A Modified K-Means Algorithm for Circular Invariant Clustering

Dimitrios Charalampidis, *Member*, *IEEE*

Abstract—Several important pattern recognition applications are based on feature vector extraction and vector clustering. Directional patterns are commonly represented by rotation-variant vectors $\boldsymbol{F}_d$ formed from features uniformly extracted in M directions. It is often desirable that pattern recognition algorithms are invariant under pattern rotation. This paper introduces a distance measure and a K-means-based algorithm, namely, Circular K-means (CK-means) to cluster vectors containing directional information, such as $\boldsymbol{F}_d$, in a circular-shift invariant manner. A circular shift of $\boldsymbol{F}_d$ corresponds to pattern rotation, thus, the algorithm is rotation invariant. An efficient Fourier domain representation of the proposed measure is presented to reduce computational complexity. A split and merge approach (SMCK-means), suited to the proposed CK-means technique, is proposed to reduce the possibility of converging at local minima and to estimate the correct number of clusters. Experiments performed for textural images illustrate the superior performance of the proposed algorithm for clustering directional vectors $\boldsymbol{F}_d$, compared to the alternative approach that uses the original K-means and rotation-invariant feature vectors transformed from $\boldsymbol{F}_d$.

Index Terms—Clustering, algorithms, similarity measures.

1 Introduction

Texture segmentation and object recognition and grouping have attracted great interest from researchers over the past years. Texture segmentation finds a large number of applications in medicine, remote sensing [21], and industry. Object grouping is useful for identifying similar objects in images. The term *object* may be used to describe high level structures such as vehicles and buildings or low level image components such as edges and junctions. Both texture segmentation and object grouping require that the image be segmented into several regions. Generally, the first step in the segmentation process is feature extraction: Multiple features are extracted from image regions to form vectors representing those regions. Feature vector clustering is usually the second step in the segmentation process. Since "similar" vectors correspond to similar regions, clustering results in a segmented image. Overviews of clustering algorithms can be found in [1], [2], [3], [4], [5], [6]. One of the most widely used clustering algorithms is K-means and its variations [6], [7]. Other clustering approaches include neural network-based techniques [8], [9], [10], [11].

In most applications, it is highly desirable that the segmentation process is invariant under image rotation. Commonly, one searches for features that are rotation invariant. Feature invariance under rotation is achieved either by directly applying a rotation invariant feature extraction technique [22] or by transforming rotation variant into rotation invariant vectors. In the second case, M features, $\{f_m, m = 0, 1, \ldots, M-1\}$, may be uniformly extracted from M directions $\theta_m = 360^\circ m/M$ to form a rotation variant M-dimensional vector $\boldsymbol{F}_d = [f_0, \ldots, f_{M-1}]$. The steerability property of filters can also be used for multidirectional information extraction, resulting in vectors in the form of $\boldsymbol{F}_d$, in order to characterize image components [20]. Traditionally, this feature vector may be transformed into a rotational invariant vector [12], [13], [14], [15], [24].

This paper concentrates on the development of a distance measure and a modified form of the K-means clustering algorithm, namely, Circular K-means (CK-means), in order to cluster vectors having the form of $\boldsymbol{F}_d$ in a circular-invariant manner. An efficient Fourier domain representation of the proposed measure is presented to reduce computational complexity. It should be mentioned that, since the proposed algorithm is based on K-means, it may exhibit certain disadvantages similar to the traditional algorithm, such as uncertainty regarding the correct number of clusters in the given data set. Nevertheless, solutions similar to the ones used for the traditional algorithm, such as split and merge techniques or utilization of clustering indices [16], [17], may be adopted. In this work, a split and merge approach (SMCK-means) suited to the proposed CK-means technique is proposed to avoid local minima and to estimate the correct number of clusters. The proposed SMCK-means is appropriate for texture applications that require clustering, such as texture segmentation and generation of key vectors required for building databases used for rotational invariant texture retrieval. The contribution and the advantages of this work compared to previous techniques are the following.

Information Retention. The problem with transforming rotation variant to rotation invariant feature vectors is that such a transformation results in loss of information. For instance, the Discrete Fourier Transform (DFT) coefficient magnitudes $|\mathrm{DFT}\{\boldsymbol{F}_d\}|$ of vector $\boldsymbol{F}_d$ defined above are invariant under image rotation by increments of $360^\circ/M$, since such a rotation causes a circular shift of the vector $\boldsymbol{F}_d$. Practically, there can be some preprocessing [12] so that the

• *The author is with the Department of Electrical Engineering, University of New Orleans, 2000 Lakeshore Dr., New Orleans, LA 70148. E-mail: dcharala@uno.edu.*

Manuscript received 25 Sept. 2004; revised 15 Apr. 2005; accepted 27 Apr. 2005; published online 13 Oct. 2005.
Recommended for acceptance by J. Goutsias.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0509-0904.

DFT magnitudes are invariant under any rotation. However, the DFT phase information is ignored. In previous techniques [14], [24], there has been an effort to transfer as much rotation invariant information as possible from the original to the transformed feature vectors. For instance, in [14], discrimination between textures is facilitated through microfeatures that include phase and direction information. In [24], there is a discussion of how the feature set consisting of DFT magnitudes could be enhanced by the DFT phases normalized by the phase of the first harmonic. The proposed work provides a solution to the aforementioned problems. The experimental results illustrate that those techniques using rotation invariant vectors transformed from variant ones, while being efficient, provide poorer clustering results compared to the proposed approach.

Generality. The proposed technique clusters vectors in the form of $\boldsymbol{F}_d$ containing multidirectional information. Multidirectional feature vectors can be generally used in most applications that require directional information, including texture analysis, object shape extraction, and junction shape/type identification. On the other hand, some feature extraction techniques can only be used for particular applications. For instance, junctions can be determined by line finding techniques, yet these techniques are not necessarily applicable in texture analysis.

Completeness. The proposed technique examines all common clustering problems, including avoiding local minima and estimating the correct number of clusters. At the same time, these properties are in the context of circular-shift invariance.

The paper is organized as follows: In Section 2, the distance measure and the K-means-based clustering algorithm employing the proposed distance measure are introduced. Section 3 describes the proposed split and merge approach. Section 4 presents examples to demonstrate the effectiveness of the new algorithm. Finally, Section 5 closes with some concluding remarks.

2 Circular Shift Invariant K-Means

This section presents the circular invariant clustering algorithm. First, the distance measure is developed. Then, the algorithmic steps are presented. Finally, the computational complexity and convergence of the algorithm are discussed.

2.1 Development of the Distance Measure

In the following, characters that denote vectors and matrices are lowercase and uppercase, respectively. The superscript of a vector or matrix specifies its dimensions. For instance, $\mathbf{X}^{NM}$ is a matrix of size $N \times M$, while $\mathbf{x}^N$ is a vector of size N. A vector is defined as a single column.

