Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 410–422
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Faster and more robust point symmetry-based K-means algorithm

Kuo-Liang Chung*,[1], Jhin-Sian Lin

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei, Taiwan 10672, ROC

Received 6 August 2004; received in revised form 20 May 2005; accepted 21 September 2005

Abstract

Based on the recently published point symmetry distance (PSD) measure, this paper presents a novel PSD measure, namely symmetry similarity level (SSL) operator for K-means algorithm. Our proposed modified point symmetry-based K-means (MPSK) algorithm is more robust than the previous PSK algorithm by Su and Chou. Not only the proposed MPSK algorithm is suitable for the symmetrical intra-clusters as the PSK algorithm does, the proposed MPSK algorithm is also suitable for the symmetrical inter-clusters. In addition, two speedup strategies are presented to reduce the time required in the proposed MPSK algorithm. Experimental results demonstrate the significant execution-time improvement and the extension to the symmetrical inter-clusters of the proposed MPSK algorithm when compared to the previous PSK algorithm.

Keywords: Clustering; Inter-cluster; Intra-cluster; K-means algorithm; Point symmetry; Robustness; Speedup

1. Introduction

Clustering plays an important role in data analysis and pattern classification. It has many applications in codebook design [1,2], data compression [3], data mining [4], image segmentation [5], and so on. Clustering aims to partition a set of data points into some nonoverlapping subsets [6]. In the past three decades, many efficient clustering algorithms [1,7–15] have been developed. Among these developed clustering algorithms, the K-means algorithm is the oldest and the most popular one due to its simplicity and effectiveness.

In order to improve the performance of the K-means algorithm, several improved K-means algorithms have been developed in the past several years. In Ref. [1], instead of initially assigning each point to the closest center, Kövesi et al. presented a stochastic K-means algorithm to improve the clustering result. Based on the kd-tree data structure [16], Kanungo et al. [11] presented an improved K-means algorithm which can speed up the time performance while preserving the same clustering result as in the K-means algorithm. Based on code vector activity detection approach, Kaukoranta et al. [17] presented a faster K-means algorithm, which can be used to speed up the codebook construction by using the generalized Lloyd algorithm, and has the same clustering result as in the K-means algorithm. Considering the distribution of points for the case of symmetrical inter-clusters, Su and Chou [14] adopted the idea of symmetry feature [18–21] and presented an efficient point symmetry-based K-means (PSK) algorithm based on their proposed point symmetry distance (PSD) measure. Simulation results show that their proposed PSK algorithm has a better clustering result when compared to the K-means algorithm for symmetrical intra-cluster case. The motivation of this research are twofold: (1) presenting speedup strategies to reduce the execution time required in the previous PSK algorithm significantly and (2) presenting a new symmetry similarity level (SSL) operator to handle both the intra-cluster case and the inter-cluster case.

This paper first surveys the previous PSD measure [14] and explains why the PSD measure cannot handle the case

* Corresponding author. Tel.: +886 2 27376771; fax: +886 2 27376777.
E-mail address: klchung@cs.ntust.edu.tw (K.-L. Chung).
[1] Supported in part by the National Science Council of ROC under contracts NSC92-2213-E-011-079 and NSC93-2213-E-011-023.

0031-3203/$30.00
doi:10.1016/j.patcog.2005.09.015

of symmetrical inter-clusters well. Next, a novel SSL operator is presented to calculate the symmetry level between the data point p_i and the data point p_j relative to the cluster centroid c_k. When compared to the previous PSD measure, the proposed SSL operator not only can measure the orientation symmetry between p_i and p_j with respect to c_k as in the PSD measure, but also can measure the distance symmetry between the line segment $\overline{p_i c_k}$ and the line segment $\overline{c_k p_j}$. In addition, a simple constraint is suggested to enhance in the proposed SSL operator to handle both the case of symmetrical intra-clusters and the case of symmetrical inter-clusters. Further, two speedup strategies are presented to reduce the computation time required in the proposed modified PSK (MPSK) algorithm. In order to speed up the computation of the proposed SSL operator, a two-phase speedup strategy is presented. Since the proposed MPSK clustering algorithm includes the coarse-tuning step, which is realized by the K-means algorithm, a speedup strategy is also presented to improve the code vector activity detection approach [17] such that the coarse-tuning step can be performed in a faster way. Experimental results demonstrate the significant execution-time improvement and the extension to the symmetrical inter-clusters of the proposed MPSK algorithm when compared to the previous PSK algorithm by Su and Chou.

The remainder of this paper is organized as follows: In Section 2, the previous PSD measure is surveyed. In addition, one example is given to demonstrate the clustering power of the previous PSD measure for the case of symmetrical intra-clusters. In Section 3, the related problems that the PSD measure may occur are pointed out. In Section 4, the proposed SSL operator and the two-phase speedup strategy are presented. In Section 5, the proposed whole MPSK clustering algorithm is presented. In addition, a speedup strategy is described to speed up the coarse-tuning step in the MPSK algorithm. In Section 6, some experimental results are demonstrated to show the computational and robust advantages of the proposed MPSK clustering algorithm. In Section 7, some concluding remarks are addressed.

2. The past PSD measure

In this section, first the PSD measure by Su and Chou [14] is surveyed. Next, an example of symmetrical intra-clusters demonstrates the excellent applicability of the PSD measure.

In natural scenes, symmetry is an important feature [21,22]. Since the K-means algorithm cannot handle the case of intra-clusters well, recently, Su and Chou [14] presents a PSD measure and plugs it into the K-means algorithm to handle the case of intra-clusters efficiently.

Given N data points, say $\{p_i \mid \text{for} 1 \leqslant i \leqslant N\}$, using the K-means algorithm, let the temporary obtained K cluster centroids be denoted by $\{c_k \mid \text{for} 1 \leqslant k \leqslant K\}$. The PSD measure between the data point p_i and the data point p_j relative to the cluster centroid c_k is defined as

$$d_s(p_j, c_k) = \min_{\forall i \neq j \text{ and } 1 \leqslant i \leqslant N} \frac{\|(p_j - c_k) + (p_i - c_k)\|}{\|p_j - c_k\| + \|p_i - c_k\|}, \quad (1)$$

where $\|\cdot\|$ denotes the 2-norm distance.

An example is used to demonstrate how the PSD measure works well for the case of symmetrical intra-clusters. Fig. 1(a) illustrates two symmetrical intra-clusters, C_1 and C_2, where the data points are denoted by black dots and c_1 and c_2 are two centroids of the cluster C_1 and the cluster C_2, respectively. The positions of c_1 and c_2 are $c_1 = (5, 8)$ and $c_2 = (9.5, 8)$. p_1, p_2, and p_3 are three data points and their positions are $p_1 = (8, 7)$, $p_2 = (2, 9)$, and $p_3 = (12.5, 9.5)$, respectively. After running the K-means algorithm in Fig. 1(a), the data point p_1 in Fig. 1(a) would be assigned to the cluster C_2 because the data point p_1 is closer to c_2 than c_1. Fig. 1(b) shows the unsatisfactory clustering result by running the K-means algorithm in Fig. 1(a). In Fig. 1(b), the first unsatisfactory clustering result C_1 is denoted by squares and the second unsatisfactory clustering result C_2 is denoted by triangles. According to the visual inspection, the data point p_1 should be assigned to the cluster C_1 due to the symmetrical distribution of data points in C_1. The efficient PSD measure proposed by Su and Chou can indeed handle the case of symmetrical intra-clusters. By Eq. (1), for the data point p_1, it yields

$$d_s(p_1, c_1) = \frac{\|(p_1 - c_1) + (p_2 - c_1)\|}{\|(p_1 - c_1)\| + \|(p_2 - c_1)\|} = \frac{0}{\sqrt{10} + \sqrt{10}} = 0$$

and

$$d_s(p_1, c_2) = \frac{\|(p_1 - c_2) + (p_3 - c_2)\|}{\|(p_1 - c_2)\| + \|(p_3 - c_2)\|} = \frac{\sqrt{2.5}}{\sqrt{3.25} + \sqrt{11.25}} = 0.31.$$

Because $d_s(p_1, c_1) < d_s(p_1, c_2)$ and $d_s(p_1, c_1)$ is less than the specified threshold θ, e.g. $\theta = 0.18$ [14], the data point p_2 is said to be the most symmetrical point of p_1 relative to c_1, thus we have

$$p_2 = Arg\, d_s(p_1, c_1).$$

Consequently, assigning the data point p_1 to the cluster C_1 is a good decision. Fig. 1(c) depicts two satisfactory resulting clusters when applying the PSD measure to Fig. 1(a).

