A Robust Eigenspace Method for High-Speed Massively Parallel Vision Systems

Toshiharu Mukai

Bio-Mimetic Control Research Center
The Institute of Physical and
Chemical Research (RIKEN)
Nagoya 463, Japan

Noboru Ohnishi

Dept. of Information Engineering
Faculty of Engineering
Nagoya University
Nagoya 464-01, Japan

Abstract

Image processing systems, each consisting of massively parallel photodetectors and digital processing elements on a monolithic circuit, are recently being developed by several researchers. This approach is promising for applications in robot vision and visual inspection. Some earlyvision-like processing algorithms, such as edge-detection, smoothing and thinning, are installed in the vision systems. However, they are not sufficient for applications because their output is in the form of pattern information, so that, in order to respond to input, some feature values are required to be extracted from the pattern. In the present paper, we propose a method for extracting feature values associated with images in a massively parallel vision system. The feature can be a complex one, such as the rotation angle of an object, if the feature value changes continuously as the image changes. The method is based on the eigenspace method, but modified to be robust. In addition, we propose some computation accelerating methods and report some experimental results.

1 Introduction

Image processing systems, each consisting of massively parallel photodetectors and digital PEs (Processing Elements) on a monolithic circuit, are recently being developed by several researchers [1, 2, 3]. This approach is promising for applications in robot vision and visual inspection. The advantages of the approach are very high-speed processing and flexibility. Each PE has one corresponding photodetector directly connected to it. This connection and massively parallel structure enable high-speed processing and the digital architecture provides programmability which gives flexibility to the system. For example, SPE-8 [3, 4] can complete various image processing tasks within 1 millisecond.

In these vision systems, many processors are integrated on the same silicon chip. As a result, a single processor is constructed with the number of transistors as small as possible. Thus there is a trade-off between the number of processors and their versatility of operation. For example, in SPE-8 where 8 PEs are integrated in a chip, local memory is limited to a 24 bits. In the programmable artificial retina [1], the array of PEs is 65×76, but the memory size is just 3 bits. Furthermore, connection lines in the chip should be reduced. Thus instruction lines are reduced and, as a result, these systems have SIMD (Single Instruction Multi Data) architecture for processing. In other words, processes in all PEs must be executed under the same instruction set. Connection between PEs is also limited. In the developed systems, each PE is connected to its 4-neighbors and output from the entire system is obtained only from the periphery of the PE array.

Some earlyvision-like processing algorithms, such as edge-detection, smoothing and thinning, are installed for massively parallel vision systems. However, they are not sufficient for practical applications. When such vision systems are used in an application, some decision or reaction must be made according to the output from the vision system. In order to use the high-speed property of massively parallel vision systems to full advantage, output and decision-making must be completed in a short period of time. However, the output of earlyvision-like processing algorithms is in the form of pattern information, so that both transmission and processing take a long period of time. Therefore it is useful to extract some scalar values or vectors representing abstract features of the image from the pattern information. In the presently developed systems, only simple features, such as the centroid, are extracted.

We propose a method for extracting information associated with images in massively parallel vision. The method is robust to perturbations, only using local values, though the magnitudes of perturbations are limited to small values. Complex features, such as the rotation angle of an object, can be extracted only if the feature value changes continuously as the image changes.

The method is based on the eigenspace method. The usual eigenspace method is used after segmentation of the window including the object concerned from the entire image and normalization of the size of the window, so that the object has a predetermined size and position. However, in massively parallel vision, these processes are not suitable because they cannot be executed in parallel using only local information, so the processing time is lengthened.

We extend the eigenspace method to make it robust. Then a variety of conditions such as position, size and illumination can be absorbed to some extent. The usual eigenspace method is to discriminate all samples but our new method is for discriminating sets corresponding to feature values while neglecting the variance within each set. Some studies have been conducted on set discrimination in image processing, [5, 6, 7, 8]. We propose a novel method for set discrimination, the results of which are associated to feature values. The association is similar to a previous method [9, 10], but our method is specifically for use in massively parallel vision. In addition, we propose a method for accelerating premeasurement calculation. Lastly, some experimental results are shown.