The novel distance measure introduced in this work is based on the Mahalanobis distance. Other commonly used distance measures, such as the Euclidean, are special cases of the Mahalanobis. The square of the Mahalanobis distance between a vector $\mathbf{x}_j^N$ and a centroid $\mathbf{m}_l^N$ is defined as

$$d^2\left(\mathbf{x}_j^N, \mathbf{m}_l^N\right) = \left(\mathbf{x}_j^N - \mathbf{m}_l^N\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \left(\mathbf{x}_j^N - \mathbf{m}_l^N\right) \tag{1}$$

or

$$\begin{aligned} d^2\left(\mathbf{x}_j^N, \mathbf{m}_l^N\right) = {} & \left(\mathbf{x}_j^N\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \mathbf{x}_j^N \\ & - 2\left(\mathbf{x}_j^N\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \mathbf{m}_l^N + \left(\mathbf{m}_l^N\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \mathbf{m}_l^N, \end{aligned} \tag{2}$$

where superscript T denotes transpose, index l identifies the lth cluster, index j identifies the jth data vector, and $\mathbf{K}_l^{NN}$ is the inverse of the lth cluster's covariance matrix $\mathbf{C}_l^{NN} (\mathbf{K}_l^{NN} = (\mathbf{C}_l^{NN})^{-1})$. In order to calculate the minimum Mahalanobis distance between vector $\mathbf{x}_j^N$ and centroid vector $\mathbf{m}_l^N$, with respect to all possible circular shifts of $\mathbf{x}_j^N$, the following circulant matrix is constructed:

$$\begin{aligned} & \mathbf{X}_j^{NN} = \\ & \left[\mathbf{x}_j^N * \boldsymbol{\delta}_0 \quad \mathbf{x}_j^N * \boldsymbol{\delta}_1 \quad \mathbf{x}_j^N * \boldsymbol{\delta}_2 \quad \ldots \quad \mathbf{x}_j^N * \boldsymbol{\delta}_k \quad \ldots \quad \mathbf{x}_j^N * \boldsymbol{\delta}_{N-1}\right], \end{aligned} \tag{3}$$

where

$$\boldsymbol{\delta}_k = [\underbrace{0 \ldots 0}_{k} \ 0 \ 1 \ 0 \ldots . \ 0]^{\mathrm{T}}. \tag{4}$$

Operator * corresponds to circular convolution. Then, the square of the Mahalanobis distance for N possible circular shifts of vector $\mathbf{x}_j^N$ is defined as

$$\begin{aligned} \mathbf{d}_{j,l}^N = {} & diag\left\{\left(\mathbf{X}_j^{NN}\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \mathbf{X}_j^{NN}\right\} - 2\left(\mathbf{X}_j^{NN}\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \mathbf{m}_l^N \\ & + \left(\mathbf{m}_l^N\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \mathbf{m}_l^N \mathbf{1}^N, \end{aligned} \tag{5}$$

where $diag\{\mathbf{Y}^{NN}\}$ is defined as a column vector that consists of the N diagonal elements of $\mathbf{Y}^{NN}$ and $\mathbf{1}^N$ is an all-ones column vector. Let vector $\mathbf{a}_l^N$ and parameter b_l be defined as

$$\begin{aligned} \mathbf{a}_l^N &= \mathbf{K}_l^{NN} \mathbf{m}_l^N \\ b_l &= \left(\mathbf{m}_l^N\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \mathbf{m}_l^N. \end{aligned} \tag{6}$$

These are constant for a given center vector and covariance matrix. Thus, $\mathbf{d}_{j,l}^N$ can be expressed as

$$\mathbf{d}_{j,l}^N = diag\left\{\left(\mathbf{X}_j^{NN}\right)^{\mathrm{T}} \mathbf{K}_l^{NN} \mathbf{X}_j^{NN}\right\} - 2\left(\mathbf{X}_j^{NN}\right)^{\mathrm{T}} \mathbf{a}_l^N + b_l \mathbf{1}^N. \tag{7}$$

Based on the previous equation, the cross-correlation vector between vectors $\mathbf{a}_l^N$ and $\mathbf{x}_j^N$, $\mathbf{r}_{j,l}^N$, is defined as

$$\begin{aligned} \mathbf{r}_{j,l}^N &= \left(\mathbf{X}_j^{NN}\right)^{\mathrm{T}} \mathbf{a}_l^N, \\ \text{or } \ r_{j,l}^N(k) &= \sum_{i=0}^{N-1} a_l(i) x_j(i-k). \end{aligned} \tag{8}$$

The cross-correlation $\mathbf{r}_{j,l}^N$ can be calculated using the Fourier Transform:

$$\mathbf{r}_{j,l}^N = \Im^{-1}\left\{\Im\left\{\mathbf{a}_l^N\right\} \circ \Im\left\{\mathbf{x}_j^N\right\}^*\right\}, \tag{9}$$

where operator $\circ$ corresponds to element-wise matrix multiplication, while $\Im(\mathbf{x}^N)$ and $\Im^{-1}(\mathbf{x}^N)$ are, respectively, the N-point Fourier Transform and Inverse Fourier Transform operators on vector $\mathbf{x}^N$. Furthermore,

$$\begin{aligned}\mathbf{e}_{j,l}^{N} &= diag\left\{\left(\mathbf{X}_j^{NN}\right)^{\mathrm{T}}\mathbf{K}_l^{NN}\mathbf{X}_j^{NN}\right\}\\ &= diag\left\{\mathbf{F}\mathbf{F}^{\mathrm{H}}\left(\mathbf{X}_j^{NN}\right)^{\mathrm{T}}\mathbf{F}\mathbf{F}^{\mathrm{H}}\mathbf{K}_l^{NN}\mathbf{F}\mathbf{F}^{\mathrm{H}}\mathbf{X}_j^{NN}\mathbf{F}\mathbf{F}^{\mathrm{H}}\right\} \qquad (10)\\ &= diag\left\{\Im_v\left\{\Im_h^{-1}\left\{\mathbf{P}_{j,l}^{NN}\right\}\right\}\right\},\end{aligned}$$

where

$$\mathbf{\Phi}_j^{NN} = \left[\Im\left\{\mathbf{x}_j^{N}\right\}\ldots.\Im\left\{\mathbf{x}_j^{N}\right\}\right], \qquad (11)$$

$$\mathbf{Q}_l^{NN} = \Im_v^{-1}\left\{\Im_h\left\{\mathbf{K}_l^{NN}\right\}\right\}, \qquad (12)$$

$$\mathbf{P}_{j,l}^{NN} = \mathbf{\Phi}_j^{NN}\circ\mathbf{Q}_l^{NN}\circ\left(\mathbf{\Phi}_j^{NN}\right)^{\mathrm{H}}. \qquad (13)$$

In the above equations, superscript H denotes Hermitian transpose, $\mathbf{F}^{NN}$ is the $N\times N$ Fourier Transform matrix, where $F(i,m) = e^{-2\pi\, j(i-1)(m-1)}$, while $\Im_h(\mathbf{X}^{NN})$ and $\Im_v(\mathbf{X}^{NN})$ are, respectively, the N-point Fourier Transform operators applied on matrix $\mathbf{X}^{NN}$ row-wise and column-wise.

