Real-Time 100 Object Recognition System [*]

Shree K. Nayar[†], Sameer A. Nene[†], and Hiroshi Murase[‡]

[†]Department of Computer Science
Columbia University
New York, N.Y. 10027

[‡]NTT Basic Research Laboratory
Morinosato Wakamiya, Atsugi-shi
Kanagawa 243-01, Japan

Abstract

A real-time vision system is described that can recognize 100 complex three-dimensional objects. In contrast to traditional strategies that rely on object geometry and local image features, the present system is founded on the concept of appearance matching. Appearance manifolds of the 100 objects were automatically learned using a computer-controlled turntable. The entire learning process was completed in 1 day. A recognition loop has been implemented that performs scene change detection, image segmentation, region normalizations, and appearance matching, in less than 1 second. The hardware used by the recognition system includes no more than a CCD color camera and a workstation. The real-time capability and interactive nature of the system have allowed numerous observers to test its performance. To quantify performance, we have conducted controlled experiments on recognition and pose estimation. The recognition rate was found to be 100 % and object pose was estimated with a mean absolute error of 2.02 degrees and standard deviation of 1.67 degrees.

1 Introduction

The problem of 3D object recognition has been intensely studied in the past three decades. A panoply of theoretical results, algorithms and systems have resulted from this investigation. Despite this immense effort, we are yet to see a system that can recognize a large number of complex objects with robustness and efficiency. The lesson learned thus far is that the recognition of 3D objects in entirely unstructured environments is a hard problem. What can be accomplished in a somewhat structured setting where the problems of segmentation and occlusion are not as severe? We present in this paper a real-time system that recognizes any of 100 complex objects in less than 1 second using no more than a color camera and a workstation.

The versatility of a recognition system is determined to a large extent by the power of the underlying representations used. Vision research has placed significant emphasis on the development of compact and descriptive shape representations [24, 2, 15]. This has led to the creation of a variety of novel representations, including, generalized cylinders [3], superquadrics [1][22], extended gaussian images [6], parametric bicubic patches [15] and differential geometric representations [4], only to name a few. While these representations are all useful in specific application domains, each has been found to have its 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 extrinsic factors such as illumination. This observation has motivated us to take an extreme approach to visual representation. What we advocate is not a representation of shape but rather appearance [12], 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 exists a wide range 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

[*]This research was conducted at the Center for Research in Intelligent Systems, Department of Computer Science, Columbia University and at the NTT Basic Research Laboratory, Atsugi, Japan. It was supported by an NSF National Investigator Award, by ARPA Contract DACA-76-92-C-007 and the NTT Basic Research Laboratory.

easier to acquire through an automatic learning phase than to create manually. The 100 objects in our system's database were automatically learned in less than 1 day. Each object is represented in a low-dimensional subspace as a continuous appearance manifold that is parametrized by object pose.

Given an image consisting of objects of interest, we assume that the objects are not occluded and can be segmented from the remaining scene. Each 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 the appearance subspace. The closest manifold reveals the identity of the object and exact position of the closest point on the manifold determines its pose in the scene. Two efficient schemes have been tested for determining the closest manifold point, one is based on binary search [19] and other uses an input-output mapping network [9].

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 here and elsewhere 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 are less efficient in the case of appearance matching [11].

We begin with a brief overview of appearance matching and its use for color object recognition. The algorithm we have developed for searching for the closest manifold point in a high-dimensional subspace is described and its complexity discussed. Next, we detail the structure of our recognition loop which is fully automated and enables a user to interact with the system in the laboratory. Finally, we conclude with experiments that demonstrate the recognition rate and pose estimation accuracy of our system.

2 Appearance Matching: Overview

Before we describe the recognition system, we briefly review the notion of parametric appearance matching as introduced in [12]. 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. The visual learning problem is viewed 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-Loeve transform [21] [5] [10]. 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 for human faces in [25][27] and for edges and lines in [7][8]. 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.

