Machine Vision and Applications
© Springer-Verlag 2000

# Appearance-based visual learning and object recognition with illumination invariance

**Kohtaro Ohba**[1], **Yoichi Sato**[2], **Katsusi Ikeuchi**[2]

[1] Mechanical Engineering Laboratory, MITI, 1-2 Namiki, Tsukuba 305-8564, Japan; Tel: +81-298-61-7264, Fax: +81-298-61-7201, e-mail: kohba@mel.go.jp
[2] Institute of Industrial Science, The University of Tokyo, 7-22-1 Roppongi, Minato-ku, Tokyo 106-0032, Japan

**Abstract.** This paper describes a method for recognizing partially occluded objects under different levels of illumination brightness by using the eigenspace analysis. In our previous work, we developed the "eigenwindow" method to recognize the partially occluded objects in an assembly task, and demonstrated with sufficient high performance for the industrial use that the method works successfully for multiple objects with specularity under constant illumination. In this paper, we modify the eigenwindow method for recognizing objects under different illumination conditions, as is sometimes the case in manufacturing environments, by using additional color information. In the proposed method, a measured color in the RGB color space is transformed into one in the HSV color space. Then, the hue of the measured color, which is invariant to change in illumination brightness and direction, is used for recognizing multiple objects under different illumination conditions. The proposed method was applied to real images of multiple objects under various illumination conditions, and the objects were recognized and localized successfully.

**Key words:** Assembly tasks – Object recognition – Visual learning – Eigenspace – Illumination invariance

## 1 Introduction

Object recognition has a wide variety of military, civilian, and industrial applications. Some of the representative applications include bin-picking, automatic target recognition, surveillance and monitoring, and industrial inspection. Some of the earlier work in this domain include [1–6]. Despite the long history of research, these applications still provide a challenge to computer vision researchers. The main difficulties include requirement for high-performance computing, difficulty in segmentation, and difficulty in obtaining appropriate object models.

Recently, visual learning methods based on the eigenspace analysis have shown a potential to solve some of these difficulties [7–10]. In the eigenspace analysis, object models are *learned* from a series of images taken in the same environment as in the recognition mode. Thus, the difficulty of object modeling is avoided in the analysis. Furthermore, since an object model is stored as a vector in a low-dimensional eigenspace, and since objects are recognized by comparing the model with image vectors, computation for object recognition can remain effectively low enough to achieve real-time performance.

Although promising, the current eigenspace analysis is based on the assumption that objects are not occluded in images. Therefore, to apply the eigenspace analysis for partially occluded objects, we proposed to divide appearances into small windows, referred to as *eigenwindows* [11] and to apply eigenspace analysis to each eigenwindow. The basic idea is that, even if some of the windows are occluded, a large number of windows are still visible, so that the object can be recognized and localized in the images.

One drawback of the eigenwindow method is that only a limited number of images can be used for learning object models, and therefore, all possible illumination directions cannot be taken into account. Actually, this problem is quite important for the industrial inspection. Therefore, the object may be illuminated from a different direction in the recognition mode, resulting in incorrect recognition results.

In this paper, to overcome that drawback, we propose to use the color measurement *hue*, which is illumination-invariant, in the eigenwindow method. To demonstrate the effectiveness of the proposed method, we applied the method to real images taken under different illumination directions and brightness.

## 2 Eigenwindow method

In this section, we briefly review the eigenwindow method that we have proposed [11] to overcome limitations of the original eigenspace analysis, such as image shift, occlusion, noise, and scaling.

*Correspondence to*: K. Ohba

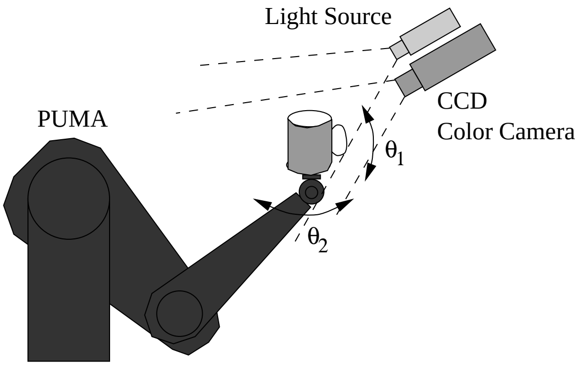

**Fig. 1.** Experimental setup

### *2.1 Eigenspace technique*

Let $M$ be the number of the images $z_1, z_2, \cdots, z_M$ in a training set related to each rotation of viewpoints $\theta_1$ and $\theta_2$, as shown in Fig. 1. Each image $z_i$, with dimensions $N \times N$, has been converted into a column vector of length $N^2$.

By subtracting the average brightness $c$ of all the images, we obtain the training matrix of the size of $N^2$ by $M$; $Z = [z_1 - c, z_2 - c, \cdots, z_M - c]$. This covariance matrix $Q = ZZ^T$ provides a series of eigenvalues $\lambda_i$ and eigenvectors $e_i (i = 1, \cdots, N^2)$, where each corresponding eigenvalue and eigenvector pair satisfies; $\lambda_i e_i = Qe_i$.