2 Association of images and feature values using the eigenspace method

The purpose of this study is to establish association between images and feature values. An example of a feature value is the rotation angle of an object in the image. We assume that template images and their association to feature values are stored in advance. For the association, some mapping from image space to feature space can be used. This idea itself is quite simple. However, images have such large dimensionality that calculation to find the prestored template image nearest the input image is too complex for real-time application such as visual feedback. In addition, in massively parallel vision, the memory size available for storing template images is very limited.

Therefore some compression techniques are required. When vision is used in a real-time application, consecutive frame images are obtained one after another and changes between them can be assumed to be small. We assume that changes in associated feature values are small, too.

The eigenspace method is suitable for this situation. In this method, template images are compressed in advance. More specifically, images are mapped into a smaller dimensional space, the so-called eigenspace, where the scatter of the mapped images is guaranteed to be maximal. Then mapped images are stored as compressed template images.

In the eigenspace method, an image with size $a \times b$ is converted to a $p(= a \times b)$-dimensional vector lexicographically. We denote the vector as $\boldsymbol{x}$. Note that there is one-to-one mapping between an $a \times b$-image and a p-vector.

Assume that template images $\boldsymbol{x}_i$ are given, where i $(i = 1, 2, \ldots, n)$ denotes the index of the template image. We compute the covariance matrix

$$S = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T, \qquad (1)$$

where $\boldsymbol{\mu}$ is the mean image of all template images. Then the normalized eigenvectors $\boldsymbol{v}_i$, where $\boldsymbol{v}_i$ corresponds to the ith largest eigenvalue, are computed. When we want to determine a compressed space, the so-called eigenspace, with the dimensionality q, where usually $q \ll p$, q largest eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_q$ are used for compression. In other words, $p \times q$-matrix W is defined as

$$W = [\boldsymbol{v}_1 \boldsymbol{v}_2 \ldots \boldsymbol{v}_q] \qquad (2)$$

and p-vector $\boldsymbol{x}$ is mapped to q-vector $\boldsymbol{y}$ as

$$\boldsymbol{y} = W^T(\boldsymbol{x} - \boldsymbol{\mu}). \qquad (3)$$

Template images $\boldsymbol{x}_i$ are compressed to $\boldsymbol{y}_i$, then $\boldsymbol{y}_i$ are stored as templates. In this small dimensional space, memorization and association of compressed images to feature values are much easier. We want to point out that, for our purpose, $-W^T\boldsymbol{\mu}$ in (3) is not necessary, because it merely changes the offset. Hence we use

$$\boldsymbol{y} = W^T\boldsymbol{x} \qquad (4)$$

instead of (3).

In our design, the eigenvectors, $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_q$, are stored in the massively parallel vision system for mapping input image into the eigenspace. Eigenvector $\boldsymbol{v}_i$ has the same dimensionality with input images, so that each element of $\boldsymbol{v}_i$ is stored in the corresponding PE. The upper limit of q depends on the local memory size of a PE.

In measurement, we map input image $\boldsymbol{x}$ to $\boldsymbol{y}$ in the eigenspace using (4), then find the associated feature value. The PE array is programmed to calculate

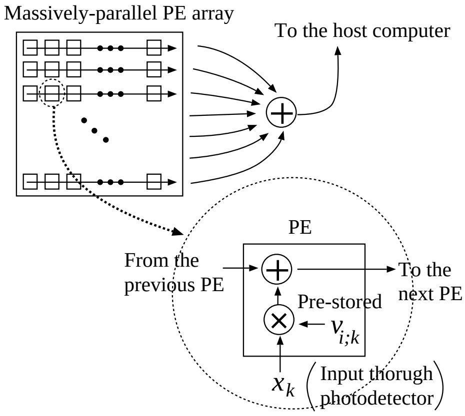


Figure 1: The summation of $v_{i;k}x_k$ in row-directional pipelines.