3. Possible problems occurred in the PSD measure

In this section, three observations are given to point out the three problems that the PSD measure may occur. The three possible problems existed in the PSD measure are (1) lacking the distance difference symmetry property, (2) leading to an unsatisfactory clustering result for the case of symmetrical

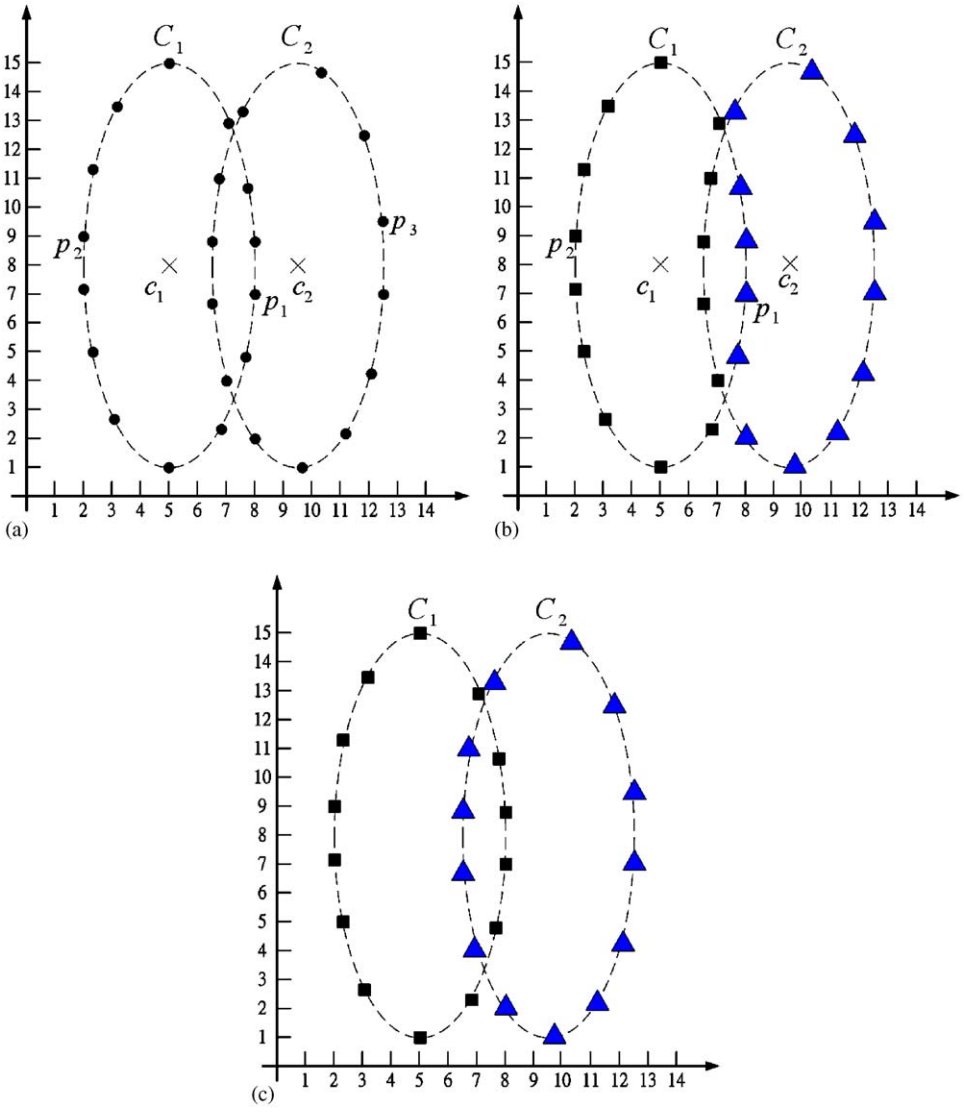

Fig. 1. An example to show the power of PSD measure for the case of intra-clusters. (a) Two symmetrical intra-clusters, C_1 and C_2; (b) two unsatisfactory resulting clusters after running the K-means algorithm in (a); (c) two satisfactory resulting clusters when applying the PSD measure to (a).

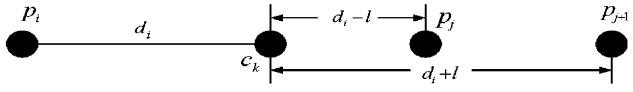

Fig. 2. An example for indicating the distance difference symmetry problem.

inter-clusters, and (3) lacking the closure property. The first and third properties will degrade the symmetrical robustness of PSD measure.

An example is given to illustrate the first possible problem. In Fig. 2, there are four data points, namely the centroid c_k and the three data points p_i, p_j, and p_{j+1} at locations $c_k = (0, 0)$, $p_i = (-d_i, 0)$, $p_j = (d_i - l, 0)$, and $p_{j+1} = (d_i + l, 0)$, respectively. It is known that the unit vectors of $\overrightarrow{p_i c_k}$, $\overrightarrow{c_k p_j}$, and $\overrightarrow{c_k p_{j+1}}$, i.e. $\overrightarrow{p_i c_k}/\|\overrightarrow{p_i c_k}\| = \overrightarrow{c_k p_j}/\|\overrightarrow{c_k p_j}\| = \overrightarrow{c_k p_{j+1}}/\|\overrightarrow{c_k p_{j+1}}\|$, are equivalent and the two related distance differences $\|\overline{p_i c_k} - \overline{c_k p_j}\|$ $(=l)$ and $\|\overline{p_i c_k} - \overline{c_k p_{j+1}}\|$ $(=l)$ are equivalent too. In Fig. 2, the most symmetrical point of p_i relative to the centroid c_k is the data point p_j or the data point p_{j+1}. By Eq. (1), we have

$$d_s(p_i, c_k) = \min\left\{\frac{\|(p_i - c_k) + (p_j - c_k)\|}{\|(p_i - c_k)\| + \|(p_j - c_k)\|}, \frac{\|(p_i - c_k) + (p_{j+1} - c_k)\|}{\|(p_i - c_k)\| + \|(p_{j+1} - c_k)\|}\right\} = \min\left\{\frac{l}{2d_i - l}, \frac{l}{2d_i + l}\right\} = \frac{l}{2d_i + l},$$

so the data point p_{j+1} is selected as the most symmetrical point of p_i relative to the centroid c_k. This indicates that the PSD measure favors the far data point when we have more than two candidate data points and this may degrade the symmetrical robustness. The first observation is given to indicate the first problem.

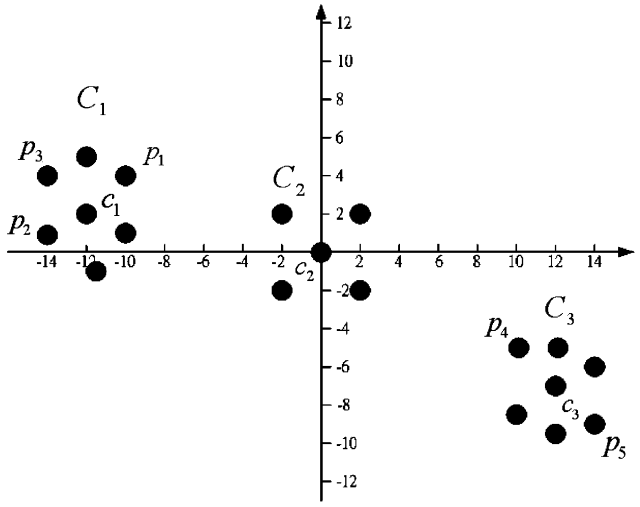

Fig. 3. An example of symmetrical intra/inter clusters.

Observation 1. *The PSD measure may lack the distance difference symmetry property, and it may degrade the symmetrical robustness.*

Next, a case of symmetrical inter-clusters is considered to explain why the PSD measure may generate an unsatisfactory clustering results. In Fig. 3, we have three symmetrical intra-clusters C_1, C_2 and C_3 where for each symmetrical intra-cluster, each data point relative to its own centroid c_k can find the most symmetrical data point in that cluster. In Fig. 3, C_1 and C_3 are two symmetrical inter-clusters because each data point belonging to C_1 (C_3) can find the most symmetrical data point belonging to C_3 (C_1) with respect to the centroid c_2.

