ICPR 96

Learning by a Generation Approach to Appearance-based Object Recognition

Hiroshi Murase
NTT Basic Research Labs
Morinosato Wakamiya, Atsugi
243-01, JAPAN

Shree K. Nayar
Colombia University
New York, NY 10027
USA

E-mail: murase@apollo3.brl.ntt.jp, nayar@cs.columbia.edu

Abstract

We propose a methodology for the generation of learning samples in appearance-based object recognition. In many practical situations, it is not easy to obtain a large number of learning samples. The proposed method learns object models from a large number of generated samples derived from a small number of actually observed images. The learning algorithm has two steps: (1) generation of a large number of images by image interpolation, or image deformation, and (2) compression of the large sample sets using parametric eigenspace representation. We compare our method with the previous methods that interpolate sample points in eigenspace, and show the performance of our method to be superior. Experiments were conducted for 432 image samples for 4 objects to demonstrate the effectiveness of the method.

1 Introduction

Appearance-matching techniques are becoming popular in machine vision. For example, Pentland et. al proposed the eigenface representation for human face recognition [1]. Recently, a new representation of object appearance called parametric eigenspace [2] was proposed by Murase and Nayar. In parametric eigenspace, an object is represented as a continuous appearance manifold in a low-dimensional subspace parameterized by object pose, illumination direction, and other relevant parameters. This method is invariant to parameter changes such as a pose change and gives good experimental results for object recognition. The parametric eigenspace representation has found several important applications. These include learning object models [2], real-time recognition of 3D objects, real-time positioning and tracking of 3D objects by a robot manipulator, object detection from cluttered scenes [3], and illumination planning for robust object recognition [4]. Recently, a recognition system with 100 complex rigid objects in its database was developed that is solely based on appearance matching. The sheer efficiency of appearance matching enables the system to accomplish both recognition and pose estimation in real time using nothing more than a standard workstation equipped with an image sensor.

In the context of large systems, the primary bottleneck has turned out to be the learning stage, which includes the acquisition of a large image set. Poggio et al. [5] proposed the virtual view for learning in their face recognition system. Daffertshofer et. al [6] used deformed patterns in the matching stage. Rubinstein et. al [7] proposed a recognition method for distorted patterns generated by Lie transformation groups. A related idea can be seen in character recognition in the analysis-by-synthesis approach. Murase et al. [8] have shown that the recognition accuracy of the pattern matching method improves when deformed characters are added to the dictionary. The basic idea underlying these approaches is the generation of feature patterns to deal with deformations.

In appearance-based object recognition using the parametric eigenspace approach, each object is represented as a separate manifold in eigenspace that is parameterized by pose and illumination parameters. The accuracy of the learning stage is determined by the number of sample images used to compute an appearance manifold. This leads to the following question: Can we decrease the number of images needed for constructing the appearance manifold for any given object? We examine the case where the number of available learning samples is very small, as is the case with many practical applications of object recognition.

For illumination parameters, Nayar and Murase have shown that the dimensionality of the illumination man-

ifold [9] for Lambertian surface is 3. This means that three images of the same object due to illumination from different directions is enough to construct the eigenspace manifold that corresponds to all possible illumination directions.

In this paper, we focus on the geometric parameters that effect the appearance of an object. Our method generates new appearance patterns due to variations in geometric parameters to decrease the number of the learning samples that need to be measured. First, many appearance variations are generated from a small numbers of measured samples by image interpolation, or image deformation. Next, this large number of generated images is compressed using the parametric eigenspace representation. The computed parametric eigenspace is very effective in applications where appearance is parametrized; for example, when objects are rotated in 3-D or 2-D, or objects are composed of parts that are connected to each other at joints with a small number of degrees of freedom. The combination of the generation of learning samples and the parametric eigenspace representation makes it possible to learn objects accurately even when the number of available learning samples is small.

The structure of the paper is as follows. The sample generation method is described in section 2, learning by generation in section 3, the parametric eigenspace representation in section 4, and experiments are reported in section 5.

