ELSEVIER

Pattern Recognition Letters 18 (1997) 375–384

Pattern Recognition Letters

Detection of 3D objects in cluttered scenes using hierarchical eigenspace

Hiroshi Murase [a,*], Shree K. Nayar [b]

[a] NTT Basic Research Laboratories, 3-1, Morinosato Wakamiya, Atsugi-Shi, Kanagawa, 243-01, Japan
[b] Department of Computer Science, Columbia University, New York, USA

Received 20 August 1996; revised 29 November 1996

Abstract

This paper proposes a novel method to detect three-dimensional objects in arbitrary poses and sizes from a complex image and to simultaneously measure their poses and sizes using appearance matching. In the learning stage, for a sample object to be learned, a set of images is obtained by varying pose and size. This large image set is compactly represented by a manifold in compressed subspace spanned by eigenvectors of the image set. This representation is called the parametric eigenspace representation. In the object detection stage, a partial region in an input image is projected to the eigenspace, and the location of the projection relative to the manifold determines whether this region belongs to the object, and what its pose is in the scene. This process is sequentially applied to the entire image at different resolutions. Experimental results show that this method accurately detects the target objects. © 1997 Elsevier Science B.V.

Keywords: Object recognition; Segmentation; Eigenvectors; Hierarchical representation

1. Introduction

Detection of three-dimensional (3D) objects has wide applications such as visual search of a target in security systems or target detection in recognition systems. There are two approaches used for object detection. One uses local features such as edges or corners and matches them with 3D models (Besl and Jain, 1985; Chin and Dyer, 1986; Poggio and Edelman, 1990; Weng et al., 1993). This method might handle 3D rotation and scaling of objects; however, extraction of geometric features from noisy natural scenes is not easy. The other approach uses template matching such as image correlation (matched filtering) or image subtraction. This approach is insensitive to noise and small distortions. Our method is based on this approach.

Template matching is a fundamental task in image processing. Even if we limit the discussion to search problems, many vision algorithms using template matching have been proposed. For example, feature detection using template matching in pyramids (Rosenfeld and Vanderbrug, 1977; Tanimoto, 1981), using matched-filters (Liu and Caelli, 1988), or using modular eigenspaces (Pentland et al., 1991), were proposed. Caelli and Liu (1988) showed that a small number of templates is enough to detect a pattern using template matching. However, these methods were developed for two-dimensional template matching, so they cannot deal directly with 3D objects in a 3D scene. A 3D object has many appearances (Weng

[*] Corresponding author. E-mail: murase@eye.brl.ntt.jp.

PII S0167-8655(97)00017-2

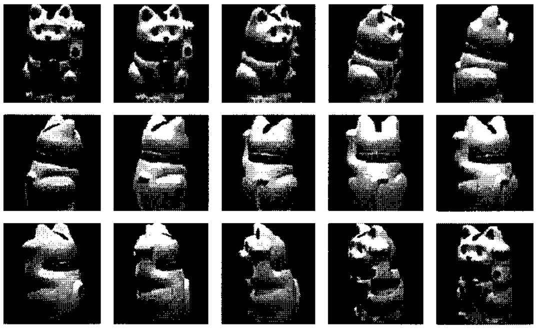


Fig. 1. A variety of appearances when varying the pose of one object.

et al., 1993) depending on the pose and distance between the camera and the object. Fig. 1 shows that a variety of appearances can be seen even for one object. If we store all variations of the object appearance and sequentially match them with the whole subpart of the input image using conventional template matching, a vast amount of memory and computation time is required. Our method is related to this exhaustive template matching, however, we use a new compact representation that makes the computation of image correlation quick and efficient. This representation is called parametric eigenspace. This approach makes it possible to detect a 3D object in an arbitrary pose and position in the scene.

The idea of a parametric eigenspace was first applied for isolated object recognition (Murase and Nayar, 1994, 1995a). We extend this idea to object detection (Murase and Nayar, 1995b), which solves the complex situation in which there is an object with a complicated background. This representation uses two fundamental ideas: KL (Karhunen–Loeve) expansion and a manifold representation. The KL expansion is a well-known technique to approximate images in the low-dimensional subspace spanned by eigenvectors of the image set. This technique is based on principal component analysis (Fukunaga, 1990; Oja, 1983), and has been applied to pattern recognition problems such as character recognition (Murase et al., 1981) and human face recognition (Sirovich and Kirby, 1987; Pentland et al., 1991). We call this subspace the eigenspace. Calculation in the eigenspace reduces computation time. Secondly, an appearance manifold conveniently represents continuous appearance changes due to changes in parameters such as object pose or object size. The combination of these two ideas yields a new continuous and compact representation of 3D objects. We used this representation for partial image matching and hierarchical matching at image resolutions to detect target objects.