Equation (10) uses the property [18] that, since $\mathbf{X}_j^{NN}$ is circulant, the matrix products $\mathbf{F}^{\mathrm{H}}(\mathbf{X}_j^{NN})\mathbf{F}$ and $\mathbf{F}^{\mathrm{H}}(\mathbf{X}_j^{NN})^{\mathrm{T}}\mathbf{F}$ result in diagonal matrices representing, respectively, the Fourier Transform and the conjugate Fourier Transform coefficients of $\mathbf{x}_j^N$. Another property used in (10) is that, if a matrix $\mathbf{Y}^{NN}$ is left multiplied with a diagonal matrix $\mathbf{\Lambda}^{NN}$, the product $\mathbf{Y}^{NN}\mathbf{\Lambda}^{NN}$ is equivalent with the element-wise multiplication of the diagonal elements of $\mathbf{\Lambda}^{NN}$ with each row of $\mathbf{Y}^{NN}$. Similarly, the product $\mathbf{\Lambda}^{NN}\mathbf{Y}^{NN}$ is equivalent with the element-wise multiplication of the diagonal elements of $\mathbf{\Lambda}^{NN}$ with each column of $\mathbf{Y}^{NN}$. Therefore, the minimum distance square between $\mathbf{x}_j^N$ and $\mathbf{m}_l^N$ can be defined as

$$D_{j,l} = \min_k\left\{\mathbf{d}_{j,l}^{N}\right\} = \min_k\left\{\, e_{j,l}(k) - 2\, r_{j,l}(k)\right\} + b_l \qquad (14)$$

and is circular-shift invariant. In the case of Euclidean distance, $\mathbf{K}_l^{NN} = \mathbf{I}^{NN}$, thus

$$\begin{aligned}\mathbf{a}_l^N &= \mathbf{m}_l^N\\ b_l &= \left(\mathbf{m}_l^N\right)^{\mathrm{T}}\mathbf{m}_l^N\\ diag\left\{\left(\mathbf{X}_j^{NN}\right)^{\mathrm{T}}\mathbf{X}_j^{NN}\right\} &= \left(\mathbf{x}_j^N\right)^{\mathrm{T}}\mathbf{x}_j^N\mathbf{1}^N\end{aligned} \qquad (15)$$

and

$$\begin{aligned}D_{j,l} &= \min_k\left\{\mathbf{d}_{j,l}^{N}\right\} = \min_k\\ &\left\{\left(\mathbf{x}_j^N\right)^{\mathrm{T}}\mathbf{x}_j^N - 2\,\Im^{-1}\left\{\Im\left\{\mathbf{m}_l^N\right\}\Im\left\{\mathbf{x}_j^N\right\}^{*}\right\}\right\} + \left(\mathbf{m}_l^N\right)^{\mathrm{T}}\mathbf{m}_l^N\\ &= \left(\mathbf{x}_j^N\right)^{\mathrm{T}}\mathbf{x}_j^N + \left(\mathbf{m}_l^N\right)^{\mathrm{T}}\mathbf{m}_l^N - 2\max\left\{\,\mathbf{r}_{j,l}^N\right\}.\end{aligned} \qquad (16)$$

Based on the previous discussion, the jth pattern is assigned to the cluster that provides the minimum distance measure:

$$D_j = \min_l\left\{D_{j,l}\right\}. \qquad (17)$$

2.2 CK-Means Algorithm

Essentially, CK-means employs a circular-shift invariant distance measure to assign each vector to a cluster based on the current cluster centroids and covariance matrices. Then, each one of the vectors associated to a particular cluster is shifted by the appropriate shift that minimized its distance from the cluster. Finally, the centroids and covariance matrices are updated similarly to the traditional K-means algorithm. Next, the algorithmic steps of the core CK-means algorithm are presented:

1. *Initialization (Iteration 0)*. Calculate the Fourier Transform domain matrix $\mathbf{\Phi}_j^{NN}$ as in (11). This calculation may be performed only once since $\mathbf{\Phi}_j^{NN}$ is iteration independent. In the case of Euclidean distance, $(\mathbf{x}_j^N)^{\mathrm{T}}\mathbf{x}_j^N$ is also iteration independent and needs to be calculated only once. Also, initialize the number of clusters to L as well as the centroid $\mathbf{m}_l^N(0)$ and inverse covariance matrix $\mathbf{K}_l^{NN}(0)$ for all L clusters.
2. *Iteration t*.
   a. Calculate $\mathbf{P}_{j,l}^{NN}$, $\mathbf{a}_l^N$, and b_l, for $l = 1,\ldots,L$. These calculations are performed only once in the beginning of each iteration, since $\mathbf{P}_{j,l}^{NN}$, $\mathbf{a}_l^N$, and b_l remain unchanged in a single iteration.
   b. For each vector $\mathbf{x}_j^N$ and lth cluster, calculate $D_{j,l}$ as in (14) for Mahalanobis or (16) for Euclidean distance.
   c. If the centroid that provides the minimum $D_{j,l}$ is l_o, then circularly shift $\mathbf{x}_j^N$ by k_{j,l_o} to obtain $\mathbf{x}_j^N * \boldsymbol{\delta}_{k_{j,l_o}}$, where k_{j,l_o} is the shift k that minimizes D_{j,l_o} in (14) for Mahalanobis or (16) for Euclidean distance.
   d. Update the cluster centers and covariance matrices:

$$\mathbf{m}_l^N(t) = \frac{1}{J_l}\sum_{j_l}\left(\mathbf{x}_{j_l}^N * \boldsymbol{\delta}_{kj,l_o}\right), \qquad (18)$$

$$\begin{aligned}&\mathbf{C}_l^{NN}(t) = \frac{1}{J_l}\\ &\sum_{j_l}\left(\mathbf{x}_{j_l}^N * \boldsymbol{\delta}_{kj,l_o} - \mathbf{m}_l^N(t)\right)\left(\mathbf{x}_{j_l}^N * \boldsymbol{\delta}_{kj,l_o} - \mathbf{m}_l^N(t)\right)^{\mathrm{T}},\end{aligned} \qquad (19)$$

   where J_l is the total number of vectors associated to the lth cluster and j_l identifies vectors associated to the lth cluster.
   e. Stop if the measure sum $D_{sum} = \sum_{j=1}^{J} D_j$ does not decrease more than a specified threshold determined by the user. Otherwise, go to Step 2a.

