Parametric Appearance Representation*

Shree K. Nayar and Hiroshi Murase
Department of Computer Science
Columbia University, New York, N.Y. 10027

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 an 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. Appearance representation has numerous applications in visual perception. As examples, a real-time recognition system with 20 complex objects and a real-time visual 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 laid significant emphasis on the development of compact and descriptive shape representations for object recognition [Requicha 80, Besl and Jain 85, Nalwa 93]. This has lead to the creation of a variety of novel representations, including, generalized cylinders [Binford 87], superquadrics [Barr 81][Pentland 86], extended gaussian images [Horn 84], parametric bicubic patches [Nalwa 93] and differential geometric representations [Brady et al. 85], 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 motivated us to take an extreme approach to visual representation. What we seek is not a representation of shape but rather appearance [Murase and Nayar 92], 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 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.

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,

*Hiroshi Murase is with the NTT Basic Research Laboratory, Musashino-Shi, Tokyo 180, Japan. This work was supported by ARPA contract DACA-76-92-C-007, an NSF National Investigator Award, and a David and Lucile Packard Fellowship.

when occlusion effects are not negligible shape models offer solutions in the form of partial matching that appearance representation does not easily lend itself to.

Parametric appearance representations have been applied to a variety of problems, including, object model acquisition [Murase and Nayar 93], efficient object recognition [Murase and Nayar 92], illumination planning for robust object recognition [Murase and Nayar 94], visual positioning and tracking [Nayar et al. 94], and temporal inspection of complex parts. Here, we summarize a few of these applications. The results demonstrate that the techniques underlying appearance modeling and matching are general. This has led to the development of a comprehensive software package [Nene et al. 94] for appearance matching that is presently being used by several research institutions.

2 Computing Appearance Models

We begin by presenting a procedure for acquiring appearance models. In subsequent sections, this procedure is applied to 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):

$$\mathbf{q} = [q_1, q_2, ..., q_m]^{\mathrm{T}} \qquad (1)$$

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

$$\mathbf{i} = [i_1, i_2, ..., i_N]^{\mathrm{T}} \qquad (2)$$

In any 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 set to be of the same size. For instance, in a recognition task an object region is segmented from the scene and scale normalized [Murase and Nayar 93] 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: $\hat{\mathbf{i}}_j = \mathbf{i}_j / \| \mathbf{i}_j \|$.

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:

$$\{ \hat{\mathbf{i}}_1, ..., \hat{\mathbf{i}}_2, ..., \hat{\mathbf{i}}_M \} \qquad (3)$$

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

Note that the above image vectors $\hat{\mathbf{i}}_j$ represent unprocessed brightness image (barring the normalizations). Alternatively, processed images such as smoothed images, first derivatives, second derivatives, Laplacian, or 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 other 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 [Oja 83], 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 [Hummel 79] [Lenz 87], representing human face images [Sirovich and Kirby 87], and recognizing face images [Turk and Pentland 91]. 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 ensures 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 columnwise:

$$\mathbf{P} \triangleq \left\{ \hat{\mathbf{i}}_1 - \mathbf{c},\ \hat{\mathbf{i}}_2 - \mathbf{c},\ \ldots\ldots,\ \hat{\mathbf{i}}_M - \mathbf{c} \right\} \tag{4}$$

$\mathbf{P}$ 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*:

$$\mathbf{Q} \triangleq \mathbf{P}\,\mathbf{P}^{\mathrm{T}} \tag{5}$$

$\mathbf{Q}$ 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 (see [Oja 83]). We have used a fast implementation [Murase and Nayar 92] of the algorithm proposed by Murakami and Kumar [Murakami and Kumar 82]. On a Sun IPX workstation this implementation enables us to compute, for instance, 20 eigenvectors of a set of 100 images (each 128x128 in size) 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 $\{\lambda_1 \geq \lambda_2 \geq \ldots\ldots \geq \lambda_K\}$, 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. In all 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:

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

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 [Rogers 90]. The resulting manifold can be expressed as:

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

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 $F = \{\mathbf{f}^1, \mathbf{f}^2, \ldots\ldots, \mathbf{f}^P\}$. In this case, the eigenspace is computed using image sets of all the workspaces.