In Fig. 3, there are 19 data points and the centroid of first cluster C_1 is c_1; the centroid of second cluster C_2 is c_2, and the centroid of third cluster C_3 is c_3. Let $p_1 = (-10, 4)$, $p_2 = (-14, 1)$, $p_3 = (-14, 4)$, $p_4 = (10, -5)$, $p_5 = (14, -9)$, $c_1 = (-12, 2)$, $c_2 = (0, 0)$, and $c_3 = (12, -7)$, then for the data point p_4, by Eq. (1), we have

$$d_s(p_4, c_1) = \min_{1 \leqslant i \leqslant 16, i \neq 4} \frac{\|(p_4 - c_1) + (p_i - c_1)\|}{\|p_4 - c_1\| + \|p_i - c_1\|} = \frac{\|(p_4 - c_1) + (p_3 - c_1)\|}{\|(p_4 - c_1)\| + \|(p_3 - c_1)\|} = \frac{\sqrt{425}}{\sqrt{533} + \sqrt{8}} = 0.80,$$

$$d_s(p_4, c_2) = \min_{1 \leqslant i \leqslant 16, i \neq 4} \frac{\|(p_4 - c_2) + (p_i - c_2)\|}{\|p_4 - c_2\| + \|p_i - c_2\|} = \frac{\|(p_4 - c_2) + (p_1 - c_2)\|}{\|(p_4 - c_2)\| + \|(p_1 - c_2)\|} = \frac{1}{\sqrt{125} + \sqrt{116}} = 0.05$$

and

$$d_s(p_4, c_3) = \min_{1 \leqslant i \leqslant 16, i \neq 4} \frac{\|(p_4 - c_3) + (p_i - c_3)\|}{\|p_4 - c_3\| + \|p_i - c_3\|} = \frac{\|(p_4 - c_3) + (p_5 - c_3)\|}{\|(p_4 - c_3)\| + \|(p_5 - c_3)\|} = \frac{0}{\sqrt{8} + \sqrt{8}} = 0.$$

From the above three PSD values, for the data point p_4, we have $d_s(p_4, c_1) = 0.8$, $d_s(p_4, c_2) = 0.05$, and $d_s(p_4, c_3) = 0$ corresponding to the centroids c_1, c_2, and c_3, respectively. Since $d_s(p_4, c_3)$ is the smallest, which is less than the specified threshold θ (=0.18), among the three PSD values, the data point p_4 is thus assigned to the cluster C_3 and it matches our visual inspection. The above example for the data point p_4 really reflects the power of the PSD proposed by Su and Chou [14] when handling the case of symmetrical intra-clusters.

We now consider the data point p_1, by Eq. (1), it yields

$$d_s(p_1, c_1) = \min_{1 \leqslant i \leqslant 16, i \neq 1} \frac{\|(p_1 - c_1) + (p_i - c_1)\|}{\|p_1 - c_1\| + \|p_i - c_1\|} = \frac{\|(p_1 - c_1) + (p_2 - c_1)\|}{\|(p_1 - c_1)\| + \|(p_2 - c_1)\|} = \frac{1}{\sqrt{8} + \sqrt{5}} = 0.20,$$

$$d_s(p_1, c_2) = \min_{1 \leqslant i \leqslant 16, i \neq 1} \frac{\|(p_1 - c_2) + (p_i - c_2)\|}{\|p_1 - c_2\| + \|p_i - c_2\|} = \frac{\|(p_1 - c_2) + (p_4 - c_2)\|}{\|(p_1 - c_2)\| + \|(p_4 - c_2)\|} = \frac{1}{\sqrt{116} + \sqrt{125}} = 0.05$$

and

$$d_s(p_1, c_3) = \min_{1 \leqslant i \leqslant 16, i \neq 1} \frac{\|(p_1 - c_3) + (p_i - c_3)\|}{\|p_1 - c_3\| + \|p_i - c_3\|} = \frac{\|(p_1 - c_3) + (p_5 - c_3)\|}{\|(p_1 - c_3)\| + \|(p_5 - c_3)\|} = \frac{\sqrt{481}}{\sqrt{605} + \sqrt{8}} = 0.80.$$

From the above three PSD values, since $d_s(p_1, c_2)$ (=0.05) is the smallest, which is less than the specified threshold θ (=0.18), among the three concerning PSD values, the data point p_1 should be assigned to the cluster C_2, but it conflicts our visual inspection. From the data distribution of C_1, instead of assigning p_1 to C_2, it will be better to assign the data point p_1 to the cluster C_1. We thus have the second observation.

Observation 2. *The PSD measure may lead to an unsatisfactory clustering result for the case of symmetrical inter-clusters.*

Before presenting the third problem, the closure property is defined as follows.

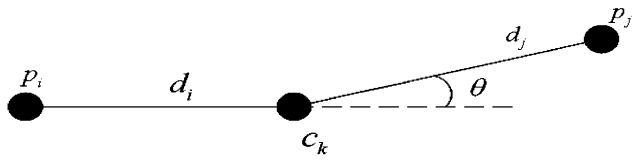

Fig. 4. An illustration for distance difference.

Property 1 (*Closure property*). *If the data point p_i is currently assigned to the cluster centroid c_k in the current iteration, the determined most symmetrical point p_j relative to c_k must have been assigned to c_k in the previous iteration.*

Based on the above example, the PSD measure tells us that the data point p_1 is currently assigned to the cluster centroid c_2 and the most symmetrical point of p_1 relative to the centroid c_2 is the data point p_4 $(=Arg\, d_s(p_1, c_2))$, but the data point p_4 has been assigned to the centroid c_3. Since the data point p_4 has not been assigned to the centroid c_2 before, it violates Property 1. We thus have the third observation.

Observation 3. *The PSD measure lacks the closure property.*

The PSD measure proposed by Su and Chou [14] is really a simple and efficient clustering algorithm for the case of symmetrical intra-clusters. However, the above three observations indicate that the PSD measure may degrade the symmetrical robustness, and may lead to unsatisfactory clustering results for the case of symmetrical inter-clusters.

4. The proposed SSL operator

In this section, a new SSL operator is presented to overcome the three possible problems occurred in the PSD measure. In addition, a two-phase speedup strategy is presented to reduce the computation load required in the proposed SSL operator.

4.1. Distance similarity level and orientation similarity level: DSL and OSL

From Observation 1, a new operator, which satisfies the distance difference symmetry property, is now defined and it is one component in the proposed SSL operator. In Fig. 4, c_k denotes the cluster centroid; p_i and p_j denote two related data points. Let $d_i = \overline{p_i c_k}$ and $d_j = \overline{p_j c_k}$, then the distance similarity level (DSL) operator for measuring the distance difference symmetry between the distance $\overline{p_i c_k}$ and the distance $\overline{p_j c_k}$ is defined by

$$DSL(p_i, c_k, p_j) = \begin{cases} 1 - \dfrac{|d_i - d_j|}{d_i} & \text{if } 0 \leqslant \dfrac{d_j}{d_i} \leqslant 2, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Note that if we replace the interval $0 \leqslant d_j/d_i \leqslant 2$ to the interval $0 \leqslant d_j/d_i \leqslant k$, $k > 2$, in Eq. (2), the number of examined symmetrical points will increase and the computational gain might be degraded.

Proposition 1. *It is true that* $0 \leqslant DSL(p_i, c_k, p_j) \leqslant 1$.

Proof. It is enough to consider three extremal cases for $d_j/d_i = 0$, $d_j/d_i = 2$ and $d_j/d_i = 1$. When $d_j/d_i = 0$, by Eq. (2), we have

$$\begin{aligned} DSL(p_i, c_k, p_j) &= 1 - \frac{|d_i - d_j|}{d_i} \\ &= 1 - \frac{|d_i - 0|}{d_i} \\ &= 0. \end{aligned}$$

When $d_j/d_i = 2$, by Eq. (2), we have

$$\begin{aligned} DSL(p_i, c_k, p_j) &= 1 - \frac{|d_i - d_j|}{d_i} \\ &= 1 - \frac{|d_i - 2 \times d_i|}{d_i} \\ &= 1 - 1 \\ &= 0. \end{aligned}$$

When $d_j/d_i = 1$, by Eq. (2), we have

$$\begin{aligned} DSL(p_i, c_k, p_j) &= 1 - \frac{|d_i - d_j|}{d_i} \\ &= 1 - \frac{|d_i - d_i|}{d_i} \\ &= 1 - 0 \\ &= 1. \end{aligned}$$

We complete the proof. □

After proving $0 \leqslant DSL(p_i, c_k, p_j) \leqslant 1$, from Eq. (2), it is clear that the larger the value of $DSL(p_i, c_k, p_j)$ is, the larger the DSL between d_i $(=\overline{p_i c_k})$ and d_j $(=\overline{p_j c_k})$ is. When $d_i = d_j$, we have $DSL(p_i, c_k, p_j) = 1$ and it means that the distance $\overline{p_u c_k}$ and the distance $\overline{p_j c_k}$ has the highest *DSL*.

