6 Parametric Appearance Representation

Shree K. Nayar
Columbia University

Hiroshi Murase
NTT Basic Research Laboratory

Sameer A. Nene
Columbia University

Abstract

In contrast to the traditional approach, the recognition problem is formulated as one of matching appearance rather than shape. For any given vision task, all possible appearance variations define its visual workspace. A set of images is obtained by coarsely sampling the workspace. The image set is compressed to obtain a low-dimensional subspace, called the eigenspace, in which the visual workspace is represented as a continuous appearance manifold. Given an unknown input image, the recognition system first projects the image to eigenspace. The parameters of the vision task are recognized based on the exact position of the projection on the appearance manifold. The proposed appearance representation has several applications in visual perception. As examples, a real-time recognition system with 20 complex objects, an illumination planning technique for robust object recognition, and a real-time visual positioning and tracking system are described. The simplicity and generality of the proposed ideas have led to the development of a comprehensive software library for appearance modeling and matching.

1. Introduction

Vision research has placed significant emphasis on the development of compact and descriptive shape representations for object recognition [39, 3, 23]. This has lead to the creation of a variety of novel representations, including, generalized cylinders [4], superquadrics [2][33], extended gaussian images [10], parametric bicubic patches [23] and differential geometric representations [5], only to name a few. While these representations are all useful in specific application domains, each has been found to have its own drawbacks. This has kept researchers in search for more powerful representations.

Will shape representation suffice? After all, vision deals with brightness images that are functions not only of shape but also other intrinsic scene properties such as reflectance and perpetually varying factors such as illumination. This observation has led to the exploration of view-based approaches to object recognition (see [37][43][14][44][38] for examples). It motivates us to take an extreme approach to visual representation. What we seek is not a representation of shape but rather appearance [19], encoded in which are brightness variations caused by three-dimensional shape, surface reflectance properties, sensor parameters, and illumination conditions. Given the number of factors at work, it is immediate that an appearance representation that captures all possible variations is simply impractical. Fortunately, there exist a wide collection of vision applications where pertinent variables are few and hence compact appearance representation in a low-dimensional space is indeed possible.

An added drawback of shape representation emerges when a vision programmer attempts to develop a practical recognition system. Techniques for automatically acquiring shape models from sample objects are only being researched. For now, a vision programmer is forced to select an appropriate shape representation, design object models using the chosen 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 shapes. It is clear that recognition systems of the future must be capable of acquiring object models without human assistance. It turns out that the appearance representation proposed here is easier to acquire through an automatic learning phase than to create manually.

The appearance of an object is the combined effect of its shape, reflectance properties, pose in the scene, and the illumination conditions. While shape and reflectance are intrinsic properties that do not change for any rigid object, pose and illumination vary from one scene to the next. 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 lighting conditions. This can be accomplished using, for example, two robot manipulators; one rotates the object while the other varies the illumination direction. The result is a 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 this 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-Loéve transform [32] [9]. It uses the eigenvectors of an image set as orthogonal bases 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 are generally sufficient to capture the significant appearance characteristics of an object, as shown in [42][43]. These eigenvectors constitute the dimensions of what we refer to as the *eigenspace*. From the perspective of machine vision, the eigenspace has an attractive property. If any two images from the set are projected to the eigenspace, the distance between the corresponding points in eigenspace is the best approximation to correlation between the images.

We have proposed a continuous and compact representation of object appearance that is parametrized by the variables, namely, object pose and illumination. This representation is referred to as the *parametric eigenspace* [18][19]. We have shown that parametric eigenspaces are useful not only for object recognition but a variety of other vision tasks. In object recognition, 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 sensor magnification and illumination intensity. The eigenspace for the image set is constructed and all object images (learning samples) are projected to it to obtain a set of points. These points lie on a *manifold* that is parametrized by pose and illumination. The manifold is constructed from the discrete points by spline interpolation [19]. For the class of objects with linear reflectance models, we have analyzed the effect of illumination on the structure of the manifold [26]. In was shown that, in the case of an ideal diffuse object with arbitrary texture, three illumination directions are sufficient to construct the entire illumination manifold. This result drastically reduces the number of images required in the learning stage.

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 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 projected to eigenspace. The closest manifold reveals the identity of the object and exact position of the closest point on the manifold determines pose and illumination direction. Two different techniques have been tested for determining the closest manifold point, one is based on binary search [30] and other uses an input-output mapping network [15]. We have achieved further speed-up in recognition by developing a comprehensive theory and a novel algorithm for pattern rejection [1].

Will appearance representation suffice? Given the large number of parameters that affect appearance, it does not suggest itself as a replacement for shape representation. In fact, our experiments on recognition and robot tracking show that appearance models are in many ways complementary to shape models. Appearance representation proves extremely effective when the task variables are few; it is efficient and circumvents time-consuming and often unreliable operations such as feature detection. On the other hand, when occlusion effects are not negligible, shape models offer solutions in the form of partial matching that is more challenging in the case appearance matching [22].

Parametric appearance models have been applied to a variety of problems besides object recognition, such as, illumination planning for robust recognition [20] [21], visual positioning and tracking [25], and temporal inspection of complex parts [27]. These applications have demonstrated that the techniques underlying appearance modeling and matching are general. This has motivated us to develop a comprehensive software package [31] for appearance matching that is presently being used at several research institutions. We conclude with a brief discussion on the salient features of appearance matching and our most recent results on the topic.

2. Computing appearance models

We begin by presenting a general procedure for acquiring appearance models. In subsequent sections, this procedure is applied to a few vision problems.

2.1. The visual workspace

Each appearance model is parametrized by the variables of the vision task at hand. In the case of object recognition, these could include object pose and illumination parameters. If the objects are non-rigid, deformation parameters would serve as additional variables. In the case of visual tracking applications, the coordinates of a hand-eye system with respect to a moving object would be pertinent variables. Without loss of generality, we define the variables of a vision task as the *visual degrees of freedom* (DOF):

$$\mathbf{q} \;=\; [\,q_1,\, q_2,\, ...,\, q_m\,]^{\mathrm{T}} \tag{1}$$

where m is the total number of DOF at work. For any vector $\mathbf{q}$, the vision sensor produces an image vector:

$$\mathbf{i} \;=\; [\,i_1,\, i_2,\, ...,\, i_N\,]^{\mathrm{T}} \tag{2}$$

In a given application, $\mathbf{q}$ has lower and upper bounds and its continuous set of values within these bounds map to a continuous domain of images $\mathbf{i}(\mathbf{q})$. This range of appearances is what we refer to as the *visual workspace* of the task. Our approach is to acquire an image set by coarsely sampling the visual workspace and then produce a compact representation of the image set that can be used not only to recognize the discrete appearances in the image set but also those that lie in between the ones in the set, i.e. a continuous representation of the entire visual workspace.

To achieve scale invariance we force all images in an acquired image set to be of the same size. For instance, in a recognition task an object region is segmented from the scene and scale normalized [18] to fit a predetermined image size. This ensures that the recognition system is invariant to magnification, i.e.

the distance of the object from the image sensor. It is also desirable that appearance representation and recognition be unaffected by variations in the intensity of illumination or the aperture of the imaging system. This can be achieved by normalizing each acquired image such that the total energy contained within is unity: $\hat{\mathbf{i}}_j = \mathbf{i}_j / \parallel \mathbf{i}_j \parallel$.

Let the number of discrete samples obtained for each degree of freedom q_l be R_l. Then the total number of images is $M = \Pi_{l=1}^{m} R_l$. The complete image set

$$\{ \hat{\mathbf{i}}_1, ..., \hat{\mathbf{i}}_2, ..., \hat{\mathbf{i}}_M \} \tag{3}$$

can be a uniform or non-uniform sampling of the visual workspace.

Note that the above vectors, $\hat{\mathbf{i}}_j$, represent unprocessed brightness images (barring the scale and brightness normalizations). Alternatively, processed images such as smoothed images, first derivatives, second derivatives, Laplacian, or even the power spectrum of each image may be used instead. In applications that employ depth sensors, the images could be range maps. The image type is selected based on its ability to capture distinct appearance characteristics of the task workspace. Here, for the purpose of description we use raw brightness images, bearing in mind that appearance models can in principle be constructed for any image type.

