

# Detectability, Uniqueness, and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects

Kohtaro Ohba and Katsushi Ikeuchi

**Abstract**—This paper describes a method for recognizing partially occluded objects for bin-picking tasks using eigenspace analysis, referred to as the “eigen window” method, that stores multiple partial appearances of an object in an eigenspace. Such partial appearances require a large amount of memory space. Three measurements, *detectability*, *uniqueness*, and *reliability*, on windows are developed to eliminate redundant windows and thereby reduce memory requirements. Using a pose clustering technique, the method determines the pose of an object and the object type itself. We have implemented the method and verified its validity.

**Index Terms**—Object recognition, multiple objects, eigenspace, detectability, uniqueness, reliability.

## 1 INTRODUCTION

OBJECT recognition has a wide variety of military and civilian applications. Some of the representative applications include bin-picking, automatic target recognition, and surveillance and monitoring. Some earlier works in this domain include [1], [2], [3], [4], [5], [6]. Despite this long history of research, these applications still provide a challenge to vision researchers. The main difficulties include: real-time requirements, difficulty in segmentation, and difficulty in obtaining appropriate models of the objects.

Recently, visual learning methods based on eigenspace analysis [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17] have shown the potential to solve some of these problems. These methods learn object models from a series of images taken in the same environment as in the recognition mode. Thus, this method by-passes the difficulty in modeling. Furthermore, since such methods store an object model as a vector in a low dimensional eigenspace and recognize objects by comparison of model and image vectors, recognition speed is very rapid and can achieve real-time performance.

Though promising, the current eigenspace analysis assumes that all the appearances are non-occluded. The “image spotting” criterion was proposed by Murase and Nayar [16] to handle the recognition of multiple objects on a complicated background, but it did not handle occlusion. In order to apply the eigenspace analysis to recognition of partially occluded objects, we propose to divide appearances into small windows, referred to as “eigen windows” [19] and to apply eigenspace analysis to each eigen window. The basic idea is that, even if some of the windows are occluded, the remaining windows are still effective and can recover the object pose. The total number of such small windows may be very large. Storing all of them may require a prohibitive amount of memory space. This paper proposes a set of measures to select only effective windows. Krumm [20] independently developed a similar idea but did not address the uniqueness and

reliability of the proposed method. Those issues are explicitly addressed here.

In Section 2, we review eigenspace analysis, discuss the limitations of eigenspace analysis, and explain how to overcome these limitations using the eigen window method. Section 3 proposes three measures to select a small set of effective windows. Section 4 shows some of the experimental results and evaluates the stability of verification. Section 5 concludes this paper.

## 2 EIGEN WINDOW METHOD

First, we will review the eigenspace technique and discuss the limitations of the technique under image shift, occlusion, noise, scaling, and illumination. Then, the eigen window method is proposed. This method is designed to overcome these problems.

### 2.1 Eigenspace Technique

Let  $M$  be the number of the images in a training set. Each image has been converted into a column vector  $\mathbf{z}_i$  of length  $N$ :

$$[\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M] \quad (1)$$

By subtracting the average image of the all images, we obtain the training matrix,

$$\mathbf{Z} = [\mathbf{z}_1 - \mathbf{c}, \mathbf{z}_2 - \mathbf{c}, \dots, \mathbf{z}_M - \mathbf{c}] \quad (2)$$

where  $\mathbf{c}$  is the average image, and the size of the matrix  $\mathbf{Z}$  is  $N \times M$ .

The sample covariance matrix  $\mathbf{Q}$ ,  $N \times N$ , is obtained as:

$$\mathbf{Q} = \mathbf{Z}\mathbf{Z}^T \quad (3)$$

This sample covariance matrix provides a series of eigenvalues  $\lambda_i$  and eigenvectors  $\mathbf{e}_i$  ( $i = 1, \dots, N$ ), where each corresponding eigenvalue and eigenvector pair satisfies:

$$\lambda_i \mathbf{e}_i = \mathbf{Q}\mathbf{e}_i \quad (4)$$

That is, matrix  $\mathbf{Q}$  can be decomposed into  $N$  orthonormal components, of which the eigenvalues are  $\lambda_i$ . Thus, each image set can be described by a set of eigenvectors with associated weight factors, i.e., eigenvalues.