When $\overline{p_i c_k} = d_i = 1.8$ and $\overline{p_j c_k} = d_j = 2$, we have $DSL(p_i, c_k, p_j) = 1 - \frac{0.2}{1.8} = 0.89$ and it indicates that the DSL between d_i and d_j is rather large. The following theorem confirms that the proposed DSL operator can preserve the distance difference symmetry property.

Theorem 1. *For $n = 1$, the DSL operator defined in Eq.* (2) *preserves the distance difference symmetry property.*

Proof. As shown in Fig. 2, it is known that $d_i = \overline{p_i c_k}$, $d_j = \overline{p_j c_k}$ $(=d_i - l)$ and $d_{j+1} = \overline{c_k p_{j+1}}$ $(=d_i + l)$, then for $n = 1$,

Eq. (2) yields

$$DSL(p_i, c_k, p_j) = 1 - \frac{|d_i - d_j|}{d_i} = 1 - \frac{|d_i - (d_i - l)|}{d_i} = 1 - \frac{l}{d_i}.$$

By the same arguments, we further have

$$DSL(p_i, c_k, p_{j+1}) = 1 - \frac{|d_i - d_{j+1}|}{d_i} = 1 - \frac{|d_i - (d_i + l)|}{d_i} = 1 - \frac{l}{d_i}.$$

Because of $DSL(p_i, c_k, p_j) = DSL(p_i, c_k, p_{j+1})$, the proposed DSL operator preserves the distance difference symmetry property. We complete the proof. □

Corollary 1. *For general n, the DSL operator defined in Eq. (2) preserves the distance difference symmetry property.*

Besides the DSL operator, we further present the second component used in the proposed SSL operator, say orientation similarity level (OSL). Applying the projection concept [23], the OSL between the two vectors, $v_i = \overrightarrow{p_i c_k} = (c_k - p_i)$ and $v_j = \overrightarrow{c_k p_j} = (p_j - c_k)$, is defined by $OSL'(p_i, c_k, p_j) = v_i \cdot v_j / \|v_i\| \|v_j\|$, $-1 \leqslant OSL'(p_i, c_k, p_j) \leqslant 1$. The larger the value $OSL'(p_i, c_k, p_j)$ is, the larger the OSL is. For example, given $p_i = (-1, 0)$, $p_j = (2, 0)$ and $c_k = (0, 0)$, we have $OSL' = \frac{2}{2} = 1$ and it indicates that the two vectors $\overrightarrow{p_j c_k}$ and $\overrightarrow{c_k p_j}$ have the same orientation. In order to confine the range of $OSL'(p_i, c_k, p_j)$ from 0 to 1, the operator $OSL'(p_i, c_k, p_j)$ is modified to be

$$OSL(p_i, c_k, p_j) = \frac{v_i \cdot v_j}{2\|v_i\|\|v_j\|} + 0.5. \quad (3)$$

It is known that $-1 \leqslant v_i \cdot v_j / \|v_i\| \|v_j\| \leqslant 1$, from Eq. (3), we have the following result.

Proposition 2. *It is true that* $0 \leqslant OSL(p_i, c_k, p_j) \leqslant 1$.

4.2. Symmetry similarity level: SSL

By Eqs. (2) and (3), we now combine the effect of $DSL(p_i, c_k, p_j)$ and the $OSL(p_i, c_k, p_j)$ to define a symmetry similarity level (SSL′) between the vector $\overrightarrow{p_i c_k}$ and $\overrightarrow{c_k p_j}$ and it is defined by

$$SSL'(p_i, c_k, p_j) = \sqrt{\frac{DSL^2(p_i, c_k, p_j) + OSL^2(p_i, c_k, p_j)}{2}} \quad (4)$$

for $1 \leqslant k \leqslant K$ and $1 \leqslant i \leqslant N$. Because of $0 \leqslant DSL(p_i, c_k, p_j) \leqslant 1$ and $0 \leqslant OSL(p_i, c_k, p_j) \leqslant 1$, it is easy to verify that $0 \leqslant SSL'(p_i, c_k, p_j) \leqslant 1$ is held. The larger the value $SSL'(p_i, c_k)$ is, the larger the SSL is. For the data point p_i with respect to the cluster centroid c_k, the operator $SSL''(p_i, c_k)$ $(=\max_{1 \leqslant j \leqslant N} SSL'(p_i, c_k, p_j))$ is used to find the most symmetrical data point p_j relative to c_k such that the value of $SSL'(p_i, c_k, p_j)$ is maximal among all the concerning data points. Naturally, the operator $SSL(p_i, c_k)$ is written by

$$SSL''(p_i, c_k) = \max_{1 \leqslant j \leqslant N} SSL'(p_i, c_k, p_j) = \max_{1 \leqslant j \leqslant N} \sqrt{\frac{DSL^2(p_i, c_k, p_j) + OSL^2(p_i, c_k, p_j)}{2}}. \quad (5)$$

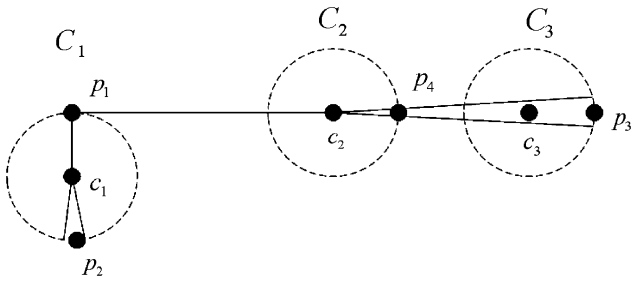

Fig. 5. Violation of closure property for $SSL''(p_i, c_k)$.

Although the $SSL''(p_i, c_k)$ operator defined in Eq. (4) satisfies the distance symmetry property, the $SSL''(p_i, c_k)$ still lacks the closure property (see Property 1). For example, in Fig. 5, suppose we have $p_1 = (-10, 0)$, $p_2 = (-9, -4)$, $p_3 = (10, 0)$, $p_4 = (2, 0)$, $c_1 = (-10, -2)$, $c_2 = (0, 0)$, and $c_3 = (8, 0)$. In addition, suppose p_3 belongs to C_3 and p_2 belongs to C_1. For the data point p_1 relative to centroid c_1, by Eq. (5), it yields

$$SSL''(p_1, c_1) = \max_{1 \leqslant j \leqslant 4, j \neq 1} \sqrt{\frac{DSL^2(p_1, c_1, p_j) + OSL^2(p_1, c_1, p_j)}{2}} = \sqrt{\frac{DSL^2(p_1, c_1, p_2) + OSL^2(p_1, c_1, p_2))}{2}} = \sqrt{\frac{0.77 + 0.9}{2}} = 0.91$$

and we have $p_2 = Arg\, SSL''(p_1, c_1)$. For the data point p_1 relative to centroid c_2, Eq. (5) yields

$$SSL''(p_1, c_2) = \max_{1 \leqslant j \leqslant 4, j \neq 1} \sqrt{\frac{DSL^2(p_1, c_2, p_j) + OSL^2(p_1, c_2, p_j)}{2}} = \sqrt{\frac{DSL^2(p_1, c_2, p_3) + OSL^2(p_1, c_2, p_3))}{2}} = \sqrt{\frac{1 + 1}{2}} = 1$$

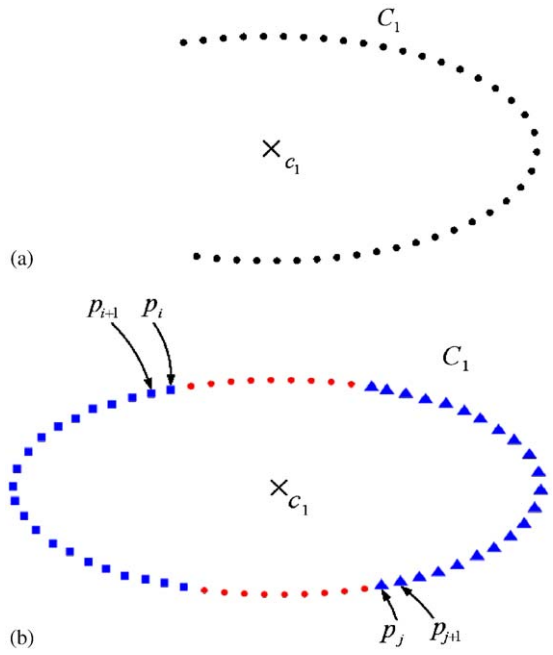

Fig. 6. An observation for cluster with closure property: (a) An incomplete symmetrical intra-cluster; (b) reconstructed complete symmetrical intra-cluster.