It can be easily shown that the proposed algorithm converges for D_{sum} since D_{sum} is reduced in each iteration. The distance measure presented in (14) and (16) is minimized for each individual pattern in Step 2c with respect to cluster and circular shift. Furthermore, (18) recalculates the cluster centers. These cluster centers are computed from shifted vectors, whose shift was the one that minimized D_j. Similarly to the traditional K-means, (18) reduces the overall distance

between vectors and corresponding cluster centers. This distance is given by D_{sum}.

Similar to the traditional algorithm, it is likely that CK-means converges to a local optimum solution. The tendency of CK-means to terminate at a local minimum is amplified due to fact that CK-means considers all possible vector shifts. More specifically, each N-dimensional vector $\mathbf{x}_j^N$ can be represented by any one out of the N possible shifted versions of $\mathbf{x}_j^N$. Several initialization techniques have been presented in the literature, including initial condition refinement techniques [25] and split and merge approaches. In this work, a split-and-merge algorithm (SMCK-means) is proposed in Section 3. This algorithm incorporates an additional technique to CK-means to estimate the correct number of clusters in the data.

2.3 Fast FFT-Based Measure Implementation and Computational Complexity

The distance measure D_j defined in (17) can take advantage of the reduced computational complexity that an efficient Fast Fourier Transform (FFT) technique can provide. A Radix-2 FFT can be used if the vector length N is a number equal to a power of 2. In general, this condition may not be satisfied. A modification can be applied to $\mathbf{e}_{j,l}^N$ in (10) and, more specifically, to matrix $\mathbf{P}_{j,l}^{NN}$ so that the $\Im_h^{-1}$ operation (10) applied on $\mathbf{P}_{j,l}^{NN}$ can be performed with an Inverse FFT. The Fourier Transform operations required for the calculation of $\mathbf{Q}_l^{NN}$ are applied only once at the beginning of each iteration. Thus, an FFT approach is not crucial for the calculation of $\mathbf{Q}_l^{NN}$.

Consider the N-point Fourier Transform of a vector $\mathbf{v}^N$, namely, $\mathbf{f}_v^N = (\mathbf{F}^{NN}\mathbf{v}^N)^{\mathrm{T}}$. An interpolated version of $\mathbf{v}^N$ can be obtained by appropriate zero-insertion in vector $\mathbf{f}_v^N$ and by applying the Inverse Fourier Transform. More specifically, zero-insertion can be defined through the following column-wise operator $\Delta^M\{\}$:

$$\Delta^M\{\mathbf{f}_v^N\} = \begin{cases} \left[f_v(1), \quad \ldots \quad f_v\left(\frac{N+1}{2}\right), \quad \mathbf{0}^{M-N}, \quad f_v\left(\frac{N+1}{2}+1\right), \quad \ldots, \quad f_v(N)\right]^{\mathrm{T}}, & N{:}odd \\ \left[f_v(1), \quad \ldots \quad f_v\left(\frac{N}{2}\right)/2, \quad \mathbf{0}^{M-N-1}, \quad f_v\left(\frac{N}{2}\right)^*/2, \quad \ldots, \quad f_v(N)\right]^{\mathrm{T}}, & N{:}even, \end{cases} \tag{20}$$

where M equals a power of two as required by FFT, and $\mathbf{0}^N$ represents an all zeros column vector of size N. Thus, if the inverse Fourier transform is applied on vector $\Delta^M\{\mathbf{f}_v^N\}$, the result is a size N vector $(\mathbf{F}^{MM}\Delta^M\{\mathbf{f}_v^N\})^{\mathrm{T}}$ which is an interpolated version of $\mathbf{v}^N$. Similarly to (20), each column of $\mathbf{P}_{j,l}^{NN}$ in (13) can be zero-inserted to obtain

$$\bar{\mathbf{P}}_{j,l}^{MN} = \Delta^M\left\{\mathbf{P}_{j,l}^{NN}\right\}, \tag{21}$$

which is an $M \times N$ matrix. Then, the $\Im_h^{-1}$ operation in (10) can be implemented using an Inverse FFT. Matrix $\Im_h^{-1}\{\bar{\mathbf{P}}_{j,l}^{MN}\}$ is also zero-inserted row-wise, to obtain

$$\bar{\bar{\mathbf{P}}}_{j,l}^{MM} = \Delta^M\left\{\Im_h^{-1}\left\{\bar{\mathbf{P}}_{j,l}^{MN}\right\}^{\mathrm{T}}\right\}^{\mathrm{T}}. \tag{22}$$

Finally, using an FFT for the $\Im_v$ operation in (10) is not crucial, since only the M diagonal elements of the resulted $M \times M$ matrix are needed. Furthermore, the autocorrelation vector in (9) can be calculated as

$$\mathbf{r}_{j,l}^M = \Im^{-1}\left\{\Delta^M\left\{\Im\{\mathbf{a}_l^N\} \circ \Im\{\mathbf{x}_j^N\}^*\right\}\right\}. \tag{23}$$

Based on the previous discussion, an efficient computation of (10) can be expressed as

$$\mathbf{e}_{j,l}^N = diag\left\{\Im_v\left\{\bar{\bar{\mathbf{P}}}_{j,l}^{MM}\right\}\right\}. \tag{24}$$

In case of the Euclidean distance, (16) can be written as

$$D_{j,l} = \left(\mathbf{x}_j^N\right)^{\mathrm{T}}\mathbf{x}_j^N + \left(\mathbf{m}_l^N\right)^{\mathrm{T}}\mathbf{m}_l^N - 2\max\left\{\mathbf{r}_{j,l}^M\right\}. \tag{25}$$

Next, the computational complexity of the proposed algorithm is presented. Considering that there is a reasonably large number of iterations, Step 1 of the algorithmic description in Section 2.2 does not require significant additional processing time. Similarly, Step 2a is only performed once at the beginning of each iteration.

Step 2b requires calculation of (23) and (24) for the case of the Mahalanobis distance. Equation (23) requires N multiplications for the element-wise vector product and $M \log_2 M$ operations for the Inverse FFT. Calculation of (24) also requires calculation of (13), (19), and (22). Calculation of $\mathbf{P}_{j,l}^{NN}$ in (19) requires $2N$ multiplications for the element-wise matrix products. Calculation of $\bar{\bar{\mathbf{P}}}_{j,l}^{MM}$ in (22) requires approximately $N \cdot M \cdot \log_2(M)$ operations. The Fourier Transform in (24) is required only for the diagonal elements of the resulted matrix, thus it requires approximately M^2 operations. All above calculations are required each iteration for all combinations of J vectors and L clusters. The computational complexity for Step 2d is $\mathrm{O}\{J \cdot M^2\}$ for Mahalanobis distance. Therefore, the algorithm's complexity is $\mathrm{O}\{J \cdot L \cdot N \cdot M \cdot \log_2(M)\}$. In the case of Euclidean distance, the algorithm's complexity is $\mathrm{O}\{J \cdot L \cdot M \cdot \log_2 M\}$. The complexity of the traditional circular variant K-means algorithm is $\mathrm{O}\{J \cdot L \cdot N^2\}$ for Mahalanobis distance and $\mathrm{O}\{J \cdot L \cdot N\}$ for Euclidean distance.