The system described here uses the parametric eigenspace representation presented in [14][12]. We assume that the illumination conditions remain more or less constant and hence object pose is the only variable of interest. An analysis of the effect of illumination on the parametric eigenspace can be found in [17]. An image set is obtained for each of the 100 objects by varying pose in small increments. Each image is normalized in brightness and scale to achieve invariance to sensor magnification and illumination intensity. Images of all objects (learning samples) are used together to construct an eigenspace. The images of each object are then projected to eigenspace to obtain a set of points. These points lie on a manifold that is parametrized by pose. The manifold is constructed from the discrete points by spline interpolation [12]. Each object is stored in the database as a collection of eigenspace points obtained by densely resampling its appearance manifold. Recognition and pose estimation of a novel image is achieved by projecting a novel object image to eigenspace and finding the closest manifold (object identity) and the closest point on this manifold (pose).

3 Learning 100 Colored Objects

We use a color sensor to enhance the discriminatory power of appearance matching. Color histograms have been shown to be effective in the recognition of objects with complex spectral variations [26]. Here, we are interested in pose invariant recognition and hence use color images in their entirety to exploit not only the color measurements but also their spatial arrangement in the scene.

A color image of each of the 100 objects learned by the system is shown in Figure 3. A variety of strategies can be used to exploit color information in appearance matching. One approach is to concatenate the three color bands (red, green, and blue) of the image into a single appearance vector prior to brightness normalization. Such a vector would capture the spectral properties of the object; brightness normalization preserves the relative contributions of the different bands. The exact order of concatenation is not important as long as the same order is used during learning and recognition. This results from the fact that the arrangement of pixels and their attributes in the appearance vector only effects the ordering of elements of the computed eigenvectors but not their values [21].

We have chosen not to concatenate the color bands into a single vector as it triples the size of the appearance vectors. This, in turn, makes the computation of eigenspaces both slower and more memory intensive. Instead, we have chosen to break up the recognition problem and compute a separate eigenspace and a set object manifolds for each of the three color bands. While adopting this approach, care must be taken to ensure that color information is exploited. If each band is brightness normalized independent of others, the relative strengths of the different colors are lost. To avoid this, we normalize each band of the color image with the total energy in all three bands. Though the resulting band vectors are not of unit magnitude, they remain invariant to the intensity of illumination while capturing the object's spectral properties.

The current implementation assumes that the objects lie on a planar surface and hence always show up in one of a finite number of stable configurations. Therefore, pose variations used for learning correspond to rotations about a single axis that is normal to the planar surface. The learning images were obtained by rotating the object on a computer-controlled turntable. A total of 48 discrete poses were used for each object, i.e. pose increments of 7.5 degrees. The learning procedure for the 100 objects, including computation of the three subspaces and construction of object manifolds, took approximately 1 day.

4 Finding the Closest Manifold Point

During recognition, an input color image is segmented and each object region is normalized in scale and brightness as in the learning stage. Each color band is projected to its respective eigenspace and an object is recognized when the projections in all three eigenspaces are close to the manifolds of the same object and the closest manifold point in the three spaces all correspond to approximately the same pose (within 2 pose degrees of each other).

Mapping an input image to eigenspace is computationally simple. The universal eigenspace for each color band has 30 dimensions. The projection of an input image to a 30D space requires 30 dot products of the image with the orthogonal eigenvectors that constitute the space. Given that the normalized images are small (128x128), all 90 dot products (three color bands) can be computed in approximately 320 msec on the DEC Alpha 3600 workstation that is currently being used to demonstrate the system. 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 [19]. This algorithm uses a carefully designed data structure to facilitate quick search through the multidimensional eigenspace in approximately $O(\,k\,log_2\,n\,)$. Figure 1 illustrates the data structure, which is created off-line as follows. Given the set of densely sampled points in eigenspace, called the *point set*, the elements of each dimension of the point set are sorted independently in ascending order. It is clear that this independent sorting causes the coordinates of any given manifold point to be scattered in different rows in the *ordered set*. To preserve connectivity between coordinates, we use two types of maps. The *forward map* maintains the mapping between the original point set and the ordered set and the *backward map* facilitates mapping in the opposite direction (see Figure 1).