2. Learning object models

The appearance of an object depends on its shape, reflectance properties, pose, distance from the camera, and the illumination conditions. The first two parameters are intrinsic properties of the object that do not vary. The correlation method is relatively robust to illumination variations when a brightness normalization process is used. On the other hand, object pose and camera distance can vary substantially from one scene to the next. Here, we represent an object using the parametric eigenspace representation that is parameterized by its pose and its distance from the camera.

2.1. Search window

First, for a given object sample to be learned, we collect a set of images by varying the pose using a computer controlled turntable. Then we segment the object region from each image and normalize its size to some fixed rectangle. Next, we generate several sizes of the images (i.e., scale factor $1, 1.1, 1.2, \ldots, \alpha$, where $\alpha = 1.5$) for each pose. These images are used for object learning. We refer to this image set as the learning image set (Fig. 2). Using all the generated images, we design the search

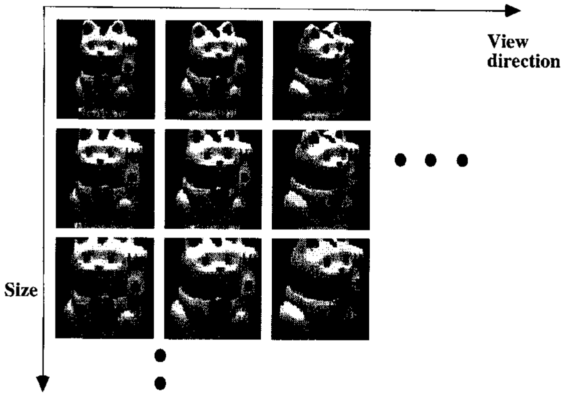
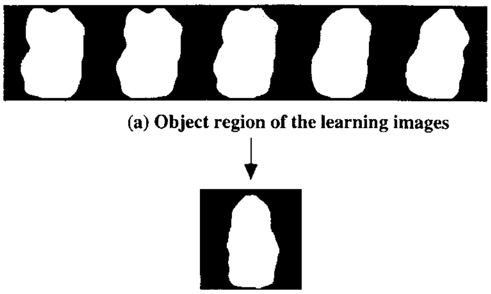


Fig. 2. A learning image set.



(a) Object region of the learning images

(b) Search window

Fig. 3. A search window.

window. The window is the AND area of the object region of all images in the learning image set. Fig. 3 shows an example of the search window constructed using the learning image set. This search window is introduced to eliminate the background region and extract only the parts of the object region in the learning stage. In the object detection stage, this search window is used to scan the entire input image.

2.2. Eigenspace

Each learning image is first masked by the search window, then represented by the N-dimensional vector $\hat{\boldsymbol{x}}_{r,s}$ $(r = 1,\ldots,R,\ s = 1,\ldots,S)$, where the element of the vector is a pixel value of the image inside the window, N is the number of pixels, r is a pose parameter, and s is a size parameter. Here, R and S are the respective total number of discrete poses and sizes. We normalize the brightness to be independent of variations in intensity of illumination or the 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 $\hat{\boldsymbol{x}}_{r,s}$ to a normalized image $\boldsymbol{x}_{r,s}$ where

$$\boldsymbol{x}_{r,s} = \frac{\hat{\boldsymbol{x}}_{r,s}}{\| \hat{\boldsymbol{x}}_{r,s} \|}.$$

The covariance matrix of this normalized image vector set is

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

Here, $\boldsymbol{c}$ is the average of all images in the learning set determined as

$$\boldsymbol{c} = \frac{1}{RS} \sum_{s=1}^{S} \sum_{r=1}^{R} \boldsymbol{x}_{r,s}.$$

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

$$\lambda_i \boldsymbol{e}_i = Q\boldsymbol{e}_i.$$

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

$$\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k\} \quad (\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_k)$$

is an optimal subspace to approximate the original learning image set in the sense of an 1^2 norm. Computing the eigenvectors of a large matrix such as Q can prove to be computationally very intensive. Efficient algorithms for this are summarized in (Murase and Lindenbaum, 1995; Oja, 1983). Fig. 4 shows eigenvectors for the object shown in Fig. 2.

