Fast Template Matching Based on the Normalized Correlation by Using Multiresolution Eigenimages

Shinichi Yoshimura
Sony Corporation
Tokyo, Japan

Takeo Kanade
Carnegie Mellon University
Pittsburgh, PA, U.S.A.

Abstract

We present a fast computation method of the normalized correlation for multiple rotated templates by using multiresolution eigenimages. This method allows us to accurately detect both location and orientation of the object in a scene at faster rate than applying conventional template matching to the rotated object.

Since the correlation among slightly rotated templates is high, we first apply the Karhunen-Loéve expansion to a set of rotated templates and extract "eigenimages" from them. Each template in this set can be approximated by a linear combination of these eigenimages and it substitute for the template in computing the normalized correlation. The number of eigenimages is smaller than that of original templates and computation cost becomes small.

Second, we employ a multiresolution image structure to reduce the number of rotated templates and location search area. For the lower resolution image, the position and angle are coarsely obtained in a wide region. Then not only searching area for the position but also the range of rotation angle of templates at the next layer can be limited to the neighbor of the prior results. The precise position and angle detection is possible to repeat this strategy.

We implemented the proposed algorithm on vision system and realized computation time around 600 msec and achieved sub pixel resolution for translation and 0.3 degree maximum error for 360 degree rotation on the 512 by 480 gray scale image. Experimental results are shown to demonstrate the accuracy, efficiency and feasibility of the proposed method.

1 Introduction

We will deal with the problem of fast positioning and angle detection of the object in the two-dimensional image. This is one of the major issues of application of image processing to industrial field such as assembly, adjustment, and inspection stage in the manufacturing facilities.

For the two-dimensional image, template matching is a useful tool to detect the object or its feature [1]. Given a template which describe the object or feature to be detected, the goal is to find the location where this template and input image match best. One of the measures of the degree of match is the normalized correlation [2, 1]. Its advantage is an insensitivity to the change of intensity or gray level of the image. However, pixel by pixel multiplications of the size of template is required over the region where the object of interest may exist.

What will happen if the object is rotated in a scene? A single template does no longer work as a standard pattern and we need rotated templates. The difficulty of applying the normalized correlation to the rotated object search lies on its highly expensive computation cost. Its cost depends on the number and the size of templates. The more templates we have to detect the rotation angle precisely in wide range, the more computation is required.

Several literatures have addressed to tackle the reduction of computation in template matching problem. Many of past studies, however, have focused on a single template matching problem (non-rotated object) [3, 4, 16].

We can take advantage of the fact that rotated templates differ slightly each other and they are highly correlated. Therefore, these templates are effectively replaced with a small number of images ("eigenimages") by the Karhunen-Loéve (K-L) expansion.

The K-L expansion is a popular scheme as a data compression technique [5, 6]. Applications of the K-L expansion to computer vision are found mainly in pattern recognition, especially face recognition problems [7, 8, 9]. Further discussion about visual learning and recognition has been introduced in [10]. It addressed a construction of parametric eigenspace for learning, recognition, and pose estimation of the objects. The focus of the above studies lies on a parameter space which is built from the coefficients obtained by projection of face image onto eigenvector space.

We employ K-L expansion to replace original templates by a linear combination of eigenimages then reduce the number of templates to compute the normalized correlation.

Further reduction of computation cost is realized by creating a multiresolution image structure or "image pyramid". Feature extraction such as edge detection, and object recognition by using image pyramid have been reported [11, 12, 13]. They are mainly dealing with only spatial resolution of image, while we consider the hierarchy of rotation angle of templates beside resolution to reduce the range of angle.

At the lower resolution layer, wider range of rotation angle is needed to roughly obtain the orientation of the object. The required range of rotation angle at the next resolution layer is just the neighbor of this detected angle, thus only limited number of templates are used to compute the normalized correlation. The higher resolution of image is, the narrower the rotation angle of template can be. The search area for the location is also limited with the results of the lower layer and computation cost is saved more.

In this paper we will focus on the reduction of the number of rotated templates and the range of rotation angle of those templates by using hierarchical eigenimage structure.

Our method is suitable to the industrial application such as positioning and angle detection of components on the assembly line or inspection and adjustment stage in the manufacturing facilities. The target object may be placed randomly and prediction of its location is difficult.

2 Normalized Correlation by Use of Eigenimages

Suppose we compute the normalized correlation of P templates of size N at M searching points over the input scene. The computation cost is proportional to $NM(P+1)$.

Since slightly rotated images are similar each other and they are highly correlated, they can be represented as a linear combination of certain images in a different coordinate system. These images are defined as "eigenimages" obtained by Karhunen-Loéve expansion. The number of eigenimages in a linear combination determines the degree of approximation of the original templates. We notice that truncation of the eigenimages does not effect the degree of approximation so much.

