Learning and Recognition of 3D Objects from Appearance *

Hiroshi Murase

NTT Basic Research Labs
3-9-11 Midori-cho, Musashino-shi
Tokyo 180, Japan
murase@siva.ntt.jp

Shree K. Nayar

Department of Computer Science
Columbia University
New York, N.Y. 10027
nayar@cs.columbia.edu

Abstract

We address the problem of automatically learning object models for recognition and pose estimation. In contrast to the traditional approach, we formulate the recognition problem as one of matching visual 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 of an object and are constant, pose and illumination vary from scene to scene. We present a new compact representation of object appearance 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 large image set is compressed to obtain a low-dimensional subspace, called the eigenspace, in which the object is represented as a hypersurface. Given an unknown input image, the recognition system projects the image onto the eigenspace. The object is recognized based on the hypersurface it lies on. The exact position of the projection on the hypersurface determines the object's pose in the image. We have conducted experiments using several objects with complex appearance characteristics. We conclude with a discussion on various issues related to the learning and recognition techniques proposed in the paper.

1 Introduction

One of the primary goals of an intelligent vision system is to recognize objects in an image and compute their pose in the three-dimensional scene. Such a recognition system has wide applications ranging from autonomous navigation to visual inspection. 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 on the use of geometric (shape) models [1] [2] 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, develop object models using this 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 complicated geometric properties. It is clear that recognition systems of the future must be capable of acquiring object models without human assistance. In other words, recognition systems must be able to automatically *learn* the objects of interest.

Visual learning is clearly a well-developed and vital component of biological vision systems. If a human is handed an 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 to learn objects, it is clear that the overall appearance of the object plays a critical role in its perception. In contrast to biological systems, machine vision systems today have little or no learning capabilities. Hence, visual learning is now emerging as an topic of research interest (see [11], [10] [15], [3], [5]). The goal of this paper is to advance this important but relatively unexplored area of machine vision.

Here, we present a technique for automatically learning object models from images. The appearance

*This research was conducted at the Center for Research in Intelligent Systems, Department of Computer Science, Columbia University. This research was supported in part by the David and Lucile Packard Fellowship and in part by ARPA Contract No. DACA 76-92-C-0007.

0-8186-3692-0/93 $03.00 © 1993 IEEE

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* of the object that do not vary, 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 illumination directions and object poses. 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. Since all images in the set are of the same object, any two consecutive images are correlated to a large degree. The problem then is to compress this large image set into 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 [9] [4], this method computes the eigenvectors of an image set. The eigenvectors form an orthogonal basis for the representation of individual images in the image 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* for the image set. From the perspective of machine vision, the eigenspace has a very attractive property. When it is composed of all the eigenvectors of an image set, it is optimal in a *correlation* sense: If any two images from the set are projected onto the 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 [6] and human face recognition [13], [14]. These applications lie within the domain of pattern classification and do not address the problem of learning or using complete parametrized models of the objects of interest.

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 the intensity of illumination. The eigenspace for the image set is obtained by computing the most prominent eigenvectors of the image set. Next, all images in the object's image set (the learning samples) are projected onto the eigenspace to obtain a set of points. These points lie on a *hypersurface* that is parametrized by object pose and illumination. The hypersurface is computed from the discrete points using the cubic spline interpolation technique. 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 hypersurface in two different eigenspaces; the universal eigenspace and the object's own eigenspace. The *universal eigenspace* is computed by using the image sets of all objects of interest to the recognition system, and the *object eigenspace* is computed using only images of the object. We show that the universal eigenspace is suited for discriminating between objects, whereas the object eigenspace is better for pose estimation. Object 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 onto the universal eigenspace to identify the object. After the object is recognized, the image is projected onto the object eigenspace and the location of the projection on the object's parametrized hypersurface determines its pose in the scene.

The learning of an object requires the acquisition of a large image set and the computationally intensive process of finding eigenvectors. However, the learning stage is done off-line and hence can afford to be relatively slow. In contrast, recognition and pose estimation are often subject to strong time constraints, and our approach offers a very simple and computationally efficient solution. We have conducted extensive experimentation to demonstrate the power of the parametric eigenspace representation. We conclude with a discussion on the advantages and limitations of the proposed learning and recognition methods. The fundamental contributions of this paper can be summarized as follows. (a) The parametric eigenspace is presented as a new representation of object appearance. (b) Using this representation, object models are automatically learned from appearance by varying pose and illumination. (c) Both learning and recognition are accomplished without prior knowledge of the object's shape and reflectance.

2 Visual Learning of Objects

In this section, we discuss the learning of object models using the parametric eigenspace representation. First, we discuss the acquisition of object image sets. The eigenspaces are computed using the image sets and each object is represented as a parametric hypersurface. Throughout this section, we will use a sample object to describe the learning process. In the next section, we discuss the recognition and pose estimation of objects using the parametric eigenspace representation.