2.3. Correlation and distance in eigenspace

In this section, we discuss the relation between image correlation and distance in an eigenspace. Consider two images $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ that belong to the image set used to compute an eigenspace. Let the points $\boldsymbol{g}_m$ and $\boldsymbol{g}_n$ be the projections of two images in the eigenspace. Each image can be expressed in terms of its projection as

$$\boldsymbol{x}_m = \sum_{i=1}^{N} g_{mi} \boldsymbol{e}_i + \boldsymbol{c},$$

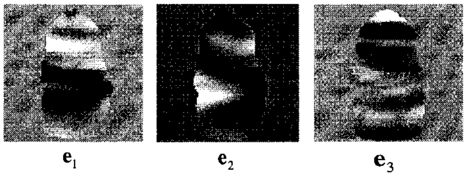


Fig. 4. Eigenvectors for a learning image set shown in Fig. 2.

where c is once again the average of the entire image set. Note that our eigenspaces are composed of only k eigenvectors. Hence, x_m can be approximated by the first k terms in the above summation:

$$x_m \approx \sum_{i=1}^{k} g_{mi} e_i + c.$$

As a result of the brightness normalization described in Section 3.2, x_m and x_n are unit vectors. The SSD (sum-of-squared-difference) measurement between the two images is related to correlation as

$$\| x_m - x_n \|^2 = (x_m - x_n)^{\mathrm{T}} (x_m - x_n) \\ = 2 - 2 x_m^{\mathrm{T}} x_n,$$

where $x_m^{\mathrm{T}} x_n$ is the correlation between the images. Alternatively, the SSD can be expressed in terms of the coordinates g_m and g_n in the eigenspace:

$$\| x_m - x_n \|^2 \approx \left\| \sum_{i=1}^{k} g_{mi} e_i - \sum_{i=1}^{k} g_{ni} e_i \right\|^2 \\ = \| g_m - g_n \|^2.$$

So we have

$$\| g_m - g_n \|^2 \approx 2 - 2 x_m^{\mathrm{T}} x_n.$$

This relation implies that the square of the Euclidean distance between the point g_m and g_n is an approximation of the SSD between the images x_m and x_n. In other words, the closer the projections are in an eigenspace, the more highly correlated are the images. We use this property of an eigenspace to calculate image correlation efficiently.

2.4. Parametric manifold

The next step is to construct the parametric manifold for the object in an eigenspace. Each image $x_{r,s}$ 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 $g_{r,s}$ in the eigenspace:

$$g_{r,s} = [e_1 \ldots e_k]^{\mathrm{T}} x_{r,s}.$$

Once again, the subscript r represents the rotation parameter and s is the size parameter. By projecting all the learning samples in this way, we obtain a set of discrete points in a universal eigenspace. Since consecutive object images are strongly correlated, their projections in an 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 (Press et al., 1988). This interpolation makes it possible to represent appearance between sample images. The resulting manifold can be expressed as $g(\theta_1, \theta_2)$ where θ_1 and θ_2 are the continuous rotation and size parameters. The above manifold is a compact representation of the object's appearance. Fig. 5 shows the parametric eigenspace representation of the object shown in Fig. 1. The figure shows only three of the most significant dimensions of the eigenspace since it is difficult to display and visualize higher-dimensional spaces. The object representation in this case is a surface since the object image set was obtained using two parameters. If we add more parameters such as rotations about other axes, this surface becomes a high-dimensional manifold.

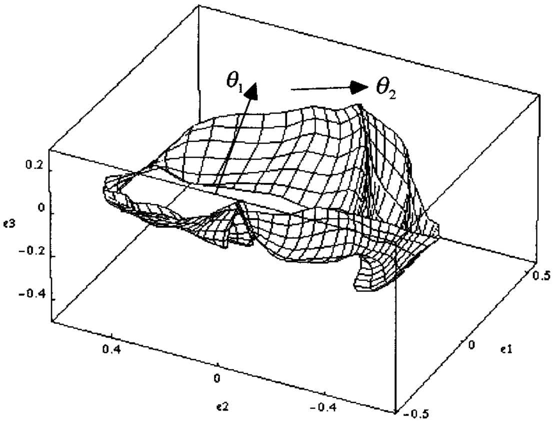


Fig. 5. A parametric eigenspace representation for the object shown in Fig. 2.