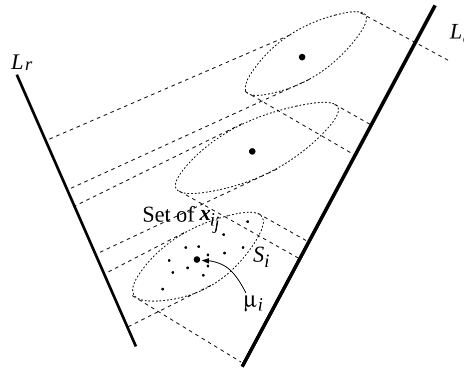


Figure 2: When $\boldsymbol{x}_{i_j}$ is mapped to L_e, it may overlap with other points from the different set, while when mapped to L_r, there is no overlap.

$W^T\boldsymbol{x}$ in (4). Then the result is output from the array to the host computer. The calculation of (4) is done in the PE array as follows. It consists of $\boldsymbol{v}_i^T\boldsymbol{x}$ and each $\boldsymbol{v}_i^T\boldsymbol{x}$ is the summation of $v_{i;k}x_k$, where $v_{i;k}$ and x_k are the kth element of the vector $\boldsymbol{v}_i$ and $\boldsymbol{x}$, respectively. Therefore, to obtain the value, after multiplication of $v_{i;k}x_k$ in each PE in parallel, their addition is necessary. The summation is calculated in each row (column)-directional pipeline as shown in Fig.1. After obtaining all outputs from one side of the PE array and adding them, we obtain the $\boldsymbol{v}_i^T\boldsymbol{x}$. By iterating this procedure q times, we obtain $W^T\boldsymbol{x}$.

In the host computer, $\boldsymbol{y}$ is mapped to the corresponding feature value. The simplest method is finding the nearest template image, but it lacks interpolation ability. So we use a parameterized curve to represent the relation between template images and feature values. When the mapped point in the eigenspace from input image is obtained, the nearest point on the curve is determined to obtain the corresponding feature value. This space is referred to as "parametric eigenspace" in [9] and we have adopted this nomenclature.

3 Robust Eigenspace Method

The problem of the eigenspace method is its lack of robustness. When an image is shifted, for example, the mapped point may move to a position distant form the original position. For our purpose, all perturbed images from the same original image should be mapped to close positions. When applications do not require real-time processing, the image can be segmented and size-normalized to provide a regulated image for the eigenspace method. However these techniques cannot be used in massively parallel vision where direct PE communication is only among 4-neighbors and fast calculation is required.

We propose a new method for this purpose, though the perturbation is restricted. We assume that the sets of perturbed images $\boldsymbol{x}_{i_j}$ $(j = 1, 2, \ldots, m_i)$, each corresponding to one feature value indexed by i, are obtained. Then we obtain the mean image $\boldsymbol{\mu}_i$ and covariance matrix S_i from the set. To discriminate sets, $\boldsymbol{\mu}_i$ can be used to represent the set and used in the conventional eigenspace method. However, if using only $\boldsymbol{\mu}_i$, distribution of members in the set is not considered, so that the best compressed space cannot be obtained. For example, in Fig.2, L_e is the space obtained by the eigenspace method and the mapped points from the different sets are overlapping. We want to find the space L_r for mapping, where overlap does not exist.