2.2. Computing eigenspaces

Images in the set tend to be correlated to a large degree since visual displacements between consecutive images are small. The obvious step is to take advantage of this and compress the large set to a low-dimensional representation that captures the key appearance characteristics of the visual workspace. A suitable compression technique is based on principal component analysis [32], where the eigenvectors of the image set are computed and used as orthogonal bases for representing individual images. Principal component analysis has been previously used in computer vision for deriving basis functions for feature detection [12] [13], representing human face images [42], and recognizing face images [43] [34]. Though, in general, all the eigenvectors of an image set are required for perfect reconstruction of any particular image, only a few are sufficient for visual recognition. These eigenvectors constitute the dimensions of the *eigenspace*, or image subspace, in which the visual workspace is compactly represented.

First, the average $\mathbf{c}$ of all images in the set is subtracted from each image. This would ensure that the eigenvector with the largest eigenvalue represents the subspace dimension in which the variance of images is maximum in the correlation sense. In other words, it is the most important dimension of the eigenspace. An image matrix is constructed by subtracting $\mathbf{c}$ from each image and stacking the resulting vectors column-wise:

$$\mathbf{P} \stackrel{\triangle}{=} \left\{ \hat{\mathbf{i}}_1 - \mathbf{c},\ \hat{\mathbf{i}}_2 - \mathbf{c},\ ......,\ \hat{\mathbf{i}}_M - \mathbf{c} \right\} \tag{4}$$

$\mathbf{P}$ is $N \times M$, where N is the number of pixels in each image and M is the total number of images in the set. To compute eigenvectors of the image set we define the *covariance matrix*:

$$\mathbf{Q} \stackrel{\triangle}{=} \mathbf{P}\,\mathbf{P}^{\mathrm{T}} \tag{5}$$

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

$$\lambda_k\,\mathbf{e}_k \;=\; \mathbf{Q}\,\mathbf{e}_k \tag{6}$$

Calculation of the eigenvectors of a matrix as large as $\mathbf{Q}$ is computationally intensive. Fast algorithms for solving this problem have been a topic of active research in the area of image coding/compression and pattern recognition. A few of the representative algorithms are summarized in Appendix A. In some of our systems we have used a fast implementation [31] of the algorithm proposed by Murakami and Kumar [16] and in others the STA algorithm of Murase and Lindenbaum [17]. On a Sun IPX workstation, for instance, 20 eigenvectors of a set of 100 images (each 128x128 in size) can be computed in about 3 minutes, and 20 eigenvectors of a 1000 image set in less than 4 hours. Workstations are fast gaining in performance and these numbers are expected to diminish quickly.

The result of eigenstructure decomposition is a set of eigenvalues $\{\,\lambda_k \mid k = 1, 2, ..., K\,\}$ where $\{\,\lambda_1 \geq \lambda_2 \geq \ldots.. \geq \lambda_K\,\}$, and a corresponding set of orthonormal eigenvectors $\{\,\mathbf{e}_k \mid k = 1, 2, ..., K\,\}$. Note that each eigenvector is of size N, i.e. the size of an image. These K eigenvectors constitute our eigenspace; it is an approximation to a complete Hilbert space with N dimensions. A variety of criteria have been suggested for selecting K for any given image set [32]. In most of our applications, we have found eigenspaces of 20 or less dimensions to be more than adequate.

2.3. Parametric eigenspace representation

Each workspace sample $\hat{\mathbf{i}}_j$ in the image set is projected to eigenspace by first subtracting the average image $\mathbf{c}$ from it and finding the inner product of the result with each of the K eigenvectors. The result is a point $\mathbf{f}_j$ in eigenspace:

$$\mathbf{f}_j \;=\; [\,\mathbf{e}_1, \mathbf{e}_2, \ldots.., \mathbf{e}_K\,]^{\mathrm{T}}\;(\,\hat{\mathbf{i}}_j \,-\, \mathbf{c}\,) \tag{7}$$

By projecting all images in this manner, a set of discrete points is obtained. Since consecutive images are strongly correlated, their projections are close to one another. Hence, the discrete points obtained by projecting all the discrete samples of the workspace can be assumed to lie on a manifold that represents a *continuous* appearance function. The discrete points are interpolated to obtain this manifold. In our implementation, we have used a standard quadratic B-spline interpolation algorithm [41]. The resulting manifold can be expressed as:

$$\mathbf{f}\,(\mathbf{q}) \;=\; \mathbf{f}\,(\,q_1, q_2, \ldots\ldots., q_m\,) \tag{8}$$

It resides in a low-dimensional space and therefore is a compact representation of appearance as a function of the task DOF $\mathbf{q}$. The exact number of task DOF is of course application dependent. It is worth pointing out that multiple visual workspaces (for instance, multiple objects in a recognition task) can be represented in the same eigenspace as set of manifolds $F=\{\mathbf{f}^1, \mathbf{f}^2, ..., \mathbf{f}^P\}$. In this case, the eigenspace is computed using image sets of all the visual workspaces. The above representation is called the *parametric eigenspace*.

2.4. Correlation and distance in eigenspace

Before we proceed to describe appearance recognition, it is worthwhile to discuss some relevant properties of the eigenspace representation. Consider two images $\hat{\mathbf{i}}_m$ and $\hat{\mathbf{i}}_n$ that belong to the image set used to compute an eigenspace. Let the points $\mathbf{f}_m$ and $\mathbf{f}_n$ be the eigenspace projections of the two images. It is well-known in pattern recognition theory [32] that each of the images can be expressed in terms of its projection:

$$\hat{\mathbf{i}}_m = \sum_{i=1}^{N} f_{m\,i}\,\mathbf{e}_i + \mathbf{c} \qquad (9)$$

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

$$\hat{\mathbf{i}}_m \approx \sum_{i=1}^{K} f_{m\,i}\,\mathbf{e}_i + \mathbf{c} \qquad (10)$$

As a result of the brightness normalization described in section 2.1, $\hat{\mathbf{i}}_m$ and $\hat{\mathbf{i}}_n$ 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 is related to correlation $\hat{\mathbf{i}}_m{}^{\mathrm{T}}\,\hat{\mathbf{i}}_n$ between the images as:

$$\begin{aligned} \|\, \hat{\mathbf{i}}_m - \hat{\mathbf{i}}_n \,\|^2 &= (\,\hat{\mathbf{i}}_m - \hat{\mathbf{i}}_n\,)^{\mathrm{T}}\,(\,\hat{\mathbf{i}}_m - \hat{\mathbf{i}}_n\,) \\ &= 2 - 2\,\hat{\mathbf{i}}_m{}^{\mathrm{T}}\,\hat{\mathbf{i}}_n \end{aligned} \qquad (11)$$

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 $\mathbf{f}_m$ and $\mathbf{f}_n$ using (10):

$$\| \; \hat{\mathbf{i}}_m \; - \; \hat{\mathbf{i}}_n \; \|^2 \approx \| \; \sum_{i=1}^{K} f_{m\,i} \, \mathbf{e}_i \; - \; \sum_{i=1}^{K} f_{n\,i} \, \mathbf{e}_i \; \|^2 \tag{12}$$

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

$$\begin{aligned} \| \; \sum_{i=1}^{K} f_{m\,i} \, \mathbf{e}_i \; - \; \sum_{i=1}^{K} f_{n\,i} \, \mathbf{e}_i \; \|^2 \;\; &= \;\; \| \; \sum_{i=1}^{K} (\, f_{m\,i} \; - \; f_{n\,i} \,) \, \mathbf{e}_i \; \|^2 \\ &= \;\; \| \; \mathbf{f}_m \; - \; \mathbf{f}_n \; \|^2 \end{aligned} \tag{13}$$

The last simplification results from the eigenvectors being orthonormal; ${\mathbf{e}_i}^{\mathrm{T}} \, \mathbf{e}_j \; = \; 1$ when $i \; = \; j$, and 0 otherwise. From (12) and (13), we get:

$$\| \; \hat{\mathbf{i}}_m \; - \; \hat{\mathbf{i}}_n \; \|^2 \approx \| \; \mathbf{f}_m \; - \; \mathbf{f}_n \; \|^2 \tag{14}$$

The above relation implies that the square of the Euclidean distance between points $\mathbf{f}_m$ and $\mathbf{f}_n$ is an approximation to the SSD between images $\hat{\mathbf{i}}_m$ and $\hat{\mathbf{i}}_n$. 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 computational vision, where, correlation is frequently used as a measure of similarity between images.