The above representation is called the *parametric eigenspace*. It posses an important property. Consider two images $\hat{\mathbf{i}}_r$ and $\hat{\mathbf{i}}_s$ that belong to the image set used to compute an eigenspace. Let points $\mathbf{f}_r$ and $\mathbf{f}_s$ be eigenspace projections of the two images. It is well-known in pattern recognition theory [Oja 83] [Murase and Nayar 92] that the distance between the two points in eigenspace is an approximation to the correlation between the two images:

$$\|\, \hat{\mathbf{i}}_r - \hat{\mathbf{i}}_s \,\|^2 \approx \|\, \mathbf{f}_r - \mathbf{f}_s \,\|^2 \tag{9}$$

The closer the projections, the more similar are the images in l^2. The eigenspace is an optimal subspace for computing correlation between images. This motivates us to develop an appearance representation based on principal component analysis.

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 workspace samples.

The parametric eigenspace representation enables us to accomplish image matching in a very efficient manner. Since the eigenspace is optimal for computing the 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:

$$\mathbf{f}_c = [\mathbf{e}_1, \mathbf{e}_2, \ldots\ldots, \mathbf{e}_K]^{\mathrm{T}} (\hat{\mathbf{i}}_c - \mathbf{c}) \qquad (10)$$

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

$$d = \min_{\mathbf{q}} \| \mathbf{f}_c - \mathbf{f}(\mathbf{q}) \| \qquad (11)$$

If d is within some pre-determined threshold value (selected based on the noise characteristics of the image sensor), we conclude that the $\hat{\mathbf{i}}_c$ 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. In practice, the manifold is stored in memory as a list of K-dimensional points obtained by densely resampling $\mathbf{f}(\mathbf{q})$. The closest point to $\mathbf{f}_c$ on $\mathbf{f}(\mathbf{q})$ (or even a set of manifolds, F) can be determined either by exhaustive search (if the list of manifold points is small), binary search, or indexing. In [Nene and Nayar 93] an algorithm is developed that results in near-constant search time of approximately 20 msec on a Sun IPX workstation. Alternatively, $\mathbf{q}_c$ can be determined from $\mathbf{f}_c$ by training a regularization network of the type described in [Poggio and Girosi 90].

4 Real-Time Object Recognition

Appearance representation has been used for object recognition [Murase and Nayar 93] [Murase and Nayar 92]. 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 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 degree-of-freedom 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. Figure 2 shows four toy cars 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.

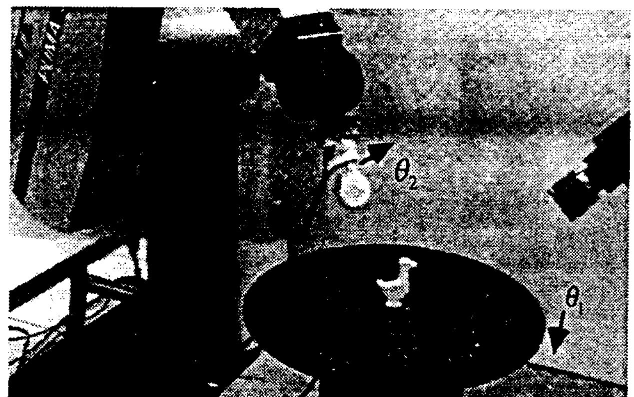
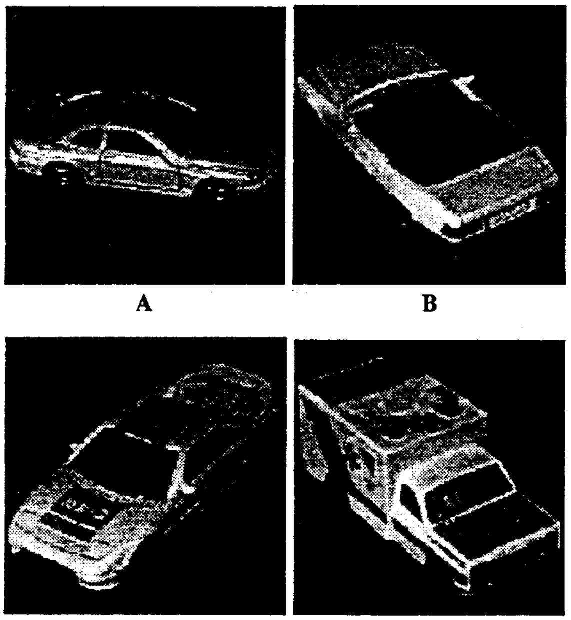


Figure 1: Setup used to automatically acquire object appearance models for recognition and pose estimation. The object is placed on a motorized turntable.

