Study on Algorithms for Part Recognition Problem in the Smart Surface

SHUAI, Jiangbing

(出版者 / Publisher)
法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編 / 法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)
8

(開始ページ / Start Page)
43

(終了ページ / End Page)
48

(発行年 / Year)
2013-03

(URL)
https://doi.org/10.15002/00009542

Study on Algorithms for Part Recognition Problem in the Smart Surface

Jiangbing Shuai
The Graduate School of Computer and Information Sciencesin at the Hosei University
E-mail: edisonshuai@gmail.com

***Abstract*—This study looks into the part recognition problem in the smart surface and proposes a modified Modular Principle Component Analysis (MPCA) algorithm for it. The smart surface aims at designing a microrobotics system for conveying, sorting and positioning. The part recognition is the first important step. In this research, the part recognition block is analyzed and simulated. The previous approach and algorithm by Boutoustous, et. al. PCA and MPCA are implemeted for the same two sets of experiements. Aiming at a better performance in terms of differentiation rate, computing time, memory cost, a modified modified MPCA algorithm is proposed. It extracts local and global information of the parts in order to reduce the effects caused by local array distribution and global array profile, and chooses a suitable eigenspace in order to reduce memory cost and improves accuracy. Two sets of experiments, i.e. three simple models and three complicated models, are conducted. The experimental results are analyzed and comparisons are made, which indicates that the performance of the proposed approach, the modified MPCA is more effective and robust than others.**

Keywords-modular Principal Component Analysis; modified modular Principal Component Analysis; part recognition; smart surface; Micro-Electro-Mechanical Systems (MEMS)

I. RESEARCH BACKROUND

Smart surface refers to the Micro-Electro-Mechanical Systems (MEMS) based on an array of micromodules in possession of intelligence. In the past decades, MEMS community has been drawing intense research interest because of their potential applications in macro scale [1] and micro scale. D. El Baz [2] proposed synchronous algorithm and asynchronous algorithm to differentiate parts in a smart surface. Nevertheless, high computational costs are major problems of this technique.

2D topology array distributed on the smart surface can be approximately regarded as a binary image. Principal Component Analysis (PCA) is a classical algorithm, which has widely used in dimension reduction, feature extraction, image recognition and other fields, and the related algorithms emerge in endlessly [3]. In recent years, the study of PCA has mainly focused on two aspects: one is to improve the PCA algorithm, such as two-dimensional PCA [4] and modular PCA (MPCA) [5] . Traditional PCA works over the one-dimensional vectors patterns which usually cause high dimensionality problem and suffer from small sample size problem. However, MPCA can easily solve these problems [6]. The other is to apply PCA to other algorithms for data preprocessing, such as independent component analysis [7, 8] and linear discriminent analysis [4]. The preprocessing by PCA can avoid data redundancy by reduction of dimension.

A new improvement for part recognition using modified MPCA is presented in this paper. The proposed algorithm is different from conventional MPCA in feature subspace computation. It not only takes the local features which are beneficial for differentiation into consideration, but also chooses an appropriate eigenspace dimension. If eigenspace dimension is too small, the information is not enough to describe the features of the global image. If eigenspace dimension is too large, it is easily affected by the noise and also cost too much memory without the gain of the performances. Therefore, it is wise to find out a way to choose eigenspace dimension. The experimental results performed over both three simple models and three complicated models show that the proposed algorithm works reasonably for the simple shape of parts recognition, and it outperforms the pervious method for the complicated shape recognition, in terms of differentiation rate, execution time, and memory cost.

The rest of this paper is organized as follows. Section 2 briefly introduces the part recognition problem in the smart surface, and interprets the previous approach by Kahina Boutoustous. Section 3 describes the PCA algorithm and MPCA algorithm. Section 4 specifies the modified MPCA method, then analyzes the performance theoretically. Section 5 presents experimental results over both three simple models and three complicated models obtained by applying the improved algorithm, MPCA, PCA, and previous method to part recognition, then analyzes the experiment results. Finally, conclusions are drawn.

II. THE PART RECOGNITION PROBLEM

Multi-function, such as conveying, sorting, and positioning object, is the objective of designing a smart surface. Part recognition is the first step to achieve other complicated functions. Due to the advantages in macro scale, fully decentralized systems are introduced to smart surface.