3. Image recognition

Our goal here is to develop an efficient method for recognizing an unknown input image $\hat{\mathbf{i}}_c$. A brute force solution would be to compare the input image with all images corresponding to discrete workspace samples. Such an approach is equivalent to exhaustive template matching. Clearly, this is impractical from a computational perspective given the large number of images we are dealing with. Further, the input image $\hat{\mathbf{i}}_c$ may not correspond exactly to any one of the images obtained by sampling the visual workspace; $\hat{\mathbf{i}}_c$ may lie in between discrete samples.

The parametric eigenspace representation enables us to accomplish image matching in a very efficient manner. Since the eigenspace is optimal for computing correlation between images, we can project the current image to eigenspace and simply look for closest point on the appearance manifold. Image recognition proceeds as follows. We will assume that $\hat{\mathbf{i}}_c$ has already been normalized in scale and brightness to suit the invariance requirements of the application. The average $\mathbf{c}$ of the visual workspace is subtracted from $\hat{\mathbf{i}}_c$ and the resulting vector is projected to eigenspace to obtain the point:

$$\mathbf{f}_c \;\; = \;\; [\, \mathbf{e}_1, \, \mathbf{e}_2, ..., \mathbf{e}_K \,]^{\mathrm{T}} \, (\, \hat{\mathbf{i}}_c \; - \; \mathbf{c}) \tag{15}$$

The matching problem then is to find the minimum distance d_r between $\mathbf{f}_c$ and the manifold $\mathbf{f}(\mathbf{q})$:

$$d_r \;\; = \; \min_{\mathbf{q}} \; \| \; \mathbf{f}_c \; - \; \mathbf{f} \, (\mathbf{q}) \; \| \tag{16}$$

If d_r is within some pre-determined threshold value (selected based on the noise characteristics of the image sensor), we conclude that $\hat{\mathbf{i}}_c$ does belong to the appearance manifold $\mathbf{f}$. Then, parameter estimation is reduced to finding the coordinate $\mathbf{q}_c$ on the manifold corresponding to the minimum distance d_r. In practice, the manifold is stored in memory as a list of K-dimensional points obtained by densely resampling $\mathbf{f}(\mathbf{q})$. Therefore, finding the closest point to $\mathbf{f}_c$ on $\mathbf{f}(\mathbf{q})$ (or even a set of manifolds, F) is reduced to the classical nearest-neighbor problem.

4. Finding the closest manifold point

Mapping an input image to eigenspace is computationally simple. As mentioned earlier, the eigenspaces are typically less than 20 in dimensions. The projection of an input image to a 20-D space requires 20 dot products of the 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. What remains to be addressed is an efficient way of finding the closest manifold point. One approach is to use an exhaustive search algorithm. This is clearly inefficient both in memory and time; all the sampled manifold points need to be stored, and the distance of the input point with respect to each manifold point must be computed. The computational complexity is $O(\,K\,n\,)$ 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 [30]. 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 [15] uses three-layered radial basis function (RBF) networks proposed by Poggio and Girosi [36] to learn the mapping between input points and manifold parameters (object number and pose). The complexity of the network approach depends on the number of networks used and their sizes. In [15] a new framework is introduced that uses the wavelet integral transform for finding the smallest RBF network to accomplish any given input-output mapping. The performance of the network based scheme is generally comparable to that of the binary search approach. The network implicitly interpolates, or reconstructs, manifolds from the discrete eigenspace points $\mathbf{f}_j$ and therefore does not require the use of spline interpolation followed by the resampling of manifolds. This advantage however comes with a slight sacrifice in parameter estimation accuracy [15].

5. Object recognition and pose estimation

We have used appearance models for 3-D object recognition and pose estimation [18] [19]. During model acquisition, each object is placed on a computer-controlled turntable (see Fig.1) and its pose is varied about a single axis, namely, the axis

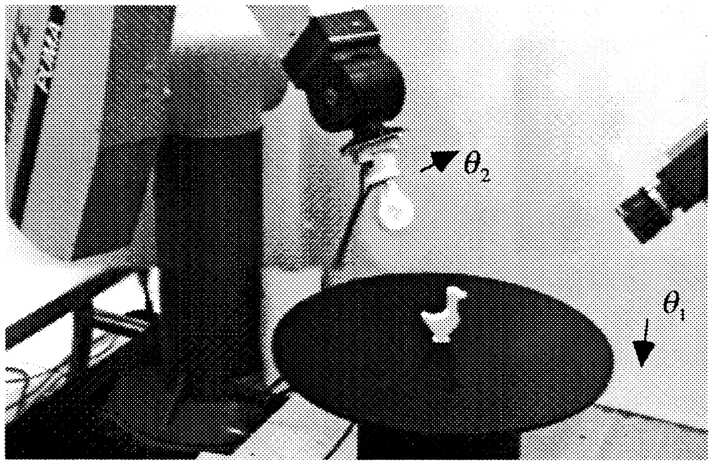
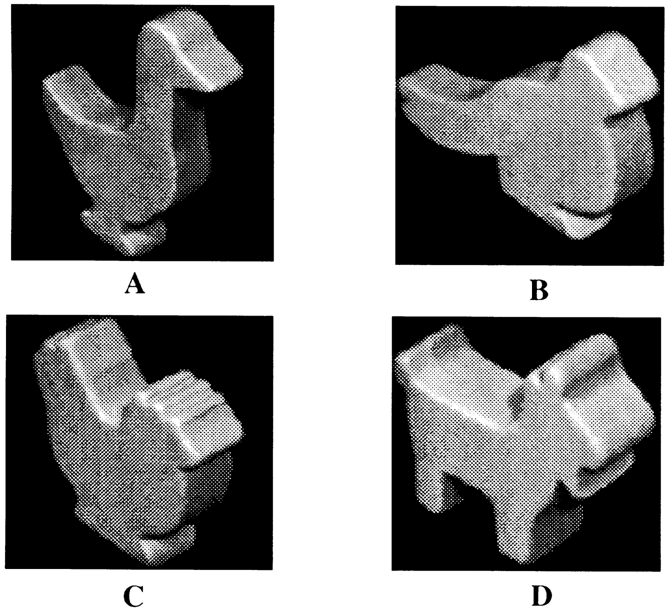


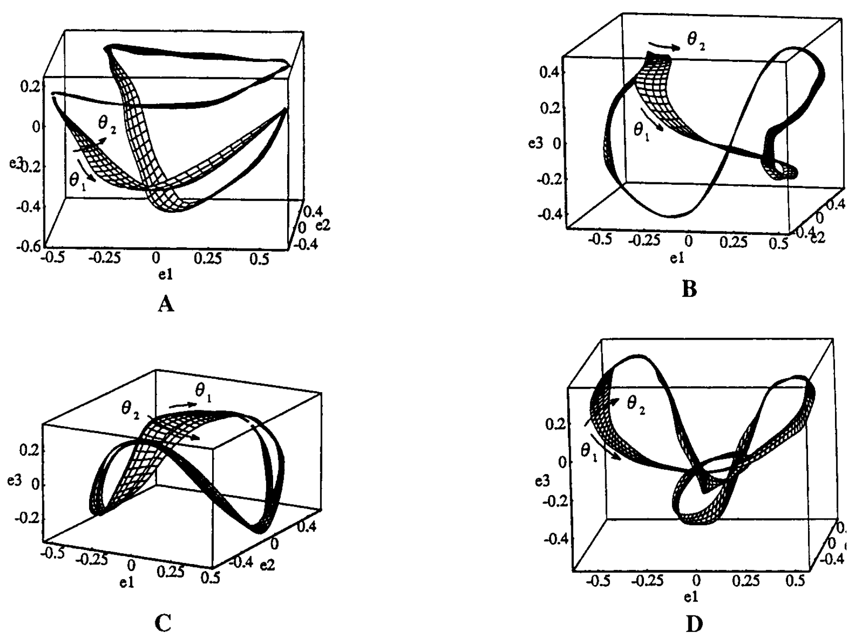
Figure 1: *Setup used to automatically acquire object appearance models for recognition and pose estimation. The object is placed on a motorized turntable.*

of rotation of the turntable. 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. The object is illuminated by the ambient lighting of the environment that is expected to remain more or less unchanged between model acquisition 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 DOF robot manipulator (see Fig.1) with a light source mounted on its end-effector. Images of the object are sensed using a 512×480 pixel CCD camera and digitized using an Analogics frame-grabber board. Fig.2 shows four toys and their respective appearance models. For each object, 90 poses and 5 source directions were used (a total of 450 images, each 128x128 pixels in size after segmentation and scale normalization). The manifolds reside in 10-D eigenspaces and are parameterized by a single pose parameter θ_1 and a single illumination direction parameter θ_2.

