf{Q}^{(p)}$, corresponding to the k largest eigenvalues, constitute the eigenspace of object p. As an example, Figure 3 shows six eigenvectors (shown as images) computed from the image set in Figure 2. The eigenvectors are displayed in descending order of their eigenvalues.

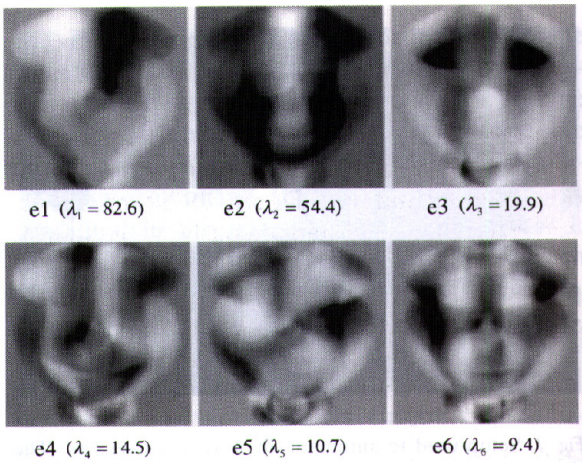


Fig. 3. Eigenvectors corresponding to the six largest eigenvalues computed for the image set shown in Figure 2.

2.3 Parametric Eigenspace: Appearance Representation

The appearance representation of object p is constructed in universal eigenspace as follows. Each learning sample $\mathbf{x}_{r,l}^{(p)}$ in the image set of p is projected to the eigenspace by first subtracting the average image $\mathbf{c}$ from it and finding the dot product of the result with each of the eigenvectors (dimensions) of the universal eigenspace:

$$\mathbf{g}_{r,l}^{(p)} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]^{\mathrm{T}} (\mathbf{x}_{r,l}^{(p)} - \mathbf{c}) \tag{10}$$

Once again, the subscript r represents rotation and l the illumination direction. By projecting all learning samples in this manner, we obtain a set of discrete points in universal eigenspace.

Since consecutive images are strongly correlated, their projections in eigenspace are close to one another. The exact reasons for this proximity are given in the following section. This is in contrast to the geometrical "aspects" (Koenderink and van Doorn 1979) of an object that change instantaneously with viewing direction at accidental views, causing a visual event. Such aspect changes are caused by the sudden appearance of previously occluded object features such as faces. In our appearance representation, changes in object aspect seldom cause drastic changes in correlation. On the other hand, such changes may be expected when the object is either highly specular or has high-frequency texture. In such cases, an incremental pose or illumination variation can cause dramatic changes in image brightness. In the absence of such effects, however, the discrete points $\mathbf{g}_{r,l}^{(p)}$ describe a smoothly varying manifold in eigenspace:

$$\mathbf{g}^{(p)}(\theta_1, \theta_2, \ldots, \theta_m) \tag{11}$$

where, $\theta_1, \theta_2, \ldots, \theta_m$ are continuous pose and illumination parameters. The above manifold is referred to as the *parametric eigenspace representation*; it is a compact representation of the appearance of object p. In practice, the number of parameters used for pose and illumination can vary. Therefore, depending on the application, the appearance representation may be a curve, surface, or volume in k-dimensional space; we shall refer to this geometrical "variety" as simply a manifold. As stated earlier, in our experiments we rotate the object about a single axis. Further, for ease of implementation, illumination direction is confined to a single plane in three-dimensional space. Thus, the appearance representation is a bivariate manifold:

$$\mathbf{g}^{(p)}(\theta_1, \theta_2) \tag{12}$$

In the present implementation, we have used a standard cubic-spline interpolation algorithm (Press et al. 1988) to compute the above manifold from the points $\mathbf{g}_{r,l}^{(p)}$. If the manifolds of two objects intersect in universal eigenspace, the intersection corresponds to poses of the two objects for which their images are very similar in appearance. Such images are inherently ambiguous; they simply do not contain sufficient information for unique object identification.

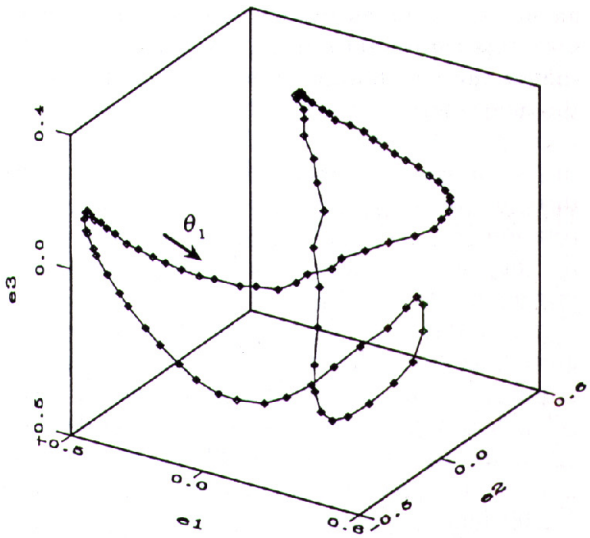


Fig. 4. Parametric eigenspace representation computed using the image set shown in Figure 2. Only the three most prominent dimensions of the eigenspace are displayed here. The dots correspond to projections of learning samples. Since illumination is constant in this case, appearance is given by a curve with a single parameter (rotation) rather than a surface.

Using the above procedure, a manifold is also constructed in the object's eigenspace. Learning samples are projected onto this space to obtain the discrete points:

$$\mathbf{f}_{r,l}^{(p)} = \left[\mathbf{e}_1^{(p)}, \mathbf{e}_2^{(p)}, \ldots, \mathbf{e}_k^{(p)}\right]^{\mathrm{T}} \left(\mathbf{x}_{r,l}^{(p)} - \mathbf{c}^{(p)}\right) \tag{13}$$

where, $\mathbf{c}^{(p)}$ is the average of all images in the object image set. Using cubic splines, the points $\mathbf{f}_{r,l}^{(p)}$ are interpolated to obtain the manifold:

$$\mathbf{f}^{(p)}(\theta_1, \theta_2) \tag{14}$$

This continuous manifold enables us to find poses of the object that are not included in the learning samples. It also enables us to compute accurate pose estimates under illumination directions that lie in between the discrete ones used in the learning stage.

Figure 4 shows the eigenspace representation of the object in Figure 1. This eigenspace was computed using the image set in Figure 2. The figure shows only three of the most significant eigenvectors since it is difficult to display and vi-

sualize higher-dimensional spaces. The appearance representation in this case is a curve rather than a surface since illumination is constant for the image set in Figure 2. The dots on the curve correspond to projections of individual images in the learning set. The continuous curve passing through the discrete points is parametrized by rotation, θ_1. A closed curve is obtained since the object is rotated a full 360 degrees with increments of 4 degrees.