In order to differentiate part in a smart surface under fully distributed architecture, a large number of micromodules are put in array on the smart surface, and each micromodule comprises a sensor, a processing unit, and an actuator, which is able:

- to locally sense the state of the object (for example the presence/absence of the object).
- to decide its actions by itself in communication with its local neighbors.

Supervisor: Prof. Runhe Huang

- to have its own intelligence, that it doesn't need a central control unit.
- to well divided up so as to allow good functionality of global system.

A. The previou approach by Kahina Boutoustous [9]

To differentiate part on the smart surface, K. Boutoustous had proposed a fully distributed. The main process is conducted in the following two stages:

1) Offline stage

After rotation and translation of the training models, the images taken by the camera are discretized in matrices by "1" if the sensor is covered by the object and "0" otherwise. Then unique masks will be calculated the values of each criterion, and save the criteria values to a database. The unique parts refer to the sets of eliminating 90° rotations and mirrors.

2) Online stage

After reconstruction and calculation of criteria values, comparison of criteria values which are calculated in offline stage is executed until their difference is less than a given threshold. The online time is not ended until the type is successfully differentiated. Then the comparison block sends the information, such as the result and the position of the object (coordinates of the bounding rectangle) on the Smart Surface, to the control block. Thus the control block takes the responsibility from now on, either to give the object a small movement in order to retry differentiation or to move the object to its destination.

Online stage process is described as bellow:

```
Start time
repeat
  1. Each micromodule exchange local information
     with its four direct neighbors
  2. Do a union bit by bit of received information
until (the first micromodule get the matrix
     representation of the object)
  3. Calculate criteria values
  4. Match them with the criteria values stored in
     database
  5. Generate a DifferentiationResult
if DifferentiationResult > Threshold
  6. Send the DifferentiationResult to the control
     block
else
  7. Move the object to another place
  8. Reconstruct the part and recognize it again
end if
End time
```

B. Memory cost

Each micromodule has a memory limitation due to its micro scale integration. This section discuses the memory needed by one criterion.

First, the memory needed by one micromodule using one criterion computed. Each micromodule knows all the models. Let m(c_i, p_j) be the memory needed to store the value of criterion c_i of part p_j, which refers to the part used in offline stage. Therefore, all the models need m_1 bytes memory, defined by:

$$m_1 = \sum_{j=1}^{N_u} m(c_i, p_j) \tag{1}$$

where N_u is the number of unique parts [10], m_2 is the maximum size of a part given by:

$$m_2 = x \times y \tag{2}$$

The value of a criterion of the part P, which is going to be differentiated in real time, is m_3=m(c_i, P). Thus the total memory footprint needed by one micromodule for one criterion c_i is:

$$\begin{aligned} m &= m_1 + m_2 + m_3 \\ &= \sum_{j=1}^{N_u} m(c_i, p_j) + x \times y + m(c_i, P) \end{aligned} \tag{3}$$

C. Execution time

Suppose CC_i the best combination of criteria. Let $t_{1(c_i)}$ be the execution time of the criterion c_i. The total computation time of all criteria is:

$$T_1 = \sum_{c_j \subset CC_i} t_1(c_j) \tag{4}$$

Let t_2 be the (constant) comparison time. n is the number of parts to be differentiated. Thus, the execution time is:

$$\begin{aligned} T &= \sum_{c_j \subset CC_i} (t_1(c_j) + nt_2) \\ &= nt_2 |CC_i| + \sum_{c_j \subset CC_i} t_1(c_j) \end{aligned} \tag{5}$$

D. Discription of criteria

The differentiation criteria must be simple and must be easy to implement. There are totally 17 kinds of criteria in [10], and 8 of them are introduced in this paper. Descriptions of the criterion are as follows respectively:

- A: The number of 1 having at least three neighbors to 0 and forming a right angle.
- P: The number of 1 at least one neighbor at 0.
- S: The number of 1 of the part.
- R: The sum of the number of V shape angles.
- F: The sum of all Manhattan distances between 0.
- M: The sum of the number of bits that change.
- Y: The product of all Manhattan distances between 1.
- D: The sum of 1 located on both diagonals.