3. Image spotting

3.1. Image spotting using the parametric eigenspace

Consider an image of a scene that includes one or more of the objects that we have learned, on a complicated background. We assume that the objects are not occluded by other objects in the scene when viewed from the camera direction.

First, the search window is scanned on the entire input image area $(1 \leqslant x \leqslant X,\ 1 \leqslant y \leqslant Y)$ and a sequence of the subimages is made. Here, X and Y are sizes of the input image. The search window eliminates the background effect and extracts only a subpart of the input images, namely, inside the object region. Each subimage is normalized with respect to brightness as described in the previous section. The normalized subimage at position (x, y) is represented by vector $\boldsymbol{p}(x, y)$. Next, $\boldsymbol{p}(x, y)$ is projected into the eigenspace by

$$\boldsymbol{h}(x, y) = [\boldsymbol{e}_1 \dots \boldsymbol{e}_k]^{\mathrm{T}} \boldsymbol{p}(x, y).$$

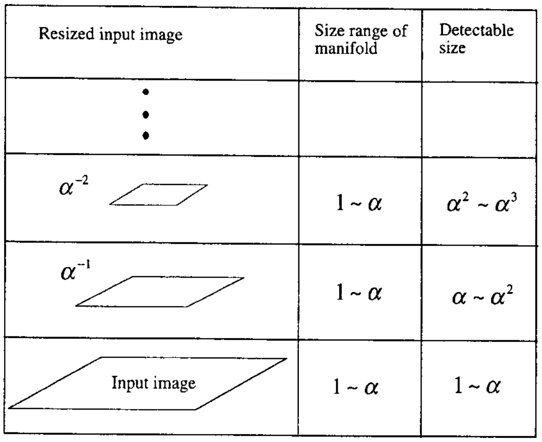
If this subimage is one of the learning set, the projected point $\boldsymbol{h}(x, y)$ will be located on the manifold $\boldsymbol{g}(\theta_1, \theta_2)$. Next, we compute the distance between the projected point and the manifold, using

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

If the distance $d(x, y)$ is less than some predetermined threshold value, the position (x, y) is a candidate for the object. After finding the candidate, the minimum peak of the distance around this position is searched, because the distance of the subimage at (x, y) is similar to that of the subimage around this position since these images are correlated to each other. Finally, we can conclude that the position that minimizes the distance is of the object. The pose and size parameters can be estimated by the parameters θ_1 and θ_2 that minimize the distance.

3.2. Hierarchical image spotting

We assume weak perspective image projection. This means the size of the object is a function of the distance between a camera and the object. As shown in the previous section, the parameter θ_2 in the manifold can deal with size variation of the object region. However, the dynamic range should be limited, because the effective window area, that is, the area used for correlation, becomes small if we cover a large range of the size parameter using the parametric eigenspace representation. In our experiment, we set the dynamic range of the size parameter of the manifold to around 1.5 $(1 \leqslant \theta_2 \leqslant 1.5)$. This range in size variation is not sufficient for many applications. Here, to cover a wider range of size variations, we apply this process hierarchically. The input image is resized as $1, \alpha^{-1}, \alpha^{-2}, \dots,$ and the same object detection procedure is applied for each resized image. Here, α is set to the maximum value of the parameter θ_1. As a result, this method can cover size variations continuously. Fig. 6 shows the range to cover the size for each resized input image.



Resized input image	Size range of manifold	Detectable size
• • •		
α^{-2}	$1 \sim \alpha$	$\alpha^2 \sim \alpha^3$
α^{-1}	$1 \sim \alpha$	$\alpha \sim \alpha^2$
Input image	$1 \sim \alpha$	$1 \sim \alpha$

Fig. 6. Hierarchical scaling of the input image for arbitrary size.