We replace templates with their approximation form to compute the normalized correlation. The reduction of computation cost is from $NM(P+1)$ to $NM(K+2)$, where K represents the number of eigenimages. The main contribution of the proposed method is to reduce the number of templates to compute the normalized correlation and its reduction rate is $(K+2)/(P+1)$. Therefore, if K is smaller enough than P, the cost is saved in a great deal.

2.1 Eigenimages of Templates

Each one in a set of templates can be approximated as a linear combination of their eigenimages obtained by applying the Karhunen-Loéve expansion. Those eigenimages are basis vectors which span subspace of the original template space. We will regard a two-dimensional image as a vector.

Let $\mathbf{x}_i$, $i = 1, 2, \ldots, P$ be P templates of size N with different rotation angle. First, these templates are normalized in the sense that the average intensity of whole pixels is zero. Normalized templates $\tilde{\mathbf{x}}_i$, $i = 1, 2, \ldots, P$ are described as

$$\tilde{\mathbf{x}}_i = [\tilde{x}_{i1}, \tilde{x}_{i2}, \ldots, \tilde{x}_{iN}]^T. \tag{1}$$

The average of templates

$$\left.\begin{aligned} \mathbf{c} &= [c_1, c_2, \ldots, c_N]^T \\ c_j &= \tfrac{1}{P}\textstyle\sum_{i=1}^{P} \tilde{x}_{ij},\ j = 1, 2, \ldots, N \end{aligned}\right\} \tag{2}$$

is subtracted from each normalized template and the covariance matrix of these templates is defined as

$$\mathbf{A} = \frac{1}{P}\sum_{i=1}^{P}(\tilde{\mathbf{x}}_i - \mathbf{c})(\tilde{\mathbf{x}}_i - \mathbf{c})^T. \tag{3}$$

This covariance matrix has a size of $N \times N$ and requires very expensive computation cost to obtain its eigenvalues and eigenvectors. Our interest is limited only in the major eigenvectors which correspond to the large eigenvalues. Since the number of linearly independent eigenvectors of covariance matrix $\mathbf{A}$ is at most $P-1$, we solve eigenvalue problem [14]

$$\tilde{\mathbf{A}}\tilde{\mathbf{e}}_j = \tilde{\lambda}_j \tilde{\mathbf{e}}_j \tag{4}$$

instead of

$$\mathbf{A}\mathbf{e}_j = \lambda_j \mathbf{e}_j. \tag{5}$$

In (4), i, j-th component of matrix $\tilde{\mathbf{A}}$ is given as

$$\tilde{A}_{ij} = \frac{1}{P}(\tilde{\mathbf{x}}_i - \mathbf{c})^T(\tilde{\mathbf{x}}_j - \mathbf{c}),\ i, j = 1, 2, \ldots, P \tag{6}$$

and the size of $\tilde{\mathbf{A}}$ is $P \times P$. Eigenvalues are obtained in order of magnitude, i.e.,

$$\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_{P-1}. \tag{7}$$

The $P-1$ major eigenvalues of covariance matrix $\mathbf{A}$ are identical to those of matrix $\tilde{\mathbf{A}}$,

$$\lambda_j = \tilde{\lambda}_j,\ j = 1, 2, \ldots, P-1, \tag{8}$$

and their associative eigenvectors $\mathbf{e}_j$ of covariance matrix $\mathbf{A}$ can be obtained from the eigenvectors $\tilde{\mathbf{e}}_j$,

$$\tilde{\mathbf{e}}_j = [\tilde{e}_{1j}, \tilde{e}_{2j}, \ldots, \tilde{e}_{Pj}]^T \tag{9}$$

as

$$\mathbf{e}_j = \frac{1}{\sqrt{\tilde{\lambda}_j P}} \sum_{i=1}^{P} \tilde{e}_{ij}(\tilde{\mathbf{x}}_i - \mathbf{c}),\ j = 1, 2, \ldots, P-1. \tag{10}$$

Since eigenvectors $\tilde{\mathbf{e}}_j$ are chosen to be orthonormal, eigenvectors $\mathbf{e}_j$ are also orthonormal,

$$\mathbf{e}_i^T \mathbf{e}_j = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}. \tag{11}$$

Here we call an eigenvector $\mathbf{e}_j$ as an eigenimage.