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 a zero brightness value and the object region is re-sampled such that the larger of its two dimensions fits the image size we have selected for the image set representation. We now have a scale normalized image. This image is written as a vector $\hat{\mathbf{x}}$ by reading pixel brightness values from the image in a raster scan manner:

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

The appearance of an object depends on its shape and reflectance properties. These are intrinsic properties of the object that do not vary. The appearance of the object also depends on the pose of the object and the illumination conditions. Unlike the intrinsic properties, object pose and illumination are expected to vary from scene to scene. If the illumination conditions of the environment are constant, the appearance of the object is affected only by its pose. Here, we assume that the object is illuminated by the ambient lighting of the environment as well as one additional distant light source whose direction may vary. Hence, all possible appearances of the object can be captured by varying object pose and the light source direction with respect to the viewing direction of the sensor. We will denote each image as $\hat{\mathbf{x}}_{r,l}^{(p)}$ where r is the rotation or pose parameter, l represents 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** and can be expressed as:

$$\left\{ \hat{\mathbf{x}}_{1,1}^{(p)}, \ldots..., \hat{\mathbf{x}}_{R,1}^{(p)}, \hat{\mathbf{x}}_{1,2}^{(p)}, \ldots..., \hat{\mathbf{x}}_{R,L}^{(p)} \right\} \tag{2}$$

Here, R and L are the total number of discrete poses and illumination directions, respectively, used to obtain the image set. 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 the object image sets:

$$\begin{array}{l} \{ \hat{\mathbf{x}}_{1,1}^{(1)}, \ldots..., \hat{\mathbf{x}}_{R,1}^{(1)}, \hat{\mathbf{x}}_{1,2}^{(1)}, \ldots..., \hat{\mathbf{x}}_{R,L}^{(1)}, \\ \hat{\mathbf{x}}_{1,1}^{(2)}, \ldots..., \hat{\mathbf{x}}_{R,1}^{(2)}, \hat{\mathbf{x}}_{1,2}^{(2)}, \ldots..., \hat{\mathbf{x}}_{R,L}^{(2)}, \\ \\ \hat{\mathbf{x}}_{1,1}^{(P)}, \ldots..., \hat{\mathbf{x}}_{R,1}^{(P)}, \hat{\mathbf{x}}_{1,2}^{(P)}, \ldots..., \hat{\mathbf{x}}_{R,L}^{(P)} \} \end{array} \tag{3}$$

We assume that the imaging sensor used for learning and recognizing objects has a linear response, i.e. image brightness is proportional to scene radiance. We would like our recognition system to be unaffected by variations in the intensity of illumination or the aperture of the imaging system. This can be achieved by normalizing each of the images in the object and universal sets, such that, the total energy contained in the image is unity, i.e. $\| \mathbf{x} \| = 1$. This brightness normalization transforms each measured image $\hat{\mathbf{x}}$ to a normalized image $\mathbf{x}$:

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

where:

$$x_n = \frac{1}{\sigma}(\hat{x}_n), \quad \sigma = \sqrt{\sum_{n=1}^{N} (\hat{x}_n)^2} \tag{5}$$

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

The images 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. In this paper, 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 it and is used to vary the illumination direction. In our experiments, we have used a turntable to rotate the object in a single plane (see Fig. 1). This gives us pose variations about a single axis. A robot manipulator is used to vary the illumination direction. If the recognition system is to be used in an environment where the illumination (due to one or several sources) is not expected to change, the image set can be obtained by varying just object pose.

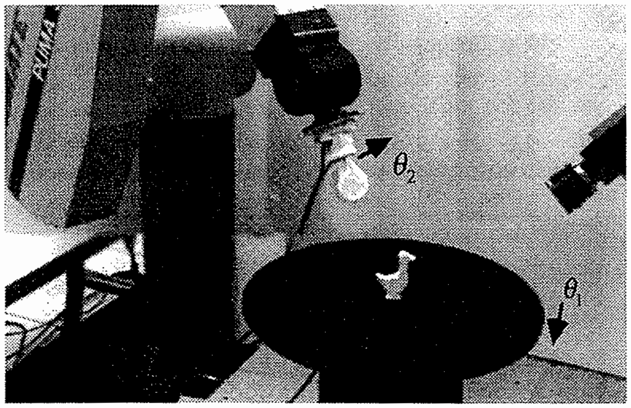


Figure 1: Setup used for automatic acquisition of object image sets. The object is placed on a motorized turntable.

2.2 Computing Eigenspaces

Consecutive images in an object image set tend to be correlated to a large degree since pose and illumination variations between consecutive images are small. Our first step is to take advantage of this correlation and compress large image sets into low-dimensional representations that capture the gross appearance characteristics of objects. A suitable compression technique is the Karhunen-Loeve expansion [4] where the eigenvectors of the image set are computed and used as orthogonal basis functions for representing individual images. Though, in general, all the eigenvectors of an image set are required for the perfect reconstruction of an object image, only a few are sufficient for the representation of objects for recognition purposes. We compute two types of eigenspaces; 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, we first subtract the average of all images in the universal set 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. The average $\mathbf{c}$ of all images in the universal image set is determined as:

$$\mathbf{c} \;=\; \frac{1}{RLP}\sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{l=1}^{L}\mathbf{x}_{r,l}{}^{(p)} \qquad (6)$$

A new image set is obtained by subtracting the average image $\mathbf{c}$ from each image in the universal set:

$$\mathbf{X} \triangleq \left\{\mathbf{x}_{1,1}{}^{(1)}-\mathbf{c},\ \ldots,\ \mathbf{x}_{R,1}{}^{(1)}-\mathbf{c},\ \ldots,\ \mathbf{x}_{R,L}{}^{(P)}-\mathbf{c}\right\} \qquad (7)$$