III. PCA AND MPCA

A. PCA

PCA for recognition problems was proposed by M. Turk and A. Pentland [11]. It is a classical dimension reduction and feature extraction algorithm whose theoretical basis is discrete K–L transform. The algorithm description is as follows:

1) Data preprocess

Let M images, $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_M$, be training samples. Each image size is $a \times b$. a and b are the row and column pixel number respectively. Transform each image into vector $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \ldots, \mathbf{\Gamma}_M$.

2) Computation of feature subspace

The average image is defined by:

$$\psi = \frac{1}{M}\sum_{i=1}^{M}\mathbf{\Gamma}_i \qquad (6)$$

Each image differs from the average by the vector $\mathbf{\Phi}_i = (\mathbf{\Gamma}_i - \mathbf{\Psi})$, $i=1, 2, \ldots, M$. The covariance matrix is obtained as

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{\Phi}_i\mathbf{\Phi}_i^T \qquad (7)$$

The eigenvectors, $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$, corresponding to the largest d ($d \ll M$) eigenvalues of $\mathbf{C}$ are so called eigenspace.

3) Recognition

In the recognition processes using PCA, there are the following steps:

a) Project all training samples onto eigenspace, then obtain the vectors $\mathbf{\Omega}_i = [\mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_d]^T\mathbf{\Phi}_i$, $i=1, 2, \ldots, M$.

b) Obtain the vector $\mathbf{\Omega}_{test} = [\mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_d]^T(\mathbf{\Gamma}_{test} - \mathbf{\Psi})$ of the test sample $\mathbf{\Gamma}_{test}$ in the eigenspace.

c) Calculate Euclidean distance of image vectors by

$$\varepsilon_i = \| \mathbf{\Omega}_{test} - \mathbf{\Omega}_i \|_2, \quad i=1, 2, \ldots, M \qquad (8)$$

The smaller the value is, the less difference of the image is.

B. MPCA

Define PCA, as a global algorithm, is sensitive to noise. Furthermore, variation on shape sometimes affects only some regions of the images. From these points of view, we expect that the MPCA reaches better recognition rate than the conventional PCA. Rajkiran Gottumukkal et al. [6] presented modular PCA which segment each image into the same size subimages, and apply PCA to these subimages. MPCA can reduce the influence of local noise on image and express more local features which are used for recognition. It is noted that the number of divided subimages must not too large or too small, or the global information of the image may be lost. The algorithm can be described as follows:

1) Data preprocess

Assume that the same training samples are processed in MPCA algorithm. Each image in the training sets is segmented into $p \times q$ subimages, where p and q are divided horizonal and vertical numbers respectively. Then the training samples increase from M to $M' = M \times p \times q$. Then we represent the divided submatrices by $\mathbf{A}_{ij}$, $i=1, 2, \ldots, M$, $j=1,2,\ldots,p \times q$, the j^{th} submatrix of the i^{th} matrix. Each submatrix $\mathbf{A}_{ij}$ is transformed from $(a / p) \times (b / q)$ matrix to a vector $\mathbf{\Gamma}_{i,j}$ by placing the submatrix columns consecutively. Then PCA is applied to each subimage.

2) Computation of feature subspace

The average image is given by

$$\psi = \frac{1}{M'}\sum_{i=1}^{M}\sum_{j=1}^{p \times q}\mathbf{\Gamma}_{ij} \qquad (9)$$

Each image differs from the average by the vector $\mathbf{\Phi}_{i,j} = (\mathbf{\Gamma}_{ij} - \mathbf{\Psi})$, $i=1, 2, \ldots, M$, $j=1, 2, \ldots, p \times q$. The covariance matrix is obtained as

$$\mathbf{C} = \frac{1}{M'}\sum_{i=1}^{M}\sum_{j=1}^{p \times q}\mathbf{\Phi}_{ij}\mathbf{\Phi}_{ij}^T \qquad (10)$$

The eigenvectors, $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$, corresponding to the largest d ($d \ll M$) eigenvalues of $\mathbf{C}$ are so called eigenspace.