and we have $p_3 = Arg\, SSL''(p_1, c_2)$. For the data point p_1 relative to centroid c_3, Eq. (5) yields

$$SSL''(p_1, c_3) = \max_{1 \leqslant j \leqslant 4, j \neq 1} \sqrt{\frac{DSL^2(p_1, c_3, p_j) + OSL^2(p_1, c_3, p_j)}{2}} = \sqrt{\frac{DSL^2(p_1, c_3, p_3) + OSL^2(p_1, c_3, p_3))}{2}} = \sqrt{\frac{0.01 + 1}{2}} = 0.71$$

and we have $p_3 = Arg\, SSL''(p_1, c_2)$. Because of $SSL''(p_1, c_2) > SSL''(p_1, c_1) > SSL''(p_1, c_3)$, the data point p_1 would be assigned to the cluster C_2. Unfortunately, the data point p_3 $(= Arg\, SSL''(p_1, c_2)) \in C_3$, which is the most symmetrical point of p_1 relative to c_2, does not belong to C_2 originally. Assigning p_1 to C_2 makes the movement of the c_2 some large. It indicates that the above SSL'' operator lacks the closure property and it would result in an unsatisfactory clustering result. In next subsection, the SSL'' operator in Eq. (5) is modified to be the proposed SSL operator satisfying the closure property.

4.3. A constraint for closure property

Our main idea is based on the observation in Fig. 6. Fig. 6(a) illustrates an incomplete symmetrical intra-cluster. According to the symmetry property, the data point p_i in Fig. 6(b), which is not in the cluster C_1 originally, the most symmetrical data point of p_i relative to the centroid c_1 is the data point p_j. Thus we assign the data point p_i to the centroid c_1. Since the data point p_j belongs to the cluster C_1, the assignment of p_i to the cluster C_1 is a reasonable assignment from our visual system. Therefore, for the testing data point p_i, we restrict the candidate symmetrical points p_j that must belong to the cluster centroid c_1 when computing $SSL(p_i, c_1)$. For the data point p_i relative the centroid c_1, this restriction can help us to search more suitable symmetrical point p_j and it leads to a computation-saving effect because we ignore the candidate most symmetrical point p_j which is not in the cluster C_1.

Following the above restriction and the operator $SSL''(p_i, c_k)$ in Eq. (5), the SSL with closure property for the data point p_i relative to the centroid c_k can be modified as

$$SSL(p_i, c_k) = \max_{p_j \varepsilon C_k} \sqrt{\frac{DSL^2(p_i, c_k, p_j) + OSL^2(p_i, c_k, p_j)}{2}} \quad (6)$$

for $1 \leqslant k \leqslant K$ and $1 \leqslant i \leqslant N$. Using the proposed SSL operator in Eq. (6) for the data point p_1, Fig. 5 yields

$$SSL(p_1, c_1) = \sqrt{\frac{DSL^2(p_1, c_1, p_2) + OSL^2(p_1, c_1, p_2))}{2}} = \sqrt{\frac{0.77 + 0.9}{2}} = 0.91,$$

$$SSL(p_1, c_2) = \sqrt{\frac{DSL^2(p_1, c_2, p_4) + OSL^2(p_1, c_2, p_4))}{2}} = \sqrt{\frac{0.04 + 1}{2}} = 0.72$$

and

$$SSL(p_1, c_3) = \sqrt{\frac{DSL^2(p_1, c_3, p_4) + OSL^2(p_1, c_2, p_3))}{2}} = \sqrt{\frac{0.01 + 1}{2}} = 0.71.$$

Because of $SSL(p_1, c_1) > SSL(p_1, c_2) > SSL(p_1, c_3)$ and the most symmetrical point p_2 $(= Arg\, SSL(p_1, c_1))$ have been assigned to C_1, the data point p_1 would be assigned to the cluster C_1. This clustering result meets our visual inspection.

4.4. Two-phase approach to speed up the computation of SSL

After presenting the proposed SSL operator to measure the SSL, it is known that the proposed SSL operator has some good properties such as the closure property and the robustness property. Checking Eq. (6) again, since the SSL operator contains two components, i.e. the DSL operator and the OSL operator, two thresholds, namely α and β, must be specified to speed up the computation of the proposed SSL

operator. The main idea of the first speedup strategy is that if the candidate symmetrical data point violates the threshold α or the threshold β, the candidate symmetrical data point can be discarded in the early stage; otherwise, Eq. (6) is computed further. On the other hand, if the calculated value of $DSL(p_i, c_k, p_j)$ is less than the threshold α or the calculated value of $OSL(p_i, c_k, p_j)$ is less than the threshold β, we do not need to compute Eq. (6) further, i.e. the computations of one squared root operation, one addition, one division, and one maximal selection can be discarded. Plugging the above speedup strategy into the whole clustering algorithm can lead to a significant computation-saving effect.

Now the threshold α is defined. Let us return to Fig. 4 again. Suppose we confine the tolerance rate of the distance difference to be less than 40% between the two line segments d_i and d_j, i.e. $|d_i - d_j|/d_i \leqslant 0.4$, and confine the tolerance rate of the angle orientation to be less than 20° between the two vectors v_1 $(=c_k - p_i)$ and v_2 $(=p_j - c_k)$. According to the definition of $DSL(p_i, c_k, p_j)$ (see Eq. (2)), the threshold α can be defined by

$$\begin{aligned}\alpha &= 1 - \frac{|d_i - d_j|}{d_i}\\ &= 1 - 0.4\\ &= 0.6.\end{aligned}$$

According to the definition of $OSL(p_i, c_k, p_j)$ (see Eq. (3)), the threshold β can be defined by

$$\begin{aligned}\beta &= \frac{v_i \cdot v_j}{2\|v_i\|\|v_j\|} + 0.5\\ &= \frac{\|v_i\|\|v_j\|\cos\theta}{2\|v_i\|\|v_j\|} + 0.5\\ &= \frac{\cos\theta}{2} + 0.5\\ &= \frac{\cos 20^\circ}{2} + 0.5\\ &= \frac{0.94}{2} + 0.5\\ &= 0.97.\end{aligned}$$

For this case, in the first phase, if the calculated value of $DSL(p_i, c_k, p_j)$ is less than the threshold α (=0.6) or the calculated value of $OSL(p_i, c_k, p_j)$ is less than the threshold β (=0.97), the computation of the final value for $SSL(p_i, c_k, p_j)$ can be discarded in the second phase.

5. The proposed MPSK algorithm

Based on the proposed two-phase SSL operator described in Section 4, this section presents the proposed MPSK algorithm. Besides being suitable for the symmetrical intra-clusters as the PSK algorithm [14] does, the proposed MPSK algorithm is also suitable for the symmetrical inter-clusters.

5.1. The proposed MPSK algorithm

The proposed five-step MPSK algorithm is listed below. Specifically, in order to speed up the coarse-tuning step in the MPSK algorithm, a modified version of the code vector activity detection approach [17] is presented in next subsection. The complete MPSK algorithm is presented as follows.

Step 1: (Initialization). Give N data points, we choose K data points randomly as the initial cluster centroids.

Step 2: (Coarse-tuning). Apply the K-means algorithm to update the selected K cluster centroids until the K cluster centroids are converged to fixed points or the terminating criteria is satisfied.

Step 3: (Fine-tuning).

Step 3.1: (Pruning impossible candidate symmetrical data points). For each data point p_i, find out the set Sb_{ik} of all possible candidate symmetrical data points p_j's relative to each c_k such that $DSL(p_i, c_k, p_j) \geqslant \alpha$ $(=0.6)$ and $OSL(p_i, c_k, p_j) \geqslant \beta$ $(=0.97)$ are held, $1 \leqslant i, j \leqslant N$ and $1 \leqslant k \leqslant K$, where p_j belongs to the kth cluster already.

Step 3.2: (Searching the most symmetrical data points). For the data point p_i, find out the cluster centroid c_{k^*} such that the value of $SSL(p_i, c_{k^*})$ is the largest and the most symmetrical point p_j relative to c_{k^*} belongs to Sb_{ik^*}. If such a cluster centroid c_{k^*} does not exist, then the data point p_i would be assigned to the k^{**}th cluster with the shortest Euclidean distance; otherwise the data point p_i is assigned to the k^*th cluster.