To reduce the memory requirement, we ignore eigenvectors corresponding to small eigenvalues $e_i (i > l)$. These eigenvectors do not significantly affect object recognition results. Once we obtain the remaining eigenvectors, we can construct the eigenvector matrix, $E = [e_1, e_2, \cdots, e_l]$, which projects an image $z_i$ (dimension $N^2$) into the eigenspace as an *eigenpoint* $g_i$ (dimension $l$).

$$g_i = E^T(z_i - c). \tag{1}$$

The eigenspace analysis can drastically reduce the dimension of the images ($N^2$) to the eigenspace dimension ($l$), while preserving enough dominant features to reconstruct the original images.

### *2.2 Eigenwindow technique*

To reduce the disturbance effects such as image shift and occlusion, we propose to select small windows in the original images. Each of the selected small windows is then analyzed by using the eigenspace analysis as described in the previous section. We call this method *the eigenwindow method*. Figure 2 shows the overview of the method.

#### 2.2.1 Training eigenwindows

The training set of eigenwindows is given as $F = [F^1, F^2, \cdots, F^M]$, where $F^i$ denotes the collection of eigenwindows from the $i$th training image. Each $F^i$ has the form $[f_1 - c, f_2 - c, \cdots, f_{n_i} - c]$, where $f_j$ denotes the $j$th eigenwindow in the $i$th training image; $n_i$ denotes the number of eigenwindows in the $i$th image; and $c$ is the average intensity value across all eigenwindows in the whole training set. In Fig. 2, the white square denotes one of the training eigenwindows.

#### 2.2.2 Matching operation

From an input image, a set of input eigenwindow images is obtained; $G = [g_1 - c, g_2 - c, \cdots, g_n - c]$, such as the white window in the lower left image in Fig. 2.

The similarity between a training eigenwindow and an input eigenwindow is evaluated by using the distance between them in the eigenspace. Given an input eigenpoint $\psi_k$ projected from input eigenwindow $g_k$ using Eq. 1, we try to find a training eigenpoint $\hat{\phi}_k$ from all training eigenpoints $\phi$ projected from all training eigenwindows $f$. The training eigenpoint is the one with the maximum similarity defined as $\hat{\phi}_k = \arg\min_{\forall\phi}(\|\psi_k - \phi\|)$, where $\|x\|$ denotes the norm of $x$ using L1-norm or L2-norm. We denote the eigenwindow that is projected to $\hat{\phi}_k$ as $\hat{f}_k$. The eigenwindow $\hat{f}_k$ corresponds to the input eigenwindow $g_k$.

#### 2.2.3 Voting operation

The previous matching operation selects a set of training eigenwindows; $[\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n]$ corresponding to input eigenwindows. We now sort the selected training eigenwindows into each group, which contains windows that come from the same training images, from a corresponding training image; $\left[\hat{F}^1, \hat{F}^2, \cdots, \hat{F}^M\right]$, where $\hat{F}^i = \{\hat{f} | \hat{f}$ *comes from training image* $i\}$.

We then prepare a pose space for voting from the established correspondences. In this operation, we consider only translation, and therefore the space is two-dimensional, corresponding to a $[x, y]$ location of eigenwindows. The size of the pose space is set to be twice that of the input image size, e.g., $256 \times 256$ in our examples. The pose space is prepared separately for each group $\hat{F}^i$.

Using each correspondence, we can compute the difference of the training eigenwindow's location $X(\hat{f}_k)$ and the input eigenwindow's location $X(g_k)$. ($X(f)$ represents $[x, y]$ location of the eigenwindow $f$.) The difference is given as $X(g_k) - X(\hat{f}_k)$.

Then, in the pose space, the cell that represents this distance, $X(g_k) - X(\hat{f}_k)$, gets a vote. To avoid the digitization error, all of the $5 \times 5$ neighbor cells around the center cell get a vote from a single correspondence. We repeat this operation, using all correspondences for a group $\hat{F}^i$ (all the correspondences from the same training image.) Then, we obtain a resulting pose space for each of the groups $\hat{F}^i$.

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, we can eliminate noise peaks and extract only prominent peaks.

#### 2.2.4 Pose determination

The number of the prominent peaks in the pose space is equal to the number of objects that have roughly the same rotation, but have a different translation.

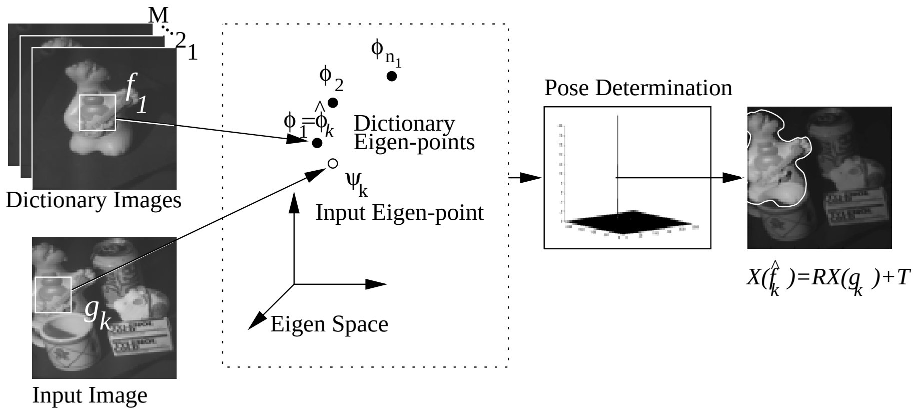

**Fig. 2.** Eigenwindow technique

By retrieving voted pairs, we further divide the group $\hat{F}^i$ into sub-groups, each of which belongs to each prominent peak, i.e., an isolated object in the input image.