If the number of images  $M$  is much smaller than the number of pixels  $N$ , the implicit sample covariance matrix  $\tilde{\mathbf{Q}} = \mathbf{Z}^T\mathbf{Z}$  can be used instead of the sample covariance matrix  $\mathbf{Q}$  to calculate the first  $M$  eigenvectors [21].

For the sake of memory efficiency, we will ignore small eigenvalues and their corresponding eigenvectors using a threshold value,  $T_s$ :

$$W_k = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^N \lambda_i} \geq T_s \quad (5)$$

where  $k$  is sufficiently smaller than the original dimension  $N$ .

From this reduced set of eigenvectors, the matrix  $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k]$  is constructed to project an image,  $\mathbf{z}_i$  (dimension  $N$ ) into the eigenspace as an eigen point,  $\boldsymbol{\zeta}_i$  (dimension  $k$ ).

$$\boldsymbol{\zeta}_i = \mathbf{E}^T(\mathbf{z}_i - \mathbf{c}) \quad (6)$$

This eigenspace analysis can drastically reduce the dimension of the images ( $N$ ) to the eigenspace dimension ( $k$ ) while keeping several of the most effective features that summarize the original images.

• K. Ohba is with the Mechanical Engineering Laboratory, Ministry of International Trade and Industries, Tsukuba, Japan. Email: kohba@mel.go.jp.

• K. Ikeuchi is with the Institute of Industrial Science, The University of Tokyo, Tokyo, Japan. Email: ki@iis.u-tokyo.ac.jp.

Manuscript received 1 July 1996; revised 9 June 1997. Recommended for acceptance by P. Flynn.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 105311.

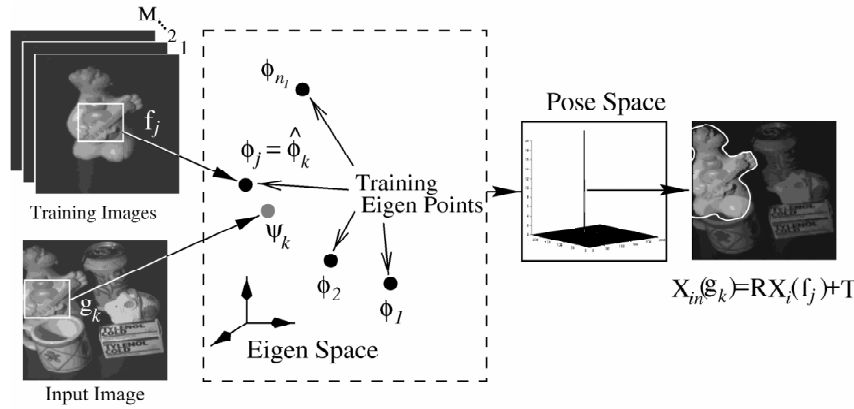


Fig. 1. Eigen window technique.

## 2.2 Limitations of the Eigenspace Technique

The eigenspace representation, which is a collection of points in the eigenspace, is very sensitive to image conditions—background noise, image shift, occlusion of objects, scaling of the image, and illumination changes.

As an effort to reduce these disturbance effects in the eigenspace, Murase and Nayar segmented only a window circumscribing the object using the movement of the object. Unfortunately, however, the usual bin-picking scenario does not provide such convenient clues for segmenting out a target region. Moreover, it often occurs that one window contains other objects due to the cluttered environment typical in a bin-picking scenario. Thus, we need a method to overcome these limitations.

## 2.3 Eigen Window Technique

To reduce the disturbance effects, we propose to apply small windows to the original images and to project all of them into the eigenspace. We refer to this method as the “eigen window” technique. Fig. 1 shows an overview of the technique.

### 2.3.1 Training Eigen Windows

The training set of eigen windows is given as:

$$\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_M] \quad (7)$$

where  $\mathbf{F}_i$  denotes the collection of eigen windows from the  $i$ th training image. Each  $\mathbf{F}_i$  has the form

$$[\mathbf{f}_1 - \mathbf{c}, \mathbf{f}_2 - \mathbf{c}, \dots, \mathbf{f}_{n_i} - \mathbf{c}]$$

where  $\mathbf{f}_j$  denotes the  $j$ th eigen window in the  $i$ th training image;  $n_i$  denotes the number of eigen windows in the  $i$ th image; and  $\mathbf{c}$  is the average eigen window across all eigen windows in the whole training set. In Fig. 1, the white square denotes one of the training eigen windows. A strategy for selection of these optimal eigen windows in each training image will be discussed in Section 3.