3.3. Computational cost

Here, we discuss the computational cost by estimating the number of operations such as multiplications for each calculation of distance $d(x, y)$. Let the number of pixels in the search window be N, and the number of the templates that corresponds to the number of the possible poses and sizes be M. The term k is the dimension of the eigenspace. In general, $N > M \gg k$. If we ignore the second order, the number of operations for each step for the conventional correlation technique is as follows: (1) N multiplications for normalization. (2) NM multiplications for the correlation. (3) M comparisons to find the maximum correlation. The total number of operations for each calculation of distance $d(x, y)$ is $N + NM + M$. On the other hand, if we apply the parametric eigenspace method, the operation in each step is as follows. (1) N multiplications for normalization. (2) Nk multiplications for the projection. (3)

Mk multiplications for distance calculation. (4) M comparisons to find the minimum distance. The total number of operations is $N + Nk + Mk + M$, hence, the dominant factor is $(N + M)k$. Assume that N is 6,500 (the number of pixels in the search window), and M is 3,600 ($R = 360$, $S = 10$), and k is 10. These numbers are picked from the example in our experiments (see Section 5). The results show there are 111,100 operations for our method, and 23,410,100 operations for the exhaustive correlation method. At the same time, the memory size for the templates can be reduced from NM to Nk words.

4. Experiments

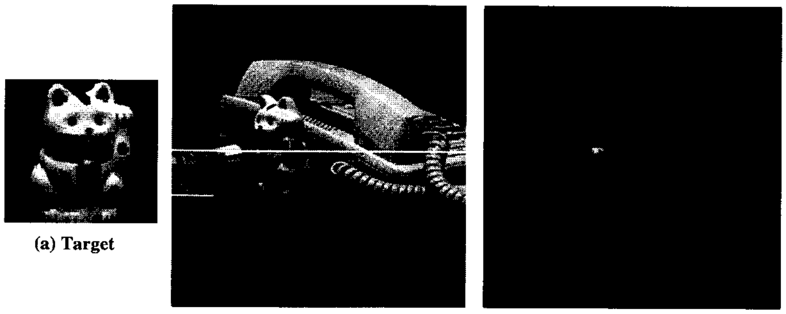
We conducted several experiments using complex objects to verify the effectiveness of the parametric eigenspace representation. This section summarizes some of our results.

For example, we demonstrated five kinds of target objects: a toy cat, a juice can, two paper boxes, and a human face. In the learning step, the object is placed on a motorized turntable and its pose is varied on a single axis, namely, the axis rotation of the turntable. Most objects have a finite number of stable configurations when placed on a planar surface. For such objects, a turntable is adequate as it can be used to vary poses for each of the object's stable configurations. For a human face, we used a rotating stool instead of a turntable. When learning, we used a black background to make it easy to segment the object region from the background. Images of the object are sensed using a 512 × 480 pixel CCD camera and are digitized to 8 bits per pixel. We took 45 images of different poses for each object for learning. The object region is segmented by the simple thresholding technique and its size is normalized to 128 × 128 pixels. Then, we compute a search window and the parametric eigenspace representation for each object (see Fig. 2). In this example, the window has 5400 pixels.

Next, we constructed a manifold in the eigenspace according to the procedure in Section 3, and densely resampled the manifold by 360 poses for the rotation parameter and 10 steps for the size parameter. There were 3,600 total resampled points on the manifold, which are used for searching the manifold. In our experiments, we calculated the distance from the manifold by finding the nearest neighbor distance from these resample points.

To test the algorithm, we used 30 images where the target object was placed on the complicated background. We applied the procedure explained in Section 4. Fig. 7(c) shows the distance map for the image example shown in Fig. 7(b) and the target object shown in Fig. 7(a). Here, a white pixel indicates an area of small distance value, and the object is possibly there. The computation time is two minutes using a SUN workstation SS10. Fig. 8 shows the results for several input images. Our method correctly detected 28 out of 30 examples. Errors in detection occurred when a part of the background was similar in texture to the target object.

We evaluated the number of dimensions of the eigenspace by changing the number of dimensions and plotting the graph of the distance from the manifold. If there are too few dimensions, the repre-


(a) Target

(b) Input image

(c) Distance from the manifold

Fig. 7. An example of image spotting. (a) A target object, (b) an input image, (c) distance map from the manifold.