Given a novel input point $\mathbf{f}_c$, all manifold points within ϵ from the novel point are determined using two binary searches on each dimension of the ordered set. The result is a range of rows (indices) for each dimension (dark shaded areas in Figure 1). Next, using the forward and backward maps, possible candidates are determined as ones with coordinates that are within ϵ from those of the novel point. Exhaustive search on this short list of candidate points reveals the closest manifold point and the corresponding pose parameter. In [19], code for the above algorithm is given and the algorithm is demonstrated to be easier to implement than most existing ones with similar complexity. In the recognition system, the above search algorithm is applied independently to the three color bands. An object is recognized when it is detected in all three color bands. The total search time is approximately 140 msec on the DEC workstation.

The second approach [9] uses three-layered radial basis function (RBF) networks [23] to learn the mapping between input points and task parameters. The complexity of the network approach depends on the number of networks used and their sizes. In [9] a novel framework is introduced that uses the wavelet integral transform for finding the smallest RBF network to accomplish a given

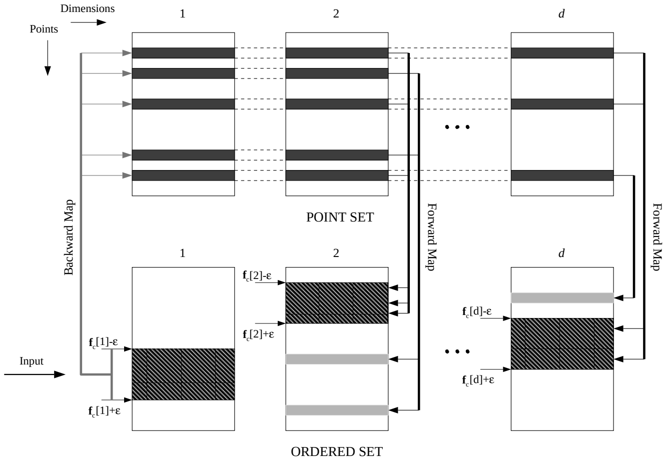


Figure 1: Data structure used to facilitate binary search through high-dimensional eigenspace for the closest manifold point.

input-output mapping. The performance of the network-based scheme is often comparable to that of the binary search approach. The network implicitly interpolates, or reconstructs, manifolds from the discrete points $\mathbf{f}_j$ and therefore does not require the use of spline interpolation followed by resampling. This advantage however comes with a slight sacrifice in parameter estimation accuracy.

5 Real-Time Recognition System

The structure of the real-time recognition loop is illustrated in Figure 2. What we have implemented is an infinite loop that enables a human to interactively test the recognition system in the laboratory. Images from the color sensor are continuously checked for scene changes. Each new frame is subtracted from the previous one and a significant change is declared when the number of pixels with significant brightness variation is large. This triggers a second change detector that does exactly the same as the first one but waits for the scene to stop changing. Once the scene has stabilized, an image is digitized and object regions are thresholded away from the black background of the scene. A sequential labeling algorithm is applied to the resulting binary image to obtain a segmented image. The largest labeled region is regarded as the object of interest. The region is normalized in scale and its color bands are normalized in brightness as described in section 3. The normalized vectors are projected to the three eigenspaces and recognition and pose estimation are done using the search algorithm described in the previous section. A template of the recognized object and its pose in degrees are overlaid on the live image of the scene.

If input projections in eigenspace are distant from all 100 manifolds in at least one of the three bands, a question mark is displayed to indicate that the object in the scene is not one of those in the database. Table 1 shows the time taken by the DEC workstation to execute each component of the recognition loop. A complete recognition cycle takes approximately 700 msec. Note that this performance is obtained without the use of any customized image processing hardware. We estimate that video-rate recognition can be attained using customized hardware built with embedded processors such as the i960.