3 Split and Merge Algorithm

3.1 Estimating the Number of Clusters

In this paper, the technique for estimating the correct number of clusters presented in [13] is tailored to the proposed CK-means algorithm. The algorithm selects the number of clusters based on a modified version of the Variance Ratio Criterion (*VRC*) index. The Circular-Invariant Variance Ratio Criterion (*CIVRC*) is given by:

$$CIVRC_L = \frac{(J-L)BCD_L}{(L-1)WCD_L}. \tag{26}$$

If different values of the number of clusters L are considered, the estimated number of clusters is the one for which index $CIVRC_L$ is maximized. In (26), BCD_L is the "between clusters distance" and WCD_L is the "within clusters distance," J is the total number of vectors, and L is the number of clusters. The "within" and "between" cluster distances considering L clusters are, respectively, defined as:

$$WCD_L = \sum_{l=1}^{L} J_l \sum_{j=1}^{N} diag\{\ \mathbf{C}_l^{NN}\} \quad \Big| \text{for } L \text{ clusters} \tag{27}$$

and

$$BCD_L = TD - WCD_L, \tag{28}$$

where

$$TD = J \sum_{j=1}^{N} diag\{\ \mathbf{C}_1^{NN}\} \quad \Big| \text{ for one cluster} \tag{29}$$

and J_l is the number of vectors $\mathbf{x}_j^N$ associated to the lth cluster. The within cluster distance WCD_L is the total distance of all vectors from their corresponding centroid, while TD describes the total distance of all vectors from a single centroid calculated from all vectors. Essentially, TD is a measure of the "extent" of the vector set in the N-dimensional space. Index $CIVRC_L$ is proportional to the ratio $BCD_L/WCD_L = TD/WCD_L - 1$; therefore, TD acts as a normalization factor. This is desired, since the decision regarding splitting or merging clusters should depend on the relative distance among vectors (considering all vectors), and not on the absolute distance between them.

From (29), the TD can be expressed as

$$\begin{aligned} TD &= \sum_{j=1}^{J} \left(\mathbf{x}_j^N - \mathbf{m}_{\mathrm{o}}^N\right)^{\mathrm{T}} \left(\mathbf{x}_j^N - \mathbf{m}_{\mathrm{o}}^N\right) \\ &= \sum_{j=1}^{J} \left(\mathbf{x}_j^N\right)^{\mathrm{T}} \mathbf{x}_j^N - J\left(\mathbf{m}_{\mathrm{o}}^N\right)^{\mathrm{T}} \mathbf{m}_{\mathrm{o}}^N, \end{aligned} \tag{30}$$

where $\mathbf{m}_{\mathrm{o}}^N$ is the centroid considering one cluster. Similarly, WCD_L can be expressed as

$$\begin{aligned} WCD_L &= \sum_{l=1}^{L} \sum_{j_l=1}^{J_l} \left(\mathbf{x}_{j_l}^N - \mathbf{m}_l^N\right)^{\mathrm{T}} \left(\mathbf{x}_{j_l}^N - \mathbf{m}_l^N\right) \\ &= \sum_{j=1}^{J} \left(\mathbf{x}_j^N\right)^{\mathrm{T}} \mathbf{x}_j^N - \sum_{l=1}^{L} J_l \left(\mathbf{m}_l^N\right)^{\mathrm{T}} \mathbf{m}_l^N, \end{aligned} \tag{31}$$

where J_l is the number of vectors associated with the lth cluster. Then,

$$BCD_L = TD - WCD_L = \sum_{l=1}^{L} J_l \left(\mathbf{m}_l^N\right)^{\mathrm{T}} \mathbf{m}_l^N - J\left(\mathbf{m}_{\mathrm{o}}^N\right)^{\mathrm{T}} \mathbf{m}_{\mathrm{o}}^N. \tag{32}$$

The centroids $\mathbf{m}_l^N$ are found in a circular invariant manner using the CK-means approach. Furthermore, although circular shift invariance considers that any shifted version of vector $\mathbf{x}_j^N$ is equivalent to $\mathbf{x}_j^N$, after a centroid $\mathbf{m}_l^N$ is found the relative shift between the vectors associated to $\mathbf{m}_l^N$ is fixed. However, the relative shift between different centroids is not fixed. Therefore, $\mathbf{m}_{\mathrm{o}}^N$ could be directly calculated by combining centroids $\mathbf{m}_l^N$ in a circular invariant manner. For that purpose, the CK-means algorithm presented in Section 2.2 treats centroids $\mathbf{m}_l^N$ as regular vectors and clusters them considering one cluster. Then, (18) is replaced by:

$$\mathbf{m}_{\mathrm{o}}^N(t) = \frac{1}{L} \sum_{l=1}^{L} (J_l/J)\left(\mathbf{m}_l^N * \boldsymbol{\delta}_{k_l}\right), \tag{33}$$

where k_l is the appropriate shift for centroid $\mathbf{m}_l^N$. The weight (J_l/J) has been included to reflect the particular contribution of $\mathbf{m}_l^N$, which is proportional to the number of vectors J_l associated to it. It should be mentioned that, due to the high (yet nonlinear) dependence of $\mathbf{m}_{\mathrm{o}}^N$ on $\mathbf{m}_l^N$ compared to the traditional case, the degrees of freedom shown in the denominator of (26) are considered to be L instead of $L-1$.

The circular invariance of $CIVRC_L$ is discussed next. Index $CIVRC_L$ depends on WCD_L and BCD_L. The only parts of WCD_L and BCD_L that could depend on the circular shift of vectors are $(\mathbf{m}_l^N)^{\mathrm{T}}\mathbf{m}_l^N$ and $(\mathbf{m}_{\mathrm{o}}^N)^{\mathrm{T}}\mathbf{m}_{\mathrm{o}}^N$ since terms $(\mathbf{x}_j^N)^{\mathrm{T}}\mathbf{x}_j^N$ are scalars with no shift dependence. Nevertheless, centroids $\mathbf{m}_l^N$ are computed in a circular invariant manner using CK-means. Furthermore, $\mathbf{m}_{\mathrm{o}}^N$ is also computed using CK-means from centroids $\mathbf{m}_l^N$. Therefore, $CIVRC_L$ is circular invariant.