Since the training set is sampled along the rotation dimension, there exists a side effect of small object rotation due to the finite sampling interval. To obtain the rotation and the translation precisely, we refine the pose estimate via a least squares minimization, using the pairs in each subgroup. $X(\hat{f}_k) = RX(g_k) + T$, where $R$ and $T$ denote the small rotation and translation, respectively.

### *2.3 Selection of effective eigenwindows*

In the eigenwindow method, it is very important and also difficult to select an optimal set of training eigenwindows. If all of the initially selected windows are used as the eigenwindows, two problems occur: 1) the number of eigenwindows becomes very large, and storing them requires a large amount of memory, and 2) due to the similarity among eigenwindows, the matching process becomes erroneous.

In this section, we introduce three criteria for selecting the optimal set of eigenwindows: *detectability*, *uniqueness*, and *reliability*.

The detectability measures the ease of detecting a window in an entire image. For example, a window containing corners of an object is much easier to detect than one containing a planar region.

Although some windows are easy to detect, they may be similar to each other. This situation arises when the target object has multiple similar corners. To select truly distinct windows, we introduce a global goodness measure called the uniqueness measure.

In addition, the reliability measure is used to select windows which do not appear and disappear with small variation of object pose such as orientation and translation.

Using these three measures, we can obtain the optimal set of eigenwindows. See detail criteria in the paper [11].

## 3 Illumination invariance

In our previous work [11], we have shown that the eigenwindow method can successfully recognize and localize an object in b/w input images which contain multiple objects with specularity, even if the input images contain a significant amount of noise, occlusion, image shifting, and scaling change.

However, the method was based on the assumption that the location and brightness of a light source are fixed. Therefore, the method did not take into account shading variation such as highlights on object surfaces. For instance, if an object exhibits specularity, the object appearance can change drastically with different illumination directions, which confounds recognition and localization of the object.

To overcome this limitation, we propose to use an illumination-invariant measure for the eigenwindow method. By using the illumination-invariant measure, the eigenwindow method can be used successfully for recognizing and localizing multiple objects under different illumination conditions.

### *3.1 Illumination invariance: hue*

Instead of black-and-white intensity images, we use RGB color images in the modified eigenwindow method. Actually, a variety of research has been carried out on the color indexing in the past [12–19], but we would like to use the *hue* criterion for its simplicity, and a color image measured in the RGB color space is converted to an HSV image (H: Hue, S: Saturation, V: Value). In these three parameters, the hue parameter is the value which represents color information, e.g., without brightness. Therefore, the hue is not affected by change of the illumination brightness and direction if the following two conditions hold: 1) the light source color can be expected to be almost white, and 2) a saturation value of object color is sufficiently large.

The original color of object $X$ is transferred to be $X' = s \cdot X + t \cdot I$ by the change in diffuse shading and specularity as shown in Fig. 3. $s$ and $t$ represent a relative strength of the diffuse reflection component and the specular reflection component of the color $X'$, respectively. If the two conditions mentioned above are true, then the hue of $X'$ remains the same as that of $X$.

In Fig. 3, object color is represented by three color components $S_1$, $S_2$, and $S_3$. In the RGB color space, those three color components are red, green and blue, for example $(S_1, S_2, S_3) = (R, G, B)$. Then, the light source color $I$ is given as $I = (1, 1, 1)$. To define hue, saturation and intensity, one pair from three components, red, green, and blue,

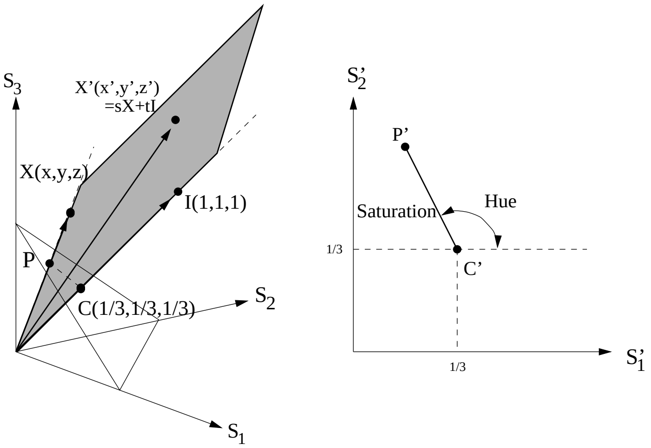

**Fig. 3.** HSV space

have to be assigned to $S'_1$ and $S'_2$. Usually, red and green are assigned as $S'_1 = R$ and $S'_2 = G$.