It is interesting to compare the memory required to store the object image set and that required for the manifold representation. Consider, learning images obtained for 100 discrete rotations and 100 discrete illumination directions. This gives us an object image set with 10,000 images where each image includes, say, 128×128 pixels. In contrast, the manifold is described by 10,000 discrete points in an eigenspace that has, say, 10 dimensions. In this case, the manifold representation yields a 1,600:1 compression ratio. Of course, the 10 eigenvectors corresponding to the largest eigenvalues also need to be stored as they represent the dimensions of the eigenspace. Each eigenvector however is only the size of an image.

2.4 *Correlation and Distance in Eigenspace*

Before we proceed to describe recognition and pose estimation, it is worthwhile to discuss some relevant properties of the eigenspace representation. We show in this section that the distance between two points in eigenspace is a measure of correlation between the corresponding brightness images. Consider two images $\mathbf{x}_m$ and $\mathbf{x}_n$ that belong to the image set used to compute an eigenspace. Let the points $\mathbf{g}_m$ and $\mathbf{g}_n$ be projections of the two images in eigenspace. It is well-known in pattern recognition theory (Oja 1983) that each of the images can be expressed in terms of its projection:

$$\mathbf{x}_m = \sum_{i=1}^{N} g_{m_i}\mathbf{e}_i + \mathbf{c} \tag{15}$$

where $\mathbf{c}$ is once again the average of the entire image set. The above expression states that the image $\mathbf{x}_m$ can be exactly represented as a weighted sum of all N eigenvectors of the image set. The individual coefficients g_{m_i} are the coordinates of the point $\mathbf{g}_m$. Note that our eigenspaces are composed of only k eigenvectors. Since these eigenvectors correspond to the largest eigenvalues, they represent the most significant variations within the image set. Hence, $\mathbf{x}_m$ can be approximated by the first k terms in the above summation:

$$\mathbf{x}_m \approx \sum_{i=1}^{k} g_{m_i}\mathbf{e}_i + \mathbf{c} \tag{16}$$

As a result of the brightness normalization described in section 2.1, $\mathbf{x}_m$ and $\mathbf{x}_n$ are unit vectors. The similarity between the two images can be determined by finding the *sum-of-squared-difference* (SSD) between brightness values in the images. This measure is extensively used in machine vision for template matching, establishing correspondence in binocular stereo, and feature tracking in motion estimation. It is known that SSD can be related to correlation as:

$$\begin{aligned}\|\mathbf{x}_m - \mathbf{x}_n\|^2 &= (\mathbf{x}_m - \mathbf{x}_n)^{\mathrm{T}}(\mathbf{x}_m - \mathbf{x}_n) \\ &= 2 - 2\mathbf{x}_m^{\mathrm{T}}\mathbf{x}_n\end{aligned} \tag{17}$$

where, $\mathbf{x}_m^{\mathrm{T}}\ \mathbf{x}_n$ is the correlation. Maximizing correlation, therefore, corresponds to minimizing SSD and thus maximizing similarity between the images. Alternatively, the SSD can be expressed in terms of the eigenspace points $\mathbf{g}_m$ and $\mathbf{g}_n$ using (16):

$$\|\mathbf{x}_m - \mathbf{x}_n\|^2 \approx \left\| \sum_{i=1}^{k} g_{m_i}\mathbf{e}_i - \sum_{i=1}^{k} g_{n_i}\mathbf{e}_i \right\|^2 \tag{18}$$

The right hand side of the above expression can be simplified to obtain:

$$\begin{aligned}\left\| \sum_{i=1}^{k} g_{m_i}\mathbf{e}_i - \sum_{i=1}^{k} g_{n_i}\mathbf{e}_i \right\|^2 &= \left\| \sum_{i=1}^{k} (g_{m_i} - g_{n_i})\mathbf{e}_i \right\|^2 \\ &= \sum_{i=1}^{k}\sum_{j=1}^{k} \mathbf{e}_i^{\mathrm{T}}\mathbf{e}_j \\ &\quad \times (g_{m_i} - g_{n_j})^2 \\ &= \|\mathbf{g}_m - \mathbf{g}_n\|^2\end{aligned} \tag{19}$$

The last simplification results from the eigenvectors being orthogonal; $\mathbf{e}_i^{\mathrm{T}}\mathbf{e}_j = 1$ when $i = j$, and

0 otherwise. From (18) and (19), we get:

$$\|\mathbf{x}_m - \mathbf{x}_n\|^2 \approx \|\mathbf{g}_m - \mathbf{g}_n\|^2 \tag{20}$$

The above relation implies that the square of the Euclidean distance between points $\mathbf{g}_m$ and $\mathbf{g}_n$ is an approximation of the SSD between images $\mathbf{x}_m$ and $\mathbf{x}_n$. In other words, the closer the projections are in eigenspace, the more highly correlated are the images. This property of the eigenspace makes it appealing from the perspective of machine vision, where, correlation is very often used as a measure of similarity between images.

3 Object Recognition and Pose Estimation

A brute force approach to appearance-based recognition is to compare an unknown input image with all images (corresponding to different poses and illumination directions) of all objects of interest to the recognition system. Such an approach is equivalent to exhaustive template matching. Clearly, this is impractical from a computational perspective given the large number of images involved. The parametric eigenspace representation enables us to accomplish essentially the same task but in a very efficient manner. Since the eigenspace is optimal for computing correlation between images, we can project an input image to eigenspace and simply look for the closest manifold (object).

Consider an image of a scene that includes one or more of the objects we have learned. We assume that the objects are not occluded by other objects in the scene when viewed from the sensor direction, and that the image regions corresponding to objects have been segmented away from the scene image. Each segmented image region is normalized in scale and brightness as described in section 2.1. This ensures that (a) the input image is of the same size as the eigenvectors (dimensions) of the universal and object eigenspaces; (b) the recognition system is invariant to fluctuations in the intensity of illumination; and (c) the recognition system is invariant to magnification, i.e. the distance of objects from the sensor. Since the current implementation assumes that the viewing direction is fixed, invariance to object magnification is valid only when image projection can be approximated by the weak-perspective model, i.e. scaling followed by orthographic projection (Huttenlocher and Ullman 1990).

Since the universal eigenspace is tuned to discriminate between different objects, the input image $\mathbf{y}$ is first projected to this space:

$$\mathbf{z} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]^{\mathrm{T}}(\mathbf{y} - \mathbf{c}) \tag{21}$$

The recognition problem then is to find the object p whose manifold $\mathbf{z}$ lies on. Due to factors such as image noise, aberrations in the imaging system, and digitization effects, $\mathbf{z}$ may not lie exactly on an object manifold. Therefore, we find the object p that gives the minimum distance $d_1^{(p)}$ between its manifold $\mathbf{g}^{(p)}(\theta_1, \theta_2)$ and $\mathbf{z}$:

$$d_1^{(p)} = \min_{\theta_1, \theta_2} \|\mathbf{z} - \mathbf{g}^{(p)}(\theta_1, \theta_2)\| \tag{22}$$

If $d_1^{(p)}$ is within some pre-determined threshold value, we conclude that the input image is of object p. If not, the input image is not of any of the objects learned by the system. It is important to note that the manifold representation results in more reliable recognition than using just the cluster of the points $\mathbf{g}_{r,l}^{(p)}$ in eigenspace. Manifolds of different objects can intersect each other or even be intertwined, in which cases, using nearest cluster algorithms could easily lead to incorrect recognition results.