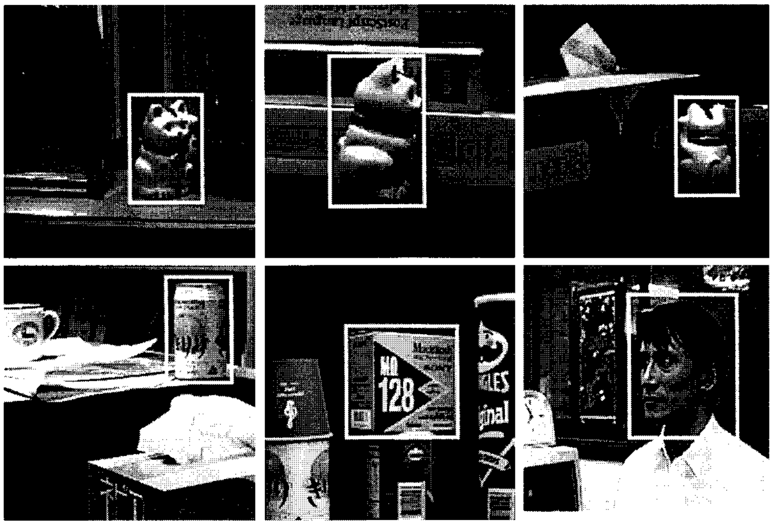


Fig. 8. Results of image spotting.

sentation is less accurately approximated. Fig. 9 shows the graph of the distance value for the place along the white line in Fig. 7(b). In Fig. 9(a) (two-dimensional eigenspace was used), there are many positions that do not belong to the object and that have a small distance. This causes errors in object detection. Figs. 9(b) and (c) show the graphs for six- and 16-dimensional eigenspace, respectively. If there are enough dimensions, the correct position of the object is detected accurately. We tested this algorithm with 30 test images, and found that 10-dimensional eigenspace is enough to detect these objects. We can evaluate the approximation of the representation by the sum of the contribution factor

$$\sum_{i=1}^{k} \lambda_i \Big/ \sum_{i=1}^{N} \lambda_i,$$

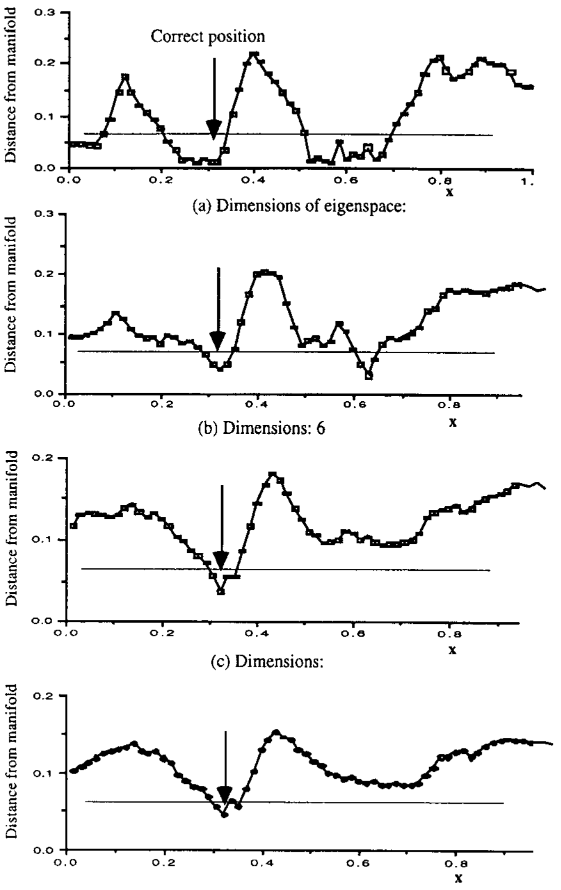
where λ_i is the ith largest eigenvalue, k is the dimension of eigenspace and N is the dimension of the original image. Fig. 10 shows a graph of the sum of the contribution factor. This is 0.7 for 10 dimensions.

Image correlation methods are robust to noise. The result for the noise pattern (SNR is 20 dB) is shown in Fig. 11. Fig. 9(d) shows the distance graph for this case. It is not very different from Fig. 7(c). Our method works well for these images.