2 Image generation

It is essential to have a large number of learning samples to increase the recognition accuracy. Image generation may help to increase the number of learning samples. There are two ways to generate the images. One is image interpolation, and the other is image deformation.

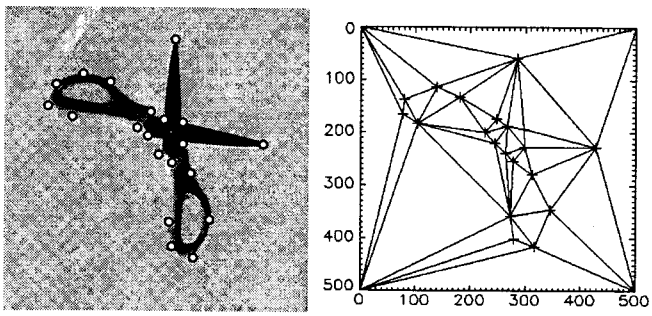
The original parametric eigenspace method interpolates sampled points in eigenspace by cubic spline interpolation. The interpolated points represent an image between two images corresponding to the two points in eigenspace. This interpolation is efficient only when the density of the sampled points is high enough, and two consecutive images are strongly correlated to each other. However, if the sampled pattern is very sparse, this interpolation does not work well because the point in the eigenspace represents an image that is a linear combination of images in a learning set. This means that it is difficult to construct a new image by any linear combination of images that are not correlated to the target image. However, if we generate the new images in image space, this problem can be solved. In this section, we show the generation in the image space, not in the eigenspace.

2.1 Image interpolation in image space

Many image interpolation techniques have been proposed, and they can be used for our purpose. The following interpolation method is one of them. We assume the control points are given and the correspondences are known. We need to provide this information only once in the learning stage, and this can be done manually. Assume that M control points in one image, I_1, are labeled (x_k, y_k) and corresponding M points in the other image, I_2, are labeled (x'_k, y'_k). We interpolate between two images, I_1 and I_2, controlled by parameter $a(0 < a < 1)$. Here, the control points in the interpolated image can be formulated as $(u_k, v_k) = a(x_k, y_k) + (1-a)(x'_k, y'_k)$. Next, we make a mapping function for the whole image area; $V(x, y)$ and $U(x, y)$ for mapping from I_1, and $V'(x, y)$ and $U'(x, y)$ for mapping from I_2. The interpolation is done in the following three steps.

(1) Triangulation

Partition of each image into triangular regions connecting neighboring control points with noncrossing line segments forms a planar graph. Delaunay triangulation is a well-known method, which can be calculated in the computation time of $O(nlogn)$, where n is the number of control points. Figure 1(a) shows the control points for the examples of scissors, and Fig. 1(b) shows a result of Delaunay triangulation for these points.



(a) (b)

Figure 1. (a) An example of control points for scissors, (b) Delaunay triangulation for the points.

(2) Linear triangulation patches

Deriving mapping function U, for example, is equivalent to determining a surface that passes through points (x_k, y_k, u_k). Here, we show a linear triangulation patch with a linear interpolant. The equation of a plane through three points $(x_1, y_1, u_1), (x_2, y_2, u_2)$, and (x_3, y_3, u_3) is

$$Ax + By + CU(x, y) + D = 0 \qquad (1)$$

where

$$A = \begin{vmatrix} y_1 & u_1 & 1 \\ y_2 & u_2 & 1 \\ y_3 & u_3 & 1 \end{vmatrix}, B = \begin{vmatrix} x_1 & u_1 & 1 \\ x_2 & u_2 & 1 \\ x_3 & u_3 & 1 \end{vmatrix}, \quad (2)$$

$$C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}, D = \begin{vmatrix} x_1 & y_1 & u_1 \\ x_2 & y_2 & u_2 \\ x_3 & y_3 & u_3 \end{vmatrix} \quad (3)$$

Each patch is calculated separately, and the surface $U(x,y)$ is constructed for the whole image area. We can calculate $V(x,y)$, $U'(x,y)$, and $V'(x,y)$ in the same way. Here, U' and V' are mapping functions from image I_2.