Once the object p in the input image $\mathbf{y}$ is recognized, $\mathbf{y}$ is projected to the object's eigenspace. This space is tuned to variations in the appearance of a single object and hence is most appropriate for pose estimation. Mapping the input image to this space gives the point $\mathbf{z}^{(p)}$. The pose estimation problem may be stated as follows: Find the rotation parameter θ_1 and the illumination parameter θ_2 that minimize the distance $d_1^{(p)}$ between $\mathbf{z}^{(p)}$ and the manifold $\mathbf{f}^{(p)}$:

$$d_2^{(p)} = \min_{\theta_1, \theta_2} \|\mathbf{z} - \mathbf{f}^{(p)}(\theta_1, \theta_2)\| \tag{23}$$

The θ_1 value obtained is the pose of the object in the input image. Figure 5(a) shows an input image of the object whose parametric eigenspace

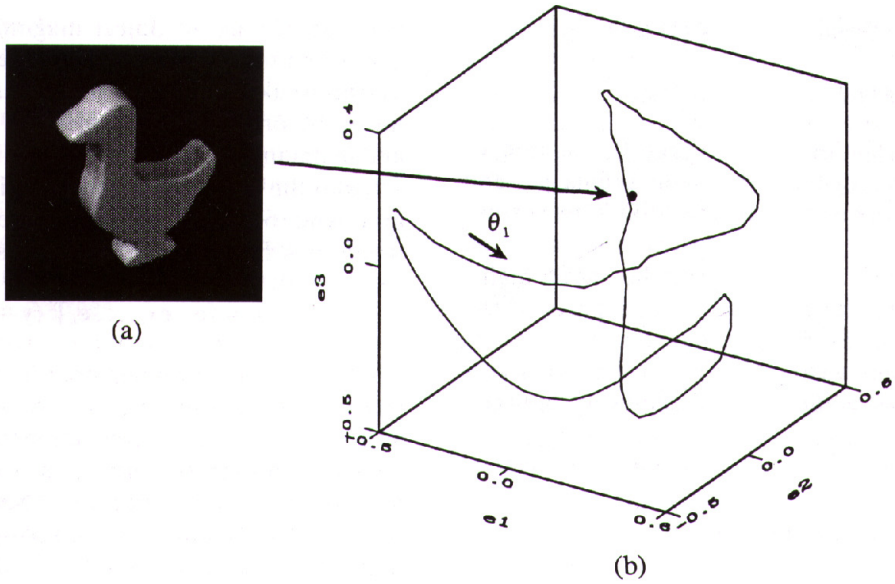


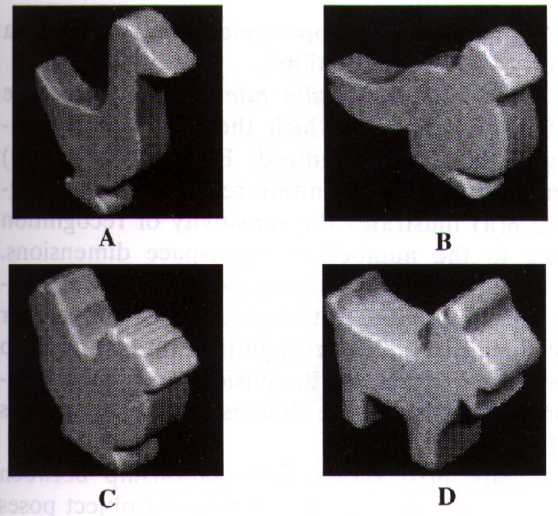
Fig. 5. (a) An input image. (b) The image is mapped to a point in object eigenspace. The location of the point on the parametric curve determines the pose of the object in the image.

was shown in Figure 4. The pose of the object in this image lies in between two consecutive poses used in the learning stage. In Figure 5(b), the input image is mapped to the object eigenspace and is seen to lie close to the parametric curve of the object.

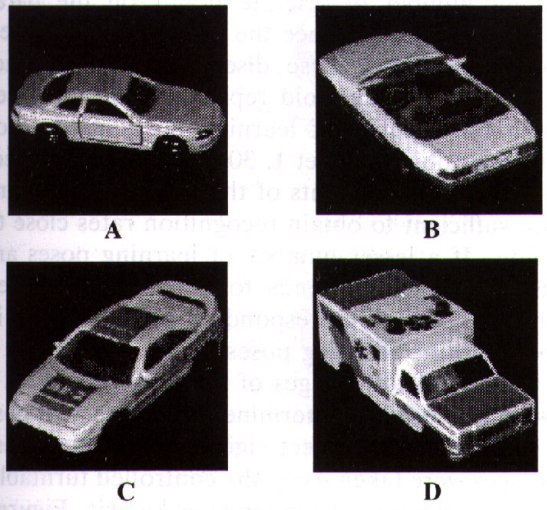
Mapping an input image to universal and object eigenspaces is computationally simple. As mentioned earlier, the eigenspaces are typically less than 20 in dimensions. The projection of an input image to a 20-dimensional space requires 20 dot products of the input image with the orthogonal eigenvectors that constitute the space. This procedure can easily be done in real-time (frame rate of a typical image digitizer) using simple and inexpensive hardware.

Once the image has been projected to an eigenspace, we need to find the manifold point that is closest to it. One approach is to use an exhaustive search algorithm that computes the distance of the input point from a large number of points uniformly sampled from the parametrized manifolds. This is clearly inefficient both in memory and time; all the sampled manifold points need to be stored, and the Euclidean distance of the input point with respect to each manifold point must be computed. The computational complexity is $O(kn)$ where n is the number of manifold points and k is the dimensionality of the eigenspace.

We have implemented two alternative schemes. The first is an efficient technique for binary search in multiple dimensions (Nene and Nayar 1994). This algorithm uses a carefully designed data structure to facilitate quick search through the multi-dimensional eigenspace in $O(k \log_2 n)$. This approach is particularly effective when the number of manifold points is relatively small. The second approach (Mukherjee and Nayar 1990) uses three-layered radial basis function (RBF) networks proposed by Poggio and Girosi (Poggio and Girosi 1991) to learn the mapping between input points and manifold parameters (object number and pose). The complexity of this method depends on the number of networks used and their sizes, and is in practice comparable to that of the binary search approach. Such a network implicitly interpolates, or reconstructs, manifolds from the discrete eigenspace points $\mathbf{g}_{r,l}^{(p)}$ and $\mathbf{f}_{r,l}^{(p)}$ and therefore does not require the use of cubic spline interpolation followed by the resampling of manifolds. This advantage however comes with a slight sacrifice in pose estimation accuracy.



(a) Object Set 1



(b) Object Set 2

Fig. 6. Two object sets used in the learning and recognition experiments. (a) Set **1** includes four objects with uniform reflectance but similar shapes. (b) Set **2** includes objects with complex reflectance and geometric properties.