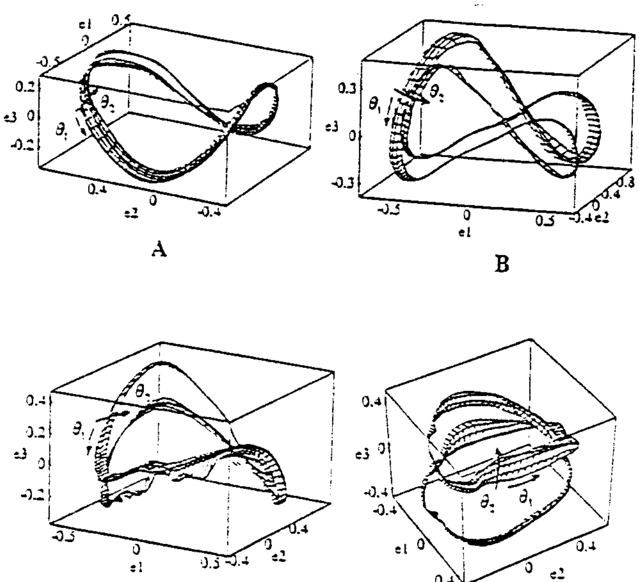
Several experiments were conducted to test recognition and pose estimation [Murase and Nayar 92]. The high accuracy, repeatability, and efficiency of recognition and pose estimation motivated us to develop a fully-automated system with 20 complex objects in its database. These objects vary from smoothly curved shapes with uniform reflectance, to fairly complex shapes with intricate textures and specularities [Murase and Nayar 92]. Developing CAD models of such objects could prove extremely cumbersome and time-consuming. Both model acquisition and recognition are done in a laboratory environment where illumination remains more or less unchanged. Each object is represented by a curve parametrized by pose in a single 20-dimensional universal eigenspace [Murase and Nayar 92].

The recognition system automatically detects significant changes in the scene, waits for the scene to stabilize, and then digitizes an image (see Figure 3). In



A B C D

(a)



A B C D

(b)

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

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 of object regions, image projection to 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 320 test images of the 20 objects taken at randomly selected but known poses. All test images were correctly identified by the system, i.e. 100% recognition rate, with an average absolute pose error of 1.59 degrees. In related work [Murase and Nayar 94], the parametric eigenspace representation was used to determine illumination conditions in a structured environment that would optimize the performance of a recognition system such as the one described above.

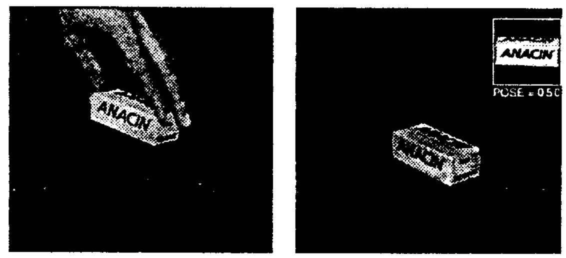


Figure 3: A real-time recognition system with 20 objects in the database [Murase and Nayar 92]. A complete recognition and pose estimation cycle takes less than 1 second on a Sun IPX workstation without the use of any customized hardware.

5 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 [Nayar et al. 94]. Our implementation uses the hand-eye system shown in Figure 4. 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.

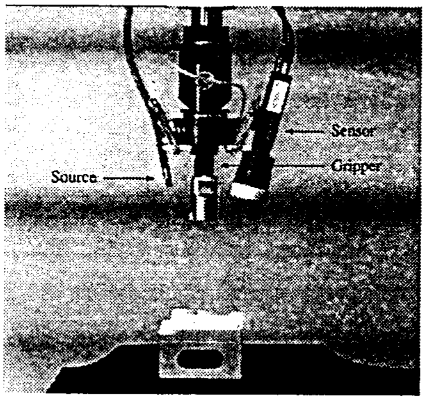


Figure 4: The hand-eye system used for visual servoing. The end-effector includes a gripper, an image sensor, and a light source. Using the parametric appearance representation real-time servoing is accomplished without the use of CAD models.