Several experiments were conducted to verify the accuracy of recognition and pose estimation [19]. For the four objects in Fig.2. a total of 1080 test images were used. These images were taken at object poses that lie in between the ones used to obtain the learning samples. We define *recognition rate* as the percentage of test images for which the object in the image is correctly recognized. Figs.3(a) and (b) summarize the recognition results for the four objects. Fig.3(a) illustrates the sensitivity of recognition rate to the number of eigenspace dimensions. Clearly, the discriminating power of the eigenspace is expected to increase with the number of dimensions. The recognition rate is found to be poor if less than 4 dimensions



(a)



(b)

Figure 2: *(a) Four objects and (b) their parametric appearance manifolds (from [19]). The manifolds reside in 10-D eigenspace but are displayed here in 3-D. They are parametrized by object pose (θ_1) and illumination direction (θ_2).*

are used but approaches unity as the dimensionality is increased to 10.

Fig.3(b) shows the relationship between recognition rate and the number of poses used for each object. 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 each 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, 30 poses of each object (12 degree increments of the turntable position) are sufficient to obtain recognition rates close to unity. If a smaller 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 four objects were also used to determine the accuracy of pose estimation. Since these images were taken using the controlled turntable, the actual pose in each image is known. Figs.3(c) and (d) show histograms of pose errors (in degrees) computed for the 1080 test images. In Fig.3(c), 450 learning samples (90 poses and 5 source directions) were used to compute an 8-D eigenspace. In Fig.3(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 degree. The sensitivity of recognition to image noise and segmentation error is analyzed in [21].

6. Automated real-time recognition system

Based on the above results, we implemented a recognition system with 20 objects in its database (see Fig.4). 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 set of 1440 images. The object appearance curves were constructed in a 20-D eigenspace. 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 eigenspace, and search for the closest object and pose is accomplished in less than 1 second on the Sun workstation. The robustness of this system was tested using

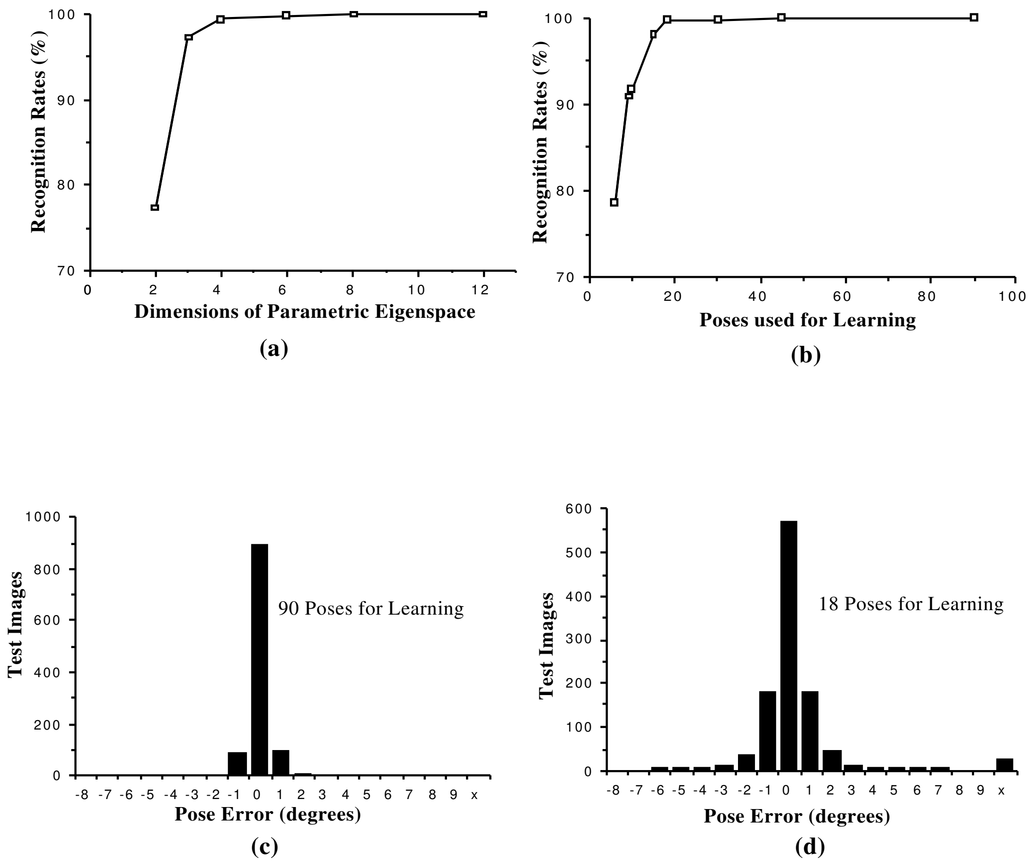


Figure 3: *Recognition and pose estimation results for the object set shown in Fig.2 (from [19]). (a) Recognition rate plotted as a function of the number of eigenspace dimensions used. (b) Recognition rate plotted as a function of the number of discrete poses of each object used in the learning stage. In both cases recognition rates were computed using 1080 test images that differ from the ones used for learning. 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 degree in the second case.*

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 Fig.4(c); the average and standard deviation of the absolute pose error were found to be 1.59 degrees and 1.53 degrees, respectively.

Recently, we have extended the capability of the above system [28]. It now includes 100 objects in its database and uses as input vectors the three bands of a color image sensor. This allows the system to distinguish between objects that are identical in shape but differ in spectral characteristics. In addition, the segmentation algorithm was modified to ensure that multiple objects (not occluding one another) can be placed in the scene and recognized simultaneously. The system is now operational and is being constantly interacted with by passers-by.

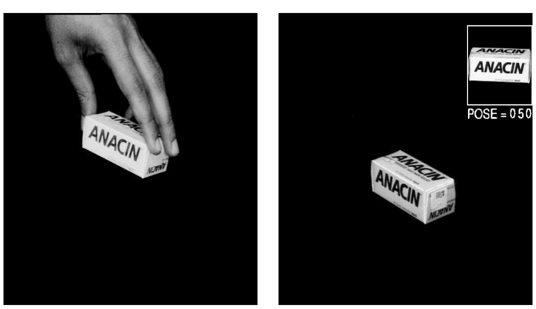
7. Structural properties of appearance manifolds

In the context of large systems, the primary bottleneck in appearance matching could turn out to be the learning stage which includes the acquisition of large image sets, the computation of eigenspaces from large covariance matrices, and the construction of parametric appearance manifolds. As described in the previous section, each object is represented as a separate manifold in eigenspace that is parametrized by pose and illumination parameters. The efficiency of the learning stage is determined by the number of sample images needed to compute an accurate appearance manifold. This brings us to the following question: What is the smallest number of images needed for constructing the appearance manifold for any given object?

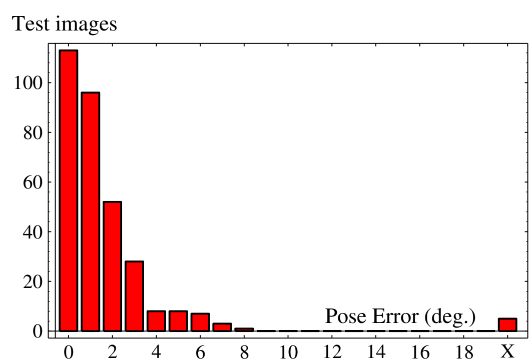
The answer lies in the structural properties of appearance manifolds. The structure of an object's manifold is closely related to its geometric and reflectance properties. In special cases, such as solids of high symmetry and solids of revolution, one can make concrete statements regarding the dimensionality of the manifold. For instance, given a fixed illumination direction and viewpoint, the manifold for a sphere of uniform reflectance is simply a point since the sphere appears the same in all its poses. This unfortunately is an extreme instance of little practical value. Under perspective projection, the relation between object shape and manifold structure is complex to say the least. A general expression that relates object pose to manifold structure would be much to hope for.