3) Recognition

Similarly, it has the following three steps.

a) Project all training samples onto eigenspace, then obtain the vectors $\mathbf{\Omega}_i = [\mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_d]^T\mathbf{\Phi}_i$, $i=1, 2, \ldots, M$.

b) Divide the test image into $p \times q$ subimage vectors $\mathbf{\Gamma}_{test,1}, \mathbf{\Gamma}_{test,2}, \ldots, \mathbf{\Gamma}_{test,p \times q}$, then obtain the vector $\mathbf{\Omega}_{test,j} = [\mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_d]^T(\mathbf{\Gamma}_{test,j} - \psi)$ of the test sample $\mathbf{\Gamma}_{test}$ in the eigenspace, $j=1, 2, \ldots, p \times q$.

c) Compute Euclidean distance between the test image vectors and the sample image vectors by:

$$\varepsilon_i = \frac{1}{p \times q}\sum_{j=1}^{p \times q}\| \mathbf{\Omega}_{i,j} - \mathbf{\Omega}_{test,j} \|_2, \quad i=1, 2, \ldots, M \qquad (11)$$

IV. MODIFIED MPCA

A. Algorithm descriptiom

In the modified MPCA algorithm, the first two processes, i.e. data processing and covariance matrix computation are the same as in the MPCA algorithm. However, the modification to the feature subspace computation is described as below.

1) Computation of feature subspace

Assume that $\lambda_1, \lambda_2, \ldots, \lambda_{n'}$ ($\lambda_1 > \lambda_2 > \ldots > \lambda_{n'}$), $n' = n / p$, are the sorted eigenvalues of $\mathbf{C}$, and $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n'}$, are the corresponding eigenvectors. The contribution of eigenvalues g is defined by the formula:

$$g = \frac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{n'}\lambda_i} \qquad (12)$$

where n' is the number of eigenvalues. Because 2D topology array distributed on the smart surface can be regarded as a binary image, the eigenvalues of the corresponding binary matrix is bound up with the size of binary matrix. Furthermore, the eigenvalues of the corresponding matrix subject to a exponential decay [12] as shown in Fig. 1.

It implies that the largest d eigenvalues contain sufficient information to depict the image. However, blind increment in contribution g may lead to a large eigenspace, which would lead to a considerable memory cost. So it is necessary to find an appropriate value r_k to evaluate the efficiency of each eigenvalue,which is defined by:

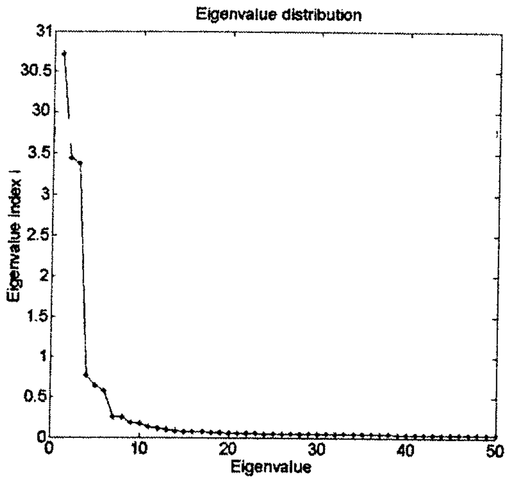

Fig. 1. The eigenvalues' exponential distribution.

$$r_k = \frac{\lambda_k}{\sum_{i=1}^{k} \lambda_i} \qquad (13)$$

where k is the eigenvalue index. r_k indicates the efficiency of eigenvalue λ_k in the largest k eigenvalues. If r_k is smaller than a given threshold, it implies that there is no need to add any more eigenvalues. Because the efficiency of the remaining eigenvalues must be lower than the threshold. Experimental results in section 5 show that the introduced r_k can efficiently prevent from memory waste problem.

The computation of feature subspace algorithm can be described as follows:

```
while contribution g ≤ c1
   Add the largest eigenvalue, and then
   calculate contribution g
end while
Set d=0
For all the selected d' eigenvalues
   if r ≥ c2 and d ≥ d'
      d=d+1, and calculate r
   else
      break
   end if
end for
```

The eigenvectors, $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$, corresponding to the largest d ($d \ll M$) eigenvalues of $\mathbf{C}$ are so called eigenspace.