The algorithmic blocks are presented in Fig. 1. The SMCK-means algorithm is significantly robust, as experiments have shown. Details of SMCK-means are presented in Sections 3.2 and 3.3.

3.2 Split Stage

In the split stage of SMCK-means, a cluster is iteratively divided into L_o subclusters using the CK-means algorithm. Essentially, a cluster is iteratively split as long as $CIVRC$ increases. Assuming that cluster l is split into L_o clusters, the WCD is updated as follows:

$$\begin{aligned} WCD_{L'} = WCD_L &- J_l \sum_{j=1}^{N} diag\{\ \mathbf{C}_l^{NN}\} \\ &+ \sum_{\lambda=1}^{L_o} J_\lambda \sum_{j=1}^{N} diag\{\ \mathbf{C}_\lambda^{NN}\}, \end{aligned} \tag{34}$$

where subscript λ identifies the subclusters resulted due to the splitting of cluster l, and L' denotes the total number of clusters after a cluster subdivision takes place.

3.3 Merge Stage

The pair of clusters (λ, l) that should be merged at a particular cluster merging iteration should be the one that provides the maximum increase in $CIVRC_{L-1}$. All cluster pairs are considered and the one (λ, l) that minimizes the following circular invariant distance measure between the corresponding centroids is chosen:

$$D_{l,\lambda} = (\mathbf{m}_l^N)^{\mathrm{T}}\mathbf{m}_l^N + (\mathbf{m}_\lambda^N)^{\mathrm{T}}\mathbf{m}_\lambda^N - 2\max\left\{\ \mathbf{r}_{l,\lambda}^N\right\}, \tag{35}$$

where

$$\mathbf{r}_{l,\lambda}^N = \Im^{-1}\{\ \Im\{\mathbf{m}_l^N\} \circ \Im\{\mathbf{m}_\lambda^N\}^*\}. \tag{36}$$

From (31), the difference in WCD if two clusters λ and l with corresponding centroids $\mathbf{m}_\lambda^N$ and $\mathbf{m}_l^N$ are merged is

$$\begin{aligned} WCD_{L-1} - WCD_L = J_l\left(\mathbf{m}_l^N\right)^{\mathrm{T}}\mathbf{m}_l^N + J_\lambda\left(\mathbf{m}_\lambda^N\right)^{\mathrm{T}}\mathbf{m}_\lambda^N \\ - (J_l + J_\lambda)\left(\mathbf{m}_{l,\lambda}^N\right)^{\mathrm{T}}\mathbf{m}_{l,\lambda}^N, \end{aligned}$$

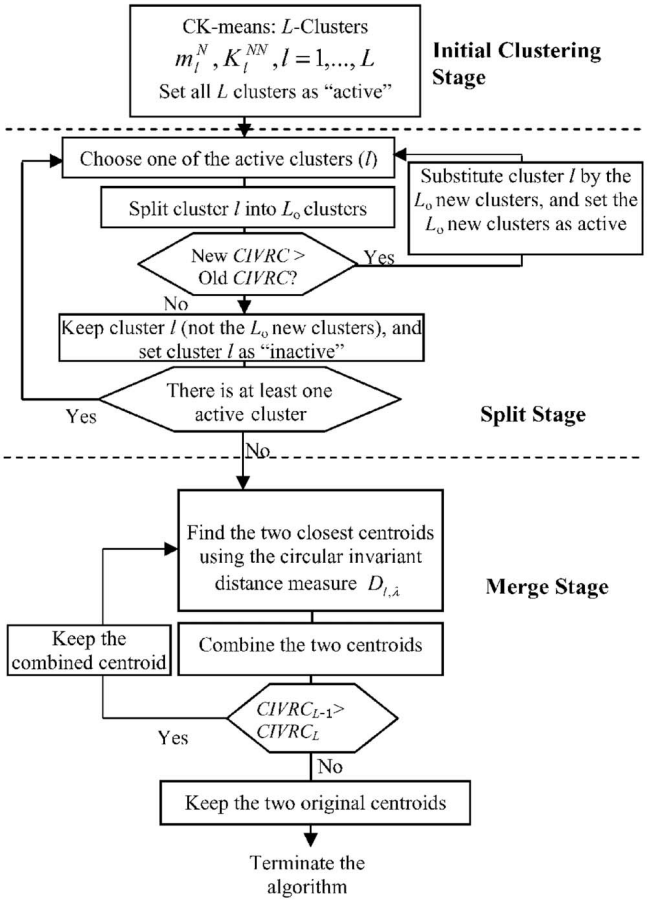


Fig. 1. The split-and-merge (SMCK-means) clustering algorithm.

where $\mathbf{m}_{l,\lambda}^N = (J_l\mathbf{m}_l^N + J_\lambda\mathbf{m}_\lambda^N)/(J_l + J_\lambda)$ is the combined centroid. Then, it can be easily shown that

$$WCD_{L-1} - WCD_L = \frac{J_\lambda J_l}{J_\lambda + J_l}\left(\mathbf{m}_l^N - \mathbf{m}_\lambda^N\right)^{\mathrm{T}}\left(\mathbf{m}_l^N - \mathbf{m}_\lambda^N\right).$$

However, for the proposed shift invariant technique, the term $(\mathbf{m}_l^N - \mathbf{m}_\lambda^N)^{\mathrm{T}}(\mathbf{m}_l^N - \mathbf{m}_\lambda^N)$ should be replaced by $D_{l,\lambda}$ presented in (35). Then, the WCD for the merge stage, after a single cluster pair merging, is updated as

$$WCD_{L-1} = WCD_L + \frac{J_\lambda J_l}{J_\lambda + J_l} D_{l,\lambda}. \tag{37}$$

Minimizing (35) is an appropriate criterion for choosing which clusters should be merged. The minimum $D_{l,\lambda}$ will introduce a small increase in WCD without tending to merge small clusters, which would be the case if minimization of $J_\lambda J_l/(J_\lambda + J_l)D_{l,\lambda}$ was used instead.

The procedure is repeated as long as the $CIVRC$ index increases. The final clustering results, including the total number of clusters, are obtained at the end of the merge stage.

4 Experimental Results

This section presents experimental results that demonstrate the effectiveness of the algorithm to cluster data in a circular invariant manner. In this work, the performance of SMCK-means as a rotation invariant clustering technique has been tested on textural images. More specifically, textural energy feature vectors are extracted from eight different textural images. Several scales may be required for successful texture analysis. However, the concentration in this paper is regarding circular invariance, thus a single scale is used. The ability of the algorithm to cluster these vectors as a semisupervised technique assuming that the number of clusters is known is examined in Section 4.1. The ability of the algorithm to estimate the correct number of clusters as a fully unsupervised technique is examined in Section 4.2. Similarly to regular split and merge techniques, the proposed method does not guarantee convergence to a global minimum. However, it increases the possibility that the algorithm will avoid a local minimum.