In contrast, the function space associated with object reflectance is more concise and hence conducive to analysis. It is possible to establish, under certain reflectance assumptions, a closed-form relationship between illumination parameters and manifold structure [26]. Given that the eigenspaces we use are linear subspaces, the class of linear reflectance functions [35][40] is of particular interest to us. It turns out that for this reflectance class the structure of the illumination manifold is completely determined from a small number of samples of the manifold. In particular, for Lambertian surfaces of arbitrary texture, the en-

(a) Object set



(b) Real-time recognition



(c) Pose estimation accuracy

Figure 4: *A real-time recognition system with 20 objects in the database [19]. A complete recognition and pose estimation cycle takes less than 1 second on a Sun IPX workstation without the use of any customized hardware.*

tire illumination manifold can be constructed from just three images taken using known illuminants. Alternatively, the dimensionality of the illumination manifold is exactly 3. This result is supported by a detailed empirical investigation reported recently in [8]. In [26] we use the above bound on the manifold dimensionality to show that novel images of the object can be recognized from just three projections on the illumination manifold without the explicit construction of the manifold. In addition, the validity of the above results for illumination by multiple sources and in the presence of interreflections caused by concave surfaces is demonstrated. This last property results from the fact that a concave Lambertian surface with all its interreflections behaves exactly like another Lambertian surface without interreflections but with a different set of normals and albedo values [24].

For ideal diffuse objects, these results have direct implications on the efficiency of both learning and recognition, as they dramatically reduce the number of images needed for appearance representation. These results stem from the observation that the image of a diffuse object under any illumination can be expressed as a linear combination of images taken using three independent basis illuminants. Such a linear combination does not generally exist for objects with nonlinear reflectance functions. For instance, a pure specular object would produce only strong highlights for each of the basis illuminants. The highlights produced by a novel source cannot in general be expressed as a linear combination of basis images. In fact, it is hard to envision non-trivial upper bounds on the dimensionality of a vector space containing illumination manifolds for the class of nonlinear reflectance functions.

8. Illumination planning for object recognition

In structured environments, vision systems are used to perform a variety of tasks, such as, inspect manufactured parts, recognize objects and sort them, or aid a robot in assembly operations. In each of these cases, the illumination of the environment can be selected to enhance the reliability and accuracy of the vision system. For instance, the robustness of the recognition system described in section 5 can be maximized by selecting a source direction that makes the objects of interest maximally different from each other in the correlation sense [20] [21].

Consider two objects, say p and q, from the set used to compute the eigenspace. For each light source direction l, we compute parametric curves for the two objects:

$$\mathbf{f}_l{}^{(p)}\,(\,\theta_1{}^{(p)}\,) \quad \text{and} \quad \mathbf{f}_l{}^{(q)}\,(\,\theta_1{}^{(q)}\,) \tag{17}$$

Here, the parameters $\theta_1{}^{(p)}$ and $\theta_1{}^{(q)}$ represent the poses of p and q, respectively. The shortest Euclidean distance between the two curves in eigenspace is computed as:

$$d_l{}^{(p,q)} \;=\; \min_{\theta_1{}^{(p)},\,\theta_1{}^{(q)}} \|\; \mathbf{f}_l{}^{(p)}\,(\theta_1{}^{(p)}) \;-\; \mathbf{f}_l{}^{(q)}\,(\theta_1{}^{(q)}) \;\| \tag{18}$$

The $\theta_1{}^{(p)}$ and $\theta_1{}^{(q)}$ values that produce the minimum distance $d_l{}^{(p,q)}$, correspond to poses of the two objects for which the objects appear most similar (in correlation) when illuminated by source l (see Fig.5). The illumination planning problem is formulated as follows: Find the source direction $\tilde{l}$ that maximizes the minimum distance ${d_l}^{(p,q)}$ between the object curves. This *max-min* strategy yields the safest illumination direction for the worst case poses that make the two objects appear most similar.

The above example includes only two objects. The *max-min* strategy is easily extended to a set of P objects. For a given illumination direction l, we now have P curves in eigenspace. The minimum distance ${d_l}^{(p,q)}$ is computed for all pairs of objects, resulting in P^2 minimum distances. The minimum of all these distances, say d_l, represents the worst case for the entire object set. The source direction $\tilde{l}$ that maximizes d_l is then the *optimal source direction* for the object set. Fig.5 shows eigenspace curves of two objects used in our experiments [21], for a particular illumination direction. The solid line segment illustrates the shortest distance between the two curves. If in a particular application the poses of the objects are fixed, the eigenspace representation of each object, for a given illumination, is reduced from a curve to a point. In that case, the optimal source direction maximizes the minimum distance between points in eigenspace that represent different objects. In [20], the above planning strategy was used to optimize the robustness of a recognition system similar to the one described in section 5.

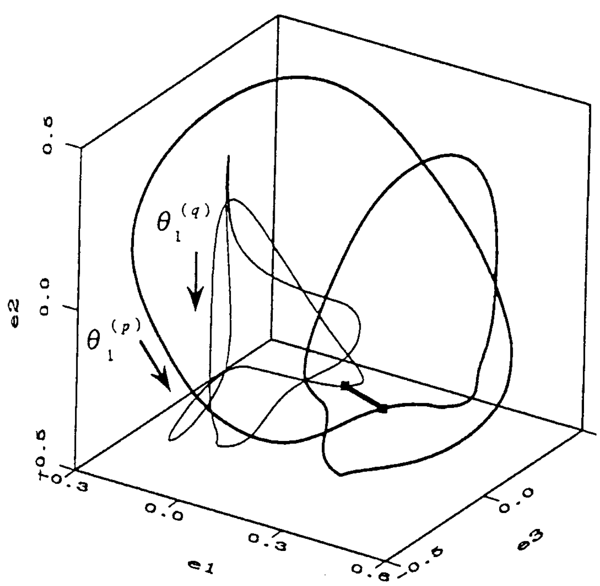


Figure 5: *Parametric eigenspace curves of two different objects obtained for a given illumination direction [21]. The shortest distance (thick line segment) between the two curves represents the worst case poses for which the objects appear most similar in the correlation sense. The optimal illumination maximizes the minimum distance between all pairs of object appearance manifolds.*

Though we have posed the planning problem as one of finding the optimal source direction, several other source characteristics such as size, distance, and spectral distribution, can be incorporated into the planning process. For instance, in [20] optimization of illumination color is described and demonstrated. The planning approach can also be used to simultaneously optimize multiple parameters. The only requirement is that these parameters be varied during the acquisition of the planning image set. Clearly, for multiple parameters, acquiring image sets, computing parametric eigenspaces, and determining the optimal parameter values can be time consuming. The planning method tends to prove impractical when more than three illumination parameters need to be jointly optimized. A small number of parameters, however, can be easily accommodated since illumination planning is typically done off-line and only once. As a result, it is generally not subject to severe time constraints.

9. Robot positioning and tracking

For a robot to be able to interact in a precise and intelligent manner with its environment, it must rely on sensory feedback. Vision serves as a powerful component of such a feedback system. It can enable a manipulator to handle task uncertainties, react to a varying environment, and gracefully recover from failures. A problem of substantial relevance to robotics is visual servoing; the ability of a robot to either automatically position itself at a desired location with respect to an object, or accurately follow an object as it moves along an unknown trajectory.

The parametric appearance representation has been used to develop an effective solution to the visual servoing problem [25]. Our implementation uses the hand-eye system shown in Fig.6. First, a sizable image window is selected that represents the appearance of the object when the robot is in the desired position. A large set of object images is then obtained by incrementally perturbing the robot's end-effector (hand-eye system) with respect to the desired position. The appearance manifold in this case represents the mapping between camera image and robot displacement, i.e. it is parametrized by the DOF of the robot end-effector.

In a positioning or tracking application, each new image is projected to eigenspace and the location of the projection on the manifold determines the robot displacement (error) with respect to the desired position. This information is relayed to the robot controller to drive it to the desired coordinates. In contrast to most previous visual servoing schemes, positioning and tracking are achieved without prior knowledge of the object's shape or reflectance, the robot's kinematic parameters, and the vision sensor's intrinsic and extrinsic parameters.