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. To compute eigenvectors of the image set we define the *covariance matrix* as:

$$\mathbf{Q} \triangleq \mathbf{X}\,\mathbf{X}^{\mathrm{T}} \qquad (8)$$

The covariance matrix is $N \times N$, clearly a very large matrix since a large number of pixels constitute an image. The eigenvectors $\mathbf{e}_i$ and the corresponding eigenvalues λ_i of $\mathbf{Q}$ are to be determined by solving the well-known eigenvector decomposition problem:

$$\lambda_i\,\mathbf{e}_i \;=\; \mathbf{Q}\,\mathbf{e}_i \qquad (9)$$

All N eigenvectors of the universal set together constitute a complete eigenspace. Any two images from the universal image set, when projected onto the eigenspace, give two discrete points. The distance between these points is a measure of the difference between the two images in the correlation sense [8]. Since the universal eigenspace is computed using images of all objects, it is the ideal space for discriminating between images of different objects.

Determining the eigenvalues and eigenvectors of a large matrix such as $\mathbf{Q}$ is a non-trivial problem. It is computationally very intensive and traditional techniques used for computing eigenvectors of small matrices are impractical. Since we are interested only in a small number (k) of eigenvectors, and not the complete set of N eigenvectors, efficient algorithms can be used. In our implementation, we have used the *spatial temporal adaptive* (STA) algorithm proposed by Murase and Lindenbaum [7]. This algorithm was recently demonstrated to be substantially more efficient than previous algorithms [7]. Using the STA algorithm the k most prominent eigenvectors of the universal image set are computed. The result is a set of eigenvalues $\{ \lambda_i \mid i = 1, 2, ..., k \}$ where $\{ \lambda_1 \geq \lambda_2 \geq ..... \geq \lambda_k \}$, and a corresponding set of eigenvector $\{ \mathbf{e}_i \mid i = 1, 2, ..., k \}$. Note that each eigenvector is of size N, i.e. the size of an image. These k eigenvectors constitute the universal eigenspace; it is an approximation to the complete eigenspace with N dimensions. We have found from our experiments that less than ten dimensions of the eigenspace are generally sufficient for the purposes of visual learning and recognition (i.e. $k \leq 10$). Later, we describe how objects in an unknown input image are recognized using the universal eigenspace.

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 similar to that used for the universal eigenspace. In this case, the average $\mathbf{c}^{(p)}$ of all images of object p is computed and subtracted from each of the object images. The resulting images are used to compute the covariance matrix $\mathbf{Q}^{(p)}$. The eigenspace for the object p is obtained by solving the system:

$$\lambda_i{}^{(p)} \mathbf{e}_i{}^{(p)} = \mathbf{Q}^{(p)} \mathbf{e}_i{}^{(p)} \qquad (10)$$

Once again, we compute only a small number ($k \leq 10$) of the largest eigenvalues $\{ \lambda_i{}^{(p)} \mid i = 1, 2, ..., k \}$ where $\{ \lambda_1{}^{(p)} \geq \lambda_2{}^{(p)} \geq ..... \geq \lambda_k{}^{(p)} \}$, and a corresponding set of eigenvector $\{ \mathbf{e}_i{}^{(p)} \mid i = 1, 2, ..., k \}$. An object eigenspace is computed for each object of interest to the recognition system.

2.3 Appearance Representation

We now represent each object as a hypersurface in the universal eigenspace as well as its own eigenspace. This new representation of appearance lies at the core of our approach to visual learning and recognition. Each appearance hypersurface is parametrized by two parameters; object rotation and illumination direction.

A parametric hypersurface for the object p is constructed in the universal eigenspace as follows. Each image $\mathbf{x}_{r,l}{}^{(p)}$ (learning sample) in the object image set is projected onto 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. The result is a point $\mathbf{g}_{r,l}{}^{(p)}$ in the eigenspace:

$$\mathbf{g}_{r,l}{}^{(p)} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_k]^{\mathrm{T}} \, (\mathbf{x}_{r,l}{}^{(p)} - \mathbf{c}) \qquad (11)$$

Once again the subscript r represents the rotation parameter and l is the illumination direction. By projecting all the learning samples in this manner, we obtain a set of discrete points in the universal eigenspace. Since consecutive object images are strongly correlated, their projections in eigenspace are close to one another. Hence, the discrete points obtained by projecting all the learning samples can be assumed to lie on a k-dimensional hypersurface that represents all possible poses of the object under all possible illumination directions. We interpolate the discrete points to obtain this hypersurface. In our implementation, we have used a standard cubic spline interpolation algorithm [12]. Since cubic splines are well-known we will not describe them here. The resulting hypersurface can be expressed as:

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

where θ_1 and θ_2 are the *continuous* rotation and illumination parameters. The above hypersurface is a compact representation of the object's appearance.