(3) Image resampling

Using mapping function (U,V) and (U',V'), we resample the images to form the interpolated image. We generate the image using the formula

$$Q(x,y) = aI_1(U(x,y),V(x,y)) + (1-a)I_2(U'(x,y),V'(x,y)) \quad (4)$$

The method above is piecewise linear mapping, where the mapping functions are continuous at the boundaries between neighboring functions, but they do not provide a smooth transition across patches. In order to obtain smoother results, the patches must be at least use $C1$ interpolants. There are several methods using N-degree polynomials. However, in many cases, piecewise linear mapping may be enough to approximate the interpolation.

2.2 Image deformation

The other type of image generation for learning object models is image deformation. The pattern is generated from a reference pattern in this case. If deformation rules of the object are given, they can be used to generate any types of deformation. Here, we show two examples.

(1) Affine transform

One simple but powerful deformation technique is affine transformations. The affine transforms cover rotation, size change, and shearing of the objects. The general representation of an affine transformation is

$$[x,y,1] = [u,v,1] \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix} \quad (5)$$

We can generate many types of useful deformed patterns by this transform. For example, Rotation by angle θ is

$$[x,y,1] = [u,v,1] \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

(2) Perspective transformation

If we consider the sensor position is close to the object, we have to consider the perspective transform [10]. Assume the view point is on the positive Z axis at $(0,0,f)$ looking toward the origin, and f is a focal length. The image plane is on the Z = 0 plane. The image plane point for any world point (x,y,z) is given by

$$(u,v) = \left(\frac{fx}{f-z}, \frac{fy}{f-z}\right). \quad (7)$$

We can generate the distorted image by non-linear image transformation parameterized by the camera position, which is formulated as above.

3 Learning by generation

If we consider all variations of object appearance, it is impossible to represent the object by small number of parameters; however, in many practical situations, the freedom of deformation is limited. In some situations, the object may be rotated along only one axis, or it may be deformed along with one joint. In this case, our approach is very powerful for learning in parametric eigenspace.

We can use different image generation rules for different geometric parameters. Some of them can generate strict deformation and some are approximated deformation.

First, we assume the simple case of two-step image generation; (i) image interpolation, and (ii) affine transformation (rotation). Figure 2 shows the image set generated from only three actually observed samples.

4 Parametric eigenspace

In the learning step, these generated images are compressed using the parametric eigenspace representation. This section shows the way to make this representation.

4.1 Eigenspace

Each learning image is represented by the N dimensional vector $\hat{\mathbf{x}}_{r,s}(r = 1,...,R, s = 1,...,S)$, where the element of the vector is the pixel value of the image, N is the number of the pixels, r is the rotation parameter, and s is the interpolation parameter. Here, R and S are the total number of discrete joint angles and rotations, respectively. We normalize the brightness to be unaffected by variations in intensity of illumination or the

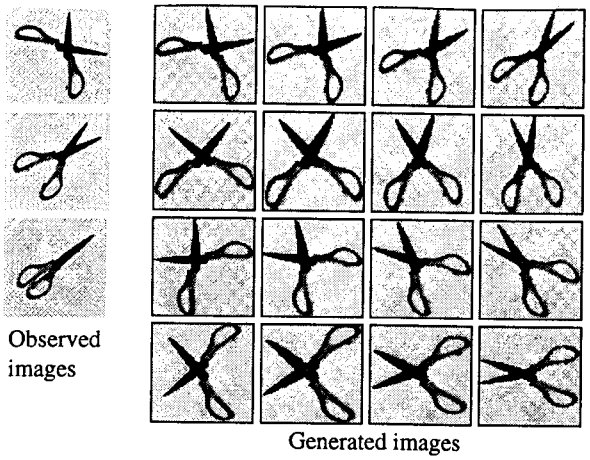


Figure 2. Examples of the generated patterns from three observed examples.

aperture of the imaging system. This can be achieved by normalizing each image such that the total energy constrained in the image is unity. This brightness normalization transforms each measured image $\hat{\mathbf{x}}_{r,s}$ into a normalized image $\mathbf{x}_{r,s}$ where

$$\mathbf{x}_{r,s} = \frac{\hat{\mathbf{x}}_{r,s}}{||\hat{\mathbf{x}}_{r,s}||} \tag{8}$$