We have conducted several positioning experiments using the Adept robot and hand-eye system shown in Fig.6. Fig.7(a) shows a printed circuit board. The box shown is the image area (128x128 pixels) used for learning and positioning. Note that the image is rather complex and includes a variety of subtle features. In this experiment, robot displacements were restricted to two dimensions (x and

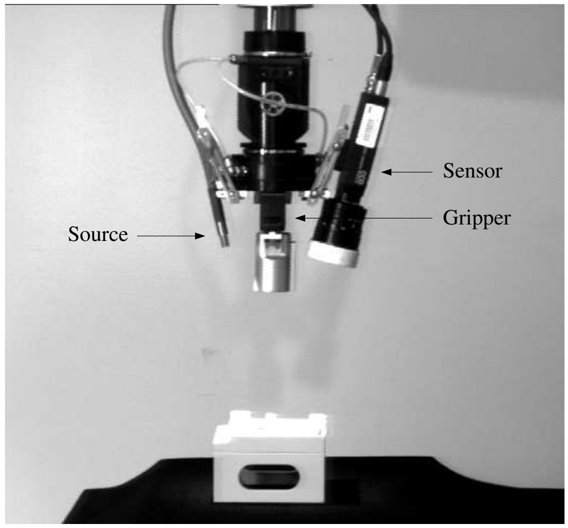


Figure 6: *The hand-eye system used for visual servoing. The end-effector includes a gripper, an image sensor, and a light source. Using the parametric appearance representation, real-time servoing is accomplished without the use of CAD models.*

y). A total of 256 images were obtained by moving the robot to 16x16 equally spaced discrete points within a 2cm x 2cm region around the desired position. A 15-D eigenspace was computed using the 256 learning images and each image was then projected to eigenspace and the resulting points were interpolated to obtain a manifold with two parameters, namely, x and y (see Fig.7(b)). The complete learning process including image acquisition, eigenspace computation, and manifold interpolation took approximately 11 minutes on a Sun IPX workstation. The parametric eigenspace is stored in memory as a set of 251x251 = 63001 points obtained by resampling the continuous manifold. A robot displacement (x,y) is stored with each manifold point.

Next, the accuracy of the positioning algorithm was tested. In these experiments, the robot was displaced by a random distance from its desired position. These random positions were uniformly distributed within the 2cm x 2cm region used for learning. Note that the random positions are generally not the same as any of the positions used while learning. The positioning algorithm was then used to estimate the robot's displacement from its desired position. This process was repeated 1000 times, each time computing the distance (error) between the robot location after positioning and the desired location. A histogram of positioning errors is shown in Fig.7(c). The average of the absolute positioning error is 0.676 mm and standard deviation is 0.693 mm. The positioning accuracy was further improved by simply using a larger number of learning images. Fig.7(d) shows the error histogram for 21x21 (441) learning images obtained within the same 2cm x 2cm displacement region. In this case, the learning process was completed in approximately 30 minutes. The average absolute error was found to be 0.151 mm and standard deviation 0.107 mm. This reflects very high positioning accuracy, sufficient for reliable insertion of a circuit chip into its holder. This task was in fact accomplished with high repeatability using the gripper of the hand-eye system.

Similar experiments were conducted for the object shown in Fig.7(e). In this case, however, three displacement parameters were used, namely, x, y, and θ (rotation in the x-y plane). During learning the x and y parameters were each varied within a $\pm$1cm range, and θ within a $\pm$10 degree range for each (x,y) displacement. A total of 11x11x11 (1331) learning images were obtained and a 5-D eigenspace computed. The parametric eigenspace representation in this case is a three-parameter manifold in 5-D space. In Fig.7(f) a projection of this manifold is shown as a surface (x and y are the parameters, while $\theta = 0$) in 3-D. Again, this reduced representation is used only for the purpose of display. The actual manifold is stored in memory as a set of 65x65x65=274625 points.

Once again, 1000 random displacements were used in the positioning experiments. The absolute Euclidean positioning errors in x-y space are illustrated by the histogram in Fig.7(g). An average absolute error of 0.291 mm and standard deviation of 0.119 mm were computed. The absolute errors for θ were computed separately and found to have a mean value of 0.56 degrees and deviation of 0.45 degrees. These results again indicate high positioning accuracy. Fig.7(h) indicates that positioning accuracy is only marginally improved for this particular object by doubling the eigenspace dimensionality. Here, 10 eigenvectors were computed to obtain a more descriptive representation of object appearance at the cost of additional memory usage. The positioning errors have a mean of 0.271 mm and deviation of 0.116 mm, and the angular errors a mean of 0.44 degrees and deviation of 0.33 degrees. This accuracy was verified by successful insertions of a peg in the hole of the object.

10. SLAM: A Software Library for Appearance Matching

As is evident from the above results, the parametric eigenspace representation can serve as the basis for solving a variety of real-world vision problems. In view of this, a software package named SLAM [31] was developed as a general tool for appearance modeling and recognition problems. The package is coded in C++ and uses advanced object-oriented programming techniques to achieve high space/time efficiency. It has four primary modules: image manipulation, subspace computation, manifold generation, and recognition. Image manipulation includes image segmentation, scale and brightness normalization, image-vector conversions, and tools for maintaining large image databases. Subspace computation, the second module, computes eigenvectors and eigenvalues of large image sets using the approach outlined in [16]. The manifold generation module can be used for projecting image (or feature) sets to subspaces, B-spline interpolation [41] of subspace projections to produce multivariate manifolds, dense resampling of man-

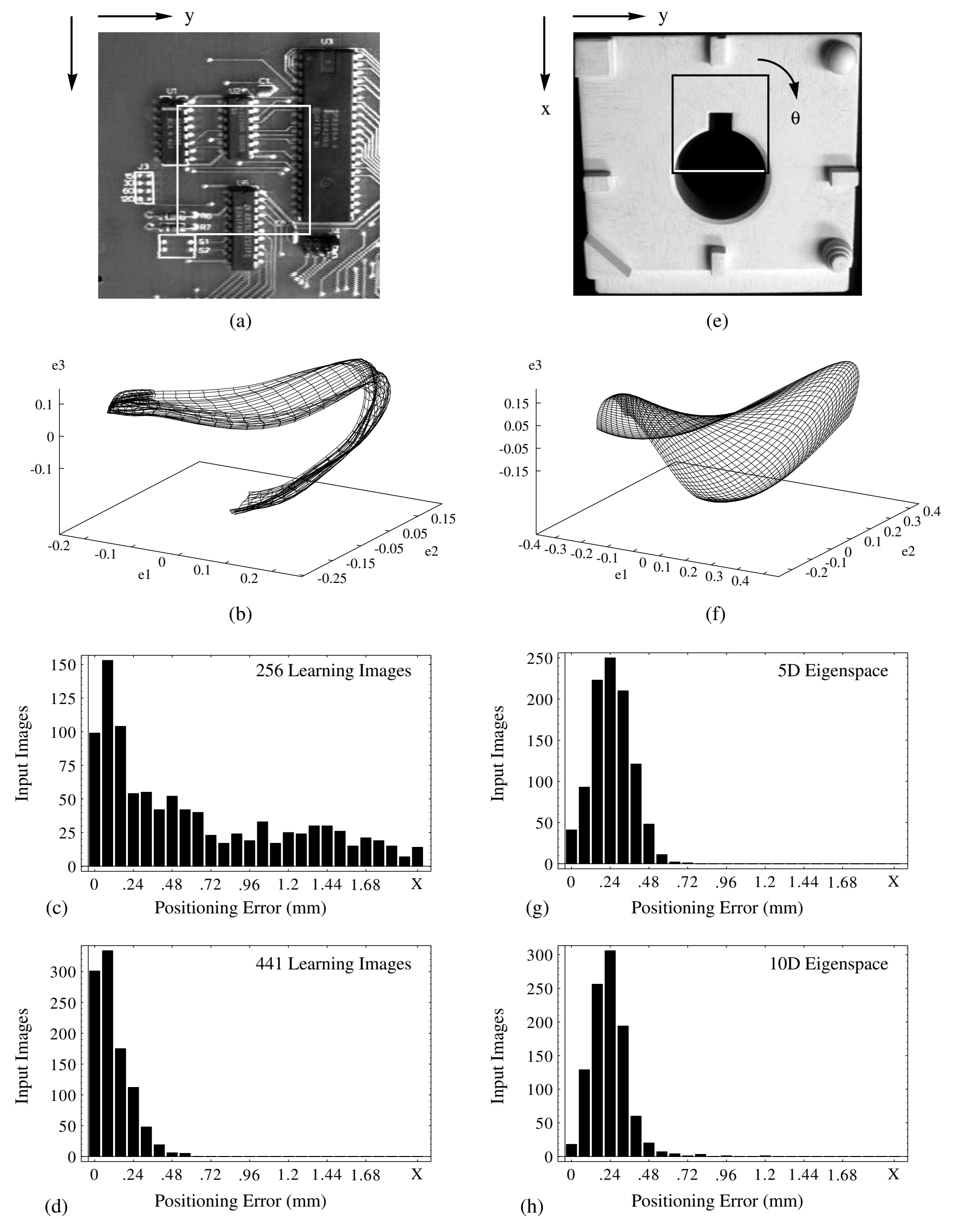


Figure 7: *Visual positioning experiments (from [25]). (a) Printed circuit board. The image window (white box) shown was used for learning and positioning. (b) Parametric appearance representation of the visual workspace displayed in 3-D. Robot displacements are in two dimensions (x and y). Histograms of absolute positioning error (in mm) for (c) 256 learning images and (d) 441 learning images. (e) Object with hole and slot. (f) Parametric appearance representation displayed in 3-D. Displacements are in three dimensions (x, y, θ). Histograms of absolute positioning error (in mm) for (g) 5-D eigenspace and (h) 10-D eigenspace.*