In this experiment, eight different textures of size 512×512 are used. A directional exponential filter is chosen to extract energy features in several orientations. The exponential filter is defined by the following impulse response:

$$g(i,m) = -i \cdot \exp\left\{-\frac{i^2}{s_i^2} - \frac{m^2}{s_m^2}\right\}, \tag{38}$$

where i, m are the filter's spatial coordinates, and s_i^2 and s_m^2 are the horizontal and vertical standard deviations of the Gaussian envelope. The oriented texture energy at location (i,m) is defined as:

$$E^\varphi(i,m) = \frac{1}{W_E^2}\sum_{m'=-W_E/2}^{W_E/2}\sum_{i'=-W_E/2}^{W_E/2}\left|I_{g^\varphi}(i+i', m+m')\right|, \tag{39}$$

where I_{g^φ} is the image filtered by the exponential filter oriented at direction φ and parameter W_E defines the window size. In order to avoid dependence on image contrast, normalized energy was used:

$$E_n^\varphi(i,m) = E^\varphi(i,m)/E^T(i,m), \tag{40}$$

where

$$E^T(i,m) = \sum_\varphi E^\varphi(i,m) \tag{41}$$

is the total energy over all directions. Energy vectors are extracted from texture blocks of size $W_E \times W_E$, where $W_E = 64$. Each vector consists of the energy calculated in 32 different orientations sampled uniformly in the interval $(\theta, 180^\circ + \theta)$, where θ specifies the starting orientation in the interval. The goal of these experiments is to evaluate the circular-shift invariant performance of the algorithm. Therefore, feature vectors should be extracted from several rotated versions of the same texture. Equivalently, several feature vectors can be extracted from the same texture block in angular intervals of length $180^\circ : (\theta, 180^\circ + \theta)$, considering a different starting point θ at a time. Ten different θ values were randomly selected, thus 10 feature vectors were extracted per block.

Fig. 2 shows texture samples from the textures used in the experiments. Fig. 3 shows polar plot examples of eight feature vectors. In general, oriented energy feature vectors are periodic with period 360 degrees. In this particular case where the filters used are antisymmetric, feature extraction results in feature vectors with period 180 degrees. Fig. 4 shows an example of four differently oriented texture samples and the corresponding feature vector polar plots. Apparently, the effect of texture rotation on the feature vector is a circular shift.

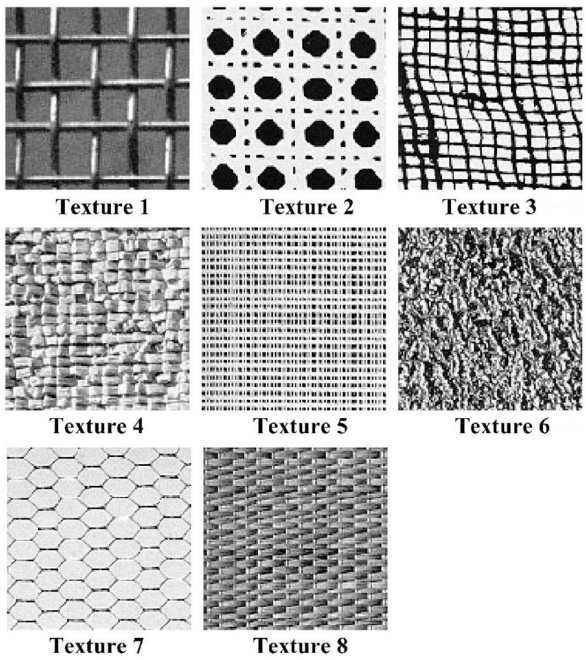


Fig. 2. Samples from the eight textures used for the texture clustering experiment.

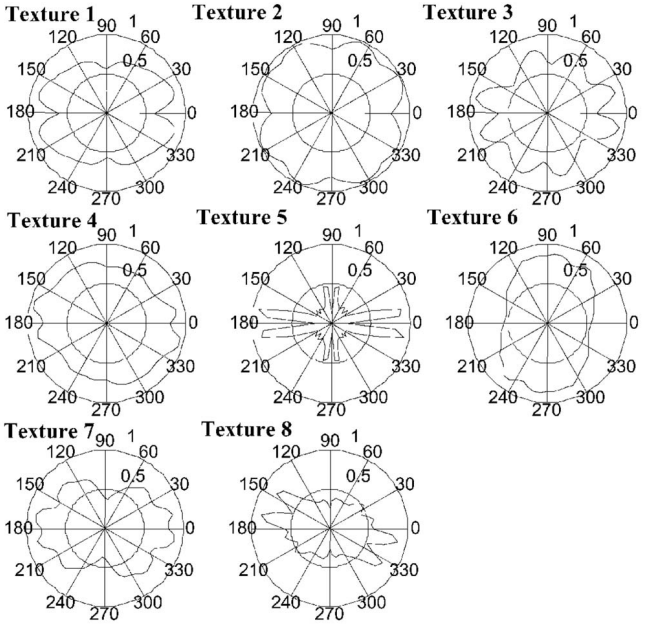


Fig. 3. Polar plots of energy extracted in multiple orientations from a 64×64 block from each of the eight textures shown in Fig. 2.

It should also be mentioned at this point that the threshold used as the terminating criterion for Step 2e of the CK-means algorithm was set equal to $J \cdot N \cdot L \cdot 10^{-5}$ for all experiments.

4.1 Percentage of Correct Clustering

In classification related applications, it is easy to measure the quality of the classification technique using the *percentage of correct classification* defined as the percentage of test vectors that, given the actual vector labels, have been correctly classified. However, in the case of clustering the validity of the technique is measured using the *Percentage of Correct Clustering* (PCC), also used in [23], which is defined as follows: Let $l = 1, 2, \ldots, L$ be the index in a set of L known labels. Furthermore, assuming that the data vectors have been clustered into L' clusters, let $l' = 1, 2, \ldots, L'$ be the index that identifies the l'th cluster. A cluster l' is associated to label l if the number of vectors labeled l contained in l' is larger compared to the number of vectors labeled with any other single label. Then, the PCC for cluster l' is defined as:

$$\mathrm{PCC}_1{}' = 100 J_l / J'_l, \tag{42}$$

where J_l is the number of label l vectors in cluster l' and J'_l is the total number of vectors in cluster l'. The overall PCC is defined as

$$\mathrm{PCC} = \frac{100}{J} \sum_{l'} \mathrm{PCC}_{l'} \quad J_{l'}, \tag{43}$$

where J is the total number of vectors.