The covariance matrix of this normalized image vector set is

$$Q = \frac{1}{RS}\sum_{s=1}^{S}\sum_{r=1}^{R}(\mathbf{x}_{r,s} - \mathbf{c})(\mathbf{x}_{r,s} - \mathbf{c})^T. \tag{9}$$

Here, c is the average of all images in the learning set determined as

$$Q = \frac{1}{RS}\sum_{s=1}^{S}\sum_{r=1}^{R}\mathbf{x}_{r,s}. \tag{10}$$

The eigenvectors $\mathbf{e}_i (i = 1, ..., k)$ and the corresponding eigenvalues λ_i of Q can be determined by solving the well-known eigenvalue decomposition problem

$$\lambda_i \mathbf{e}_i = Q\mathbf{e}_i. \tag{11}$$

Although all N eigenvectors of the planning image set are needed to represent images exactly, only a small number ($k << N$) of eigenvectors are generally sufficient for capturing the primary appearance characteristics of objects. The k-dimensional eigenspace spanned by the eigenvectors,

$$\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_k (\lambda_1 \geq \lambda_2 \geq, ..., \geq \lambda_k), \tag{12}$$

is the optimal subspace to approximate the original leaning image set in the sense of l^2 norm. Computing the eigenvectors of a large matrix such as Q can prove computationally very intensive. Figure 3 shows eigenvectors for the object in Fig. 2.

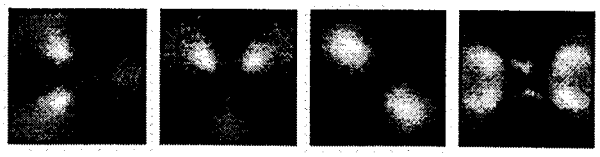


Figure 3. Eigenvectors for the patterns in Fig. 2.

4.2 Parametric Manifold

The next step is to construct the parametric manifold for the object in eigenspace. Each image $\mathbf{x}_{r,s}$ in the object image set is projected to the eigenspace by finding the dot product of the result with each of the eigenvectors of the eigenspace. The result is a point $\mathbf{g}_{r,s}$ in the eigenspace:

$$\mathbf{g}_{r,s} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_k]^T \mathbf{x}_{r,s}. \tag{13}$$

Once again the subscript r represents the rotation parameter and s is the interpolation parameter. By projecting all the learning samples in this way, we obtain a set of discrete points in universal eigenspace. Since consecutive object images are strongly correlated, their projections in eigenspace are close to one another. Hence, the discrete points obtained by projecting all the learning samples can be assumed to lie on a k-dimensional manifold that represents all possible poses and a limited range of object size variation. We interpolate the discrete points to obtain this manifold. In our implementation, we have used a standard cubic spline interpolation. This interpolation makes it possible to represent appearance between sample images. The resulting manifold can be expressed as $\mathbf{g}(\theta_1, \theta_2)$, where θ_1 and θ_2 are the continuous rotation and interpolation parameters. The above manifold is a compact representation of the object's appearance. Figure 4 shows the parametric eigenspace representation of the object in Fig. 2. It shows only three of the most significant dimensions of the eigenspace since it is difficult to display and visualize higher dimensional spaces. The object representation in this case is a surface since the object image set was obtained using two parameters. If we add more parameters such as rotations in other axes, this surface becomes high dimensional manifold.

The total flow of the learning involves (i) the collection of sample images, (ii) image interpolation, (iii) image deformation, (iv) eigenspace computation, (v) point interpolation in eigenspace, and (vi) manifold construction.

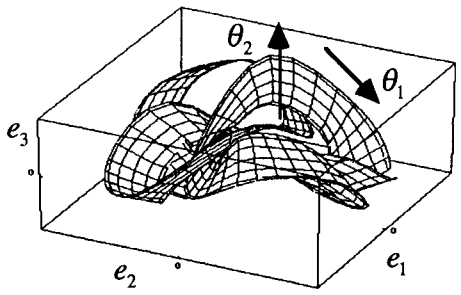


Figure 4. Parametric eigenspace representation (manifold in the eigenspace).