We conducted a simple experiment using a color test chart to see how hue is affected under different levels of illumination brightnesses. The result is shown in Figs. 4 and 5. In Fig. 5b, we can see that hue remains almost constant over a wide range of illumination brightness for many color blocks, where the hue value from 0 to $2\pi$ is assigned to the unsigned integer value from 0 to 255 to build a hue image.

However, for some color blocks, the value of hue does change with different levels of illumination brightnesses, e.g., the black-white color blocks in the last row of the color chart (color blocks #30-#35), red (color block #12) and magenta (color block #24).

That is because the saturation of color blocks #30-#35 is not sufficiently large, i.e., they are very close to gray. Also, hue has a discontinuity at 0 and $2\pi$. That is the reason for the unstable hue of the color blocks #12 and #24.

To obtain the value of hue reliably, we propose to use three criteria: *intensity value*, *saturation*, and *phase*.

### 3.1.1 Intensity value

To eliminate the background noise, we apply a threshold value for the intensity value as

$$if\ V < V_t\ then\ H = 0, \quad (2)$$

where $V$, $V_t$, and $H$ are an intensity value, the threshold value, and a hue value, respectively. If the measured color is not bright enough, the color is discarded. Then, the hue value is set to a predetermined value, i.e., 0.

### 3.1.2 Saturation

One of the problems shown in the example in Sect. 3.1 is that, if the object color is close to gray, then the hue value of the color is not stable. The reason is that, if the color is almost gray, the object color in $S'_1S'_2$ plane exists around the point $C'$ in Fig. 3. That means the hue angle cannot be determined robustly in face of the image noise. Therefore, the measured color should be discarded if the saturation value is less than a certain threshold $S_t$:

$$if\ S < S_t\ then\ H = 0, \quad (3)$$

where $S$ is the saturation value. Using the equation, measured color close to gray is discarded in the image.

### 3.1.3 Phase

The other problem shown in the example in Sect. 3.1 is that color close to red has a hue value near its discontinuity. The range of the hue value is from 0 to $2\pi$, and it has discontinuity at 0 and $2\pi$. We avoid the discontinuity effect by using the phase threshold value $\Delta P_t$ as

$$if\ H < \Delta P_t\ or\ \|H - 2\pi\| < \Delta P_t\ then\ H = 0. \quad (4)$$

In the examples shown in Figs. 4 and 5, the red color element may be neglected with this criterion.

It is important that the discontinuity of the hue value depends on the selection of the color components $S'_1$ and $S'_2$. In the next section, we discuss how to select the color components $S'_1$ and $S'_2$ to be able to find more windows.

### *3.2 How to select color components $S'_1$ and $S'_2$*

Usually, the two color components $S'_1$ and $S'_2$ are set to $R$ and $G$. But if the $R$ and $G$ factors are used for the two color components, the discontinuity of hue appears around the color red, as described in the previous section. Therefore, if red is the most important component for recognizing the objects, the use of $R$ and $G$ for $S'_1$ and $S'_2$ is not desirable.

In this section, we show how to choose the $S'_1$ and $S'_2$ from RGB components so that we can select more windows to be used as eigenwindows as described in Sect. 2.

There are six combinations for the selection of $S'_1$ and $S'_2$ from the RGB components. Figure 6 shows the result of window selection by using each combination of $S'_1$ and $S'_2$. In the figure, RG represents that $S'_1 = R$ and $S'_2 = G$.

The windows in the hue images were selected by using the corner detector algorithm as described in Sect. 2.3. So, if a hue image does not have enough contrast, fewer windows are selected. In this example, the largest number of windows was selected for the case of $S'_1 = R$ and $S'_2 = B$ in Fig. 6. Intuitively, that result indicates that there are not many green color components in the example image.

Several examples of hue images are shown in Fig. 7. In those examples, we can see that the hue value remains almost constant on the object surface with large shading change. For instance, in Fig. 7c and d, the hue of the yellow duck's surface appears to be constant even though the color image of the duck has a wide range of intensity. In those examples, background, gray color, and green color have been eliminated according to the Eqs. 2, 3 and 4, respectively.

## 4 Experimental results

The proposed method was used for recognition and localization of objects in three test cases. In the first case, the same illumination condition was used for both training and input images. In the second case, input images were taken under different levels of illumination brightnesses. In the last case, input images were taken with different light source locations.