In a similar manner, a hypersurface is also constructed in the object's eigenspace by projecting the learning samples onto this space:

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

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

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

Once again, θ_1 and θ_2 are the rotation and illumination parameters, respectively. This continuous parameterization 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 illumination directions used in the learning stage. Fig.2 shows the parametrized eigenspace representation of the object shown in Fig.1. The figure shows only three of the most significant dimensions of the eigenspace since it is difficult to display and visualize higher dimensional spaces. The object representation in this case is a curve, rather than a surface, since the object image set was obtained using a single illumination direction while the object was rotated about a single axis. The discrete points on the curve correspond to projections of the learning samples in the object image set. The continuous curve passing through the points is parametrized by the rotation parameter θ_1 and is obtained using the cubic spline algorithm.

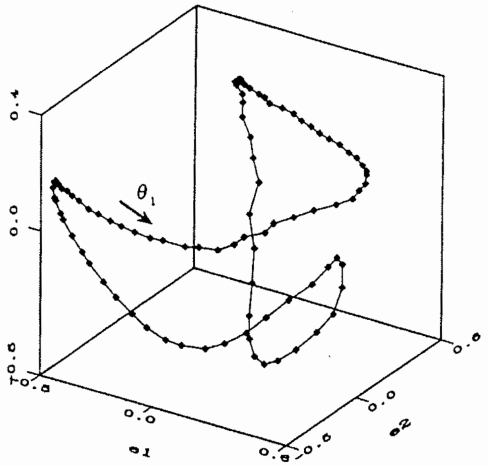


Figure 2: Parametric eigenspace representation of the object shown in Fig.1. Only the three most prominent dimensions of the eigenspace are displayed here. The points shown correspond to projections of the learning samples. Here, illumination is constant and therefore we obtain a curve with a single parameter (rotation) rather than a surface.

3 Recognition and Pose Estimation

Consider an image of a scene that includes one or more of the objects we have learned using the parametric eigenspace representation. 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. First, each segmented image region is normalized with respect to scale and brightness as described in section 2.1. This ensures that (a) the input image has the same dimensions as the eigenvectors (dimensions) of the parametric eigenspace, (b) the recognition system is invariant to object magnification, and (c) the recognition system is invariant to fluctuations in the intensity of illumination.

As stated earlier in the paper, the universal eigenspace is best tuned to discriminate between different objects. Hence, we first project the normalized input image $\mathbf{y}$ to the universal eigenspace. First, the average $\mathbf{c}$ of the universal image set is subtracted from $\mathbf{y}$ and the dot product of the resulting vector is computed with each of the eigenvectors that constitute the universal space. The k coefficients obtained are the coordinates of a point $\mathbf{z}$ in the eigenspace:

$$\mathbf{z} = [\mathbf{e}_1, \mathbf{e}_2, \ldots.., \mathbf{e}_k]^{\mathrm{T}} (\mathbf{y} - \mathbf{c}) \qquad (15)$$

$$= [z_1, z_2, \ldots.., z_k]^{\mathrm{T}} \qquad (16)$$

The recognition problem then is to find the object p whose hypersurface the point $\mathbf{z}$ 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 hypersurface. Hence, we find the object p that gives the minimum distance $d_1{}^{(p)}$ between its hypersurface $\mathbf{g}^{(p)}(\theta_1, \theta_2)$ and the point $\mathbf{z}$:

$$d_1{}^{(p)} = \min_{\theta_1, \theta_2} \| \mathbf{z} - \mathbf{g}^{(p)}(\theta_1, \theta_2) \| \qquad (17)$$

If $d_1{}^{(p)}$ is within some pre-determined threshold value, we conclude that the input image is of the object p. If not, we assume that input image is not of any of the objects used in the learning stage. It is important to note that the hypersurface representation of objects results in more reliable recognition than if the object is represented as just a cluster of the points $\mathbf{g}_{r,l}{}^{(p)}$ in eigenspace. The hypersurfaces 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 in the input image is recognized, we project the input image $\mathbf{y}$ to the eigenspace of the object. This eigenspace is tuned to variations in the appearance of a single object and hence is ideal for pose estimation. Mapping the input image to the object eigenspace gives the k-dimensional point:

$$\mathbf{z}^{(p)} = \left[\mathbf{e}_1{}^{(p)}, \mathbf{e}_2{}^{(p)}, \ldots.., \mathbf{e}_k{}^{(p)}\right]^{\mathrm{T}} (\mathbf{y} - \mathbf{c}^{(p)}) \qquad (18)$$

$$= \left[z_1{}^{(p)}, z_2{}^{(p)}, \ldots.., z_k{}^{(p)}\right]^{\mathrm{T}} \qquad (19)$$

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 $d_2{}^{(p)}$ between the point $\mathbf{z}^{(p)}$ and the hypersurface $\mathbf{f}^{(p)}$ of the object p:

$$d_2{}^{(p)} = \min_{\theta_1, \theta_2} \| \mathbf{z} - \mathbf{f}^{(p)}(\theta_1, \theta_2) \| \qquad (20)$$

The θ_1 value obtained represents the pose of the object in the input image. Fig. 3(a) shows an input image of the object whose parametric eigenspace was shown in Fig. 2. This input image is not one of the images in the learning set used to compute the object eigenspace. In Fig. 3b, the input image is mapped to the object eigenspace and is seen to lie on the parametric curve of the object. The location of the point on the curve determines the object's pose in the image. Note that the recognition and pose estimation stages are computationally very efficient, each requiring only the projection of an input image onto a low-dimensional (generally less than 10) eigenspace. Customized hardware can therefore be used to achieve real-time (frame-rate) recognition and pose estimation.