4 Experiments

The setup used to conduct experiments was described in section 2.1. The object is placed on a computer-controlled turntable (see Figure 1) and its pose is varied about a single axis, namely, the axis of rotation of the turntable. We assume that the object is illuminated by the ambient lighting of the environment that is not expected to change between the learning and recognition stages. This ambient illumination is of relatively low intensity. The main source of brightness is an additional light source whose direction can vary. Illumination is varied using a 6 degree-of-freedom robot manipulator (see Figure 1) with a light source mounted on its end-effector. Images of the object are sensed using a 512×480 pixel CCD camera and are digitized using an Analogics frame-grabber board. This setup enables us to automatically acquire image sets. Since hundreds of images are obtained for each object, a substantial amount of storage memory is required. Note that an eigenspace can be computed only after the complete image set is obtained. We have used a 1.6 Gbyte hard disk to store the image sets. These images of course can be discarded once the objects have been learned, i.e. after the low-dimensional parametric manifolds are constructed.

Experiments were conducted on two sets of objects that are shown in Figure 6(a) and Figure 6(b). Set **1** includes objects with uniform reflectance but shapes that appear very similar for certain poses. We will see that for such poses, subtle differences in shading and occluding contours are sufficient for the recognition system to correctly identify the objects. Set **2** includes objects with complex appearance characteristics. These are good examples of objects whose shape and reflectance properties are very difficult to model with any reasonable precision. In addition, their images include strong specular reflections as well as complex interreflections, physical processes that are known to be difficult to analyze. Again, we will see that appearance-based recognition does remarkably well in identifying and estimating the poses of such complex objects. Table 1 summarizes the poses and light source directions used to acquire object image sets. For each object, we have used 5 different light source directions, and 90 poses for each source direction. This gives us a total of 1800 images in the universal image set and 450 images in each object image set. Each of these images is automatically normalized in scale and brightness. Each normalized image is 128×128 pixels in size.

Table 1. Image sets for each of the two object sets shown in Figure 6. The 1080 test images used for recognition are different from the 1800 images used for learning.

Learning Samples	Test Samples for Recognition
4 Objects	4 Objects
5 Light Source Directions	3 Light Source Directions
90 Poses	90 Poses
1800 Images	1080 Images

4.1 Appearance Representation

The STA algorithm (Murase and Lindenbaum 1992) was used to compute 10 eigenvectors corresponding to the 10 largest eigenvalues for each of the above universal and object sets. The algorithm is implemented and executed on a Sun SPARC 2 workstation. Once the eigenspaces are computed, individual images are projected onto the universal and object eigenspaces. This process is efficient since it only requires the dot product of each learning sample with the 10 dimensions of the eigenspaces. The spline interpolation algorithm outlined in (Press et al. 1988) was used to construct parametric manifolds from the discrete points.

Figure 7 shows parametric manifolds computed for all objects in Figure 6. Once again, the manifolds are displayed as surfaces in three-dimensional eigenspaces. Since three dimensions are not sufficient to illustrate the object discriminating power of the universal eigenspace, we have shown only object eigenspaces. The parameters of the surfaces are pose (θ_1) and illumination direction (θ_2). The surfaces are narrow since we have used only 5 light source positions in the learning stage.

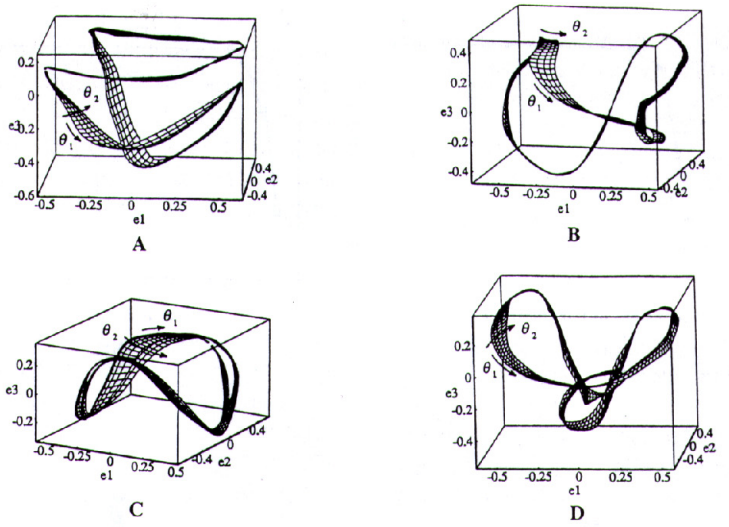
4.2 Recognition and Pose Estimation Results

These experiments were conducted independently for object sets **1** and **2**. For each set, a total of 1080 test images were used. These images are detailed in Table 1 and were taken at object poses that lie in between the ones used to obtain the learning samples. Each test image is first normalized in scale and brightness and then projected to universal eigenspace. The binary search algorithm (Nene and Nayar 1994) is used to search for the closest manifold point, this process takes approximately 60 msec on a Sun SPARC workstation.

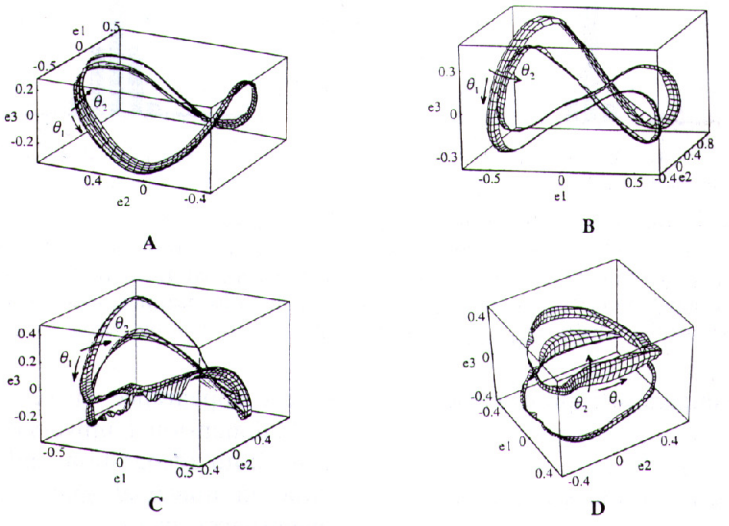
We define *recognition rate* as the percentage of test images for which the object in the image is correctly recognized. Figures 8(a) and (b) summarize the recognition results for set **1**. Figure 8(a) illustrates the sensitivity of recognition rate to the number of eigenspace dimensions. Clearly, the discriminating power of the universal space is expected to increase with the number of dimensions. The recognition rate is found to be poor if less than 4 dimensions are used but approaches unity as the dimensionality approaches 10.