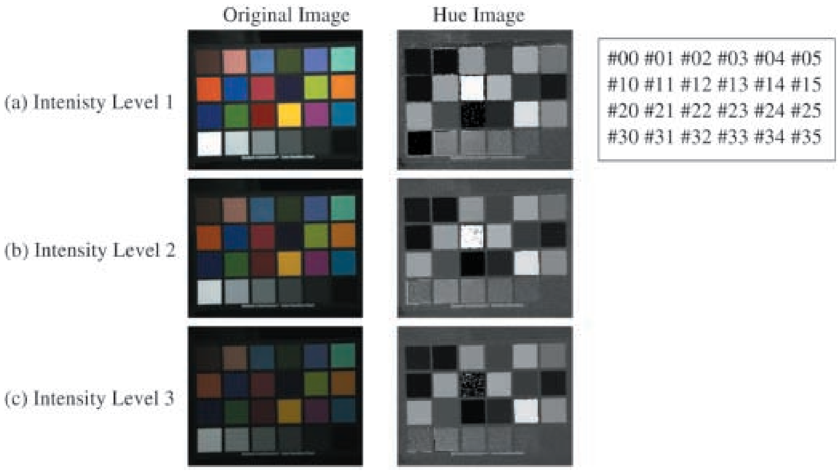

**Fig. 4.** Illumination Constancy with a Color Test Pattern Image.

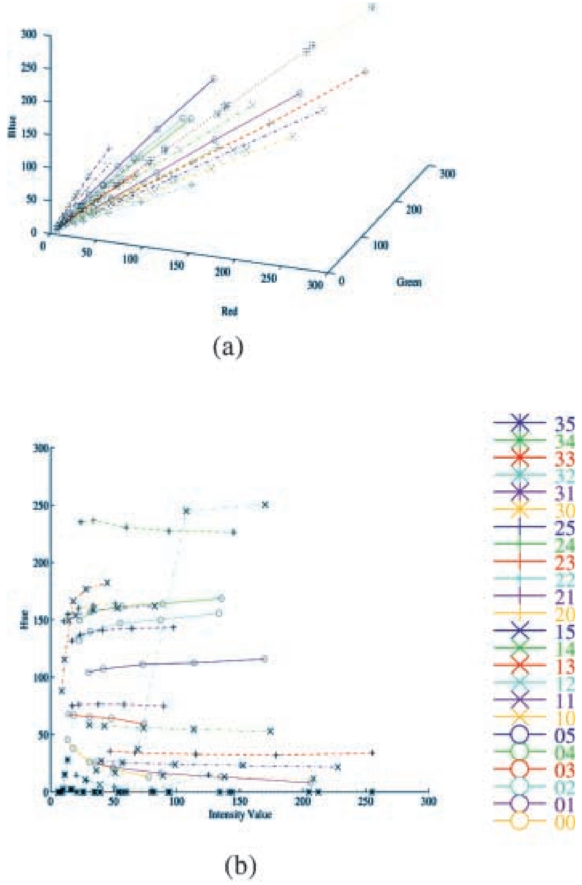

**Fig. 5a,b.** Color elements. **a** RGB space; **b** hue-intensity space

### *4.1 Object recognition and localization with hue image*

First, a set of training eigenwindows was obtained as described in Sect. 2. The training images were taken at $\theta_1 = [-20, 0, 20]$ and $\theta_2 = [0, 10, 20, \cdots, 350]$ for three different objects, *mug*, *bird*, and *tylenol*. We refer to the original images as $type(\theta_1, \theta_2)$. For example, the image $mug(-20, 60)$ denotes the image for the mug taken at the position $\theta_1 = -20deg$ and $\theta_2 = 60deg$. For each of the objects, 108 images were taken by using the experimental setup shown in Fig. 1.

Then, eigenwindows were selected in each training image by using the detectability, similarity and reliability measurements as described in Sect. 2.3. The number of eigenwindows for each of the objects was initially more than 8,000. After the three measurements were applied, less than 2,000 of the training eigenwindows were finally obtained. Then, these eigenwindows were projected to produce eigenpoints according to the Eq. 1.

One input image containing multiple objects was taken as shown in left-hand side of Fig. 8. In the input image, there are seven objects, *duck*, *mug*, *barney*, *bird*, *stop-sign*, *tylenol*, and *tylenol-cold*. First, eigenwindows were selected in the input image by using the detectability measure. Then, we established correspondences between the input eigenwindows and the training eigenwindows by using the similarity between their eigenpoints.

The recognition and localization results are shown in the middle column in Fig. 8. The figures in the right column show the resulting pose spaces. Also, the obtained affine parameters and standard deviations $\delta$ in the pose space are shown by the calculation of $\mathbf{X}(\hat{\mathbf{f}}_k) - \mathbf{X}(\mathbf{g}_k)$. As we can see, each object's type, pose, and location were successfully obtained.

### *4.2 Effect of illumination brightness change*

The same training eigenwindow set was applied to input images taken under a wide range of illumination brightness. Figure 9 shows the result.

The original color images are shown in the left column, and the computed hue images are shown in the middle column. The localization and recognition results are shown in the right column. The affine parameters and standard deviations of the pose space are also given in the figure.

The hue images did not change significantly with different levels of illumination brightness. The main difference between the hue image for the brightest illumination and that for the darkest illumination is that the hue values were not computed over a large portion of the object surfaces. This is because intensity values were so small that hue values were set to zero as the background value according to Eq. 2.