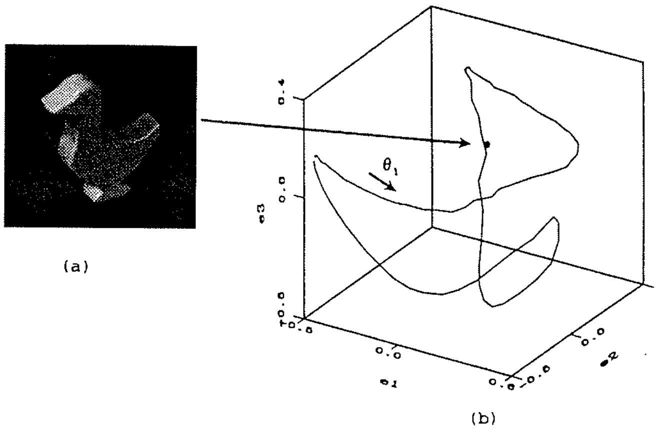


Figure 3: (a) An input image. (b) The input image is mapped to a point in the object eigenspace. The location of the point on the parametric curve determines the pose of the object in the input image.

4 Experimentation

We have conducted several experiments using complex objects to verify the effectiveness of the parametric eigenspace representation. This section summarizes some of our results. Fig. 1 in section 2 shows the set-up used to conduct the experiments reported here. 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 through software and can be varied with an accuracy of about 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.

We assume that the object is illuminated by the ambient lighting conditions of the environment that are 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. In most of our experiments, the source direction was varied manually. We are currently using a 6 degree-of-freedom robot manipulator (see Fig. 1) with a light source mounted on its end-effector. This enables us to vary the illumination direction via software. Images of the object are sensed using a 512×480 pixel CCD camera and are digitized using an Analogics frame-grabber board.

Table 1 summarizes the number of objects, light source directions, and poses used to acquire the image sets used in the experiments. For the learning stage, a total of 4 objects were used. These objects (cars) are shown in Fig. 4(a). 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 as described in section 2. Each normalized image is 128×128 pixels in size. The universal and object image sets were used to compute the universal and object eigenspaces. The parametric eigenspace representations of the four objects in their own eigenspaces are shown in Fig. 4(b).

Table 1: Image sets obtained for the learning and recognition stages. The 1080 test images used for recognition are different from the 1800 images used for learning.

Learning samples	Test samples for recognition
4 objects 5 light source directions 90 poses	4 objects 3 light source directions 90 poses
1800 images	1080 images

A large number of images were also obtained to test the recognition and pose estimation algorithms. All of these images are different from the ones used in the learning stage. A total of 1080 input (test) images were obtained. The illumination directions and object poses used to obtain the test images are different from the ones used to obtain the object image sets for learning. In fact, the test images correspond to poses and illumination directions that lie in between the ones used for learning. Each input image is first normalized and then projected onto the universal eigenspace. The object in the image is identified by finding the hypersurface that is closest to the input point in the universal eigenspace. Unlike the learning process, recognition is computationally simple and can be accomplished on a Sun SPARC 2 workstation in less than 0.2 second.

To evaluate the recognition results, we define the *recognition rate* as the percentage of input images for which the object in the image is correctly recognized. Fig. 5(a) illustrates the sensitivity of the recognition rate to the number of dimensions of the universal eigenspace. Clearly, the discriminating power of the universal eigenspace is expected to increase with the number of dimensions. For the objects used, the recognition rate is poor if less than 4 dimensions are used but approaches unity as the number of dimensions approaches 10. In general, however, the number of dimensions needed for robust recognition is expected to increase with the number of objects learned by the system. It also depends on the appearance characteristics of the objects used. From our experience, 10 dimensions are sufficient for representing objects with fairly complex appearance characteristics such as the ones shown in Fig. 4. Fig. 5(b) shows the relationship between recognition rate and the number of object poses used during the learning stage. For the four objects used in our experiments, we found that 30 poses of each object (12 degree increments of the turntable position) are sufficient to obtain recognition rates close to unity. Again, the number of learning poses needed depends on the objects used. If an object has high-frequency reflectance variations (texture) a larger number of learning samples will be required.

Once the object is recognized, the input image is projected onto the object's eigenspace and its pose is computed by finding the closest point on the parametric hypersurface. Once again, we use all 1080 input images of the 4 objects. Since these images were obtained using the controlled turntable, the actual object pose in each image is known. Fig. 6(a) and (b) shows histograms of the errors (in degrees) in the poses computed for the 1080 images. The error histogram in Fig. 6(a) is for the case where 450 learning samples (90 poses and 5 source directions) were used to compute the object eigenspace. The eigenspace used has 8 dimensions. The histogram in Fig. 6(b) is for the case where 90 learning samples (18 poses and 5 source directions) were used. The pose estimation results in both cases were found to be remarkably accurate. In the first case, the average of the absolute pose error computed using all 1080 images is found to be 0.5 degrees, while in the second case the average error is found to be 1.2 degrees.

Finally, Figure 7 shows recognition and pose estimation results for an image sequence of a moving car.