Note that all the projected points of these eigen windows are represented in a common eigenspace as shown in Fig. 1. Each point in the space has the label of the original eigen window and original training image.

### 2.3.2 Matching Operation

From an input image, a set of sub-window images is obtained:

$$\mathbf{G} = [\mathbf{g}_1 - \mathbf{c}, \mathbf{g}_2 - \mathbf{c}, \dots, \mathbf{g}_n - \mathbf{c}] \quad (8)$$

such as the white window in the lower left image in Fig. 1.

The similarity between a training eigen window and an input

eigen window is evaluated using their distance in the eigenspace. Given an input eigenspace point  $\psi_k$  projected from an input eigen window  $\mathbf{g}_k$  using (6), we try to find one corresponding point with maximum similarity among all of the dictionary points  $\phi$  projected from any training eigen window  $\mathbf{f}$ .

That is,

$$\hat{\phi}_k = \arg \min_{\phi} (\|\psi_k - \phi\|) \quad (9)$$

where  $\|\mathbf{x}\|$  denotes the norm of  $\mathbf{x}$  using L1-norm, L2-norm, or Mahalanobis distance, etc. We used the L1-norm to save computation time. We will denote the eigen window that projects to  $\hat{\phi}_k$  as  $\hat{\mathbf{f}}_k$ , and consider it as the corresponding training eigen window to the input window  $\mathbf{g}_k$ . In other words, each eigen windows  $\mathbf{g}_k$  in the input image is matched to its nearest training eigen window  $\hat{\mathbf{f}}_k$  with this equation.

### 2.3.3 Voting Operation

The previous matching operation selects a set of training eigen windows,  $[\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \dots, \hat{\mathbf{f}}_n]$  corresponding to input eigen windows,  $[\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n]$ . We will partition this set into groups such that they come from the same image, that is all eigen windows in one group,

$$[\hat{\mathbf{F}}_1, \hat{\mathbf{F}}_2, \dots, \hat{\mathbf{F}}_M] \quad (10)$$

where

$$\hat{\mathbf{F}}_i = \{\hat{\mathbf{f}} \mid \hat{\mathbf{f}} \text{ comes from image } i\} \quad (11)$$

We then prepare a pose space for voting from the correspondences. In this operation, we consider only the effect of translation, and therefore the space is two dimensional. Here, the size of the pose space is twice the size of the input image size, i.e.,  $256 \times 240$ . One pose space is prepared for each group,  $\hat{\mathbf{F}}_i$ .

Each correspondence between input eigen windows,  $\mathbf{g}_k$ , and training eigen window,  $\mathbf{f}_j (= \hat{\mathbf{f}}_k)$ , within the group  $\hat{\mathbf{F}}_i$  provides an estimate of the object's position. Let  $\mathbf{X}_{in}(\mathbf{g}_k)$  and  $\mathbf{X}_i(\mathbf{f}_j)$  denote the position of the input eigen window  $\mathbf{g}_k$  in the input image and that of the training eigen window  $\mathbf{f}_j$  in the  $i$ th image, respectively. Then, the difference,  $\Delta\mathbf{X} = (\Delta x, \Delta y)^T = \mathbf{X}_{in}(\mathbf{g}_k) - \mathbf{X}_i(\mathbf{f}_j)$  provides an estimate of the translation of the object. The cell that represents this distance in the two-dimensional pose space gets a vote. If several local appearances in the training image can be seen in the

input image, their estimates of translation should combine to form one prominent peak in the pose space. In order to absorb the digitization error, a  $5 \times 5$  window of cells around the center cell all get votes from a single correspondence. We repeat this operation using all the correspondences in the group (all the correspondences from the same training image).

### 2.3.4 Pose Determination

Some small peaks in the pose space are due to noise; other prominent peaks are due to actual objects in an input image. By thresholding these peaks, the system eliminates noise peaks and extracts prominent peaks. The number of the prominent peaks in the space is equal to the number of objects that have roughly the same rotation, but a different translation. By retrieving voted pairs in these prominent peaks, the system further divides this group into sub-groups so that each sub-group belongs to one prominent peak, and thus, one isolated object in the input image.