The experimental results show that the proposed method works even when input images are taken with different levels

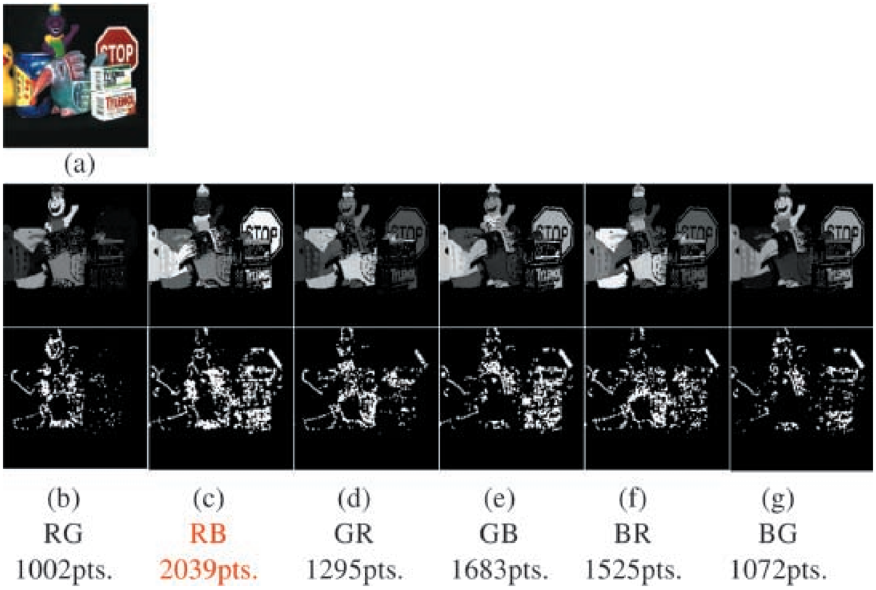

**Fig. 6a–g.** Image invariance and window selection. **a** Original image; **b** hue image and feature points with RG; **c** RB; **d** GR; **e** GB; **f** BR; **g** BG

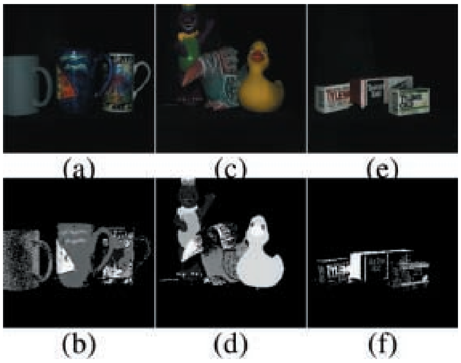

**Fig. 7a–f.** Some example hue images. **a** Original image of mug; **b** hue images of mug; **c**,**d** Birds; **e**,**f** Tylenol

of illumination brightness. The object was recognized and localized successfully.

### 4.3 Effect of different light source positions

The proposed method was also applied to input images taken with different light source locations. As the light source position changes, the appearance of objects in input images changes drastically. Therefore, changing light source position makes recognition and localization of objects even harder than changing illumination brightness.

In this experiment, four different light source positions were used as shown in Fig. 10. The left column images of Fig. 11 show the input images taken with each of the four light source positions. The middle column images show the obtained hue images. The right column images present the recognition and localization results. The affine parameters and standard deviations of pose space are also shown in the figure.

Note that, in this experiment, there was no ambient illumination. Hence, the appearance of the objects changes significantly with different light source positions. Nevertheless, the mug was correctly recognized and localized, except

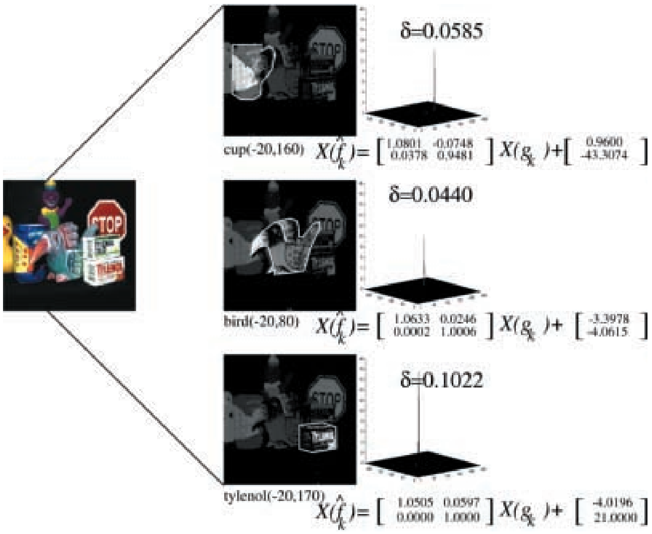
δ=0.0585
cup(-20,160) $X(\hat{\xi}_k)=\begin{bmatrix}1.0801 & -0.0748\\ 0.0378 & 0.9481\end{bmatrix}X(g_k)+\begin{bmatrix}0.9600\\ -43.3074\end{bmatrix}$
δ=0.0440
bird(-20,80) $X(\hat{\xi}_k)=\begin{bmatrix}1.0633 & 0.0246\\ 0.0002 & 1.0006\end{bmatrix}X(g_k)+\begin{bmatrix}-3.3978\\ -4.0615\end{bmatrix}$
δ=0.1022
tylenol(-20,170) $X(\hat{\xi}_k)=\begin{bmatrix}1.0505 & 0.0597\\ 0.0000 & 1.0000\end{bmatrix}X(g_k)+\begin{bmatrix}-4.0196\\ 21.0000\end{bmatrix}$