2.2 Approximation of Templates by Eigenimages

A template $\tilde{\mathbf{x}}_i$ can be represented as a linear combination of eigenimages $\mathbf{e}_j$, $j = 1, 2, \ldots, P-1$ with average image vector $\mathbf{c}$ as

$$\tilde{\mathbf{x}}_i = \mathbf{c} + \sum_{j=1}^{P-1} p_{ij}\mathbf{e}_j \tag{12}$$

where coefficients $\mathbf{p}_i^{(P-1)} = [p_{i1}, p_{i2}, \ldots, p_{iP-1}]^T$ are given by

$$\mathbf{p}_i^{(P-1)} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{P-1}]^T(\tilde{\mathbf{x}}_i - \mathbf{c}). \tag{13}$$

If only $K(\leq P-1)$ eigenimages are used, $\tilde{\mathbf{x}}_i$ is approximated by a linear combination of K eigenimages as

$$\tilde{\mathbf{x}}_i \approx \tilde{\mathbf{x}}_i^{(K)} \tag{14}$$

where

$$\tilde{\mathbf{x}}_i^{(K)} \triangleq \mathbf{c} + \sum_{j=1}^{K} p_{ij}\mathbf{e}_j. \tag{15}$$

The degree of approximation is measured as follows. Error between normalised template $\tilde{\mathbf{x}}_i$ and its approximation $\tilde{\mathbf{x}}_i^{(K)}$ is

$$\begin{aligned} \mathbf{v}_i^{(K)} &= \tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^{(K)} \\ &= \sum_{j=K+1}^{P-1} p_{ij}\mathbf{e}_j. \end{aligned} \tag{16}$$

The expectation of square error of $\mathbf{v}_i$ over i becomes

$$\begin{aligned} \epsilon^{(K)} &= E(\|\mathbf{v}_i^{(K)}\|^2) \\ &= \sum_{j=K+1}^{P-1} E(p_{ij}^2). \end{aligned} \tag{17}$$

From the definition of parameters p_{ij} (13), $\epsilon^{(K)}$ is now

$$\begin{aligned} \epsilon^{(K)} &= \sum_{j=K+1}^{P-1} E(\mathbf{e}_j^T(\tilde{\mathbf{x}}_i - \mathbf{c})(\tilde{\mathbf{x}}_i - \mathbf{c})^T\mathbf{e}_j) \\ &= \sum_{j=K+1}^{P-1} \lambda_j. \end{aligned} \tag{18}$$

Since eigenimages $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K, \ldots, \mathbf{e}_{P-1}$ are selected in order of magnitude of eigenvalues, square error $\epsilon_{(K)}$ is a summation of only small eigenvalues $\lambda_{K+1}, \ldots, \lambda_{P-1}$, then $\epsilon^{(K)}$ is also small.

We use a cumulative proportion

$$\mu^{(K)} = \frac{\sum_{j=1}^{K} \lambda_j}{\sum_{j=1}^{P-1} \lambda_j} \tag{19}$$

as a measurement for degree of approximation by a linear combination form of eigenimages. This measurement shows how many eigenimages contribute to approximate template well in the sense of minimum mean square error between original template and its linear combination form.

2.3 Normalized Correlation with Eigenimage Approximation

The normalized correlation between template $\tilde{\mathbf{x}}_i$ and a portion of input image $\tilde{\mathbf{y}}$ is given as

$$C(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}) = \frac{\tilde{\mathbf{x}}_i^T \tilde{\mathbf{y}}}{\|\tilde{\mathbf{x}}_i\| \, \|\tilde{\mathbf{y}}\|} \tag{20}$$

where $\tilde{\mathbf{y}}$ has the same size of template $\tilde{\mathbf{x}}_i$.

The main issue of our new template matching method is a replacement of template $\tilde{\mathbf{x}}_i$ with its approximation or linear combination form of eigenimages. First we replace the template by its complete linear combination form. From (12) and (20),

$$\begin{aligned} C(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}) &= C(\tilde{\mathbf{x}}_i^{(P-1)}, \tilde{\mathbf{y}}) \\ &= \frac{(\mathbf{c} + \sum_{j=1}^{P-1} p_{ij}\mathbf{e}_j)^T \tilde{\mathbf{y}}}{\|\tilde{\mathbf{x}}_i^{(P-1)}\| \, \|\tilde{\mathbf{y}}\|} \\ &= \frac{\mathbf{c}^T\tilde{\mathbf{y}} + \sum_{j=1}^{P-1} p_{ij}(\mathbf{e}_j^T \tilde{\mathbf{y}})}{\|\tilde{\mathbf{x}}_i^{(P-1)}\| \, \|\tilde{\mathbf{y}}\|}. \end{aligned} \tag{21}$$