Although we consider only translation, since the training set is sampled along the rotation dimension, there will be a small rotation effect in object pose due to the sampling interval. To recover this small rotation and the precise translation value, the system computes a final pose estimate via weighted least square minimization. The pairs in each sub-group are weighted by their uniqueness  $1/\|\mathbf{g}_k - \mathbf{f}_j\|$  to compute the pose parameters by solving:

$$\mathbf{X}_m(\mathbf{g}_k) = \mathbf{R}\mathbf{X}_i(\mathbf{f}_j) + \mathbf{T} \quad (12)$$

$$\begin{bmatrix} x_{\mathbf{g}_k} \\ y_{\mathbf{g}_k} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_{\mathbf{f}_j} \\ y_{\mathbf{f}_j} \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \quad (13)$$

where  $\mathbf{R}$  and  $\mathbf{T}$  denote the small rotation and translation, respectively.

## 3 SELECTION OF EFFECTIVE EIGEN WINDOWS

One of the issues in the eigen window technique is the selection of the optimal set of eigen windows. If all the eigen windows are utilized,

- 1) the number of eigen windows becomes very large and storing them requires a large amount of memory space, and
- 2) due to the similarity among eigen windows, the matching process becomes increasingly prone to error.

In this section, we will introduce three criteria to select the optimal set of windows: *detectability*, *uniqueness*, and *reliability*. The *detectability* measures how easy it is to detect one window within a large image. For example, a window containing corners of an object is much easier to detect than those containing a planar region. Although some windows are easy to detect, they may be similar with each other, e.g., the target object may have multiple similar corners. In order to select truly discriminative windows, we will also introduce a global measure of *uniqueness*. Finally, a *reliability* measure selects windows that remain stable within a range of object poses. Through applying these three measures, we will be able to prune the full set of possible eigen windows down to an optimal set.

### 3.1 Detectability: Local Goodness

Window selection may be considered as selecting feature points for object tracking. In [18], the following  $2 \times 2$  matrix was proposed to measure the trackability of a window:

$$\mathbf{G} = \sum_{\mathbf{x} \in \mathcal{R}} \begin{pmatrix} \frac{\partial I}{\partial \mathbf{x}} \end{pmatrix} \begin{pmatrix} \frac{\partial I}{\partial \mathbf{x}} \end{pmatrix}^T \quad (14)$$

where  $\mathcal{X} = (x, y)^T \in \mathcal{R}$  and  $I$  denotes the image intensity. This

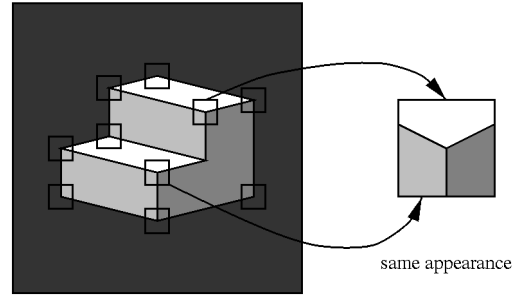


Fig. 2. A problem in corner detection with trackability.

matrix  $\mathbf{G}$ , has two eigenvalues  $\lambda_1, \lambda_2$ . The window is accepted as a good one if the relation

$$\min(\lambda_1, \lambda_2) > \lambda \quad (15)$$

holds, where  $\lambda$  is a predefined threshold. This measure works well for detecting all important corners.

### 3.2 Uniqueness: Global Goodness

The *detectability* measure selects windows containing features that are easily detected. Unfortunately, the *detectability* measure does not guarantee the global *uniqueness* of the window (see Fig. 2).

The global goodness of windows can be determined as the *uniqueness* of an eigen window. This *uniqueness* of each window may be measured using the similarity among eigen windows. As was discussed in Section 2.3.2, the similarity between training and input eigen windows was evaluated using their distance in the eigenspace, namely,

$$S_{l,m} = \|\phi_l - \phi_m\| \quad (16)$$

This measure,  $S_{l,m}$  denotes the similarity of the two eigen points  $\phi_l$  and  $\phi_m$  in the eigenspace, which are projected from  $\mathbf{f}_l$  and  $\mathbf{f}_m$ , respectively. We can use the same measure for evaluating the global goodness of a window, i.e., by evaluating the similarity among training eigen windows in eigenspace.