4.3 Object recognition using the Parametric Eigenspace

First, an input image is normalized with respect to brightness as described in the previous section. The normalized sub-image is represented by vector y. Next, y is projected into the eigenspace by

$$\mathbf{h} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_k]^T \mathbf{y}. \quad (14)$$

If this image belongs to the learned object, the projected point $\mathbf{y}$ will be located on the manifold $\mathbf{g}(\theta_1, \theta_2)$. Next, we compute the distance between the projected point and the manifold using

$$d = min_{\theta_1,\theta_2} ||\mathbf{h}(x,y) - \mathbf{g}(\theta_1,\theta_2)||. \quad (15)$$

The rotation and joint angle parameters can be estimated by the parameters θ_1 and θ_2 that minimize the distance.

5 Interpolation in eigenspace vs. interpolation in image domain

When interpolation is performed in eigenspace, the image corresponding to the interpolated points is equivalent to one that is generated by the linear combination of the original image set. If two consecutive images are not strongly correlated, the interpolation in eigenspace is not accurate.

Here, we can compare two cases. The learning samples are the three shown in Fig. 5. We denote this as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$, where $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ are observed images.

Assume a novel image is $\mathbf{y}$, and this image is represented by the linear sum of the three images. The minimum case of the mean square error is then

$$\varepsilon = min(||\mathbf{Xp} - \mathbf{y}||), \quad (16)$$

where the elements of vector $\mathbf{p}$ are the coefficients of each image for the linear sum. This value is the lower bound of the distance between the manifold and the point. This means that even if we make the manifold using any good interpolation function in the eigenspace, the distance is greater than this value. $\mathbf{Xp} = \mathbf{y}$ is an over determinant linear system, so $\mathbf{p}$ which minimize $||\mathbf{Xp} - \mathbf{y}||$ is formulated as $\mathbf{p} = \mathbf{X}^*\mathbf{y}$, where $\mathbf{X}^*$ is the pseudoinverse of $\mathbf{X}$.

We compare two cases: (1) the pattern is interpolated only in the eigenspace, and (2) the pattern is interpolated in the both image space and eigenspace. Figure 5 shows the lower bound ε for both cases. If we do not apply our method, ε is low only when the novel pattern is around the learning samples. On the other hand, ε is low for all values of the parameter in our method. We can see from this experiment that the image interpolation improves the accuracy of the parametric eigenspace representation even when the number of samples is very low.

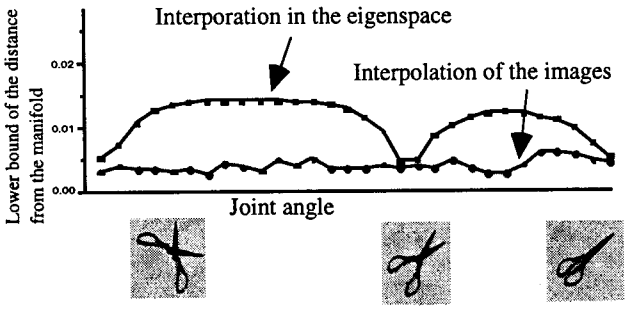


Figure 5. Lower bound of the distance from the manifold.

6 Experiments

(1) Image sets

Our learning framework can be applied to many cases where the object appearance changes in a parametric way. For the experiments, we used a typical object set as follows. Each object in the set is composed of two parts that are connected to each other at a joint, and the joint angle is unkown. The orientation of the object is unkown. The camera is viewing the object from above. This is a common situation for object recognition in the factory. Experiments were conducted on the set of four objects shown in Fig. 6.