By rewriting

$$\begin{aligned} \mathbf{q}^{(K)} &= [q_1, q_2, \ldots, q_K]^T \\ &= [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K]^T \tilde{\mathbf{y}}, \end{aligned} \tag{22}$$

we have

$$C(\tilde{\mathbf{x}}_i^{(P-1)}, \tilde{\mathbf{y}}) = \frac{\mathbf{c}^T\tilde{\mathbf{y}} + \sum_{j=1}^{P-1} p_{ij}q_j}{\|\tilde{\mathbf{x}}_i^{(P-1)}\| \, \|\tilde{\mathbf{y}}\|} \tag{23}$$

$$= \frac{\mathbf{c}^T\tilde{\mathbf{y}} + \mathbf{p}_i^{(P-1)^T}\mathbf{q}^{(P-1)}}{\|\tilde{\mathbf{x}}_i^{(P-1)}\| \, \|\tilde{\mathbf{y}}\|}.$$

Here we introduce the new measurement

$$D(\tilde{\mathbf{x}}_i^{(K)}, \tilde{\mathbf{y}}) = \frac{\mathbf{c}^T\tilde{\mathbf{y}} + \mathbf{p}_i^{(K)^T}\mathbf{q}^{(K)}}{\|\tilde{\mathbf{x}}_i^{(K)}\| \, \|\tilde{\mathbf{y}}\|} \tag{24}$$

as an approximation of the normalised correlation (20).

If $K = P - 1$, i.e., no eigenimages are truncated, $D(\tilde{\mathbf{x}}_i^{(K)}, \tilde{\mathbf{y}})$ is identical to $C(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}})$ and if $K < P - 1$ then $D(\tilde{\mathbf{x}}_i^{(K)}, \tilde{\mathbf{y}})$ is an approximation of $C(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}})$.

2.4 Comparison of Computation Cost

We will compare the computation cost of the normalized correlation $C(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}})$ and its approximation $D(\tilde{\mathbf{x}}_i^{(K)}, \tilde{\mathbf{y}})$. Let N and P be the total number of pixels in a template and the total number of templates, respectively. K denotes the number of selected major eigenimages. In general, we have $N \gg P > K$. We assume that the average of images $\mathbf{c}$, coefficients of template $\mathbf{p}_i^{(K)}$, and the norm of vector $\|\tilde{\mathbf{x}}_i^{(K)}\|$ are calculated in advance. At single searching point in the input image, the following computation is required to obtain the normalised correlation.

1. $C(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}})$
   - $\|\tilde{\mathbf{y}}\|$: N multiplications
   - $\tilde{\mathbf{x}}_i^T\tilde{\mathbf{y}}$: N multiplications

   Since the latter must be computed for each template, total computation cost becomes

$$N + NP = N(P + 1). \tag{25}$$

2. $D(\tilde{\mathbf{x}}_i^{(K)}, \tilde{\mathbf{y}})$
   - $\|\tilde{\mathbf{y}}\|$: N multiplications
   - $\mathbf{c}^T\tilde{\mathbf{y}}$: N multiplications
   - $\mathbf{q}^{(K)} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K]^T\tilde{\mathbf{y}}$: NK multiplications
   - $\mathbf{p}_i^{(K)^T}\mathbf{q}^{(K)}$: K multiplications

   The last term is calculated for every templates and cost is

$$2N + NK + KP \approx N(K + 2). \tag{26}$$

Therefore, the degree of reduction for computing the normalised correlation from original P rotated templates to extracted K eigenimages can be represented as

$$\frac{K+2}{P+1}. \tag{27}$$

3 Hierarchical Matching with Multiresolution Eigenimage

The more precise and the wider rotation angle is required to be detected, the more templates with slight rotation angle are needed. We build an image pyramid structure with respect not only to image size and searching area but also to the number of rotated templates to reduce computation cost and obtain precise result.

For lower resolution image, the searching area to compute the normalised correlation is wider but coarse. As resolution becomes higher, the search area is limited and "dense" using the result of prior layer.

The normalized correlation is insensitive to small difference of rotation angle at the lower resolution image layer. Its sensitivity becomes larger and larger as the resolution is higher. Once the rotation angle of the object is detected roughly in the lower layer, only a small number of templates which have angle around the detected one are needed in the next higher layer. The step size of angle is smaller as the resolution becomes higher, then the more precise angle detection is possible. The above scheme reduces the range of rotation angle or the number of templates and even smaller eigenimages are applied to compute the normalised correlation.

3.1 Algorithm

Now we will show a hierarchical structure of searching area and rotation angle of templates. We assume to have $L+1$ layer image pyramid structure. The lowest resolution layer is $(L + 1)$th and the highest one is 0th layer. This multi-resolution image is constructed by averaging over $n \times n$ region or Gaussian image pyramid [15].