The similarity between two training eigen windows,  $\mathbf{f}_l$  and  $\mathbf{f}_m$ , is evaluated using (16). If this measure is less than a certain threshold  $T_{sim}$ , then these two eigen windows,  $\mathbf{f}_l, \mathbf{f}_m$ , are removed from the training set.

The elimination of similar eigen windows can make the size of a training set smaller than the original one. This operation also makes the matching process more stable, since the matching evaluation will not consider a sum of random contributions from a large number of similar eigen windows.

### 3.3 Reliability

To achieve stable object verification, we consider the *reliability* of each eigen window in this section. The *reliability* of an eigen window can be defined as how stable its projected points are in the eigenspace, when the object is viewed within some range of rotation angles. For example, if the training set rotation sampling angle is  $10^\circ$  each, eigen window *reliability* can be evaluated within view rotations of  $\pm 5^\circ$ . The training windows at each sampling angle have already been evaluated according to *uniqueness* evaluation in the previous section.

When an object undergoes small rotations around the sampling angle, the appearance of feature points may change; some of the feature points may disappear, and some new feature points may appear. We define highly reliable feature points as those that remain present and stay within some neighborhood of the original dictionary point  $\phi_j$  in the eigenspace, which is projected from eigen window,  $\mathbf{f}_j$ . Thus, a *reliability* measure,  $R_j^{\pm\Delta}$ , is defined as follows:

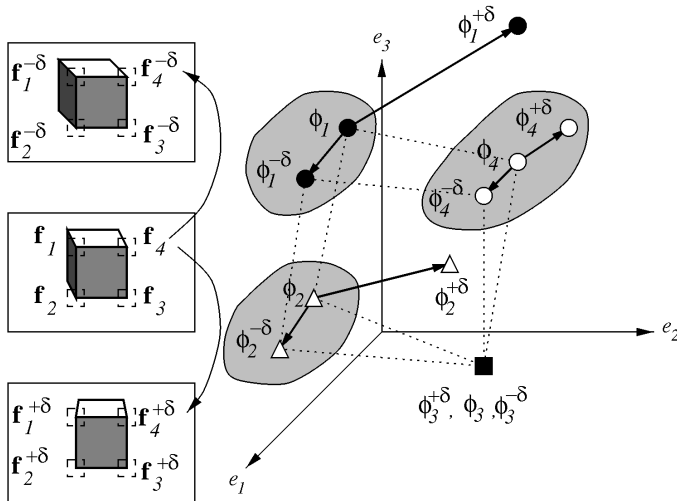


Fig. 3. Reliability evaluation in Eigen space.

$$R_j^{\pm\Delta} = \sum_{\delta \in \pm\Delta} \|\phi_j^{+\delta} - \phi_j\| \leq T_{rel} \quad (17)$$

where  $T_{rel}$  is a threshold value for the *reliability*, and  $\phi_j^{+\delta}$  denotes the eigen point, projected from the eigen window  $f_j^{+\delta}$ , at the same feature point but different sampling angle  $\delta$  (see Fig. 3).

To realize this process in consideration of the geometric relation of the eigen windows of the training images, the voting operating system in Section 2.3.3 may be useful. Here, we have an assumption, that the difference  $\delta$  is small enough to suppose the movements of each eigen window are only translation. Then, the difference,  $\mathbf{X}(f_j^{+\delta}) - \mathbf{X}(f_j)$  will make a single peak in the pose space, for which (17) holds. If there is any difference in appearance between points  $\phi_j^{+\delta}$  and  $\phi_j$ , the pose space contains some distribution of votes around the original peak. Elimination of features that cause this disturbance of the original peak enables a choice of eigen windows for which both (17) and the geometric relation of the eigen windows hold.