ifolds, and orthogonalization [11] of multiple subspaces. Finally, the recognition module includes efficient search implementations [30] that find manifold points that lie closest to novel input projections. All four modules can be accessed via an intuitive graphical interface built on X/Motif. SLAM has been licensed to several academic and industrial research institutions.

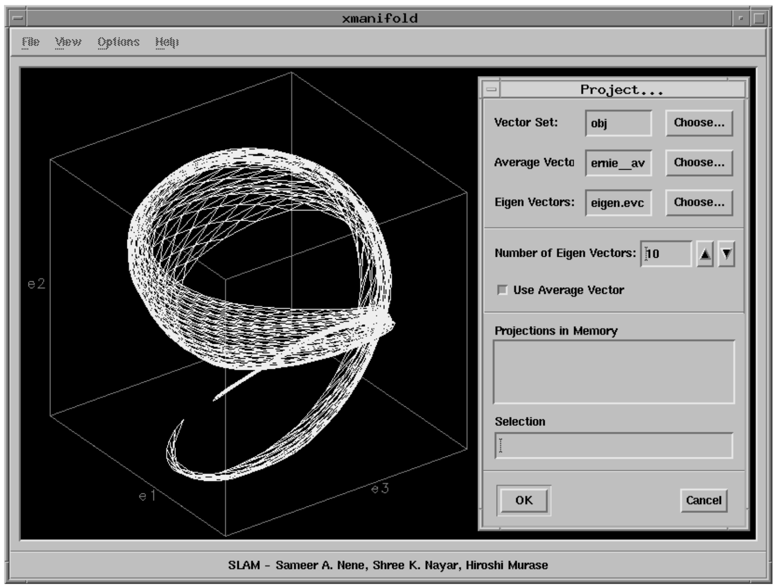


Figure 8: *The SLAM software package [31] is developed as a general tool for appearance modeling and recognition problems in vision.*

11. Discussion

In this section, we briefly discuss several issues related to the proposed learning and recognition scheme. Some of these may be viewed as merits while others as limitations that suggest research problems for the future (also see [19]).

- **Appearance Based Approach:** Both learning as well as recognition are done using just two-dimensional 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, and reliable algorithms for extracting them from images are still being actively researched. Our approach of using raw image data directly, without any significant low-level or mid-level processing, is a major advantage of the proposed approach. As stated earlier, this approach itself is not limited to brightness images and can be directly applied to processed images as well. For that matter, any sensing modality (color, infrared, range, etc.) that captures the primary visual features of a task may be used. The input vectors could even be locations and properties of features computed in images. The integration of the present scheme with previously developed geometry based recognition techniques is an interesting and open problem.

- **Shape and Reflectance:** An appealing feature of the proposed scheme is that it does not require any knowledge of the shape and reflectance properties of objects. By varying object pose and illumination (or end-effector coordinates in servoing applications), we capture the combined effect of both shape and reflectance. 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 it is appearance itself that we are representing, such phenomena need not be modeled or analyzed in isolation.

- **Segmentation and Occlusion:** We have seen that applications such as visual positioning and tracking are often not confronted with the problems of segmentation and occlusion. In such cases, it is assumed that the manipulator is close to the desired position and hence a fixed image window may be used that is more or less guaranteed to lie within the confines of the object of interest. In object recognition, learning and classification 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 can be effective for segmentation [19]. In the context of general scenes, however, segmentation poses serious problems. The method, as described here, also requires that the objects not be occluded. Since it is based on direct appearance matching, it cannot handle substantial degrees of occlusion. Segmentation and occlusion therefore present challenging research directions for appearance based recognition. Our initial investigation of this topic has resulted in a technique that performs partial matching followed by appearance voting [22].

- **Computations for Learning:** For problems that involve multiple workspaces (as in object recognition) or a large number of workspace parameters, the appearance manifolds can be expensive to compute both in time and memory requirements. It is therefore not a viable approach for the general recognition problem faced in entirely unstructured environments. However, as we have seen, it can prove as a practical approach for a variety of well-defined applications that involve a small number of parameters. Using any popular workstation, problems that involve three or less parameters can be handled with ease. Needless to say, the power and versatility of appearance matching is commensurate with the performance of the machine it is executed on. Its application domain therefore can be safely expected to broaden with time. Further, when assumptions regarding surface reflectance are feasible, we have shown that upper bounds on the dimensionality of the appearance manifold can be derived and the learning samples reduced [26].

- **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 a visual workspace is generally not as crucial as the time needed for image recognition. In contrast to learning, recognition and parameter estimation are simple and computationally very efficient, requiring only the projection of the input image to eigenspace and search for the closest manifold point. Recognition of 100 or more objects can therefore be accomplished in real-time (frame-rate of 30 Hz) using simple and inexpensive hardware [28]. In contrast, most 3-D CAD model based recognition algorithms are too slow for practical applications. The simplicity and efficiency of appearance matching makes it an attractive approach for a variety of real-world applications.

- **Efficient Pattern Rejection:** Despite the inherent efficiency of appearance matching, the present approach has complexity that at times is linear in the number of manifolds stored in the database. Recently, it was shown that the notion of *pattern rejection* [1] can be used to very quickly eliminate a large fraction of classes (manifolds) stored in the database. The result is a small set of candidates that can be viewed as a substantially reduced database for the input vector (pattern) in question. This theory of pattern rejection is applicable to not only appearance matching but in fact a large variety of well-known classification problems. It can be viewed as a complementary precursor to pattern recognition.

- **Generalized Feature Detection:** A large number of local visual features are parametric in nature, including, edges, lines, corners, and junctions. The concept of appearance matching can be used as a general framework for the design and implementation of detectors for parametrized features [29]. For robustness, the features are modeled in detail to precisely capture their appearances in the physical world. In addition, optical and sensing artifacts are incorporated to achieve realistic feature models in image domain. Each feature is then represented as a densely sampled parameterized manifold in Hilbert space. The concepts of parameter reduction by normalization, dimension reduction, pattern rejection, and efficient search are employed to achieve compact feature manifolds and efficient detection. Detectors have been implemented [29] for five specific features, namely, step edge, roof edge, line, corner, and circular disc. The tools discussed in this chapter have allowed us to generate all five of these detectors using the same procedure by simply inputing different feature models. Detailed experiments on the robustness of detection and the accuracy of parameter estimation are reported in [29].

Appendix

A. Computing eigenvectors of large image sets

Let $\mathbf{P}$ 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 $\mathbf{Q} = \mathbf{P}\,\mathbf{P}^{\mathrm{T}}$, 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 one. The first two of these algorithms are described in detail in [32].