2) Recognition

a) Divide $\mathbf{I}_{test}$ into $p\times q$ subimage vectors $\mathbf{\Gamma}_{test,1}, \mathbf{\Gamma}_{test,2}, \ldots, \mathbf{\Gamma}_{test,p\times q}$, then obtain the vector $\mathbf{\Omega}_{test,j}=[\mathbf{u}_1\mathbf{u}_2\ldots\mathbf{u}_d]^T\mathbf{\Phi}_{test,j}$, $j=1, 2,\ldots, p\times q$, of the test sample $\mathbf{\Gamma}_{test}$ in the eigenspace.

b) Compute Euclidean distance between the test image vectors and the sample image vectors as similarity measurement by:

$$\varepsilon_i = \frac{1}{p\times q}\sum_{j=1}^{p\times q} \| \mathbf{\Omega}_{i,j} - \mathbf{\Omega}_{test,j} \|_2 , \quad i=1, 2,\ldots, M \qquad (14)$$

B. Comparisons with PCA and MPCA

The author proposed modified MPCA in section 4.1, PCA and MPCA are shown in section 3.1 and 3.2 respectively.

1) Comparison with PCA

In recognition phase, PCA takes $d\times(a\times b)^2$ times addition calculation and multiplication and $M\times d$ times comparison. On the other hand, modified MPCA needs $(d\times(a\times b)^2)/(p\times q)$ times addition calculation and multiplication and $M\times d\times p\times q$ times comparison. Due to that addition calculation and multiplication time is shorter than comparison time, modified MPCA algorithm has a higher execution time than PCA algorithm.

2) Comparison with conventional MPCA

In conventional MPCA algorithm, the eigenvectors, $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$, corresponding to the largest d ($d \ll M$) eigenvalues of $\mathbf{C}$ are so called eigenspace. However, the eigenvalues of the corresponding binary matrix subject to a exponential decay. A suitable d is enough to depict the image. To prevent redundancy, we utilize Equation (12) and (13) to search the optimal d. This process also improve differentiation rate due to reduction of noise.

C. Memory cost and execution time

Let m_1' be the memory needed to store training samples in database, and d be the optimal dimensionality of feature subspace. Due to the segmentation of each image, the training samples increase from M to $M'=M\times p\times q$. And each subimage is converted into d dimension vector. Therefore, the memory needed in training phase is:

$$m_1' = M'\times d \qquad (15)$$

m_2' is the size of feature subspace given by

$$m_2' = \frac{a\times b\times d}{p\times q} \qquad (16)$$

The value of test image is

$$m_3' = p\times q\times d \qquad (17)$$

The total memory needed by one micromodule is:

$$\begin{aligned} m' &= m_1' + m_2' + m_3' \\ &= M'd + \frac{a\times b\times d}{p\times q} + p\times q\times d \end{aligned} \qquad (18)$$

Suppose t_1' and t_2' is the time cost of segmenting and projecting test image onto feature subspace respectively, and t_3' is the (constant) comparison time for one time. The total computation time is Mt_3'. Therefore, the execution time of modified MPCA is:

$$t' = t_1' + t_2' + Mt_3' \qquad (19)$$

V. EXPERIMENTAL RESULTS AND ANALYSES

For the comparison convenience, the experiments are conducted on the assumption that the smart surface is a 500×500 mm^2 square surface with a 400×400 actuator array. Two sets of experiments were conducted: a set of 3 simple models experiment and a set of 3 complicated models experiment, which have slightly differences in shape. All images were normalized to 400×400 pixels. In order to compare the results in equal conditions of rotation angle and translation distance, the experiments were performed using the same training sets and test sets. Each model is formed by 360 images with rotation from 1° to 359° and a random translation. And the experiments were performed 10 times using different training sets and test sets.

A. The experiment on the simple models

In the first experiment, the modified MPCA was applied to the part recognition of three simple models as shown in Fig. 2 and compared the performance with the previous method [9].

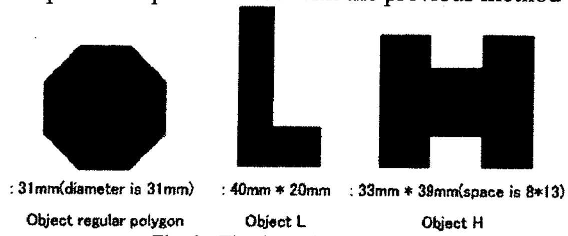

Fig. 2. The three simple models.

In the previous method, a group of criteria {*A*, *P*, *S*} were selected, and all combinations of the criteria are tested and the results in terms of the memory cost and the differentiation rates in each combination case are presented in a tree as shown in Fig. 3.