### 3.4 Example of Applying These Measures

Fig. 4 shows an example of applying these three measures to eigen windows extracted from the image in Fig. 4a. Fig. 4b depicts the center-points of eigen windows that pass the *detectability* measure. The vertical edges are selected as good windows. However, confusion often occurs among such windows along the edge. This problem is addressed by the *uniqueness* measure. Fig. 4c shows an example of the *uniqueness* evaluation in eigen-space. From the 477 eigen windows selected by the *detectability* measure, the *uniqueness* measure selects the 339 eigen windows shown in Fig. 4c. Most of the redundant windows such as along the edges in Fig. 4b are eliminated in Fig. 4c. The right figure of Fig. 4b and 4c shows the points projected in the 3D eigen space. In Fig. 4b, the distribution of projection is more uniform than that in Fig. 4c.

Fig. 4d and 4e shows the results of *reliability* evaluation within  $\pm 1$  degree and  $\pm 3$  degrees, respectively. In this case, most of the 339 feature points selected by the *uniqueness* criteria are eliminated to yield 217 and 47 points after *reliability* evaluation within  $\pm 1$  degree and  $\pm 3$  degrees, respectively.

## 4 EXPERIMENTAL RESULTS

This section presents some experimental results. First, the verification of multiple objects is shown to evaluate the validity of the

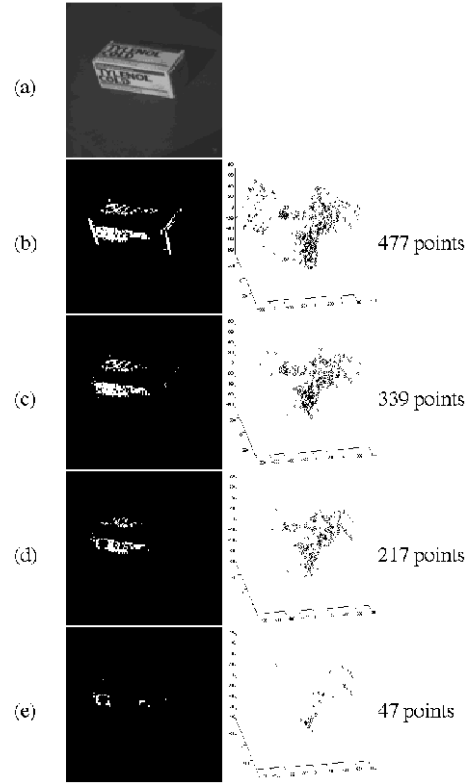


Fig. 4. Eigen window extracting process. (a) Original image. (b) Detectability. (c) Uniqueness evaluation. (d) Reliability evaluation  $\leq \pm 1$  degree. e. Reliability evaluation  $\leq \pm 3$  degrees.

algorithm in both training mode and run mode. Finally, in order to emphasize the difficulty in modeling, we intentionally use specular objects in this experiment.

### 4.1 Training Mode

In training mode, two series of images of each object are taken at  $10^\circ$  intervals using a rotary table and a CCD camera. The resolution of these images is  $128 \times 120$  pixels with intensity levels of eight bits.

- First, with the *detectability* algorithm in (14) and (15), we extract a set of feature points in each object image for sampling eigen windows. We set the size of an eigen window as  $15 \times 15$  pixels. Thus, the dimension of each eigen window is 225 ( $15 \times 15$  pixels), which is sufficiently smaller than the original image dimension of 15,360 ( $128 \times 120$  pixels). From each training image, we obtain 100 eigen windows on average. The eigenvectors and eigenvalues are calculated from the set of eigen windows using (4). The eigenvalues  $\lambda$  and  $W$  derived that a 20-dimensional subspace of the original eigenspace captures 80 percent of the original image data.
- Next, we apply the *uniqueness* measure to those eigen windows. We eliminate redundant eigen windows by evaluating the *uniqueness* among them with (16). This process selects several eigen windows that are unique. In this example, we can eliminate almost half of the eigen windows that were obtained with the *detectability* measure alone.
- Thirdly, to evaluate the *reliability* of eigen windows, we take several images around each of the sampling angles. In this case, we take two images of  $\pm 3^\circ$  around each training image (sampled at  $10^\circ$ ). Then, using (17), we evaluate the *reliability* of each eigen window, and eliminate the unstable ones, which reduces the total set of eigen windows by about one-half.