**Fig. 8.** Recognition result

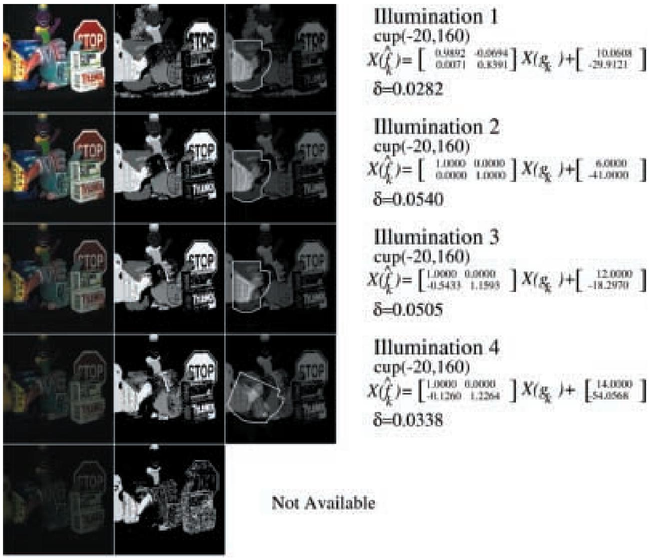
Illumination 1
cup(-20,160)
$X(\hat{\xi}_k)=\begin{bmatrix}0.9892 & -0.0694\\ 0.0071 & 0.8391\end{bmatrix}X(g_k)+\begin{bmatrix}10.0608\\ -29.9121\end{bmatrix}$
δ=0.0282
Illumination 2
cup(-20,160)
$X(\hat{\xi}_k)=\begin{bmatrix}1.0000 & 0.0000\\ 0.0000 & 1.0000\end{bmatrix}X(g_k)+\begin{bmatrix}6.0000\\ -41.0000\end{bmatrix}$
δ=0.0540
Illumination 3
cup(-20,160)
$X(\hat{\xi}_k)=\begin{bmatrix}1.0000 & 0.0000\\ -0.5433 & 1.1593\end{bmatrix}X(g_k)+\begin{bmatrix}12.0000\\ -18.2970\end{bmatrix}$
δ=0.0505
Illumination 4
cup(-20,160)
$X(\hat{\xi}_k)=\begin{bmatrix}1.0000 & 0.0000\\ -0.1260 & 1.2264\end{bmatrix}X(g_k)+\begin{bmatrix}14.0000\\ -54.0568\end{bmatrix}$
δ=0.0338
Not Available

**Fig. 9.** Object recognition results with illumination change

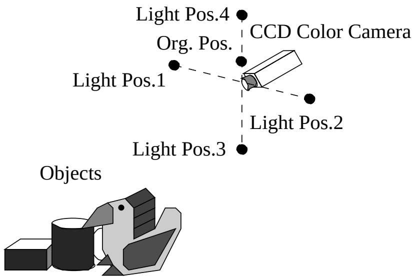

**Fig. 10.** Light source position

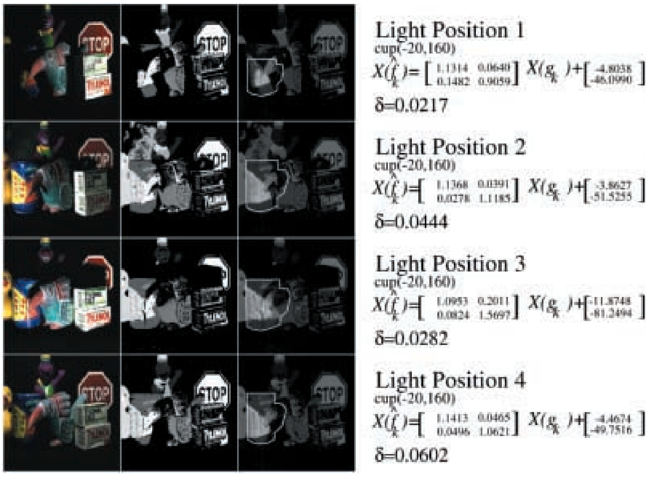
Light Position 1
cup(-20,160)
X(f̂k)=[1.1314 0.0640; 0.1482 0.9059] X(gk)+[-4.8038; -46.0990]
δ=0.0217
Light Position 2
cup(-20,160)
X(f̂k)=[1.1368 0.0391; 0.0278 1.1185] X(gk)+[-3.8627; -51.5255]
δ=0.0444
Light Position 3
cup(-20,160)
X(f̂k)=[1.0953 0.2011; 0.0824 1.5697] X(gk)+[-11.8748; -81.2494]
δ=0.0282
Light Position 4
cup(-20,160)
X(f̂k)=[1.1413 0.0465; 0.0496 1.0621] X(gk)+[-4.4674; -49.7516]
δ=0.0602

**Fig. 11.** Effect of light source direction

in the input image for the light position 1. In this case, hue values were not obtained over a large portion of the object surface because of shadow casting on the surface.

## 5 Conclusion

In this paper, we described a novel method called the eigenwindow method. The method extends the standard eigenspace analysis for the case of recognizing partially occluded objects. To reduce the redundancy among eigenwindows, we proposed three measures for selecting eigenwindows effectively: detectability, uniqueness, and reliability.