(2) Experiment 1

In the first experiment, we considered only image interpolation. For the learning image set, only three samples with different joint angles were used for each object. For test samples, we took 54 samples for each object with random joint angle (Total 432 samples). The orientation for each sample was approximately the

Figure 6. Examples of the test data.

same. Each of these images was automatically normalized in scale and brightness. Each normalized image was 64x64 pixels in size. In the learning stage, we generate the interpolated pattern from three learning samples, calculate the eigenspace, and form the manifold in the eigenspace. Here, the control points were selected manually. The manifold was formed by again interpolating among the projected points of generated samples in the eigenspace. We used 16 dimensions of eigenspace, because preliminary experiments showed this is enough. Figure 7(a) shows the recognition rates versus the number of interpolated samples. The recognition rate with our method was 100%. The rate obtained when using only observed samples for learning was 92.5%.

(3) Experiment 2

In this experiment we considered combination of two kinds of generation: interpolation and deformation. We used the same three samples for learning. First we interpolated the images in the same way as above, and rotated the patterns by affine transform. A part of the generated patterns are shown in Fig. 2. For test samples, we took 180 samples for each object with random orientation and random joint angles. This gave us a total 720 images. Each of these images was normalized in scale and brightness in the same way. Figure 7(b) shows the recognition rates versus the number of rotated samples. A recognition accuracy of 99.5% was obtained with our method.

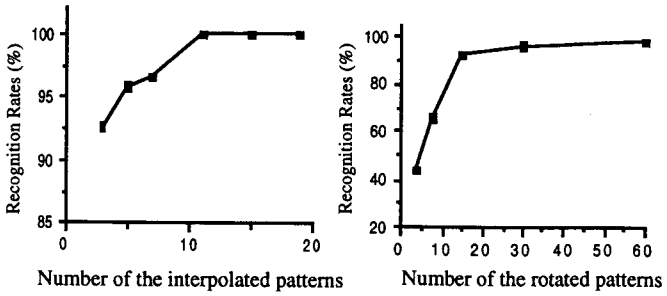


Figure 7. (a) The recognition rates versus the number of interpolated samples, (b) the recognition rates versus the number of rotated samples.

7 Conclusion

We have proposed a new learning algorithm for appearance-based object recognition. The algorithm consists of a sample generation step using image interpolation or deformation and an image compression step using parametric eigenspace representation. Experiments were conducted for 720 image samples for 4 objects, and the results show a recognition accuracy of 99.5% was obtained. In this paper, we have shown only simple deformation cases; however, we can apply our method for many types of deformation for many kinds of objects.

[Acknowledgement]

This research was conducted at the NTT Basic Research Laboratories, Japan. The authors thank Dr. Ikegami, Dr. Ishii, Dr. Hagita, and Dr. Naito for their encouragement.

[References]

[1] A. Pentland, B. Moghaddam, T. Starner, "View-based and modular eigenspaces for face recognition", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 84-91, 1991.

[2] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance", International Journal of Computer Vision, IJCV, Vol. 14, pp.5-24, 1995.

[3] H. Murase and S. K. Nayar, "Image spotting of 3D objects using parametric eigenspace representation", Scandinavian Conference on Image Analysis, SCIA95, 1995.

[4] H. Murase and S. K. Nayar, "Illumination planning for object recognition using parametric eigenspace", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 16, No. 12, pp. 1219-1227, 1994.

[5] D. Beymer and T. Poggio, "Face recognition from one example view, Proc. of International Conference on Computer Vision, pp.500-507, 1995.

[6] A. Daffertshofer and H. Haken, "A new Approach to recognition of deformed patterns, Pattern Recognition", Vol. 27, No. 12, pp. 1697-1705, 1994.

[7] J. Rubinstein, J.Segman, and Y. Zeevi, "Recognition of distorted patterns by invariance kernels", Pattern Recognition, Vol. 24, No. 10, pp. 959-967, 1991.

[8] H. Murase, F. Kimura, M. Yoshimura, Y. Miyake, "An improvement of the auto-correlation matrix in pattern matching method and its application to hand-printed HIRAGANA", Trans. IECE, vol. J64-D, No. 3, pp. 276-283, 1981.

[9] S. K. Nayar, and H. Murase, "On the dimensionality of illumination manifold in eigenspace", IEEE conf. on Robotics and Automation, 1996.

[10] D. H. Ballard, and C. M. Brown, Computer Vision, Prentice-Hall Inc., 1982.