Figure 8(b) shows the relationship between recognition rate and the number of object poses used for learning. If the pose increments used in the learning stage are small, we obtain a larger number of learning samples and hence a larger number of discrete points on the parametric manifold. Since the manifold is obtained by interpolating these discrete points, the accuracy of the manifold representation increases with the number of learning poses used. For the four objects in set **1**, 30 poses of each object (12 degree increments of the turntable position) are sufficient to obtain recognition rates close to unity. If a lesser number of learning poses are used, recognition tends to be unreliable when the test images correspond to poses that lie in between the learning poses.

The 1080 test images of the 4 objects in set **1** were also used to determine the accuracy of pose estimation using object eigenspaces. Since these images were taken using the controlled turntable, the actual pose in each image is known. Figures. 8(c) and (d) show histograms of pose errors (in degrees) computed for the 1080 test images. In Figure 8(c), 450 learning samples (90 poses and 5 source directions) were used to compute each object eigenspace. In this experiment, all the object eigenspaces used were 8-dimensional. In Figure 8(d), 90 learning samples (18 poses and 5 source directions) were used. The pose estimation results in both cases are found to be very accurate. In the first case, the average absolute pose error computed using all 1080 images is 0.5 degrees, while in the second case the average error is 1.0 degrees. Similar recognition and pose estimation experiments were conducted



(a) Object Set 1



(b) Object Set 2

Fig. 7. Appearance manifolds in object eigenspace for the objects in (a) set **1**; and (b) set **2**. For display, only the three most important dimensions of each eigenspace are shown. The manifolds are reduced to surfaces in three-dimensional space.

using object set **2**. In this case, the average absolute pose error is 0.5 degrees when 90 learning poses are used, and 1.2 degrees when 18 poses are used. The sensitivity of recognition to image noise and segmentation error is analyzed in (Murase and Nayar 1994).

4.3 An Example Application

Figure 9 shows recognition and pose estimation results for an image sequence of a moving car. Four of the 30 frames obtained are shown in Figure 9(a). A simple segmentation algorithm was implemented to extract the moving object from the background. This segmentation algorithm estimates the moving image region (white box) by subtracting an image of just the stationary background from each images in the sequence. The moving image region is normalized with respect to scale and brightness and then projected to universal eigenspace. The car model is identified and its pose computed using the manifolds shown in Figure 7(b). Figure 9(b)

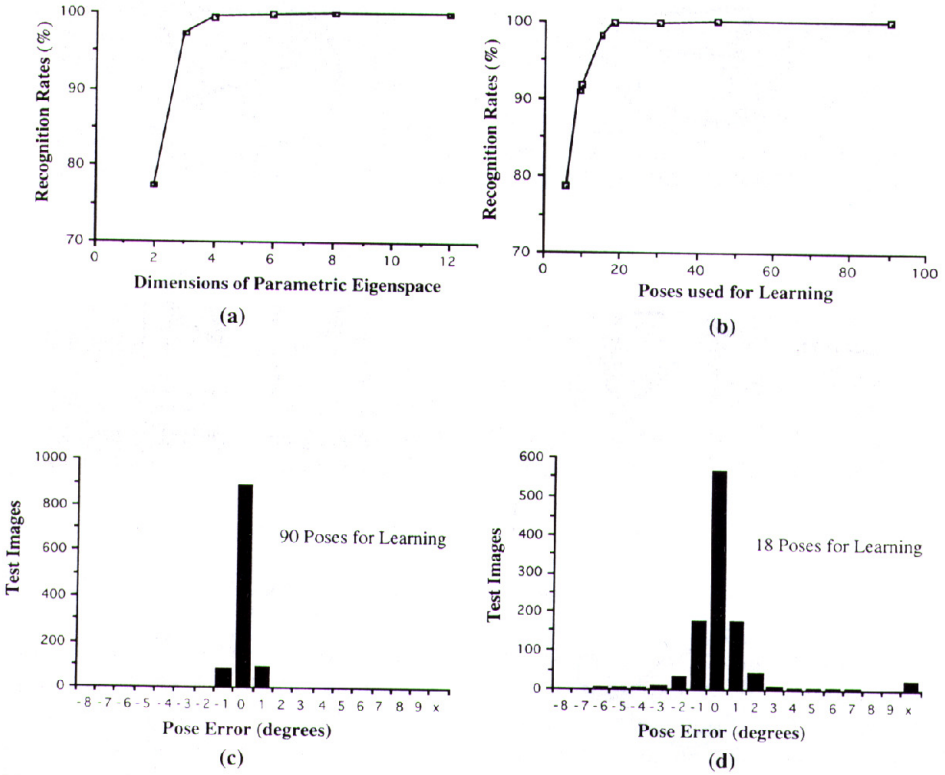


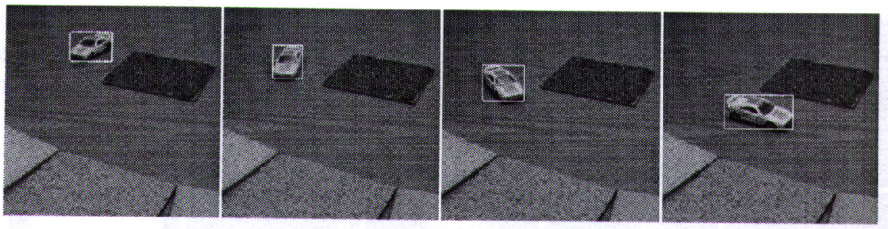
Fig. 8. Recognition and pose estimation results for object set 1. (a) Recognition rate plotted as a function of the number of universal eigenspace dimensions used. (b) Recognition rate plotted as a function of the number of discrete poses of each object used in the learning stage. In both cases recognition rates were computed using all 1080 test images detailed in Table 1. Histogram of error in computed object pose when (c) 90 poses are used for learning; and (d) 18 poses used for learning. The average absolute pose error is 0.5 degrees in the first case and 1.0 degrees in the second case.

shows the learning samples with poses closest to the computed poses.

4.4 Automated Recognition System

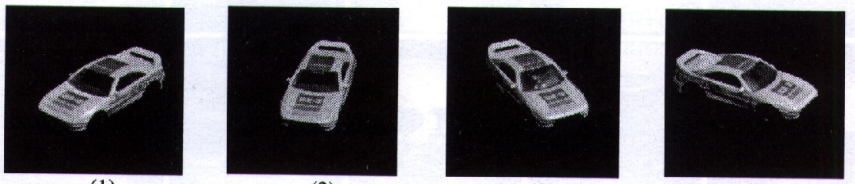
Based on the above results, we implemented a recognition system with 20 objects in its database (see Figure 10). These objects vary from smoothly curved shapes with uniform reflectance, to fairly complex shapes with intricate textures and specularities. Developing CAD models of such objects could prove extremely cumbersome and time-consuming. Both learning and recognition are done in a laboratory environment where illumination remains more or less unchanged. As a result, appearance manifolds are reduced to curves parametrized by just object pose. Each object image set includes 72 learning images (5 degree increments in pose), resulting in a universal set of 1440 images. The object appearance curves were constructed in a 20-dimensional universal eigenspace. In this case, both recognition and pose estimation are done in universal space, i.e. separate object eigenspaces were not computed. The entire learning process, including, image acquisition, computation of eigenvectors, and construction of appearance curves was completed in less than 12 hours using a Sun SPARC workstation.