Step 4: (Updating the cluster centers). After assigning these data points p_i's to the corresponding clusters, the centroids of these corresponding clusters are updated by

$$c_k^{new} = \frac{1}{|C_k|}\sum_{p_i \in C_k} p_i,$$

where C_k is the set containing the data points which have been assigned to the cluster centroid c_k and $|C_k|$ is the number of data points in C_k.

Step 5: (Continuation or termination). If all the centroids are converged to some fixed points or the number of iterations is larger than the allowable bound, stop the algorithm; otherwise go to Step 3.

Besides the two-phase speedup strategy in Step 3.1, in next subsection, a modified version of the code activity detection approach [17] is presented to speed up the computation of Step 2, i.e. the coarse-tuning step.

5.2. The speedup strategy for coarse-tuning step

According to the activity of the centroids investigated by Kaukoranta et al. [17], the centroids in each iteration are

classified into two types—active centroid and static centroid. When one current centroid in iteration i is different from the previous centroid in iteration $i-1$, the current centroid is called an active centroid; otherwise, the centroid is called a static centroid. The data points relative to its own current centroid is classified into four types—static data point, balanced data point, farther data point, and closer data point. The formal definitions of the four types are given below.

Definition 1 (*Static data point*). When the current centroid is static, the data point belonging to that centroid is called a static data point.

Definition 2 (*Balanced data point*). Let the distance between the data point and its own current centroid be denoted by d_c and let the distance between the data point and its own previous centroid be denoted by d_p, then the data point is called a static data point when its own current centroid is active and $d_c = d_p$ is held.

Definition 3 (*Farther data point*). The data point is called a farther data point when its own current centroid is active and $d_c > d_p$ is held.

Definition 4 (*Closer data point*). The data point is called a closer data point when its own current centroid is active and $d_c < d_p$ is held.

If the data point belonging to the centroid c_{k^*} is static or balanced, it is unnecessary to consider the other static centroids in the current iteration since only the other active centroids may move closer to that data point than the centroid c_{k^*} does. If the data point belonging to the centroid c_{k^*} is farther, it has to consider all centroids in the current iteration. If the data point belonging to the centroid c_{k^*} is closer, it only needs to consider the active centroids in the current iteration since only the active centroids may move closer to the data point. According to the type of the data point p_i and the activity information of all centroids, the data point p_i can determine its own nearest centroid efficiently.

Definition 5 (*Moving distance of centroid*). Let the centroid c_k be denoted by c_k^p in the previous iteration and be denoted by c_k^c in the current iteration, the moving distance from c_k^p to c_k^c is defined as $d_{c_k^p \to c_k^c}$.

After describing the efficient method by Kaukoranta et al. [17], in what follows, the proposed modified method, which can be used to speed up Step 2, i.e. the coarse-tuning step in the proposed MPSK algorithm, is presented now. We focus on the closer data point p_i belonging to the centroid c_{k^*}. In Ref. [17], when the type of data point is closer, all the active centroids must be considered. As shown in Fig. 7, the six white circles, c_1, c_2, c_3, c_4, c_5 and c_6, denote

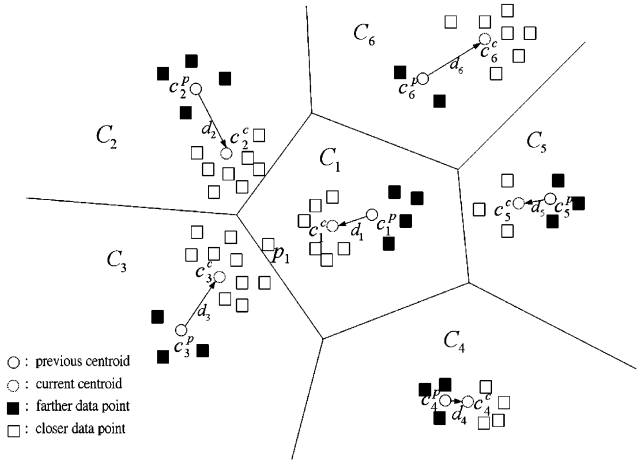

Fig. 7. An example for the closer data point p_1.

the six centroids in the previous iteration. A Voronoi diagram is depicted to separate the six clusters where within each cluster, the squares denote the related data points. In the current iteration, suppose each previous centroid is moved from the white circle to the dash-white circle and by Definition 5, the moving distance is $d_{c_k^p \to c_k^c}$. For the cluster C_1, the type of data point p_1 is closer because the current centroid moves closer to p_1 and the moving distance is $d_{c_1^p \to c_1^c} = d_1$. In the cluster C_1, the while (black) squares indicate that these data points are closer (farther) types. The main contribution of this subsection is that for the closer data point p_i relative to the centroid c_k, instead of searching the nearest centroid from all current active centroids [17], the proposed search strategy only considers the current active centroid whose moving distance between the current centroid and the previous centroid is larger than the distance $d_{c_k^p \to c_k^c}$.

According to the above search strategy, in Fig. 7, for the closer data point p_1, instead of considering the current centroids, c_1^c, c_2^c, c_3^c, c_4^c, c_5^c and c_6^c, the proposed search strategy only consider the current centroids, c_1^c, c_2^c, c_3^c, c_6^c, since $d_{c_4^p \to c_4^c} < d_{c_1^p \to c_1^c}$ and $d_{c_5^p \to c_5^c} < d_{c_1^p \to c_1^c}$.

6. Experimental results

All experiments are performed on a Pentium 4 personal computer with 2G MHz and the Windows 2000 environment. The programming language is the Borland C++ Builder version 5. Based on four sets of testing data points, the clustering effect and the execution-time performance between the previous PSK algorithm and the proposed MPSK algorithm are demonstrated.

The first data set contains ring-shaped, compact circle, and linear clusters, as shown in Fig. 8(a). This data set contains 612 data points. After running the K-means algorithm, the clustering result is shown in Fig. 8(b). Fig. 8(c) illustrates

the clustering result by using the PSK algorithm. Fig. 8(d) shows the clustering result by using the proposed MPSK algorithm. From Fig. 8(b)–(d), it is observed that for the first data set, the clustering performance of the K-means algorithm is the worst because some data points in the compact circle cluster are assigned to the ring-shaped cluster. The PSK algorithm and the proposed MPSK algorithm have the same clustering results. The second data set is shown in Fig. 9(a) which contains two crossed ellipsoidal shells. There are 628 data points in Fig. 9(a). Fig. 9(b)–(d) show the clustering result of the K-means algorithm, the PSK algorithm, and the proposed MPSK algorithm, respectively. In this example, the K-means algorithm has the worst clustering performance. The PSK algorithm and the MPSK algorithm also have the same clustering performance.

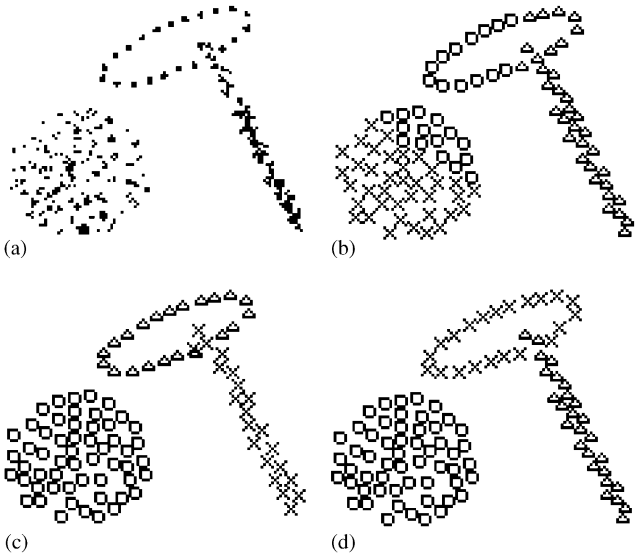

Fig. 8. Clustering performance comparison for the first data set. (a) The data set contains a combination of ring-shaped, compact circle, and linear clusters. (b) The clustering result obtained by using the K-means algorithm. (c) The clustering result obtained by using the PSK algorithm. (d) The clustering result obtained by using the proposed MPSK algorithm.

Fig. 10(a) illustrates the third data set which contains three compact circles. Among the three clustering results, the K-means algorithm and the proposed MPSK algorithm have the same clustering results (see Fig. 10(b) and (d)) which meet our visual inspection. For the PSK algorithm, some data points denoted by triangles in the leftmost compact circle have been assigned to the second compact circle (see Fig. 10(c)) and it violates our visual inspection.