In order to map images in the same set to positions in close proximity and those from different sets to separated positions, we use the Mahalanobis generalized distance which represents the distance between sets. The distance between $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}_\beta$ is

$$d(\alpha, \beta) = (\boldsymbol{x}_\alpha - \boldsymbol{x}_\beta)^T S^{-1} (\boldsymbol{x}_\alpha - \boldsymbol{x}_\beta), \tag{5}$$

where S is the covariance matrix of the set the data belong to. In the present case, we cannot determine from which set the input is obtained in advance, hence the covariance matrix of the input image cannot be determined. Hence we use the sum of all sets

$$S = \sum_{i=1}^{c} S_i = \sum_{i=1}^{c}\sum_{j=1}^{m_i} \frac{1}{m_i}(\boldsymbol{x}_{i_j} - \boldsymbol{\mu}_i)(\boldsymbol{x}_{i_j} - \boldsymbol{\mu}_i)^T, \tag{6}$$

where c is the number of sets. We define the mapping to the space where the square of the Euclidean distance of projected points is the same as the Mahalanobis generalized distance of the original points. The mapping is

$$\boldsymbol{z} = S^{-1/2}\boldsymbol{x}. \tag{7}$$

Note that $S^{-1/2}$ is symmetric because S is symmetric and can always be obtained when S is positive definite. Since S is the covariance matrix, it is semipositive definite. Moreover, if all pixel values include noise, it is supposed to be positive definite.

We convert $\boldsymbol{\mu}_i$ to $\boldsymbol{z}_i^{\mu}$ by (7) and use the usual eigenspace method about $\boldsymbol{z}_i^{\mu}$. The covariance matrix of $\boldsymbol{z}_i^{\mu}$ is obtained as

$$\begin{aligned} R &\equiv \frac{1}{c}\sum_{i=1}^{c}(\boldsymbol{z}_i^{\mu} - \bar{\boldsymbol{z}}^{\mu})(\boldsymbol{z}_i^{\mu} - \bar{\boldsymbol{z}}^{\mu})^T \\ &= \frac{1}{c}S^{-\frac{1}{2}}\left(\sum_{i=1}^{c}(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})^T\right)S^{-\frac{1}{2}}, \end{aligned} \tag{8}$$

where $\bar{\boldsymbol{z}}^{\mu} = \sum_{i=1}^{c}\boldsymbol{z}_i^{\mu}/c$ and $\bar{\boldsymbol{\mu}} = \sum_{i=1}^{c}\boldsymbol{\mu}_i/c$. Then the normalized eigenvectors $\boldsymbol{v}_i^z$ $(i = 1, 2, \ldots, q^z)$ of R, where $\boldsymbol{v}_i^z$ corresponds to the ith largest eigenvalue and q^z is the dimensionality of the eigenspace, are obtained and stored in advance. We refer to this eigenspace as "robust eigenspace".

When input image $\boldsymbol{x}$ is obtained, it is mapped into the robust eigenspace in the PE array. The ith element of the mapped image is

$$\boldsymbol{v}_i^{zT}\left(S^{-1/2}\boldsymbol{x}\right) = \left(S^{-1/2}\boldsymbol{v}_i^z\right)^T\boldsymbol{x}. \tag{9}$$

Therefore, we should store

$$\boldsymbol{u}_i \equiv S^{-1/2}\boldsymbol{v}_i^z \tag{10}$$

in the PE array in advance. This has the same dimensionality with input images and each element should be stored in each PE. Then the calculation required in the measurement is only the products of the stored vectors $\boldsymbol{u}_i$ and the input image.

To summarize, the procedure prior to measurement is as follows:

1. Calculate $S = \sum_{i=1}^{c} S_i$.
2. Obtain the $S^{-1/2}$.
3. Calculate the covariance matrix, R, in the mapped space by (8).
4. Obtain the several eigenvectors $\boldsymbol{v}_i^z$ corresponding to the several largest eigenvalues.
5. Compute $\boldsymbol{u}_i = S^{-1/2}\boldsymbol{v}_i^z$ $(i = 1, 2, \ldots, q^z)$ and store them in the PE array.