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 [45]. 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 $F(\mathbf{e})$:

$$F(\mathbf{e}) = \frac{(\,\mathbf{e}^{\mathrm{T}}\,\mathbf{Q}\,\mathbf{e}\,)}{(\,\mathbf{e}^{\mathrm{T}}\,\mathbf{e}\,)} \tag{19}$$

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:

$$\begin{aligned} \mathbf{Q}_1 &= \mathbf{Q} \\ \mathbf{Q}_s &= \mathbf{Q}_{s-1} - \lambda_{s-1}\,\mathbf{e}_{s-1}\,\mathbf{e}_{s-1}{}^{\mathrm{T}} \end{aligned} \tag{20}$$

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, described by Murakami and Kumar [16], uses the implicit covariance matrix $\tilde{\mathbf{Q}}$, where:

$$\tilde{\mathbf{Q}} \; = \; \mathbf{P}^{\mathrm{T}} \mathbf{P} \tag{21}$$

Note that $\tilde{\mathbf{Q}}$ is an $M \times M$ matrix and therefore much smaller than $\mathbf{Q}$ when the number of images in $\mathbf{P}$ 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 [16] 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:

$$\begin{aligned} \lambda_i &= \tilde{\lambda}_i \\ \mathbf{e}_i &= \tilde{\lambda}_i^{-\frac{1}{2}} \, \mathbf{P} \, \tilde{\mathbf{e}}_i \end{aligned} \tag{22}$$

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 viable, however, when more than M eigenvectors are needed.

Spatial Temporal Adaptive:

Murase and Lindenbaum [17] have proposed 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 $\mathbf{P}$ is itself expensive. Therefore, each image in $\mathbf{P}$ is divided into "blocks" and image data in each block is compressed using the discrete cosine transform (DCT) [6]. 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 $\mathbf{P}$ 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 $\mathbf{P}$. 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 compute the eigenvectors $\mathbf{e}_i$ and eigenvalues λ_i of the original covariance matrix $\mathbf{Q}$ by applying the SVD technique (equation 22). This step also requires the use of $\mathbf{P}$ which is now compressed using the DCT. Computations are once again saved by determining $\mathbf{e}_i$ in DCT domain and then transforming it back to spatial domain using the 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.

Acknowledgements

This research was conducted at the Center for Research in Intelligent Systems, Department of Computer Science, Columbia University. It was supported in parts by the NSF National Young Investigator Award, the ARPA Grant under Contract No. DACA 76-92-C-0007, and the David and Lucile Packard Fellowship. Hiroshi Murase was supported by the NTT Basic Research Laboratory.

References

[1] S. Baker and S. K. Nayar, "A Theory of Pattern Rejection," Tech. Rep. CUCS-013-95, Dept. of Computer Science, Columbia Univ., May 1995.

[2] A. H. Barr, "Superquadric and Angle Preserving Transformations," *IEEE Computer Graphics and Applications,* Vol. 1, No. 1, pp. 11-23, Jan. 1981.

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

[4] T. O. Binford, "Generalized Cylinder Representation," *Encyclopedia of Artificial Intelligence,* S. C. Sahpiro, Ed., John Wiley & Sons, New York, pp. 321-323, 1987.

[5] M. Brady, J. Ponce, A. Yuille, and H. Asada, "Describing Surfaces," *Computer Vision, Graphics, and Image Processing,* Vol. 32, pp. 1-28, 1985.

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

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

[8] R. Epstein, P. W. Hallinan, and A. L. Yuille, "5±2 Eigenimages Suffice: An Empirical Investigation of Low-Dimensional Lighting Models," *Proc. of*

IEEE Workshop on Physics Based Modeling in Computer Vision, pp. 108-116, Boston, June 1995.

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

[10] B. K. P. Horn, "Extended Gaussian Images," *Proceedings of the IEEE,* Vol. 72, No. 12, pp. 1671-1686, Dec. 1984.

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

[12] R. A. Hummel, "Feature Detection Using Basis Functions," *Computer Graphics and Image Processing,* Vol. 9, pp. 40-55, 1979.

[13] R. Lenz, "Optimal Filters for the Detection of Linear Patterns in 2-D and Higher Dimensional Images," *Pattern Recognition,* Vol. 20, No. 2, pp. 163-172, 1987.

[14] N. K. Logothetis, J. Pauls, H. H. Bulthoff, and T. Poggio, "View-dependent object recognition by monkeys," *Current Biology,* Vol. 4, No. 5, pp. 401-414, 1994.

[15] S. Mukherjee and S. K. Nayar, "Optimal RBF Networks for Visual Learning," *Proc. of Fifth Int'l. Conf. on Computer Vision,* Boston, June 1995.

[16] H. Murakami and V. Kumar, "Efficient Calculation of Primary Images from a Set of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 4, No. 5, pp. 511-515, Sept. 1982.

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

[18] H. Murase and S. K. Nayar, "Learning Object Models from Appearance," *Proc. of AAAI,* Washington D. C., July 1993.

[19] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *International Journal of Computer Vision,* Vol. 14, No. 1, pp. 5-24, 1995.

[20] H. Murase and S. K. Nayar, "Illumination Planning for Object Recognition in Structured Environments," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* Seattle, pp. 31-38, June 1994.

[21] H. Murase and S. K. Nayar, "Illumination Planning for Object Recognition Using Parametric Eigenspaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 16, No. 12, pp. 1219-1227, Jan. 1995.

[22] H. Murase and S. K. Nayar, "Image Spotting of 3D Objects Using the Parametric Eigenspace Representation," *Proc. of 9th Scandinavian Conference on Image Analysis,* pp. 325-332, June 1995.

[23] V. S. Nalwa, *A Guided Tour of Computer Vision,* Addison Wesley, 1993.

[24] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Shape from Interreflections," *International Journal of Computer Vision,* Vol. 2, No. 3, pp. 173-195, 1991.

[25] S. K. Nayar, H. Murase, and S. A. Nene, "Learning, Positioning, and Tracking Visual Appearance," *Proc. of IEEE Int'l. Conf. on Robotics and Automation,* San Diego, May 1994.

[26] S. K. Nayar and H. Murase, "On the Dimensionality of Illumination in Eigenspace," Tech. Rep. CUCS-021-94, Dept. of Computer Science, Columbia Univ., Aug. 1994. Revised Sept. 1995.

[27] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace Methods for Robot Vision," *IEEE Trans. on Robotics and Automation,* Special issue on "Vision-Based Control of Robot Manipulators," to appear in 1996. Also Tech. Rep. CUCS-06-95, Dept. of Computer Science, Columbia Univ., Mar. 1995.

[28] S. K. Nayar, S. A. Nene, and H. Murase, "Real-Time 100 Object Recognition System," Tech. Rep. CUCS-021-95, Dept. of Computer Science, Columbia Univ., Sept. 1995.

[29] S. K. Nayar, S. Baker, and H. Murase, "Parametric Feature Detection," Tech. Rep. CUCS-028-95, Dept. of Computer Science, Columbia Univ., Oct. 1995.

[30] S. A. Nene and S. K. Nayar, "Algorithm and Architecture for High Dimensional Search," Tech. Rep. CUCS-030-95, Dept. of Computer Science, Columbia Univ., Oct. 1995.

[31] S. A. Nene, S. K. Nayar, H. Murase, "SLAM: A Software Library for Appearance Matching," *Proc. of ARPA IU Workshop,* Monterey, Nov. 1994.

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

[33] A. P. Pentland, "Perceptual Organization and the Representation of Natural Form," *Artificial Intelligence,* Vol. 28, pp. 293-331, 1986.

[34] A. Pentland, B. Moghaddam, and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* Seattle, June 1994.

[35] A. P. Petrov, "Color and Grassman-Cayley coordinates of shape," in *Human Vision, Visual Processing and Digital Display II,* SPIE Proc., Vol. 1453, pp. 342-352, 1991.

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

[37] T. Poggio and S. Edelman, "A network that learns to recognize 3D objects," *Nature*, Vol. 343, pp. 263–266, 1990.

[38] A. R. Pope and D. G. Lowe, "Learning Object Recognition Models from Images," *Proc. of Fourth Int'l. Conf. on Computer Vision,* pp. 296-301, Berlin, May 1993.

[39] A. A. G. Requicha, "Representation of Rigid Solids: Theory, Methods and Systems," *Computing Surveys,* Vol. 12, No. 4, pp. 1-437-464, Dec. 1980.

[40] A. Shashua, "On Photometric Issues in 3D Visual Recognition from a Single 2D Image," Tech. Rep., Artificial Intelligence Lab., MIT, 1993.

[41] D. F. Rogers, *Mathematical Elements for Computer Graphics,* 2nd ed., McGraw-Hill, New York, 1990.

[42] 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.

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

[44] J. Weng, N. Ahuja, and T. S. Huang, "Learning recognition and segmentation of 3-d objects from 2-d images," *Proc. of Fourth Int'l Conf. on Computer Vision*, pp. 121–128, Berlin, May 1993.

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