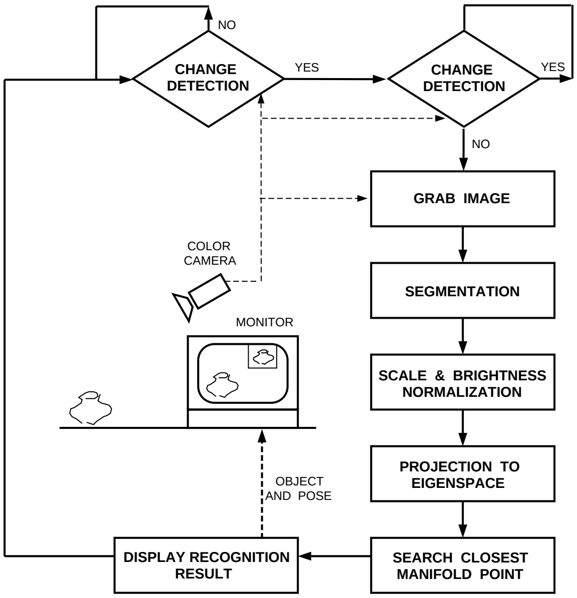


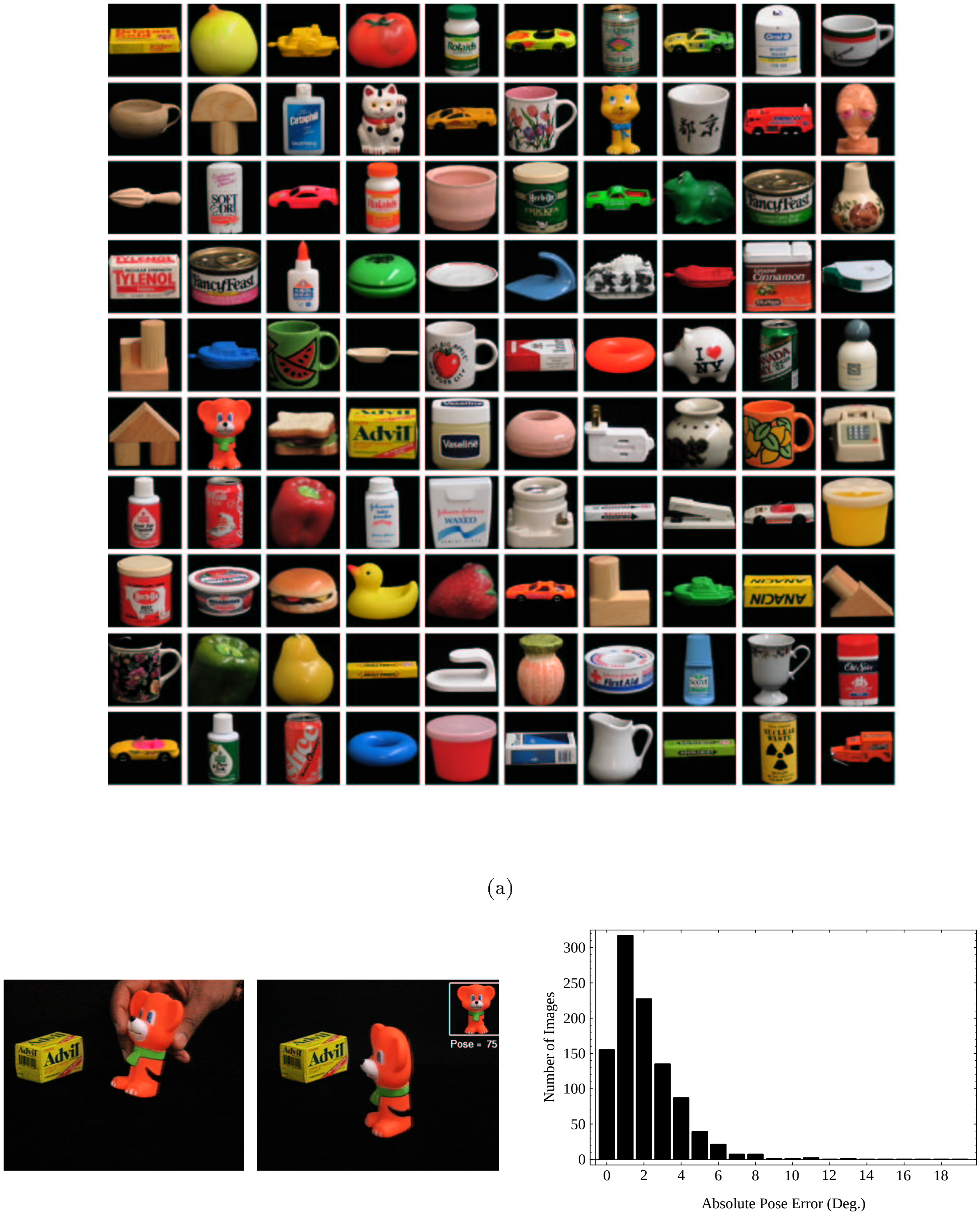
Figure 2: Components of the real-time recognition loop.