The recognition system automatically detects significant changes in the scene, waits for the scene to stabilize, and then digitizes an image. In the present implementation, objects are presented to the system one at a time and a dark background is used to alleviate object segmentation. The complete recognition process, including, segmentation, scale and brightness normalization, image projection in universal eigenspace, and search for the closest object and pose is accomplished in less than 1 second



(1) (2) (3) (4)

(a) Automatic segmentation of the moving object.



(1) (2) (3) (4)

(b) Learning sample with closest pose.

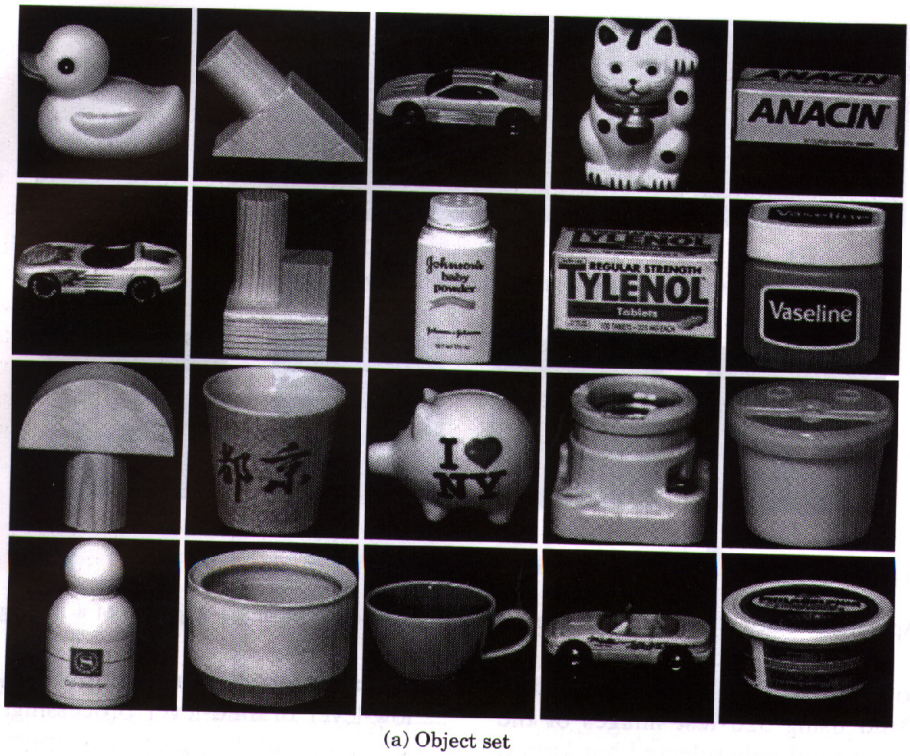
Fig. 9. Appearance-based recognition and pose estimation applied to the image sequence of a moving car.

on the Sun workstation. The robustness of this system was tested using 320 test images of the 20 objects taken at randomly selected but known poses of the objects. All test images were correctly identified by the system. A histogram of the absolute pose error is shown in Figure 10(c); the average and standard deviation of the absolute pose error were found to be 1.59 degrees and 1.53 degrees, respectively.

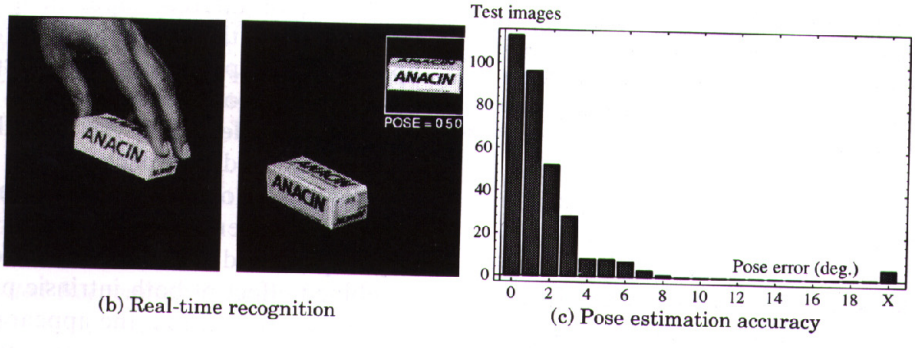
5 Discussion

In this section, we briefly discuss several issues related to the proposed learning and recognition technique. Some of these may be viewed as merits while others as limitations leading to open research problems for the future.

- **Appearance Based Approach:** Both learning as well as recognition are done using just two-dimensional brightness images. This is in strong contrast to traditional recognition algorithms that require the extraction of geometric features such as edges, lines, or geometric invariants. Such features are often difficult to compute with robustness, and reliable algorithms for extracting them from images are still being actively researched. Our approach of using raw image data directly for learning and recognition, without any significant low-level or mid-level processing, is a major advantage of the proposed approach. Interesting research directions include the use of processed, or filtered, input images, and the integration of the present appearance based approach with previously developed geometry based recognition techniques.
- **Shape and Reflectance:** An appealing feature of the proposed approach is that it does not require any knowledge of the shape and reflectance properties of objects. By varying object pose and illumination, we capture the combined effect of both intrinsic properties of an object. In addition, the appearance for any given pose and illumination may include specular highlights and complex interreflections between points on the object surface. All of these phenomena together produce the overall appearance of the object. Since we are representing object appearance, none of the above phenomena need be modeled or analyzed.
- **Segmentation and Occlusion:** Learning and recognition require the segmentation of object regions. In structured environments, the background can be controlled, in which case, simple thresholding is sufficient for robust segmentation. In the case of moving objects, simple background subtraction algorithms such as the



(a) Object set



(b) Real-time recognition

(c) Pose estimation accuracy

Fig. 10. An automated recognition system with 20 objects in the database. A complete recognition cycle takes less than 1 second on a Sun SPARC workstation.

one used for the moving car sequence (Figure 9) can be effective for segmentation. In the context of general scenes, however, segmentation poses a serious problem and can be viewed as a limitation of the proposed method. The method also requires that the objects not be occluded. Since the technique is based on direct appearance matching, it simply cannot handle substantial degrees of occlusions. This is a second limitation of the parametric eigenspace approach. Segmentation and occlusion therefore present challenging research directions for appearance based recognition.

- **Dimensionality of the Eigenspace:** The number of eigenspace dimensions needed for representation depends on the appearance characteristics of the objects as well as the number of objects of interest to the recognition system. If the objects have complex textures, a larger number of dimensions would be needed for accurate representation. Further, as the number of objects increases, a larger number of dimensions may be needed for robust recognition. The exact number of dimensions required for

any given set of objects is difficult to quantify since there are no simple relationships between an object's intrinsic properties and its eigenspace representation. Such relationships need to be explored before eigenspace representations can be optimized with respect to storage space.