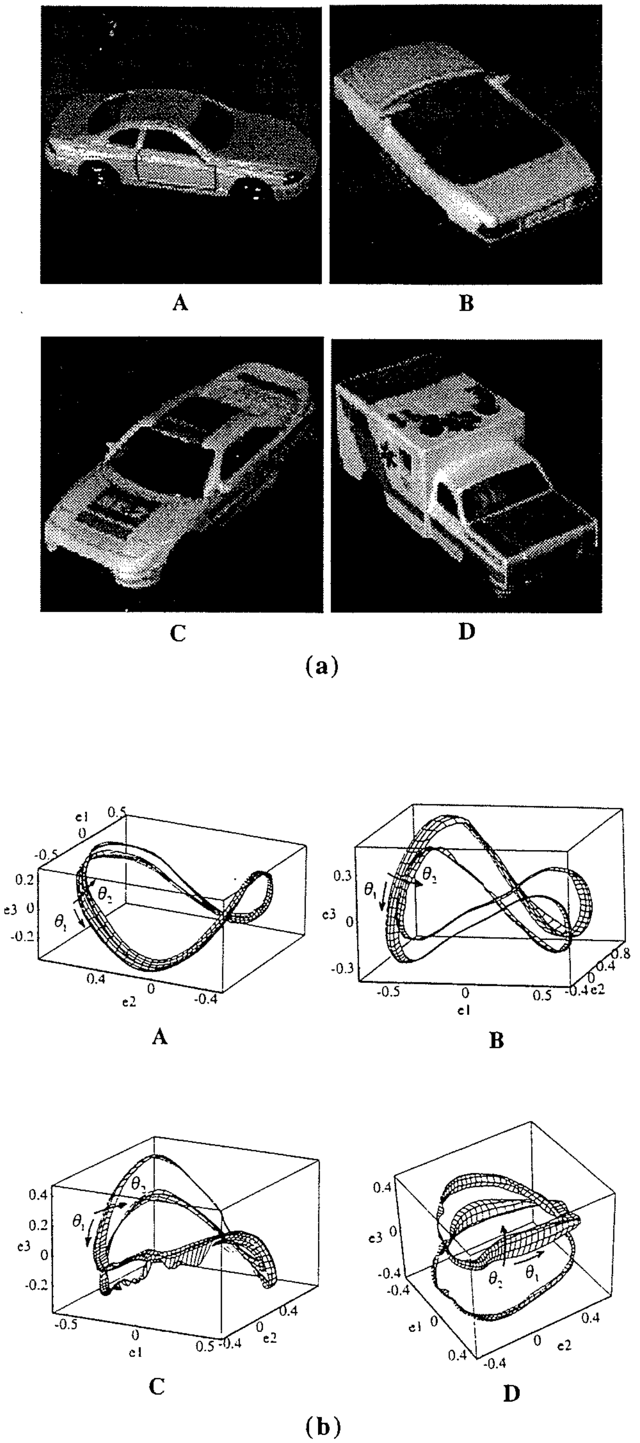


Figure 4: (a)The four objects used in the experiments. (b) The parametric hypersurfaces in object eigenspace computed for the four objects shown in (a). For display, only the three most important dimensions of each eigenspace are shown. The hypersurfaces are reduced to surfaces in three-dimensional space.

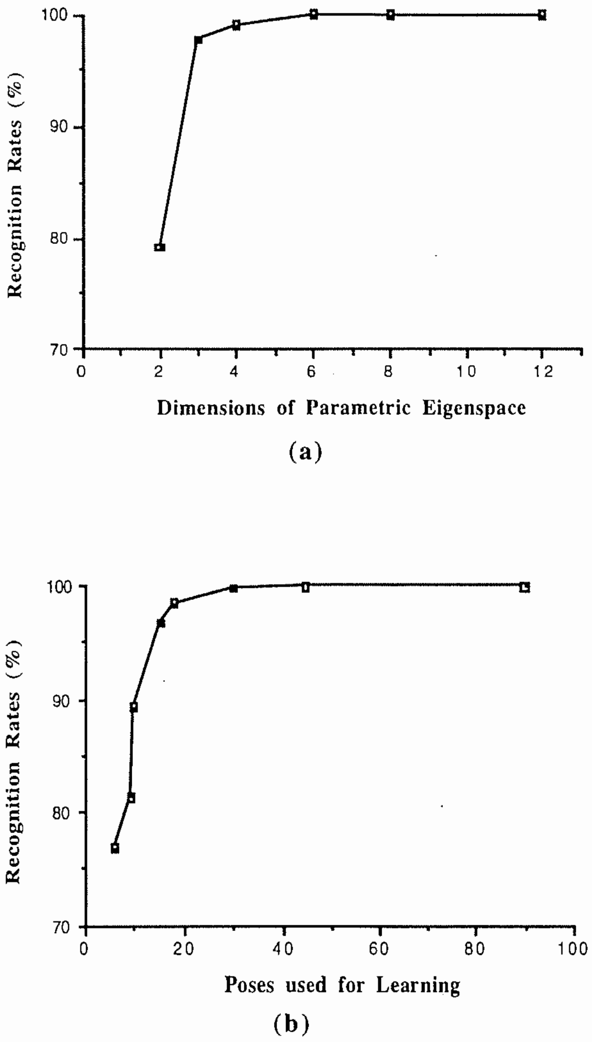


Figure 5: (a) Recognition rate plotted as a function of the number of universal eigenspace dimensions used to represent the parametric hypersurfaces. (b) Recognition rate plotted as a function of the number of discrete poses of each object used in the learning stage. In both cases the recognition rates were computed using all 1080 input images detailed in Table 1.