The final data set contains 521 data points which are distributed on two compact circles and two crossed ellipsoidal shells as shown in Fig. 11(a). After running the K-means algorithm in Fig. 11(a), there are several misclassified data points on the crossed ellipsoidal shells, which are denoted by those triangles on the lower part of one ellipsoidal shell

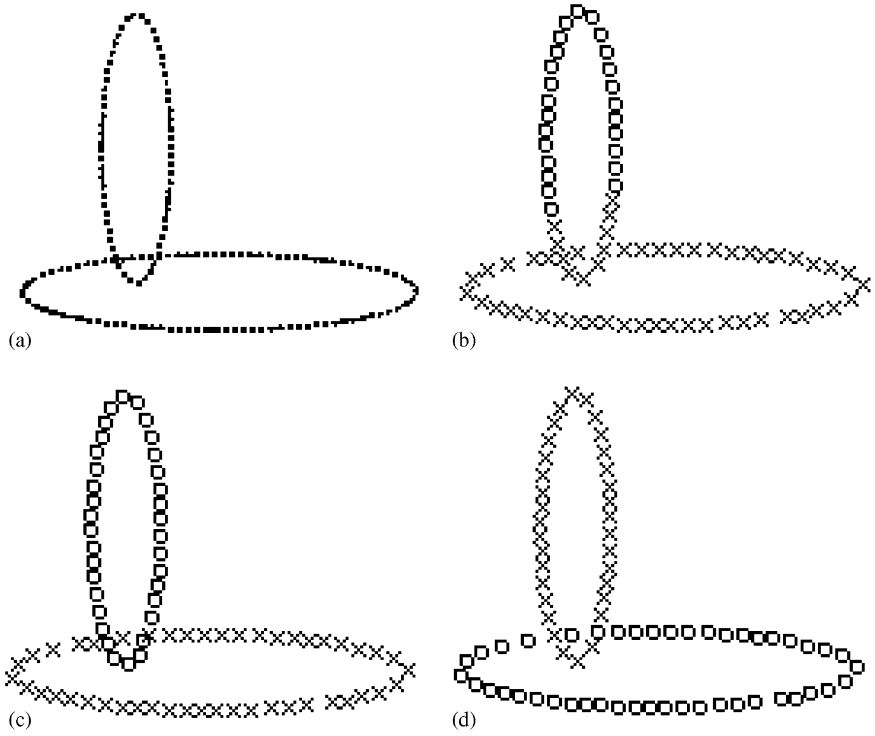

Fig. 9. Clustering performance comparison for the second data set. (a) The data set contains two ellipsoidal shells. (b) The clustering result obtained by using the K-means algorithm. (c) The clustering result obtained by using the PSK algorithm. (d) The clustering result obtained by using the proposed MPSK algorithm.

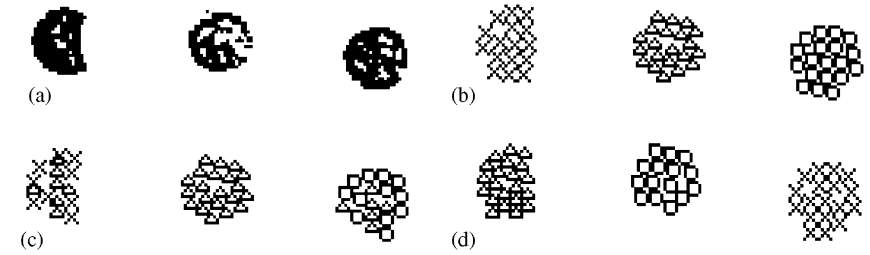

Fig. 10. Clustering performance comparison for the third data set. (a) The data set contains three compact circles. (b) The clustering result obtained by using the K-means algorithm. (c) The clustering result obtained by using the PSK algorithm. (d) The clustering result obtained by using the proposed MPSK algorithm.

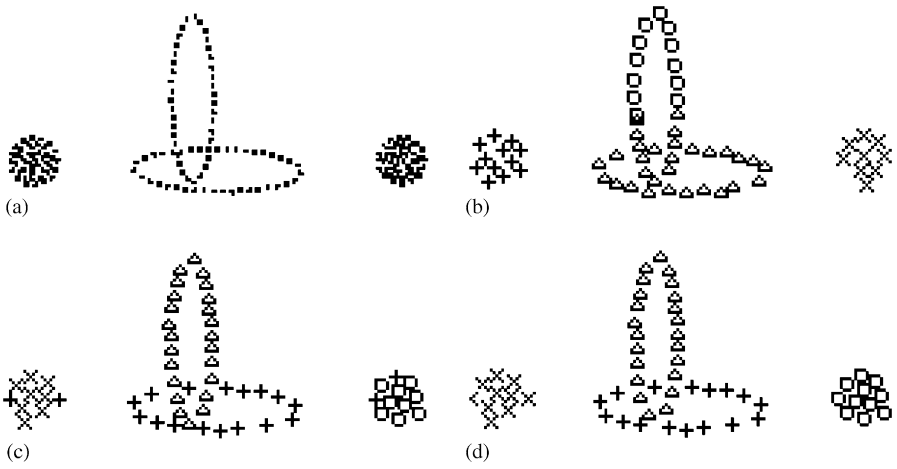

Fig. 11. Clustering performance comparison for the fourth data set. (a) The data set contains two compact circles and two crossed ellipsoidal shells. (b) The clustering result obtained by using the K-means algorithm. (c) The clustering result obtained by using the PSK algorithm. (d) The clustering result obtained by using the proposed MPSK algorithm.

Table 1
Execution-time performance comparison between the PSK algorithm and the proposed MPSK algorithm

Data set	Image size	Number of data points	Execution-time (10^{-3} s)		Improvement ratio (%)
			PSK	MPSK	
First data set	150 × 97	612	5386	210	92
Second data set	154 × 113	628	4411	216	95
Third data set	150 × 97	863	7744	340	96
Fourth data set	150 × 96	521	10627	731	93
Average					94

(see Fig. 11(b)). Fig. 11(c) shows the clustering result by using the PSK algorithm. Obviously, several misclassified data points which are denoted by '+' symbols over the two compact circles, are assigned to the bottom ellipsoidal shell and it violates our visual inspection. For the same data set, the proposed MPSK algorithm does get a satisfactory clustering result.

Based on the same four artificial data sets, Table 1 demonstrates the time performance comparison between the PSK algorithm and the proposed MPSK algorithm. Let the execution-time improvement ratio be measured by $(T_{PSK} - T_{MPSK})/T_{PSK}$ where T_{PSK} and T_{MPSK} denoted the execution-time required in the PSK algorithm and the proposed MPSK algorithm, respectively. Table 1 indicates that the execution-time improvement ratio of the proposed MPSK algorithm over the PSK algorithm is 94% in average.

Besides the above experiments, three more complicated data sets are used to illustrate the effect of our proposed

algorithm. Fig. 12(a) depicts an input real image. Fig. 12(b) depicts two crossed rubber bands. After running the K-means algorithm, Fig. 12(c) demonstrates two unsatisfactory clusters and the corresponding two centroids. After running our proposed *MPSK* algorithm, Fig. 12(d) demonstrates the satisfactory clustering result.

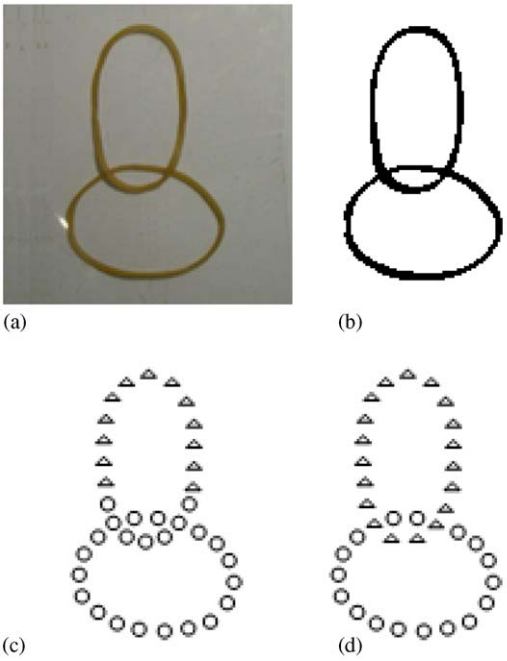

Fig. 12. Clustering performance for the fifth data set. (a) The input real image. (b) The data set contains two crossed rubber bands. (c) The clustering result obtained by using the K-means algorithm. (d) The clustering result obtained by using the proposed MPSK algorithm.