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.

Figure 5 shows results obtained for a peg-in-hole insertion task. Three displacement parameters were used, namely, x, y, and θ (rotation in the x-y plane). During image acquisition the x and y parameters were each varied within a $\pm$1cm range, and θ within a $\pm$10deg range for each (x,y) displacement. A total of 11x11x11 = 1331 images were obtained and a 5-D eigenspace computed. The appearance representation in this case is a three-parameter manifold in 5-D space. In Figure 5(b) a projection of this manifold is shown as a surface (x and y are the parameters, while $\theta = 0$) in 3-D eigenspace. 1000 random robot displacements were used to test positioning accuracy. The absolute euclidean positioning errors in x-y space are illustrated by the histogram in Figure 5(c). This high accuracy was verified by successful insertions of a peg in the object slot. In [Nayar et al. 94], these results are taken a step further. The positioning algorithm is embedded in a feedback control loop that enables the end-effector to track an object as it moves through an unknown trajectory.

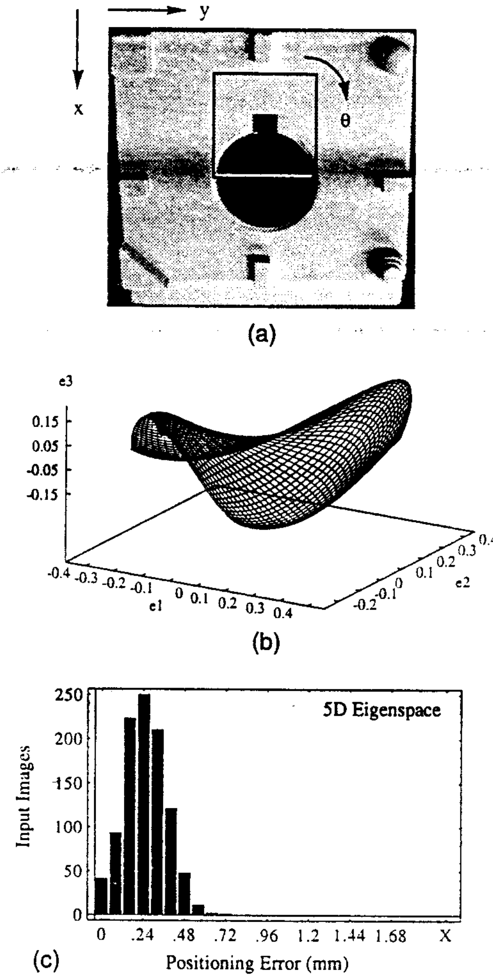


Figure 5: Visual Servoing: Peg in hole insertion. (a) Object with hole and slot. (b) Parametric eigenspace representation constructed in 5-D but displayed in 3-D. Displacements are in three dimensions (x, y, θ). (c) Histogram of absolute positioning error (in mm).

6 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 [Nene et al. 94] is developed as a general tool for appearance modeling and recognition problems in vision. 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 provides tools for maintaining large image databases. Subspace computation, the second module, computes eigenvectors and eigenvalues of large image sets using the method outlined in [Murakami and Kumar 82]. The manifold generation module can be used for projecting image (or feature) sets to subspaces, B-spline interpolation [Rogers 90] of subspace projections to produce multivariate manifolds, dense resampling of manifolds, and orthogonalization of multiple subspaces. Finally, the recognition module includes efficient search implementations [Nene and Nayar 93] that find manifold points which 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.