We have P_ℓ templates with rotation angle ranging from $-\Phi_\ell$ to $+\Phi_\ell$ at every ϕ_ℓ degrees at the ℓ th layer and obtain K_ℓ eigenimages. These eigenimages correspond to the K_ℓ largest eigenvalues of covariance matrix of these P_ℓ templates. Searching of the maximum normalised correlation is performed over M_ℓ points of this layer. It is not necessarily done at every point but every 2 or 3 points might be enough. Let θ_ℓ be the detected angle which gives the maximum normalized correlation at the position (x_ℓ, y_ℓ) in this layer.

Searching area at the next layer can be limited to the neighbor of the corresponding point of (x_ℓ, y_ℓ), e.g., $(2x_\ell \pm w_{\ell-1}, 2y_\ell \pm h_{\ell-1})$ where $w_{\ell-1}$ and $h_{\ell-1}$ are ranges of search at this $(\ell - 1)$th layer along x-direction and y-direction, respectively.

Rotation angle of templates to compute eigenimages at the $(\ell - 1)$th layer should also be the neighbor of θ_ℓ. Thus $P_{\ell-1}$ templates with rotation angle from $\theta_\ell - \Phi_{\ell-1}$ to $\theta_\ell +$

$\Phi_{\ell-1}$ are selected. The step of angle $\phi_{\ell-1}$ must be smaller than ϕ_ℓ to detect more precise angle.

Continuing the above procedure until reaching the bottom (0th) layer of image pyramid, we obtain the final result of position and rotation angle of the object accurately and precisely.

We need, however, a large amount of memories to store eigenimages in the higher resolution layer. To save memories we fix the range of rotation angle of templates in each layer except the lowest resolution one and rotate the input image by the detected angle.

Let (X_ℓ, Y_ℓ), Θ_ℓ be the detected position and angle at the ℓth layer, input image at the $(\ell-1)$th layer is rotated by $-\Theta_\ell$ centered at $(2X_\ell, 2Y_\ell)$. Suppose that $(x_{\ell-1}, y_{\ell-1})$ and $\theta_{\ell-1}$ are results at the $(\ell-1)$th layer, the location $(X_{\ell-1}, Y_{\ell-1})$ and angle $\Theta_{\ell-1}$ in the original (before rotating) input image are

$$\begin{bmatrix} X_{\ell-1} - 2X_\ell \\ Y_{\ell-1} - 2Y_\ell \end{bmatrix} = \begin{bmatrix} \cos\Theta_\ell & -\sin\Theta_\ell \\ \sin\Theta_\ell & \cos\Theta_\ell \end{bmatrix} \begin{bmatrix} x_{\ell-1} - 2X_\ell \\ y_{\ell-1} - 2Y_\ell \end{bmatrix}, \tag{28}$$

$$\Theta_{\ell-1} = \theta_{\ell-1} + \Theta_\ell. \tag{29}$$

Then, the input image at the $(\ell-2)$th layer is rotated by $-\Theta_{\ell-1}$ centered at $(2X_{\ell-1}, 2Y_{\ell-1})$. By continuing the above, the final results at the 0th layer, (X_0, Y_0) and Θ_0, are obtained as

$$\begin{bmatrix} X_0 - 2X_1 \\ Y_0 - 2Y_1 \end{bmatrix} = \begin{bmatrix} \cos\Theta_1 & -\sin\Theta_1 \\ \sin\Theta_1 & \cos\Theta_1 \end{bmatrix} \begin{bmatrix} x_0 - 2X_1 \\ y_0 - 2Y_1 \end{bmatrix}, \tag{30}$$

$$\Theta_0 = \theta_0 + \Theta_1. \tag{31}$$

3.2 Interpolation of Peak Position and Angle

The resolution of the detected position is limited to the location of pixel in a scene and the rotation angle is determined by the difference of angles between neighboring templates. We obtain both sub-pixel and "sub-angle" resolution simultaneously by applying a quadratic fitting function to correlation data within a 3 × 3 neighboring pixel region in the sense of least mean squares.

Using correlation data $g(r, c, \theta)$ over 3 × 3 region for 3 different rotation angle templates (total 27 point data), the quadratic fitting function

$$\begin{aligned} f(r,c,\theta) &= a_0 + a_1 r + a_2 c + a_3\theta + a_4 r^2 \\ &+ a_5 c^2 + a_6\theta^2 + a_7 rc + a_8 c\theta + a_9 r\theta \end{aligned} \tag{32}$$

is determined to minimize

$$E = \sum_{r,c,\theta=-1,0,1} [g(r,c,\theta) - f(r,c,\theta)]^2. \tag{33}$$

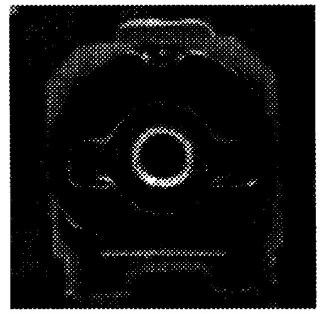

Figure 1: The target object "volume" on a print circuit board (128 × 128 pixels).