We can compute an object's pose at the same time of object detection, by estimating the pose parameter that minimizes the distance from the manifold. It was shown that the accuracy of the pose estimation for an isolated object (Murase and Nayar, 1995a) is high using the parametric eigenspace method. In our case, however, the strong feature of the object boundary cannot be used, because the boundary can be obtained after segmentation and pose estimation. We evaluate the pose estimation error for both cases for the same object: (1) isolated objects, using image matching including object boundary, and (2) objects that are not isolated, using only partial matching of the object region. Fig. 12 shows a histogram of the error for both cases for 45 test images. The accuracy of the pose estimation without using boundary information is still high, although it is a little lower than that using boundary information. Cui and Weng (1996) used a prediction-and-verification approach for the segmentation of hand gesture. This idea is useful when the boundary of the object cannot be extracted in a bottom-up approach. If our method includes this approach, the accuracy of the pose estimation might be improved.

5. Discussion

This section briefly discusses advantages and limitations of our method.



(a) Dimensions of eigenspace:

(b) Dimensions: 6

(c) Dimensions:

(d) Dimensions:16 + White noise (SNR=20dB)

Fig. 9. Distance from the manifold for each position on the white line in Fig. 7(b). (a) Using two-dimensional eigenspace, (b) using six-dimensional eigenspace, (c) using 16-dimensional eigenspace, (d) using 16-dimensional eigenspace for an image with white noise (SNR = 20 dB).

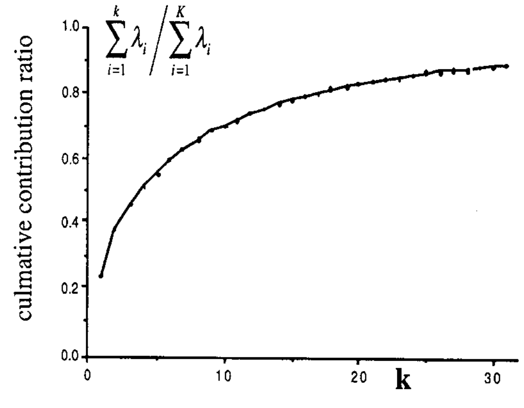


Fig. 10. Sum of the contribution ratio.

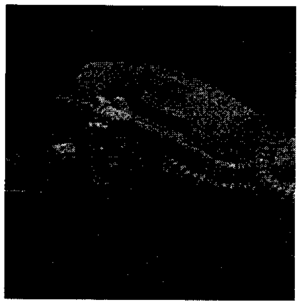


Fig. 11. An input image with white noise (SNR = 20 dB).

(1) *Learning-based approach.* Because our method uses a learning-based approach, it can recognize rotated objects and differently sized objects if these variations of the learning samples are prepared in the learning step. The feature itself is not invariant to the rotation and size. Preparing good learning samples is the important factor with our method.

(2) *Eigenspace.* We calculate eigenspace separately depending on the object. The object detection is basically a two-class problem (object and background). If we can estimate a distribution of the features in the background, we can use discriminant analysis, which may be a more efficient space than object eigenspace. However, it is usually difficult to estimate the distribution of the background, because the background is almost always unknown.

(3) *Occlusion.* The appearance-based method is basically template matching, so it is robust to small noise; however, it simply cannot handle occlusion.

(4) *Lighting.* Illumination changes the appearance, and this causes problems with our method. One way to solve this problem is to add samples with

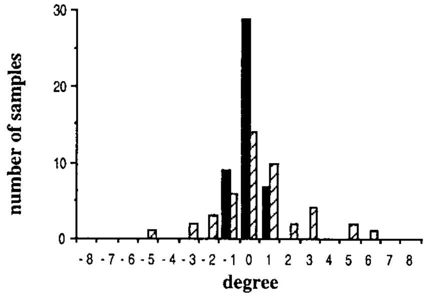


Fig. 12. Histogram for pose estimation error (the striped bar: without using boundary information; the black bar: using boundary information).

different illuminations when learning. This increases the number of parameters of the manifold in eigenspace. If the number of parameters is larger, we have to use high-dimensional eigenspace. This increases the computation time.

(5) *Local features.* One extension of our method is to extract local features and to integrate them to deal with the deformation of the shape. (Cootes and Taylor (1996) used a similar approach for shape deformation.) However, extraction of local features is not usually accurate because it can use only local information. If the object is rigid and not occluded, a global feature is more stable.