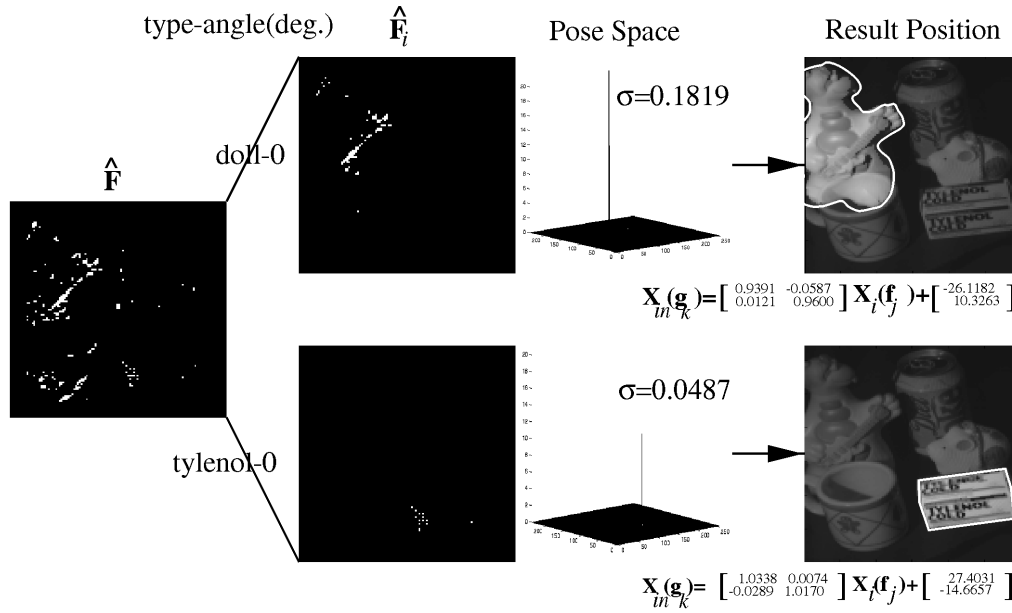


Fig. 5. Components of the verification results.

**4.2 Run Mode**

Fig. 5 shows the results in run mode. Here,  $\hat{F}$ , depicts those eigen windows that have been matched to some training eigen window. The figures in  $\hat{F}_i$  depict window positions that come from the same rotation angles of the same objects. The next column and the number  $\sigma$  depict the voting results and the standard deviation in the pose space, respectively, by the calculation of  $X_m(g_i) - X_i(f_j)$ . The number of votes represents the probability of the existence of the object at that pose angle. from these voting results, we can calculate the rotation and translation of each object using the voting elements which make a particular peak in the pose space. The final positions of each object are superimposed and high-lighted in the right-hand pictures with (12). The system correctly identifies each object and pose.

The algorithm also works well for the verification of specular objects, such as the multiple objects, which includes several bolts and BNC connectors. In this case, the image training set is composed of object training images such as those. Since specular features are all isolated edges and points, they are all good features in *detectability*. We thus did not apply the *detectability* and *reliability* measure.

The final verification and localization results are shown in Fig. 6, which shows the results superimposed on the original input image. The system identifies eight bolts out of ten and two BNC connectors in the input image. In this case, the missing bolts were not

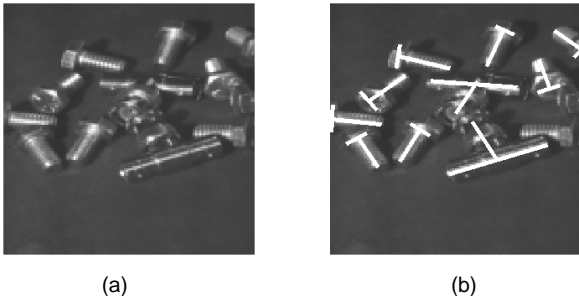


Fig. 6. Localization and verification results. (a) Original images. (b) Verification results.

bright enough to pass the *detectability* operation for detecting input eigen windows.

**5 CONCLUSION**

This paper describes a novel method, referred to as the eigen window method, to extend the standard eigenspace analysis technique to be able to recognize partially occluded objects. To reduce the redundancy among eigen windows, we have proposed three measures for selecting effective eigen windows: *detectability*, *uniqueness*, and *reliability*. We have implemented the eigen window method with these selection measures and verified the validity of the approach.