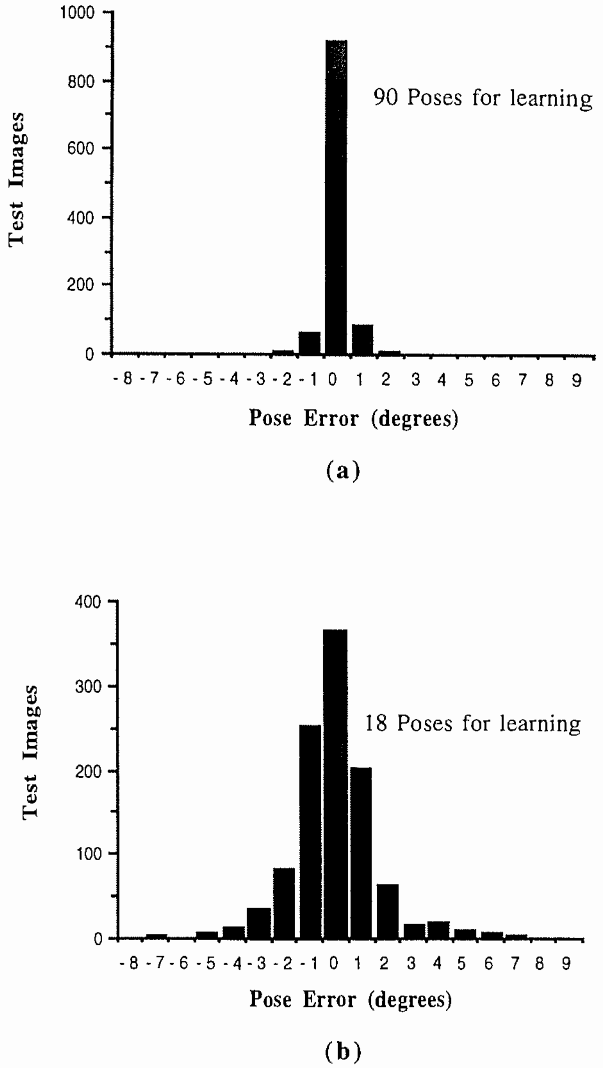


Figure 6: (a) Histogram of the error (in degrees) in computed object pose for the case where 90 poses are used in the learning stage. (b) Pose error histogram for the case where 18 poses are used in the learning stage. The average of the absolute error in pose for the complete set of 1080 test images is 0.5 in the first case and 1.2 in the second case.

Five of the 30 frames obtained are shown in Figure 7(a). A simple segmentation algorithm was implemented to extract the moving object from the background. The segmentation algorithm estimates the moving image region by subtracting an image of only the stationary background from each of the images in the sequence (see Figure 7(b)). Once the moving image region is identified, it is normalized with respect to scale and brightness and then projected to the universal eigenspace. The car model is recognized based on the hypersurface the projected image falls on. Next, the pose of the car is computed by projecting the normalized image region to the object eigenspace. Figure 7(c) shows the learning sample with pose closest to the computed pose, and Figure 7(d) shows the computed pose plotted as a function of image number.

5 Discussion

In this section, we briefly discuss several issues related to the proposed method. Some of these may be viewed as advantages while others as limitations.

- **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 geometric features are often difficult to compute with robustness. The use of raw image data directly for learning and recognition is a major advantage of the proposed approach.