In the measurement phase, $\boldsymbol{u}_i^T\boldsymbol{x}$ is calculated in the PE array using the pipeline method explained in the previous section. The output is obtained from one side of the array and it is mapped to the feature value in the host computer.

4 Accelerating Algorithm

If we straightforwardly compute the above algorithm for premeasurement, it is a lengthy process because vectors made from images have very large dimensionality. For example, even if the image is as small as 32×32 pixels, the vector is a 1024-vector and its covariance matrix is 1024×1024-matrix. Therefore matrix calculation becomes very lengthy.

We want to obtain $S^{-1/2}$ in an efficient way. Since S is a symmetric positive definite matrix, it can be decomposed as

$$S = V\Lambda V^T, \tag{11}$$

where V is an orthonormal matrix composed of the eigenvectors $\boldsymbol{v}_i$, and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$. Then $S^{-1/2}$ is composed as

$$S^{-1/2} = V\Lambda^{-1/2}V^T, \tag{12}$$

where $\Lambda^{-1/2} = \mathrm{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_p^{-1/2})$.

However there are some problems. The first one is that S computed from actual samples is semipositive definite when the number of samples is not sufficient. In this case, $\lambda_i = 0$ and we cannot obtain $\lambda_i^{-1/2}$. The number of nonzero eigenvalues is equal to the rank of S, so it does not exceed the number of samples. Therefore, in many cases where the sample number is small, zero eigenvalues exist. However, it is expected that, if we observe more images, the matrix S becomes full rank because of observation noise. So, we set a small positive value as the minimum eigenvalue and, round up the smaller eigenvalues to this minimum value.

The second problem is computation time of the decomposition (11), which is very lengthy when S is large. This problem can be solved by the method proposed in [11]. S can be expressed as follows:

$$S = \sum_{i=1}^{c}\sum_{j=1}^{m_i}\frac{1}{m_i}(\boldsymbol{x}_{j_i} - \boldsymbol{\mu}_i)(\boldsymbol{x}_{j_i} - \boldsymbol{\mu}_i)^T = XX^T. \tag{13}$$

Instead of calculating the eigenvalues and eigenvectors of S, we calculate those of

$$\tilde{S} = X^TX. \tag{14}$$

The size of $\tilde{S}$ is usually much smaller than that of S, so that the calculation is faster than that of S. In actual calculation, to obtain the eigenvalues and eigenvectors of $\tilde{S}$, we used the method in [12] composed of the Householder method and QL method. When the eigenvalues and eigenvectors of $\tilde{S}$ are $\tilde{\lambda}_i$ and $\tilde{\boldsymbol{v}}_i$, respectively, the following relation is used to obtain those of the original S.

$$\begin{aligned} \lambda_i &= \tilde{\lambda}_i \\ \boldsymbol{v}_i &= \tilde{\lambda}_i^{-1/2} X \tilde{\boldsymbol{v}}_i \end{aligned} \tag{15}$$

S is usually larger and has more eigenvalues than $\tilde{S}$. The remaining eigenvalues which cannot be obtained using the above equation are all theoretically zero. However, in this method, we assume the positive minimum value as mentioned above, instead of zero.

Then (8) and $S^{-1/2}\boldsymbol{v}_i^z$ are calculated. This also takes a long time because the matrices are very large. To solve this problem, we rescale S so that the minimum value is 1. This procedure merely changes the scale of mapping, so that the final result does not change. Then we can obtain

$$\begin{aligned} S^{-1/2} &= \sum_{i=1}^{p} \sqrt{\frac{1}{\lambda_i}} \boldsymbol{v}_i \boldsymbol{v}_i^T \\ &= I + \sum_{i=1}^{q} \left(\sqrt{\frac{1}{\lambda_i}} - 1 \right) \boldsymbol{v}_i \boldsymbol{v}_i^T, \end{aligned} \tag{16}$$

where we used $VV^T = I$ and $\lambda_i = 1$ when $i = q+1, q+2, \dots, p$. Remember that $q \ll p$. Hence, for arbitrary matrix A and $\boldsymbol{x}$, simplification is obtained as follows:

$$AS^{-1/2} = A + \sum_{i=1}^{q} \left(\sqrt{\frac{1}{\lambda_i}} - 1 \right) A \boldsymbol{v}_i \boldsymbol{v}_i^T, \tag{17}$$

$$S^{-1/2}A = A + \sum_{i=1}^{q} \left(\sqrt{\frac{1}{\lambda_i}} - 1 \right) \boldsymbol{v}_i \boldsymbol{v}_i^T A, \tag{18}$$

$$S^{-1/2}\boldsymbol{x} = \boldsymbol{x} + \sum_{i=1}^{q} \left(\sqrt{\frac{1}{\lambda_i}} - 1 \right) \boldsymbol{v}_i \boldsymbol{v}_i^T \boldsymbol{x}. \tag{19}$$

These relations greatly accelerate the calculation. For example, in (17), if A and $S^{-1/2}$ are $p \times p$-matrices, the left-hand side requires p^3 multiplications, while the right-hand side requires about $2p^2q$ multiplications.

5 Experiments

Experiments for checking discrimination ability of the proposed method are conducted. Our method was originally designed for massively parallel vision but, at present, we cannot use such a vision system with enough resolution. Thus we used a CCD camera as the input device. The input image is reduced to a 32×32 6-bit gray-scale image in order to simulate massively parallel vision ability. All calculations are performed in the host computer. Although we used the CCD camera as the input device, we consider that the usefulness of this method is proven by these experiments.

The object and one of the reduced outputs from the CCD camera is shown in Fig.3. For the templates, the object is rotated in steps of 30 degrees from -90 to +90 degrees and an image is obtained at each posture. We used two-dimensional robust eigenspace and the depth for representing each element of $\boldsymbol{u}_i$ is 7 bits. Therefore, memory size in each PE for storing precalculated data is 14 bits.

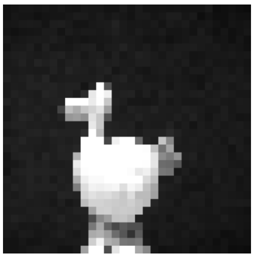
For obtaining perturbed images, we shifted original images by image processing for convenience. Each original image is shifted in x- and y-direction by $-1, 0, 1$ pixel, so that we obtain 9 images according to each rotation angle. For the small positive value used to round up eigenvalues, we selected the 50th largest eigenvalues. Distribution of template images mapped to the robust eigenspace by the proposed method is shown in Fig.4. For comparison, distribution in the conventional eigenspace obtained using only the means of sets of perturbed images is also shown in Fig.5. In these figures, 0 denotes a rotation angle of -90 degrees, 1 denotes -60 degrees, and so on. In the conventional eigenspace, points corresponding to different rotation angles are overlapping, so that it is impossible to make a mapping from this space to the rotation angle. On the other hand, in the robust eigenspace, points from the different sets are separated. This means that the robustness is improved in the proposed method.

We also performed an experiment to investigate the effect of changing background color and obtained similar results (data not shown).

6 Conclusion

A robust eigenspace method was proposed. It is suitable for a massively parallel vision system on the monolithic circuit. In addition, an accelerating algorithm for calculations needed prior to measurements was proposed and some experiments, though we used the CCD camera to simulate photodetector output, were conducted.