In above equations, c and r are x and y position in an image, respectively and θ corresponds to the rotation angle of template. Those variables are normalized to have value −1, 0, and 1. Coefficients $\mathbf{a} = [a_0, a_1, \ldots, a_9]^T$ which minimize the above energy function E in the least squares sense are obtained by solving

$$\frac{\partial E}{\partial \mathbf{a}} = \mathbf{0}. \tag{34}$$

The sub-pixel location (c_s, r_s) and sub-angle resolution θ_s are given as

$$\begin{bmatrix} r_s \\ c_s \\ \theta_s \end{bmatrix} = \begin{bmatrix} 2a_4 & a_7 & a_9 \\ a_7 & 2a_5 & a_8 \\ a_9 & a_8 & 2a_6 \end{bmatrix}^{-1} \begin{bmatrix} -a_1 \\ -a_2 \\ -a_3 \end{bmatrix}. \tag{35}$$

4 Experiments

We have conducted several experiments to verify the proposed method. Accuracy and linearity are shown first, then computation time is measured for a sample object.

The sample object to be detected is a small "volume" on the print circuit board as shown in Figure 1.

4.1 Multiresolution Eigenimages

We made a 4 layer multiresolution image structure (image pyramid) for templates by 5 × 5 Gaussian pyramid algorithm. Template size is 128 × 128 pixels, 62 × 62 pixels, 29 × 29 pixels, and 13 × 13 pixels for the 0th, 1st, 2nd, and 3rd layer, respectively.

The number and the range of rotation angle of templates are shown in Table 1.

Eigenimages in each layer are computed from a set of templates in that layer. The number of eigenimages varies layer to layer and it depends on the feature of the original template. For example, Figure 2 illustrates 5 rotated templates in the 2nd layer and Figure 3 shows the cumulative

Table 1: The number (P) and the range (θ) of rotation angle of the templates for experiment.

Layer	0th	1st	2nd	3rd
P	5	5	5	91
θ	$\pm 1.0°$	$\pm 2.0°$	$\pm 4.0°$	$\pm 180°$

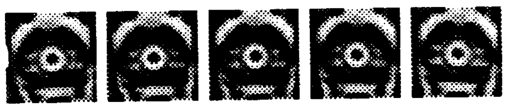

Figure 2: 5 rotated templates in the 2nd layer (29 $\times$ 29 pixels) : 2°, 1°, 0°, −1°, −2° (from left to right).

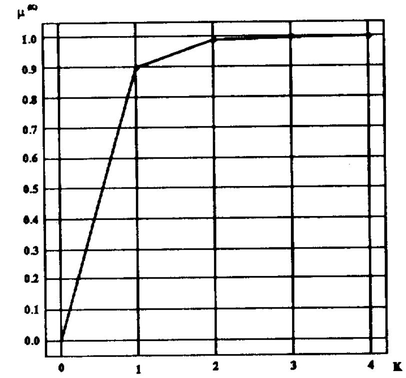

Figure 3: Cumulative proportion $\mu^{(K)}$ in the 2nd layer.

Figure 4: An average of 5 templates (left) and the chosen eigenimage (right) in the 2nd layer.

proportion as defined in (19). We set a threshold level for approximation as 0.7 for this layer, thus only one eigenimage is used to approximate the original 5 templates. Figure 4 shows the average of 5 templates and the chosen eigenimage which corresponds to the largest eigenvalue.

We demonstrate that small number of eigenimages are enough to approximate the original templates by a linear combination of eigenimages and average template even though the number of original templates is very large. Figure 5 shows the cumulative proportion in the 3rd layer for the different number of original templates. Case 1 is 91 templates ($-180° \sim +180°$, every 4 degrees), case 2 is 47 templates ($-92° \sim +92°$, every 4 degrees), and case 3 is 25 templates ($-48° \sim +48°$, every 4 degrees).

Suppose that we chose the threshold level 0.75 for the degree of approximation, the number of eigenimages become 6, 4, and 2 for each case. The ratio K/P is 0.067, 0.085, and 0.08 for P=91, P=47, and P=25, respectively. Thus, a large number of eigenimages are not required to approximate the original templates which cover wider range of rotation angle fot this sample target.

4.2 Accuracy and Linearity

The target object ("volume") is mounted on the small print circuit board and that board is placed on the translation and rotation table, which has a resolution of 0.5mm for two dimensional translation and 5 minutes (5/60 degrees) for rotation.