- **Parameters of the Manifold:** In our experiments, we have used only two parameters for object representation, one for object rotation and the other for illumination direction. A single rotation parameter is sufficient for objects that have a finite number of stable configurations. In general, however, three parameters are needed to describe the pose of an object in three dimensions. An additional two parameters would be required for varying illumination in three dimensions; only two parameters are sufficient since rotations about the source direction need not be considered. From a practical perspective, the number of parameters would be too many if arbitrary rotations and illumination directions are considered. In general, manifolds with up to three parameters can be used without much of a problem. These three parameters can be selected depending on the application at hand.
- **Computations for Learning:** In our present implementation, we use the universal eigenspace for object identification. The space is computed using image sets of all objects. If a new object is to be learned by the recognition system, the universal eigenspace must be recomputed with a universal image set that includes the image set of the new object. Since the universal eigenspace is time consuming to compute, we would like to avoid recomputing it when new objects need to be learned. One approach is to project the new object's learning samples to the previously computed universal eigenspace. Though the resulting manifold is only an approximation, it would generally be sufficient for object identification. A more reliable approach is to compute a modified universal space by orthogonalizing the previous universal space and the new object's own eigenspace. One practical method, among several others, to achieve this is Gram-Schmidt orthogonalization (Householder 1964).
- **Computations for Recognition:** Though the learning process poses large memory requirements and is computationally intensive, it is done off-line. The time taken to learn an object is generally not as crucial as the time needed to recognize it. In contrast to learning, recognition and pose estimation are simple and computationally very efficient, requiring only the projection of the input image to universal and object eigenspaces and search for the closest manifold points. Recognition and pose estimation can therefore be accomplished in real-time (frame-rate of 30 Hz) using simple and inexpensive hardware. In contrast, most model-based recognition algorithms are too slow for practical applications.
- **Applications:** We have presented appearance based learning and recognition as a general approach for visual perception. However, the parametric eigenspace representation can be used to solve a variety of specific vision problems, such as, illumination planning (Murase and Nayar 1994), visual positioning and tracking of robot manipulators (Nayar et al. 1994), and visual inspection. In many of these applications, factors such as segmentation and occlusion are not problems, and high-dimensional manifold representations are not required. For such applications, the appearance representation presented here offers powerful and efficient solutions.

6 Conclusion

We presented a new representation for machine vision called the parametric eigenspace. While representations previously used in computer vision are based on object geometry, the proposed one describes object appearance. We proposed a method for automatically learning an object's parametric eigenspace. Such learning techniques are fundamental to the advancement of visual perception. We developed efficient object recognition and pose estimation algorithms that are based on the parametric eigenspace representation. The learning and recognition algorithms were tested on objects with complex shape and reflectance properties. A statistical analysis of the errors in recognition and pose estimation demonstrate the proposed approach to be very

robust to factors such as image noise and quantization. These results suggest the feasibility of a real-time appearance based recognition system with a very large object database.

A Computing Eigenvectors of Large Image Sets

Let $\mathbf{X}$ be an $N \times M$ image matrix, where M is the total number of images and N the number of pixels in each image. We are interested in finding the eigenvectors of the covariance matrix $\mathbf{Q} = \mathbf{X}\mathbf{X}^T$, an $N \times N$ matrix. The calculation of the eigenvectors of such a large matrix is computationally intensive. Fast algorithms for solving this problem have been a topic of active research in the area of image coding and compression. Here, we briefly describe three algorithms. We refer to these as the conjugate gradient, singular value decomposition, and spatial temporal adaptive algorithms. Each algorithm may be viewed as a modification of the previous ones. The first two of these are described in detail in (Oja 1983).

Conjugate Gradient

A practical approach to computing the eigenvectors of large matrices is to use iterative methods. A reasonably efficient iterative scheme that suggests itself is the conjugate gradient method. There are several variations to the conjugate gradient approach (Yang et al. 1989). The problem is formulated as one of finding the eigenvalues and eigenvectors that maximize a scalar function. A function that is often used is the Raleigh quotient $F(\mathbf{e})$:

$$F(\mathbf{e}) = \frac{(\mathbf{e}^T\mathbf{Q}\mathbf{e})}{(\mathbf{e}^T\mathbf{e})} \tag{24}$$

Conjugate gradient is used to find the vector $\mathbf{e}_1$ that maximizes F. The corresponding value of the Raleigh quotient, $F(\mathbf{e}_1)$, is the largest eigenvalue λ_1 of the covariance matrix $\mathbf{Q}$. Once the largest eigenvalue and the corresponding eigenvector are computed in this manner, $\mathbf{Q}$ is modified to remove the dimension associated with the computed eigenvector. The Raleigh quotient is then used with the modified covariance matrix to determine the next largest eigenvalue and corresponding eigenvector. The iterative modification of $\mathbf{Q}$ can be summarized as:

$$\begin{aligned} \mathbf{Q}_1 &= \mathbf{Q} \\ \mathbf{Q}_s &= \mathbf{Q}_{s-1} - \lambda_{s-1}\mathbf{e}_{s-1}\mathbf{e}_{s-1}^T \end{aligned} \tag{25}$$

The above procedure can be repeated until a desired number of eigenvectors of $\mathbf{Q}$ are computed. Since in our case $\mathbf{Q}$ is a very large matrix ($N \times N$), each iteration of the conjugate gradient algorithm can prove expensive.

Singular Value Decomposition

If the number of images M is much smaller than the number of pixels N in each image, a much more efficient algorithm may be used. This algorithm, developed by Murakami and Kumar (Murakami and Kumar 1982), uses the implicit covariance matrix $\tilde{\mathbf{Q}}$, where:

$$\tilde{\mathbf{Q}} = \mathbf{X}^T\mathbf{X} \tag{26}$$

Note that $\tilde{\mathbf{Q}}$ is an $M \times M$ matrix and therefore much smaller than $\mathbf{Q}$ when the number of images in $\mathbf{X}$ is smaller than the number of pixels in each image. Using the conjugate gradient algorithm described above, the M eigenvectors of $\tilde{\mathbf{Q}}$ can be computed. These can be computed much faster than the first M eigenvectors of $\mathbf{Q}$ due to the disparity in the sizes of the two matrices. Using singular value decomposition (SVD), Murakami and Kumar (Murakami and Kumar 1982) show that the M largest eigenvalues and the corresponding eigenvectors of $\mathbf{Q}$ can be determined from the M eigenvalues and eigenvectors of $\tilde{\mathbf{Q}}$ as:

$$\begin{aligned} \lambda_i &= \tilde{\lambda}_i \\ \mathbf{e}_i &= \tilde{\lambda}_i^{-\frac{1}{2}}\mathbf{X}\tilde{\mathbf{e}}_i \end{aligned} \tag{27}$$

Here, λ_i and $\mathbf{e}_i$ are the i^{th} eigenvalue and eigenvector of $\mathbf{Q}$, while $\tilde{\lambda}_i$ and $\tilde{\mathbf{e}}_i$ are the i^{th} eigenvalue and eigenvector of $\tilde{\mathbf{Q}}$. Since we are only interested in the first k eigenvectors of $\mathbf{Q}$, where $k < M$, the SVD algorithm can be used. It is not useful, however, when more than M eigenvectors are needed.