Object



Reduced output from the CCD camera

Figure 3: Object and reduced output from the CCD camera

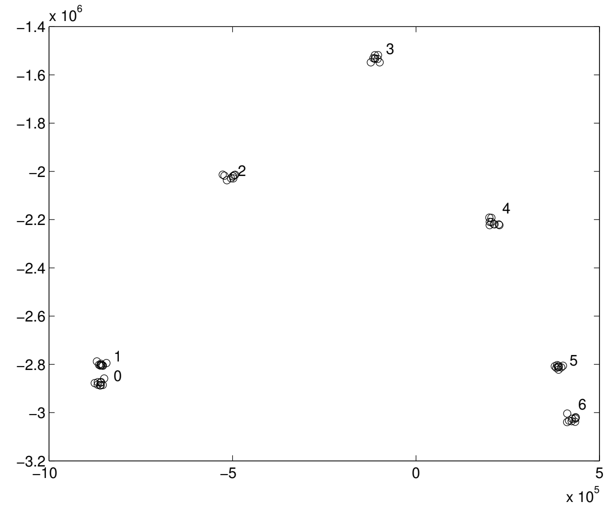


Figure 4: Distribution of mapped point by the proposed method when images are shifted.

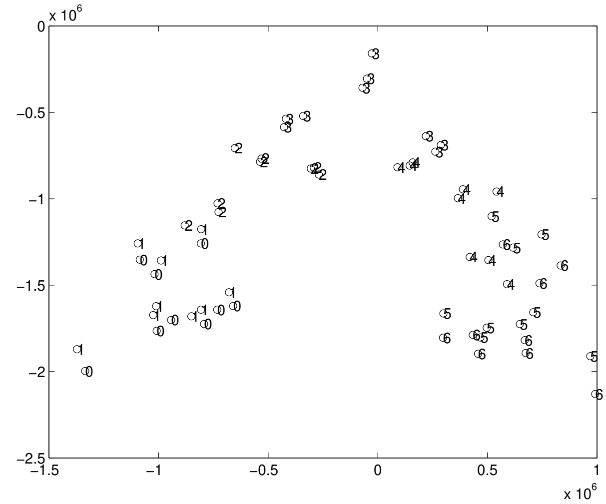


Figure 5: Distribution of mapped point by the conventional method when images are shifted.

References

[1] T. M. Bernard, B. Y. Zavidovique and F. J. Devos, "A Programmable Artificial Retina," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 7, pp. 789–798, 1993.

[2] J. E. Eklund, C. Svensson and A. Åström, "VLSI Implementation of a Focal Plane Image Processor — A Realization of the Near-Sensor Image Processing Concept," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 4, No. 3, pp. 322–335, 1996.

[3] T. Komuro, I. Ishii and M. Ishikawa, "Vision Chip Architecture Using General-Purpose Processing Elements for 1ms Vision System," *Proceedings of 4th IEEE Int. Workshop on Computer Architecture for Machine Perception (CAMP'97)*, pp. 276-279, 1997.

[4] I. Ishii, Y. Nakabo and M. Ishikawa, "Target Tracking Algorithm for 1ms Visual Feedback System Using Massively Parallel Processing," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 2309–2314.

[5] S. Baker and S. K. Nayer, "Pattern Rejection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 544–549, 1996.

[6] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 711–720, 1997.

[7] Y. Cui, D. L. Swets and J. J. Weng, "Learning - Based Hand Sign Recognition Using SHOSLIF-M," *Int'l Conf. on Computer Vision*, pp.631–636, 1995.

[8] Y. Cheng, K. Liu, J. Yang, Y. Zhuang and N. Gu, "Human Face Recognition Method Based on the Statistical Model of Small Sample Size," *SPIE Proc. Intelligent Robots and Computer Vision X: Algorithms and Techniques*, pp. 85–95, 1991.

[9] S. K. Nayar, H. Murase and S. A. Nene, "A General Learning Algorithm for Robot Vision," *Proceedings of the SPIE — The International Society for Optical Engineering*, Vol. 2304, pp. 10–17, 1994.

[10] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance," *International Journal of Computer Vision*, Vol. 14, pp. 5-24, 1995.

[11] H. Murakami and B. V. K. V. Kumar, "Efficient Calculation of Primary Images from a Set of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-4, No. 5, pp. 511–515, 1982.

[12] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, 1988.