Input image grabbed by CCD camera (Sony XC-75) is also converted to a 4 layer Gaussian pyramid. The size of input image is 512 $\times$ 480 pixels for the 0th (original resolution) layer of pyramid, 254 $\times$ 238 for the 1st layer, 125 $\times$ 117 pixels for the 2nd layer, and 61 $\times$ 57 pixels for the 3rd (the lowest resolution) layer.

Figure 6 illustrates an example of input image. The view area of this picture is approximately 16mm wide and 15mm high. The target object is located at bottom right.

The accuracy of angle detection for the proposed method is shown in Figure 7. The rotation angle is from -150 degrees to +180 degrees at every 30 degrees. The angle of the target is detected 10 times for each rotation and the average of those 10 results are plotted. Rotated templates are created by applying affine transform to 0 degree template and the real input image is realized by rotating the table. Therefore, the target object in a real scene is different from rotated templates and we have error shown in Figure 7. The error depends on the rotation angle and the maximum error is around 0.3 degrees.

We examine the linearity for both horizontal and vertical translation of the target next. Figure 8 illustrates the linearity. Translation width is 8.5mm for horizontal and 7.5mm for vertical shift and its interval is 0.5mm. Note that the frame of reference of the camera and the real world is not parallel each other, then horizontal translation of the target has y direction translation in the input image. According to these data, the linearity of two dimension is very good for the proposed method.

Resolution of angle detection to small rotation is also measured. A table is rotated by every 5 minutes (1/12 degrees) and angle is detected. Rotational center is set to around center and boundary area of the camera axis. The range of rotation angle is 44 degrees through 46 degrees. Figure 9 depicts the results for center area and boundary area. Note that sensitivity depends on the location of the

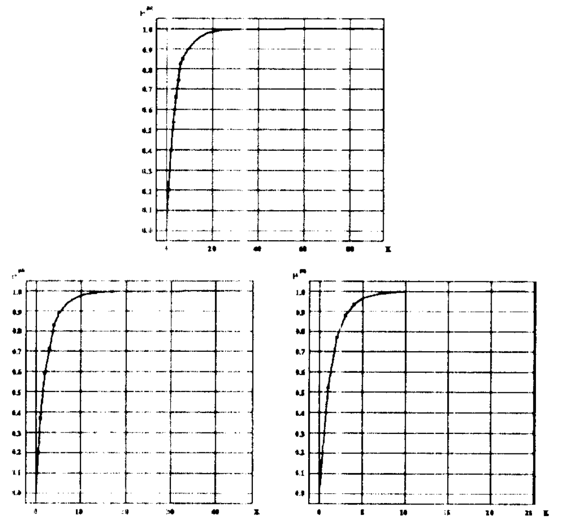

Figure 5: Cumulative proportion $\mu^{(K)}$ in the 3rd layer : P=91 (top), P=47 (left), and P=25 (right).

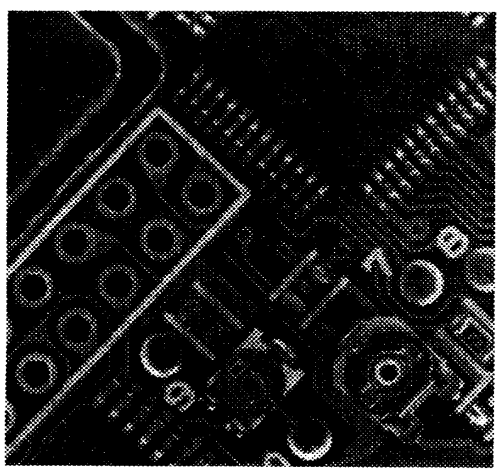

Figure 6: An example of input image (512 × 480 pixels).

target. One of the reasons is the difference of view angle from camera into the target object. The farther the object is from the center of camera axis, the more distortion is observed.

4.3 Computation Time

All experiments are realized on our machine vision system "SVS-200". It has originally designed parallel processing VLSI's (×8, clock frequency 25MHz) which con-

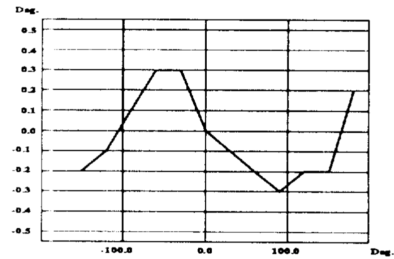

Figure 7: Accuracy of angle detection.