As shown in Figs. 13 and 14, finally the sixth data set and the seventh data set are used to evaluate the relevant performance. Figs. 13(a) and 14(b) depict two input real images. Fig. 13(b) (Fig. 14(b)) contains two crossed rubber bands and two separated rubber bands (two separated rubber bands and two sets of two crossed rubber bands). After running the K-means algorithm, Figs. 13(c) and 14(c) demonstrate the unsatisfactory clustering results and the corresponding centroids. After running our proposed *MPSK* algorithm, Figs. 13(d) and 14(d) demonstrate the relevant satisfactory clustering results.

From the above seven data sets, four artificial data sets and three real data sets, experimental clustering results indicate that our proposed MPSK algorithm works well to generate satisfactory clustering results.

7. Conclusions

The proposed MPSK algorithm has been presented. In the proposed MPSK algorithm, a new SSL operator is presented to measure the symmetry similarity level and it satisfies closure property and robustness property. In addition, two speedup strategies are presented. Plugging the proposed SSL operator and the two speedup strategies into the proposed clustering algorithm, it leads to a faster and more robust clustering algorithm to handle the intra/inter symmetrical clusters when compared the previous PSK algorithm. Experimental results confirm the computational and robust effects of the proposed MPSK algorithm.

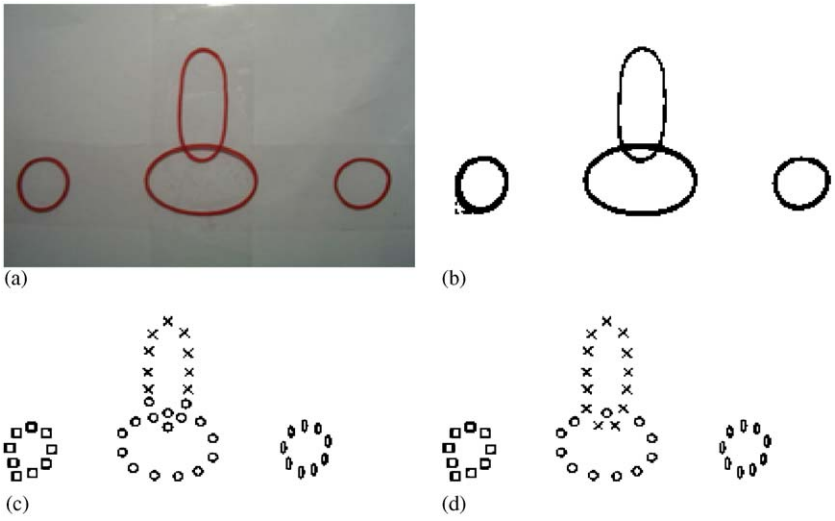

Fig. 13. Clustering performance for the sixth data set. (a) The input real image. (b) The data set contains two crossed rubber bands and two separated rubber bands. (c) The clustering result obtained by using the K-means algorithm. (d) The clustering result obtained by using the proposed MPSK algorithm.

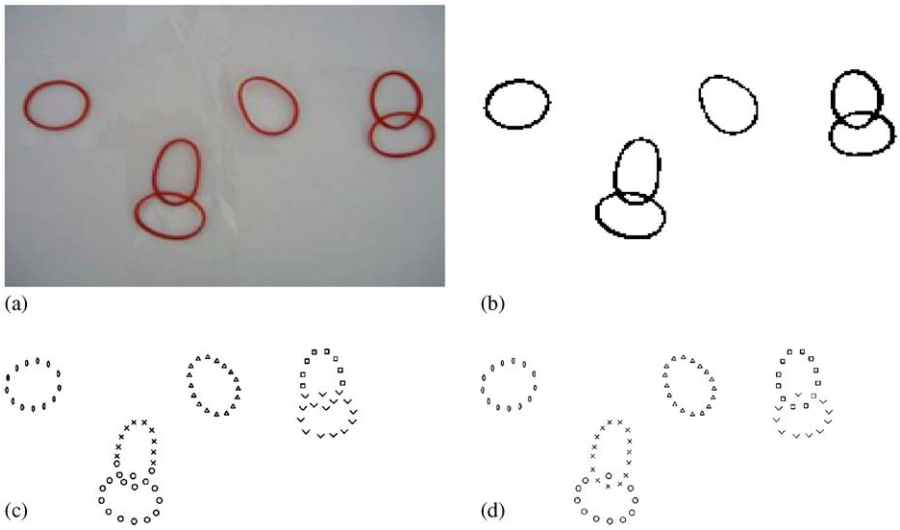

Fig. 14. Clustering performance for the seventh data set. (a) The input real image. (b) The data set contains two separated rubber bands and two sets of two crossed rubber bands. (c) The clustering result obtained by using the K-means algorithm. (d) The clustering result obtained by using the proposed MPSK algorithm.

References

[1] B. Kövesi, J.M. Boucher, S. Saoudi, Stochastic K-means algorithm for vector quantization, Pattern Recognition Lett. 22 (2001) 603–610.
[2] K.K. Paliwal, V. Ramasubramanian, Comments on modified K-means algorithm for vector quantizer design, IEEE Trans. Image Process. 9 (11) (2000) 1964–1967.
[3] K. Krishna, K.R. Ramakrishnan, M.A.L. Vector quantization using generic K-means algorithm for image compression, International Conference on Information, Communications and Signal Processing, Singapore, September 1997, pp. 9–12.
[4] R.T. Ng, J. Han, CLARANS: a method for clustering objects for spatial data mining, IEEE Trans. Knowl. Data Eng. 14 (5) (2002) 1003–1016.
[5] Y.A. Tolias, S.M. Panas, Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership functions, IEEE Trans. Syst. Man Cybern.—Part A: Syst. Humans 28 (3) (1998) 359–369.
[6] A.K. Jain, R.C. Dubes, Algorithms for Clustering, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[7] P. Bajcsy, N. Ahuja, Location- and density-based hierarchical clustering using similarity analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (9) (1998) 1011–1015.
[8] W.H.E. Day, H. Edelsbrunner, Efficient algorithms for agglomerative hierarchical clustering methods, J. Classification 1 (1) (1984) 7–24.
[9] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[10] J. Hartigan, Clustering Algorithms, Wiley, New York, 1975.
[11] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient K-means clustering algorithm: analysis and implementation, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 881–892.
[12] X. Li, Parallel algorithms for hierarchical clustering and cluster validity, IEEE Trans. Pattern Anal. Mach. Intell. 12 (11) (1990).
[13] E.M. Rasmussen, P. Willett, Efficiency of hierarchic agglomerative clustering using the ICL distributed array processor, J. Doc. 45 (1989) 1–24.
[14] M.C. Su, C.H. Chou, A modified version of the K-means algorithm with a distance based on cluster symmetry, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 674–680.
[15] C.J. Veenman, M.J.T. Reinders, E. Backer, A maximum variance cluster algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1273–1280.
[16] J.L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (1975) 509–517.
[17] T. Kaukoranta, P. Fränti, O. Nevalainen, A fast exact GLA based on code vector activity detection, IEEE Trans. Image Process. 9 (8) (2000).
[18] K. Kanatani, Comments on “symmetry as a continuous feature”, IEEE Trans. Pattern Anal. Mach. Intell. 19 (3) (1997) 246–247.
[19] W. Miller, Symmetry Groups and Their Applications, Academic press, London, 1972.
[20] H. Weyl, Symmetry, Princeton University Press, Princeton, NJ, 1952.
[21] H. Zabrodsky, S. Peleg, D. Avnir, Symmetry as a continuous feature, IEEE Trans. Pattern Anal. Mach. Intell. 17 (12) (1995) 1154–1166.
[22] D. Reisfeld, H. Wolfsow, Y. Yeshurun, Context-free attentional operators: the generalized symmetry transform, Int. Comput. Vision 14 (1995) 119–130.
[23] K. Hoffman, R. Kunze, Linear Algebra, Prentice-Hall, New Jersey, 1961.

About the Author—KUO-LIANG CHUNG received his B.S. M.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University in 1982, 1984, and 1990, respectively. From 1984 to 1986, he completed his military service. From 1986 to 1987, he was a research assistant in the Institute of Information Science, Academic Sinica. He has been a Professor and the Chairman in the Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology since 1995 and 2003, respectively. Prof. Chung received the Distinguished Professor Award from the Chinese Institute of Engineers in May 2001 and received the distinguished research award (2004–2007) from the National Science Council, ROC. He has been an IEEE senior member since 2001. His research interests include image compression, image processing, video compression, pattern recognition, coding theory, algorithms, and multimedia applications.

About the Author—JHIN-SIAN LIN received his B.S. degree in Applied Mathematics from Tatung University, ROC and received the M.S. degree from Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology. His research interests include image processing and pattern recognition.