(6) *Shape of the mask.* We used the AND region of many appearances of a target object. If the object shape has large enough volume this works well; however, if the object is made of wires or thin parts, this approach does not work. This is the limitation of this mask approach.

6. Conclusion

In this paper, we described a detection method for a three-dimensional object with a complicated background. This new image representation is called the parametric eigenspace method. The method detects an object in an arbitrary pose and size in a natural scene based on 2D image correlation, and simultaneously computes the pose and size of the object. There are a variety of appearances for a 3D object depending on its pose and position. We represent them using a compact image representation based on two key ideas. One is the KL transform, which approximates the appearance of the object image set using a small number of eigenvectors to reduce the computation time and memory size. The other is a parametric representation, which represents the continuous change in appearance by varying pose and position by a manifold to compute object pose and position. Image correlation using this representation is hierarchically computed for different sizes of input images to cover a large dynamic size range of the object.

Experimental results showed this method can accurately detect the target object. We have also shown that this method reduces the computational cost compared with the exhaustive correlation method, and is also robust to noise in input images. Future research will concentrate on increasing parameters of the manifold to handle general pose of the object in three dimensional space, and recognizing objects with large occlusion.

Acknowledgements

We would like to thank Dr. K. Ishii and Dr. S. Naito for their encouragement.

References

Besl, P.J. and R.C. Jain (1985). Three-dimensional object recognition. *ACM Comput. Surveys* 17 (1), 75–145.

Caelli, T.M. and Z. Liu (1988). On the minimum number of templates required for shift, rotation and size invariant pattern recognition. *Pattern Recognition* 21 (3), 205–216.

Chin, R.T. and C.R. Dyer (1986). Model-based recognition in robot vision. *ACM Comput. Surveys* 18 (1).

Cootes, T.F. and C.J. Taylor (1996). Locating objects of varying shape using statistical feature detectors. *ECCV96*, 465–474.

Cui, Y. and J.J. Weng (1996). Hand segmentation using learning-based prediction and verification for hand sign recognition. *CVPR96*, 88–93.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition.* Academic Press, London.

Liu, Z.Q. and T.M. Caelli (1988). Multiobject pattern recognition and detection in noisy backgrounds using a hierarchical approach. *Computer Vision, Graphics and Image Processing* 44, 296–306.

Murase, H. and M. Lindenbaum (1995). Spatial temporal adaptive method for partial eigenstructure decomposition of large images. *IEEE Trans. Image Process.*, May.

Murase, H. and S.K. Nayar (1994). Illumination planning for object recognition using parametric eigenspace. *IEEE Trans. Pattern Analysis and Machine Intelligence* 16 (12), 1219–1227.

Murase, H. and S.K. Nayar (1995a). Visual learning and recognition of 3-D objects from appearance. *Internat. J. Comput. Vision* 14, 5–24.

Murase, H. and S.K. Nayar (1995b). Image spotting of 3D objects using parametric eigenspace representation. *SCIA95*, 323–332.

Murase, H., F. Kimura, M. Yoshimura and Y. Miyake (1981). An improvement of the auto-correlation matrix in pattern matching method and its application to handprinted 'HIRAGANA'. *Trans. IECE* J64-D (3).

Oja, E. (1983). *Subspace Methods of Pattern Recognition*, Research Studies Press, Hertfordshire.

Poggio, T. and S. Edelman (1990). A network that learns to recognize three-dimensional objects. *Nature* 343, 263–266.

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

Rosenfeld, A. and G.J. Vanderbrug (1977). Coarse-fine template matching. *IEEE Trans. System, Man, and Cybernetics* 2, 104–107.

Sirovich, L. and M. Kirby (1987). Low dimensional procedure for the characterization of human faces. *J. Optical Soc. Amer.* 4 (3), 519–524.

Tanimoto, S.L. (1981). Template matching in pyramids. *Computer Vision, Graphics and Image Processing* 16, 356–369.

Pentland, A.P., B. Moghaddam and T. Starner (1991). View-based and modular eigenspaces for face recognition. *IEEE CVPR*, 84–91.

Weng, J.J., N. Ahuja and T.S. Huang (1993). Learning recognition and segmentation of 3D objects from 2D images. *IEEE ICCV*, 121–128.