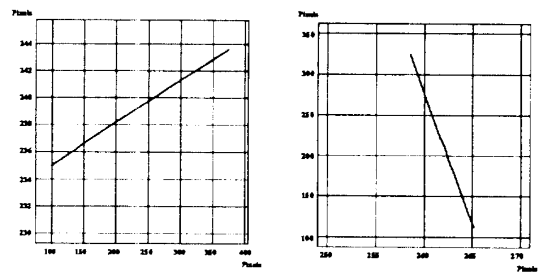

Figure 8: Linearity of two dimensional translation : Horizontal (left) and vertical (right) translation.

tains 4 processor elements each, DSP (digital signal processor) TMS320C31 (32MHz) as a main processor, and MC68EC030 (32MHz) as an I/O processor. The number of frame memories is 16 as standard, 28 as maximum. 3 camera inputs and 3 monitor outputs are standard configuration. Ethernet is available for communication with host computer and standard serial, parallel ports are also available.

Eigenimages and average of templates are stored in the host computer and they are loaded to frame memories of the SVS-200 before execution.

Table 2: Computation time for 4 layer multiresolution image.

Layer	P	K	N	M	Time
Creating 4 layer pyramid					31 msec
3rd	91	6	13^2	575	460 msec
2nd	5	1	29^2	25	28 msec
1st	5	1	62^2	25	50 msec
0th	5	1	128^2	9	35 msec
Total time					604 msec

Computation time is measured for each step and shown in Table 2. This includes the time for rotating the original image in each layer as explained in 3.1 and it is about 5 ~ 10msec. Experimental parameters are also shown in this

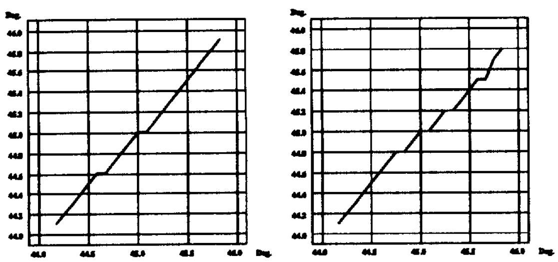

Figure 9: Resolution of angle detection : Rotation near center of image (left) and boundary area (right).

table. "N" represents the size of template and "M" is the number of searching points in each layer.

Total computation time is approximately 600 msec and it is not fast enough to the application which requires real time operation. However, reduction of the range of rotation angle or limitation of searching area will provide faster computing.

5 Conclusion

We have presented a fast template matching method using hierarchical eigenimages structure. It enables us to compute efficiently the normalized correlation as a measure of matching between input image and rotated templates.

Experimental results illustrate the accuracy of detection was a tenth pixels with respect to the position and approximately 0.3 degrees in rotation angle of the object.

Computation time on the real hardware system is around 600 msec for 4 layer multiresolution image structure.

Acknowledgements

The authors would like to thank the members of the Vision and Autonomous Systems Center of Carnegie Mellon University for their valuable comments and suggestions.

References

[1] D. H. Ballard and C. M. Brown, *Computer Vision*. New Jersey: Prentice-Hall, 1982.

[2] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic Press, 1976.

[3] G. J. VanderBrug and A. Rosenfeld, "Two-stage template matching," *IEEE Trans. Comput.*, vol. C-26, no. 4, pp. 384–393, Apr. 1977.

[4] A. Rosenfeld and G. J. VanderBrug, "Coarse-fine template matching," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-7, no. 2, pp. 104–107, Feb. 1977.

[5] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Boston: Academic Press, second ed., 1990.

[6] E. Oja, *Subspace Methods of Pattern Recogntion*. Hertfordshire, England: Research Studies Press, 1983.

[7] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer. A*, vol. 4, no. 3, pp. 519–524, Mar. 1987.

[8] M. Kirby and L. Sirovich, "Application of the karhunen-loéve procedure for the characterization of humen faces," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 1, pp. 103–108, Jan. 1990.

[9] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, (Hawaii), Jun. 1991, pp. 586–591.

[10] H. Murase and S. K. Nayar, "Parametric eigenspace representation for visual learning and recognition," Tech. Rep. CUCS-054-92, Dept. of Comput. Sci., Columbia University, New York, NY, Dec. 1992.

[11] S. L. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing," *Comput. Graphics Images Processing*, vol. 4, no. 2, pp. 104–119, Jun. 1975.

[12] S. L. Tanimoto, "Template matching in pyramids," *Comput. Graphics Images Processing*, vol. 16, no. 4, pp. 356–369, Aug. 1981.

[13] P. J. Burt, "Smart sensing with a pyramid vision machine," *Proc. IEEE*, vol. 76, no. 8, pp. 1006–1015, Aug. 1989.

[14] H. Murakami and B. V. Kumar, "Efficient calculation of primary images from a set of images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-4, no. 5, pp. 511–515, Sep. 1982.

[15] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532–540, Apr. 1983.

[16] A. Margalit and A. Rosenfeld, "Using probabilistic domain knowledge to reduçe the expected computation cost of template matching," Tech. Rep. TR-2008, Dept. of Comput. Sci., Univ. of Maryland, College Park, MD, Mar. 1988.