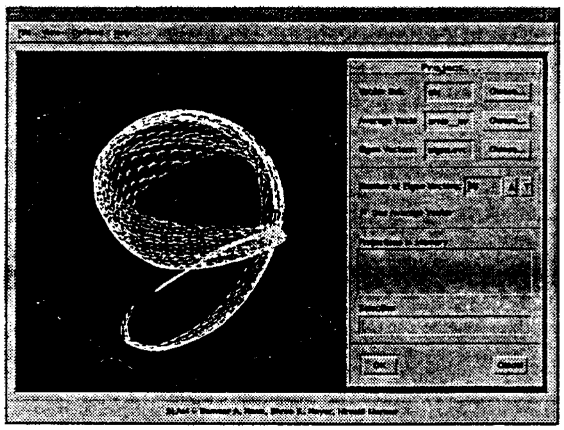


Figure 6: The SLAM software package [Nene et al. 94] is developed as a general tool for appearance modeling and recognition problems in vision.

References

[Barr 81] A. H. Barr, "Superquadric and Angle Preserving Transformations," *IEEE Computer Graphics and Applications,* Vol. 1, No. 1, pp. 11-23, Jan. 1981.

[Besl and Jain 85] P. J. Besl and R. C. Jain, "Three-Dimensional Object Recognition," *Computing Surveys,* Vol. 17, No. 1, pp. 75-145, March 1985.

[Binford 87] T. O. Binford, "Generalized Cylinder Representation," *Encyclopedia of Artificial Intelligence,* S. C. Sahpiro, Ed., John Wiley & Sons, New York, pp. 321-323.

[Brady et al. 85] M. Brady, J. Ponce, A. Yuille, and H. Asada, "Describing Surfaces," *Computer Vision, Graphics, and Image Processing,* Vol. 32, pp. 1-28, 1985.

[Horn 84] B. K. P. Horn, "Extended Gaussian Images," *Proceedings of the IEEE,* Vol. 72, No. 12, pp. 1671-1686, Dec. 1984.

[Hummel 79] R. A. Hummel, "Feature Detection Using Basis Functions," *Computer Graphics and Image Processing,* Vol. 9, pp. 40-55, 1979.

[Lenz 87] 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.

[Murakami and Kumar 82] 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.

[Murase and Nayar 93] H. Murase and S. K. Nayar, "Learning Object Models from Appearance," *Proc. of AAAI,* Washington D. C., July 1993.

[Murase and Nayar 92] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *Intl. Journal of Comp. Vision,* in press. Also Tech. Rep. CUCS-054-92.

[Murase and Nayar 94] H. Murase and S. K. Nayar, "Illumination Planning for Object Recognition in Structured Environments," *Proc. of IEEE CVPR,* Seattle, June 1994.

[Nalwa 93] V. S. Nalwa, *A Guided Tour of Computer Vision,* Addison Wesley, 1993.

[Nayar et al. 94] S. K. Nayar, H. Murase, and S. A. Nene, "Learning, Positioning, and Tracking Visual Appearance," *Proc. of IEEE Intl. Conf. on Robotics and Automation,* San Diego, May 1994.

[Nene and Nayar 93] S. A. Nene and S. K. Nayar, "Binary Search Through Multiple Dimensions," Technical Report CUCS-018-94, Dept. of Computer Science, Columbia Univ., August, 1993.

[Nene et al. 94] S. A. Nene, S. K. Nayar, H. Murase, "SLAM: A Software Library for Appearance Matching," *Proc. of ARPA IU Workshop,* Monterey, Nov. 1994. Also Tech. Rep. CUCS-019-94.

[Oja 83] E. Oja, *Subspace methods of Pattern Recognition,* Res. Studies Press, Hertfordshire, 1983.

[Pentland 86] A. P. Pentland, "Perceptual Organization and the Representation of Natural Form," *Artificial Intelligence,* Vol. 28, pp. 293-331, 1986.

[Poggio and Girosi 90] T. Poggio and F. Girosi, "Networks for Approximation and Learning," *Proc. of the IEEE,* Vol. 78, No. 9, pp. 1481-1497, September 1990.

[Requicha 80] A. A. G. Requicha, "Representation of Rigid Solids: Theory, Methods and Systems," *Computing Surveys,* Vol. 12, No. 4, pp. 1-437-464, December 1980.

[Rogers 90] D. F. Rogers, *Mathematical Elements for Computer Graphics,* 2nd ed., McGraw-Hill, New York, 1990.

[Sirovich and Kirby 87] 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.

[Turk and Pentland 91] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," *Proc. of IEEE CVPR,* pp. 586-591, June 1991.