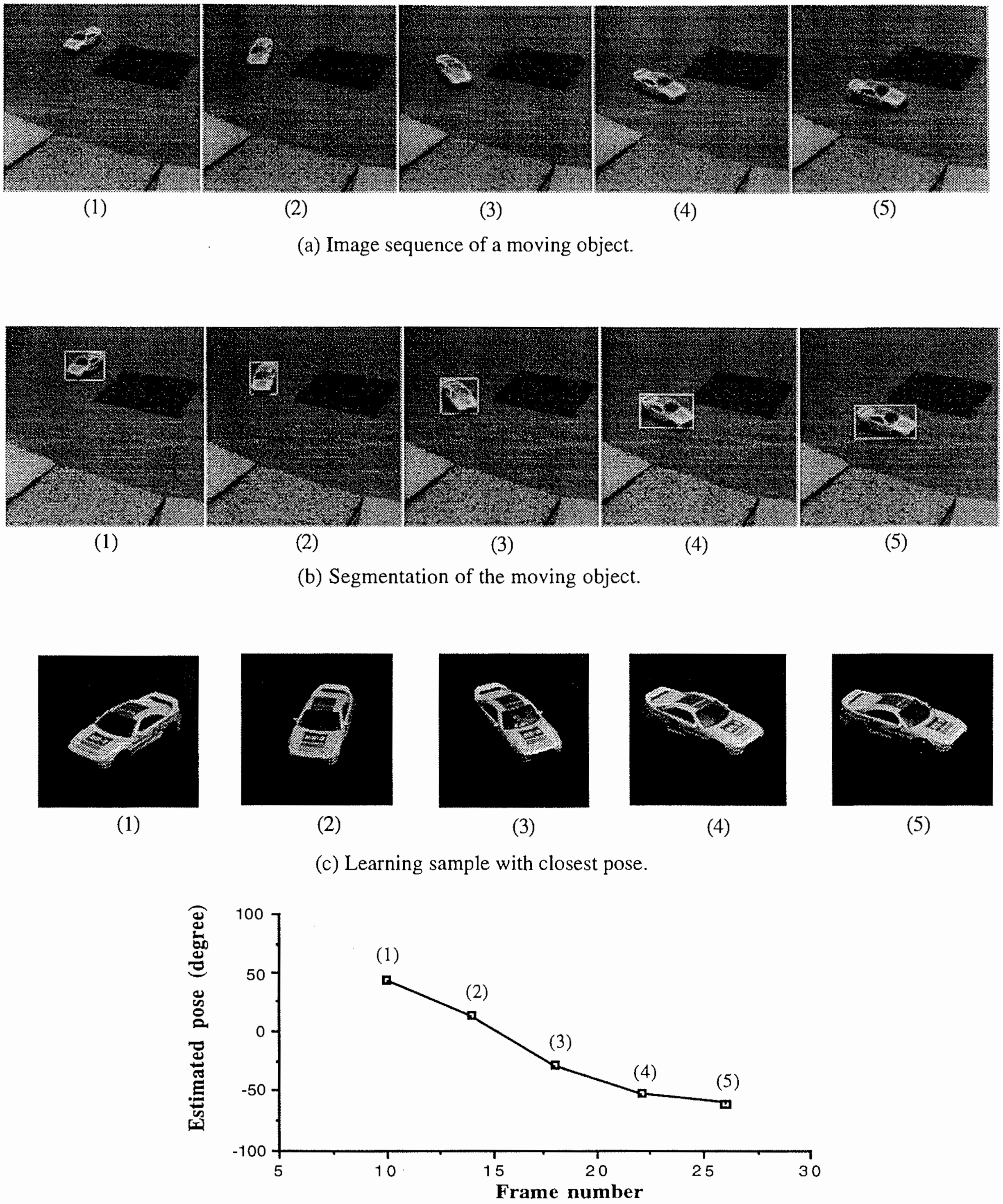
- **Segmentation and Occlusion:** Learning and recognition require the segmentation of the object region. In structured environments, the background can be controlled, in which case, simple thresholding is sufficient for robust segmentation. In the case of moving objects, simple background subtraction such as the one used in the moving car image sequence (Figure 7) can be effective for segmentation. In the context of general scenes, however, segmentation poses a serious problem and hence 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 appearance matching, it simply cannot handle occlusions. This is a second limitation of the parametric eigenspace approach.

- **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.

- **Parameters of the Hypersurface:** 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 light source axis generally 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, hypersurfaces with upto three parameters can be used without much of a problem. These three parameters can be selected depending on the application at hand.

- **Indexing into Eigenspace:** Presently, we are using an exhaustive search algorithm for finding the closest hypersurface point in eigenspace. When the number of objects and hypersurface parameters are small (as in our experiments), exhaustive search is sufficiently fast. If the number of objects and/or parameters are large, indexing-based schemes can be used to prune the search for the closest hypersurface point. This approach is currently being implemented.

- **Computational Issues:** 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 the learning stage, the recognition and pose estimation algorithms are simple and computationally very efficient, requiring only the projection of the input image onto the universal and object eigenspaces. Recognition and pose estimation can therefore be accomplished in real-time (frame-rate of 30 msec) using simple and inexpensive hardware. In contrast, most existing model-based recognition algorithms are too slow for practical applications.



(a) Image sequence of a moving object.

(b) Segmentation of the moving object.

(c) Learning sample with closest pose.

(d) Computed pose of the moving objetct.

Figure 7: Appearance-based recognition and pose estimation applied to the image sequence of a moving car.

- **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, visual positioning and tracking, and visual inspection. In many of these applications, factors such as segmentation and occlusion are not problems, and high-dimensional hypersurface representations are not required. For such problems, the appearance representation presented here offers a powerful solution.

6 Conclusion

In this paper, 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 representation describes object appearance. We presented 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 demonstrates the proposed approach to be very robust to factors, such as, image noise and quantization. We believe that the results presented in this paper are applicable to a variety of vision problems. This is the topic of our current investigation.

Acknowledgements

The authors would like to thank Daphna Weinshall for several useful comments on the paper.

References

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

[2] R. T. Chin and C. R. Dyer, "Model-Based Recognition in Robot Vision," *ACM Computing Surveys,* Vol. 18, No. 1, pp. March 1986.

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

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

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

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

[7] H. Murase and M. Lindenbaum, "Spatial Temporal Adaptive Method for Partial Eigenstructure Decompisition of Large Images," *NTT Technical Report No. 6527,* March 1992.

[8] H. Murase and S. K. Nayar, "Parametric Eigenspace Representation for Visual Learning and Recognition," Technial Report CUCS-054-92, Department of Computer Science, Columbia University, November, 1992.

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

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

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

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

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

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

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