OPERATION	**TIME* (msec)**
SEGMENTATION	140
NORMALIZATION	100
PROJECTION	320
SEARCH	140
TOTAL	**700**

* DEC ALPHA 3600 WORKSTATION

Table 1: Performances of the individual components of the recognition system.

Figure 3(b) shows the first and last frames of a complete recognition cycle. The system has been, and continues to be, extensively tested by visitors to the laboratory. We conducted a formal set of experiments to quantify the robustness and accuracy of the system. From the set of 100 objects, we picked 20 that do not possess pose ambiguities (multiple poses for which the object appears the same). A total of 1000 images of these 20 objects were taken at known poses. The recognition rate for these images turned out to be 100 %. The pose estimation accuracy is illustrated by the error histogram in Figure 3(c). The mean and standard deviation of the absolute pose error were found to be 2.02 degrees and 1.67 degrees, respectively. Given that the learning images were taken 7.5 degrees apart, these numbers indicate high performance.



(a)

(b)

(c)

Figure 3: (a) 100 objects used to train the recognition system. (b) The system is in an infinite loop that enables a user to present different objects to it. (b) Histogram of the absolute pose error (in degrees). The histogram was computed using 1000 test images of 20 objects (ones with no pose ambiguities) taken at known poses.

6 Applications of Appearance Matching

Parametric appearance models have been applied to a variety of problems besides object recognition, such as, illumination planning for robust recognition [13], visual positioning and tracking [18], and temporal inspection [16] of complex parts. The results demonstrate that the techniques underlying appearance modeling and matching are general. This has led to the development of a comprehensive software package [20] for appearance matching that is presently being used at several research institutions.

References

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

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

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

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

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

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

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

[8] 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.

[9] S. Mukherjee and S. K. Nayar, "Automatic Generation of RBF Networks for Visual Learning," *Proc. of Fifth International Conference on Computer Vision,* Boston, June, 1995.

[10] 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.

[11] 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.

[12] 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, January, 1995.

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

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

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

[16] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace Methods for Robot Vision," CUCS-06-95, Technical Report, Department of Computer Science, Columbia University, New York, February 1995.

[17] S. K. Nayar and H. Murase, "On the Dimensionality of Illumination Manifolds in Eigenspace," CUCS-021-94, Technical Report, Department of Computer Science, Columbia University, New York, August 1994.

[18] 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.

[19] S. A. Nene and S. K. Nayar, "Binary Serach Through Multiple Dimensions," Technical Report, Department of Computer Science, Columbia University, New York, (in preparation).

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

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

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

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

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

[25] 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.

[26] M. Swain and D. H. Ballard, "Color Indexing," *International Journal of Computer Vision,* pp. 11-32, November 1991.

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