By using hue, which is an illumination-invariant measure, the eigenwindow method was extended further for recognition and localization of objects in images taken under changing illumination conditions. To use hue information of input images reliably, we introduced three criteria for computing hue values: intensity value, saturation, and phase.

The proposed method was applied to real images, and the method recognized and localized objects successfully even in images taken under significantly different illumination conditions.

## References

1. Fukada Y, Doi H, Nagamine K, Inari T (1984) Relationships-Based Recognition of Structural Industrial Parts Stacked in Bin. Robotica 2: 147–154
2. Birk JR, Kelly RB, Martines HAS (1981) An Orienting Robot for Feeding Workpieces Stored in Bins. IEEE Trans Syst Man Cybern 11(2): 151–160
3. Bolles RC, Horaud P, Hannah MJ: 3DPO: Three-Dimensional Parts Orientation System. Proc. Int. Joint Conf. on Artificial Intelligence, Karlsruhe, Germany, pp. 1116–1120, 8–12 August
4. Horn BKP, Ikeuchi K (1984) The Mechanical Manipulation of Randomly Oriented Parts. Sci Am 251(2): 100–111
5. Hutchinson SA, Kak AC (1990) SPAR: A Planner that Satisfies Operational and Geometric Goals in Uncertain Environments. AI Mag 11(1): 31–61
6. Agin GJ, Duda OR (1975) SRI Vision Research for Advanced Industrial Automation. Proc. 2nd USA-Japan Computer Conf., pp. 113–117
7. Turk MA, Pentland AP (1991) Face Recognition Using Eigenfaces. Proc. CVPR 1991, pp. 586–591
8. Sclaroff S, Pentland AP (1995) Model Matching for Correspondence and Recognition. IEEE Trans Pattern Anal Mach Intell 17(6): 545–561
9. Kirby M, Sirovich L (1990) Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. IEEE Trans Pattern Anal Mach Intell 12(1): 103–108
10. Murase H, Nayar SK (1995) Visual Learning and Recognition of 3-D Objects from Appearance. Int J Comput Vision 14(1): 5–24
11. Ohba K, Ikeuchi K (1997) Detectability, Uniqueness, and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects. IEEE Trans Pattern Anal Mach Intell 19(9): 1043–1048
12. Bajcsy R, Lee SW, Leonardis A (1996) Detection of Diffuse and Specular Interface Reflections and Inter-Reflections by Color Image Segmentation. Int J Comput Vision 17(3): 241–272
13. Nayar SK, Bolle RM (1996) Reflectance Based Object Recognition. Int J Comput Vision 17(3): 219–240
14. Healey G, Slater D (1994) Global color constancy: recognition of objects by use of illumination-invariant properties of color distribution. J Opt Soc Am 11(11): 3003–3010
15. Swain MJ (1991) Color Indexing. Int J Comput Vision 7( 1): 11–32
16. Funt BV, Finlayson GD (1995) Color Constant Color Indexing. IEEE Trans Pattern Anal Mach Intell 17(5): 522–529
17. Slater D, Healey G (1996) The Illumination-Invariant Recognition of 3D Objects Using Local Color Invariants. IEEE Trans Pattern Anal Mach Intell 18(2): 206–210
18. Novak CL, Shafer SA (1990) Supervised Color Constancy Using a Color Chart. Technical Report, CMU-CS-90-140, School of Computer Science, Carnegie Mellon University, Pittsburgh Pa
19. Aubert D, Thorpe C (1990) Color Image Processing for Navigation: Two Road Trackers. Technical Report, CMU-RI-TR-90-09, The Robotics Institute, Carnegie Mellon University Pittsburgh Pa

**Kohtaro Ohba** is a research scientist at the Mechanical Engineering Laboratory of the Ministry of International Trade and Industries, Ibaragi, Japan. Dr. Ohba received the B.S. degree, the M.S. degree, and the Ph.D. degree in mechanical engineering from the Tohoku University, Japan, in 1986, 1988, and 1991, respectively. After working at Tohoku University, he joined the Mechanical Engineering Laboratory in 1997. From October of 1994 to June of 1996, he worked at the School of Computer Science, Carnegie Mellon University, Pittsburgh, USA. His current research interests include object recognition, visualization, multimedia, and human interface.

**Yoichi Sato** is an assistant professor at the Institute of Industrial Science, University of Tokyo, Tokyo, Japan. He received a BS in Mechanical Engineering from the University of Tokyo. He received an MS in Robotics in 1993 and a PhD in Robotics from the School of Computer Science, Carnegie Mellon University, in 1997. The primary areas of his research interests are computer vision (physics-based computer vision, reflectance analysis for 3D object model generation), computer graphics (virtual reality and augmented reality) and human-computer interaction.

**Katsushi Ikeuchi** is a Professor at the Institute of Industrial Science, the University of Tokyo, Tokyo, Japan. Dr. Ikeuchi received the B. Eng. degree in Mechanical Engineering from Kyoto University, Kyoto, Japan, in 1973, and the Ph.D. degree in Information Engineering from the University of Tokyo, Tokyo, Japan, in 1978. After working at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology, Electro-technical Laboratory of Ministry of International Trade and Industries, and School of Computer Science, Carnegie Mellon University, he joined the University of Tokyo in 1996.