The limitation of the eigen window method may be recognition under illumination changes, such as changes of light position and brightness. Future work will concentrate on recognizing objects under severe illumination changes.

**ACKNOWLEDGMENTS**

We would like to thank Robert Collins, Carnegie Mellon University, the anonymous reviewers, and the associate editor for their kind comments and assistance in polishing this paper. This work was done in the Computer Science Department, Carnegie Mellon University.

**REFERENCES**

- [1] Y. Fukada, H. Doi, K. Nagamine, and T. Inari, "Relationships-Based Recognition of Structural Industrial Parts Stacked in Bin," *Robotica*, vol. 2, pp. 147-154, 1984.
- [2] J.R. Birk, R.B. Kelly, and H.A.S. Martines, "An Orienting Robot for Feeding Workpieces Stored in Bins," *IEEE Trans. Systems, Man, Cybernetics*, vol. 11, no. 2, pp. 151-160, 1981.
- [3] R.C. Bolles, P. Horaud, and M.J. Hannah, "3DPO: Three-Dimensional Parts Orientation System," *Proc. Int'l Joint Conf. Artificial Intelligence*, Karlsruhe, West Germany, pp. 1,116-1,120, Aug. 8-12
- [4] B.K.P. Horn and K. Ikeuchi, "The Mechanical Manipulation of Randomly Oriented Parts," *Scientific American*, vol. 251, no. 2, pp. 100-111, 1984.
- [5] S.A. Hutchinson and A.C. Kak, "SPAR: A Planner that Satisfies Operational and Geometric Goals in Uncertain Environments," *AI Magazine*, vol. 11, no. 1, pp. 31-61, 1990.

- [6] G.J. Agin and O.R. Duda, "SRI Vision Research for Advanced Industrial Automation," *Proc. 2nd USA-Japan Computer Conf.*, pp. 113-117, 1975.
- [7] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces," *Proc. CVPR 1991*, pp. 586-591, 1991.
- [8] S. Sclaroff and A.P. Pentland, "A Model Framework for Correspondence and Description," *Proc. 4th Int'l Conf. Computer Vision*, pp. 308-313, 1993.
- [9] A.P. Pentland and B. Horowitz, "Recovery of Nonrigid Motion and Structure," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 730-742, 1991.
- [10] A.P. Pentland, B. Moghaddam, and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," *IEEE Conf. Computer Vision and Pattern Recognition*, 1994.
- [11] B. Moghaddam and A.P. Pentland, "Probabilistic Visual Learning for Object Detection," *Proc. 5th Int'l Conf. Computer Vision*, 1995.
- [12] S. Sclaroff and A.P. Pentland, "Model Matching for Correspondence and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 545-561, 1995.
- [13] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, 1990.
- [14] H. Murase and S.K. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *Int'l J. Computer Vision*, vol. 14, no. 1, pp. 5-24, 1995.
- [15] S.K. Nayar and H. Murase, "Learning, Positioning, and Tracking Visual Appearance," *Proc. IEEE Int'l Conf. Robotics and Automation*, San Diego, Calif., May 1994.
- [16] H. Murase and S.K. Nayar, "Image Spotting of 3D Objects Using Parametric Eigenspace Representation," *Proc. 9th Scandinavian Conf. Image Analysis*, pp. 325-332, June 1995.
- [17] M. Uenohara and T. Kanade, "Vision-Based Object Registration for Real-Time Image Overlay," *Proc. First Int'l Conf. Computer Vision, Virtual Reality and Robotics in Medicine*, Nice, France, Apr. 1995.
- [18] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: A Factorization Method-2. Point Features in 3D Motion," Technical Report CMU-CS-91-105, Carnegie Mellon Univ., Pittsburgh, Penn., Jan. 1991.
- [19] K. Ohba and K. Ikeuchi, "Recognition of the Multi Specularity Objects Using the Eigen Window," *Proc. Int'l Conf. Pattern Recognition*, Vienna, Aug. 1996.
- [20] J. Krumm, "Eigenfeatures for Planar Pose Measurement of Partially Occluded Objects," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, June 1996.
- [21] H. Murakami and V. Kumar, "Efficient Calculation of Primary Images from a Set of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 5, pp. 511-515, 1982.