The tree starts at node, Root and the differentiation rate is given in each node, and each link between two connected nodes denotes the added criterion and memory cost from the Root to the end of branch. For example, one single criterion *P* only have 7.90% differentiation with 0.24 KB memory cost, but the combination *PS* reaches 92.57% differentiation with 2.11 KB memory cost. The combination *APS* nevertheless reaches 97.22% differentiation with 7.45 KB memory cost. It costs more than treble memory to improve only 4.65% differentiation. Consequently, the best combination of criteria {*A*, *P*, *S*} is *PS*. This parameter is adopted in comparison with PCA, MPCA, and modified MPCA algorithm. Generally, the more criteria are used, the higher differentiation rate can obtain. However, memory cost increase along with adding criterion. Thus, it is good to find an appropriate criteria combination to balance the differentiation rate and the memory cost.

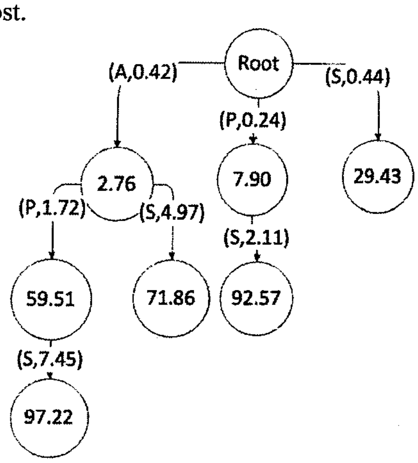

Fig. 3. Memory cost and differentiation rate tree.

TABLE 1

The comparison of previous method, PCA, and MPCA.

	previous method	PCA	MPCA
Differentiation rate (%)	92.57	91.11	92.04
Execution time (ms)	49.53	0.18	11.65
Memory cost (KB)	2.11	160.05	5.10

Table 1 gives the performance comparison using the previous method, PCA, and MPCA. It demonstrates that the differentiation rates are almost same. But previous approach performs the lowest memory cost along with the highest execution time, and PCA performs the lowest execution time with the highest memory cost. MPCA seems to receive a similar performance in terms of the differentiation rate as the previous method and PCA do. Of course, it obtains much better performance rate in the memory cost compared to PCA, and has a slight worse performance than the previous method. Both PCA and MPCA save the computation time compared the previous method.

TABLE 2

The parameters of MPCA and modified MPCA.

	MPCA	modified MPCA
Segmentation	10×10	5×5
Eigenspace dimension	3	1
Training sample number	4	4

Table 2 illustrates the parameters used in MPCA and modified MPCA. With the used parameters in the experiments, the result comparisons of MPCA and the modified MPCA are presented in Table 3.

TABLE 3

The comparison between MPCA and modified MPCA.

	MPCA	modified MPCA
Differentiation rate (%)	92.04	92.50
Execution time (ms)	11.65	5.88
Memory cost (KB)	5.10	1.61

From the results presented in Table 3, it can be observed that the modified MPCA and the MPCA have the similar differentiation rate. However, compared to MPCA, the modified MPCA gives the significant performance in terms of the execution time and the memory cost.

B. The experiment on the complicated models

In the second set of experiment, the modified MPCA was applied to the part recognition of three complicated models as shown in Fig. 4 and compared the performance with the previous method.

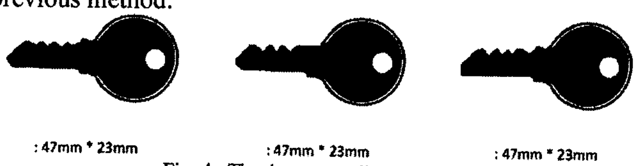

Fig. 4. The three complicated models.

In the previous method, the combination of criteria {*A*, *P*, *S*, *R*, *F*, *M*, *Y*, *D*}were selected to execute the experiment.

TABLE 4

The comparison on previous method, PCA, and MPCA.

	previous method	PCA	MPCA
Differentiation rate (%)	8.49	63.33	79.44
Execution time (ms)	542.10	1.01	88.32
Memory cost (KB)	102.91	243.25	18.80

Table 4 shows the performance comparison using previous method, PCA, and MPCA. We can see the previous method reaches only 8.49% differentiation rate along with the highest execution time and a large scale of memory cost, and PCA performs the lowest execution time with the highest memory cost. MPCA receives highest differentiation rate and lowest memory cost among them. However, the execution time is far larger than PCA. Both PCA and MPCA save the computation time compared the previous method.

TABLE 5

The parameters of MPCA and modified MPCA.