Spatial Temporal Adaptive

Murase and Lindenbaum (Murase and Lindenbaum 1992) have recently developed the spatial temporal adaptive (STA) algorithm that takes the above SVD algorithm one step further to achieve substantial improvements in computational efficiency. They observe that the computation of $\tilde{\mathbf{Q}}$ from the image matrix $\mathbf{X}$ is itself expensive. Therefore, each image in $\mathbf{X}$ is divided into "blocks" and image data in each block is compressed using the discrete cosine transform (DCT) (Chen et al. 1977). Due to spatial correlation within an image, each image block is typically represented by a small number of DCT coefficients. Further, blocks at the same location in consecutive images are often highly correlated and have the same DCT coefficients. A set of such blocks are referred to as a "superblock" and is represented by the DCT coefficients of a single block. In this manner, the image matrix $\mathbf{X}$ is compressed to obtain a small number of DCT coefficients. Individual elements of $\tilde{\mathbf{Q}}$ can then be computed from the DCT coefficients of the blocks and superblocks of $\mathbf{X}$. This procedure of computing $\tilde{\mathbf{Q}}$ saves substantial computations. Next, the conjugate gradient algorithm is used to compute the eigenvalues and eigenvectors of $\tilde{\mathbf{Q}}$. These eigenvalues and eigenvectors are used to determine the eigenvectors $\mathbf{e}_i$ and eigenvalues λ_i of the original covariance matrix $\mathbf{Q}$ by applying the SVD technique (see equation 27). This step also requires the use of $\mathbf{X}$ which is now compressed using DCT. Computations are once again saved by determining $\mathbf{e}_i$ in DCT domain and then transforming it back to spatial domain using inverse DCT.

Murase and Lindenbaum have compared the performance of the STA algorithm with the conjugate gradient and SVD algorithms described previously. Their results show the STA algorithm to be superior in performance to both algorithms, often 10 or more times faster than the SVD algorithm.

Acknowledgments

This research was conducted at the Center for Research in Intelligent Systems, Department of Computer Science, Columbia University. It was supported in part by the David and Lucile Packard Fellowship and in part by ARPA Contract No. DACA 76-92-C-0007. The authors thank Sameer A. Nene of Columbia University for help in implementing the real-time recognition system.

References

Besl, P.J., and Jain, R.C. 1985. "Three-Dimensional Object Recognition," *ACM Computing Surveys,* Vol. 17, No. 1, pp. 75–145.

Chen, W.H., Smith, H., and Fralick, S.C. 1977. "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communications,* Vol. 25, pp. 1004–1009.

Chin, R.T., and Dyer, C.R. 1986. "Model-Based Recognition in Robot Vision," *ACM Computing Surveys,* Vol. 18, No. 1, pp. 67–108.

Edelman, S. and Weinshall, D. 1991. "A self-organizing multiple-view representation of 3D objects," *Biological Cybernetics,* Vol. 64, pp. 209–219.

Edelman, S., Bulthoff, H., and Weinshall, D. 1989. "Stimulus familiarity determines recognition strategy for novel 3D objects," A.I. Memo No. 1138, AI Lab, MIT.

Fan, T.J., Medioni, G., and Nevatia, R. 1988. "Recognizing 3-D Objects Using Surface Descriptions," *Proc. of Intl. Conference on Computer Vision,* pp. 474–481, Florida.

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition,* Academic Press, London.

Householder, A.S. 1964. *The theory of matrices in numerical analysis,* Dover Publications, New York.

Huttenlocher, D.P., and Ullman, S. 1990. "Recognizing solid objects by alignment with an image," *International Journal of Computer Vision,* Vol. 5, No. 2, pp. 195–212.

Ikeuchi, K., and Suehiro, T. 1992. "Recognizing Assembly Tasks using Face-Contact Relations," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* pp. 154–160.

Koenderink, J.J., and van Doorn, A.J. 1979. "The internal representation of solid shape with respect to vision," *Biological Cybernetics,* Vol. 32, pp. 211–216.

Mukherjee, S., and Nayar, S.K. 1994. "Appearance Based Recognition of 3D Objects using RBF Networks," Technical Report, Department of Computer Science, Columbia University (in preparation).

Murakami, H., and Kumar, V. 1982. "Efficient Calculation of Primary Images from a Set of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 4, No. 5, pp. 511–515.

Murase, H., Kimura, F., Yoshimura, M., and Miyake, Y. 1981. "An Improvement of the Auto-Correlation Matrix in Pattern Matching Method and Its Application to Handprinted 'HIRAGANA'," *Trans. IECE,* Vol. J64-D, No. 3, pp. 276–283.

Murase, H., and Lindenbaum, M. 1992. "Spatial Temporal Adaptive Method for Partial Eigenstructure Decomposition of Large Images," *NTT Technical Report No. 6527.* Also *IEEE Transactions on Image Processing* (in press).

Murase, H., and Nayar, S.K., 1994. "Illumination Planning for Object Recognition in Structured Environments," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 31–38.

Nayar, S.K., Murase, H., and Nene, S.A. 1994. "Learning, Positioning, and Tracking Visual Appearance," *Proc. IEEE Conf. on Robotics and Automation,* pp. 3237–3244.

Nene, S.A., and Nayar, S.K. 1994. "Binary Search Through Multiple Dimensions," Technical Report CUCS-018-94, Department of Computer Science, Columbia University.

Oja, E. 1983. *Subspace methods of Pattern Recognition,* Research Studies Press, Hertfordshire.

Poggio, T., and Girosi, F. 1990. "Networks for Approximation and Learning," *Proceedings of the IEEE,* Vol. 78, No. 9, pp. 1481–1497.

Poggio, T., and Edelman, S. 1990. "A networks that learns to recognize three-dimensional objects," *Nature,* Vol. 343, pp. 263–266.

Press, W., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. 1988. *Numerical Recipes in C,* Cambridge University Press, Cambridge.

Sirovich, L., and Kirby, M. 1987. "Low dimensional procedure for the characterization of human faces," *Journal of Optical Society of America,* Vol. 4, No. 3, pp. 519–524.

Tarr, M., and Pinker, S. 1989. "Mental rotation and orientation-dependence in shape recognition," *Cognitive Psychology,* Vol. 21, pp. 233–282.

Turk, M.A., and Pentland, A.P. 1991. "Face Recognition Using Eigenfaces," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* pp. 586–591.

Ullman, S., and Basri, R. 1991. "Recognition by Linear Combination of Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 13, No. 10, pp. 992–1006.

Yang, X., Sarkar, T.K., and Arvas, E. 1989. "A Survey of Conjugate Gradient Algorithms for Solution of Extreme Eigen-Problems of a Symmetric Matrix," *IEEE Transactions on Acoustics, Speech and Signal Processing,* Vol. 37, No. 10, pp. 1550–1555.