	MPCA	modified MPCA
Segmentation	10×10	10×10
Eigenspace dimension	20	4
Training sample number	180	180

Table 5 illustrates the parameters used in MPCA and modified MPCA. With the used parameters in the experiments, the result comparisons of MPCA and the modified MPCA are presented in Table 6.

TABLE 6

The comparison between MPCA and modified MPCA.

	MPCA	modified MPCA
Differentiation rate (%)	79.44	87.78
Execution time (ms)	88.32	84.62
Memory cost (KB)	18.80	3.76

From the results presented in Table 6, it can be observed that the modified MPCA and the MPCA have the similar execution time. However, the modified MPCA gives the significant performance in terms of the differentiation rate and the memory cost.

C. *Experimental results analyses*

The experimental results over three simple models and three complicated models show that the modified MPCA approach outperforms others in differentiation rate and has significant better performance in the memory cost. However, the execution time is relative higher than PCA approach due to the image segmentation process but still far less than the execution time in the previous approach. In conclusion, the modified MPCA works reasonably for the simple shape of parts recognition. MPCA performs better in differentiation rate than the previous method owe to the extraction of local features from the segmented images. The proposed approach, the modified MPCA results in a better performance than MPCA, which is because of finding the appropriate eigenspace dimension. If eigenspace dimension is too small, the information is not enough to describe the features of the global image. If eigenspace dimension is too large, it is easily affected by the noise and also cost too much memory without the gain of the performances. Therefore, it is wise to find out an appropriate choose the eigenspace dimension by using the previous known eigenvalues gradually.

VI. CONCLUSION

In this paper, the modified MPCA algorithm is proposed for part recognition in a smart surface. The modified MPCA is different from MPCA in feature subspace computation. In comparison with PCA, although the proposed algorithm has a relative higher computation cost due to the segmentation process. The memory cost is significantly improved. Compared to the previous approach, regarding the relative simple shape objects, the differentiation rate does not change a lot. However, for the complicated shape objects, the previous method almost does not work reasonably, the modified MPCA works well with the significant differentiation rate and small memory cost although it trades with a relative higher computation cost.

REFERENCES

[1] T. Tsao, F. Jiang, R. Miller, Y.-C. Tai, B. Gupta, R. Goodman, S. Tung, and C.-M. Ho, "An integrated MEMS system for turbulent boundary layer control," in *Solid State Sensors and Actuators, 1997. TRANSDUCERS '97 Chicago., 1997 International Conference on*, 1997, vol. 1, pp. 315 –318 vol.1.

[2] D. El Baz, V. Boyer, J. Bourgeois, E. Dedu, and K. Boutoustous, "Distributed part differentiation in a smart surface," *Mechatronics*, vol. 22, no. 5, pp. 522–530, Aug. 2012.

[3] Y. Zhao, X. Shen, N. D. Georganas, and E. M. Petriu, "Part-based PCA for facial feature extraction and classification," in *Haptic Audio visual Environments and Games, 2009. HAVE 2009. IEEE International Workshop on*, 2009, pp. 99 –104.

[4] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 1, pp. 131 –137, Jan. 2004.

[5] C. Vermillion, J. Sun, and K. Butts, "Model predictive control allocation for overactuated systems - stability and performance," in *Decision and Control, 2007 46th IEEE Conference on*, 2007, pp. 1251 –1256.

[6] R. Gottumukkal and V. K. Asari, "An improved face recognition technique based on modular PCA approach," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 429–436, Mar. 2004.

[7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711 –720, Jul. 1997.

[8] P. Comon, "Independent component analysis, A new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[9] K. Boutoustous, G. J. Laurent, E. Dedu, L. Matignon, J. Bourgeois, and N. Le Fort-Piat, "Distributed control architecture for smart surfaces," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 2018 –2024.

[10] E. Dedu, K. Boutoustous, and J. Bourgeois, "An Exhaustive Comparison Framework for Distributed Shape Differentiation in a MEMS Sensor Actuator Array," in *Parallel and Distributed Computing, 2008. ISPDC '08. International Symposium on*, 2008, pp. 429 –433.

[11] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[12] M. L. Braun, "Accurate Error Bounds for the Eigenvalues of the Kernel Matrix," *J. Mach. Learn. Res.*, vol. 7, pp. 2303–2328, Dec. 2006.