For this experiment, all possible combinations of two, three, and four textures out of all eight textures are considered. (For instance, considering four out of eight texture combinations yields a total of 70 clustering experiments.) The average PCC is found in each case. Moreover, clustering using all eight textures is performed.

For these experiments, the split and merge technique presented in Section 2 is used for both CK-means and original K-means. However, the original K-means uses the original VRC index and the BCD and WCD as shown in (27)-(29).

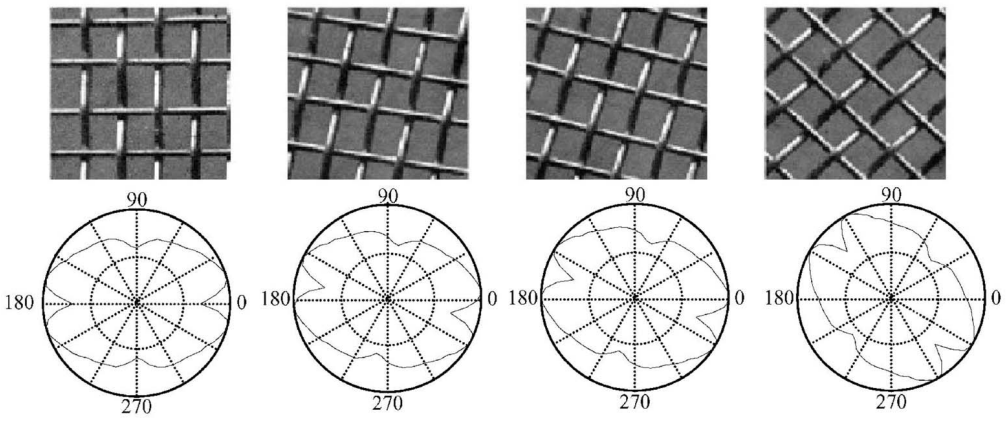


Fig. 4. Texture samples in four different orientations and corresponding polar plots of energy extracted.

TABLE 1
Average Percentage of Correct Clustering for the Three Approaches and Corresponding Timings (Secs)

		Average PCC			Average Timings Per Iteration		
ANC	CR	Proposed	FFT-magn.	MODF – 5	Proposed	FFT-magn.	MODF – 5
2	1|1	99.6%	96.7%	96.0%	0.05	$0.9 \cdot 10^{-3}$	$0.3 \cdot 10^{-3}$
2	2|1	99.7%	96.7%	94.3%	0.04	$0.6 \cdot 10^{-3}$	$0.2 \cdot 10^{-3}$
3	1|1|1	99.1%	96.3%	90.6%	0.10	$1.4 \cdot 10^{-3}$	$0.6 \cdot 10^{-3}$
3	4|3|2	99.6%	95.7%	90.1%	0.08	$0.9 \cdot 10^{-3}$	$0.5 \cdot 10^{-3}$
4	1|1|1|1	98.4%	94.7%	86.3%	0.18	$2.4 \cdot 10^{-3}$	$0.8 \cdot 10^{-3}$
4	4|2|1|1	98.8%	93.4%	91.6%	0.12	$2.1 \cdot 10^{-3}$	$0.6 \cdot 10^{-3}$
8	1|1|1…|1	93.8%	90.6%	84.4%	0.74	$8.7 \cdot 10^{-3}$	$3.3 \cdot 10^{-3}$

ANC: Actual Number of Clusters, CR: Cluster Ratios.

Since only the PCC is examined here, the number of clusters is assumed to be the actual one. This is achieved by modifying the merge stage of the algorithm so that it terminates at the point where the number of clusters equals the Actual Number of Clusters (ANC). The ANC is equivalent to the number of textures from which feature vectors have been obtained. For that purpose, it is assumed that textures are homogeneous in the textural sense. This is a valid assumption as can be observed from the samples shown in Fig. 2.

Table 1 presents comparison results for three different circular invariant approaches, namely, the proposed CK-means algorithm, the original K-means using the FFT magnitudes of the energy feature vectors, and the original K-means using the maximum orientation-differences (MODF) [13] features. A MODF feature is defined as the maximum value considering all vector element differences for a given angular distance. In this example, angular distances of $m \cdot 18^\circ, m = 1, 2, \ldots, 5$ are used. Most often, small angular distances do not provide significant information since such orientation-difference values are expected to be small, thus, a larger number of angular distances does not necessarily contribute positively in correctly clustering data.

In Table 1, different cluster ratios (CR) are used. For instance, 4 | 3 | 1 indicates that the data set consists of vectors belonging to three clusters, one of which has four parts, another has three parts, and the third has one part. Similarly, 1 | 1 | 1 indicates that the number of vectors per cluster is the same for all three clusters.

Table 1 illustrates that SMCK-means provides accurate clustering results. More specifically, the proposed technique provides 3-5 percent higher PCC than the original K-means where the FFT magnitudes of the original vectors are used. The original K-means that uses MODF-5 does not provide significantly high PCC. Nevertheless, a relatively good performance is achieved for a considerably small number of energy features.

Table 1 also presents timing results. For the purpose of consistency, the average time required per iteration is presented for each technique. The algorithms have been implemented in MATLAB 6.1 and the results have been obtained on a P4, 3.2GHz, 2GB RAM computer. The timings depend on the vectors' dimensionality, and on the number of vectors and centroids, as was discussed in Section 2.3. However, the results also depend on the particular implementation and programming tools used. Nevertheless, it can be concluded that there is a tradeoff between clustering accuracy and clustering speed. It should also be mentioned that in the case where the approach presented in Section 2.3 had not been used, CK-means would have been of order N^3 (instead of $N \cdot M \cdot \log_2(M)$), resulting in much higher time requirements.

Table 2 presents results similar to Table 1 for the proposed method and for original K-means using FFT magnitudes, but, in this case, only texture combinations that involve texture 5 are considered. Thus, from Table 2, the advantage of CK-means for more challenging cases, such as the ones involving texture 5, is apparent. Table 2 shows that the CK-means provides a higher PCC than the FFT-magnitude and MODF-5 approaches by up to 10 percent and 14 percent, respectively.

4.2 Estimating the Correct Number of Clusters

This experiment is performed for the same set of eight textures, and the results are presented in Table 3. As mentioned previously, the original K-means also uses the split and merge approach introduced in this work. However, the original VRC index is employed that uses the original forms of BCD and WCD as shown in (27)-(29).

Table 3 presents the average number of clusters, the percentage of times each approach found the ANC correctly, the percentage of times that each approach erroneously estimated that the number of clusters is: one cluster more than the ANC, one cluster less than the ANC, and different by more than one clusters compared to the ANC. Table 3 illustrates that the proposed technique is most of the times successful in identifying the correct number of clusters, while when incorrect, it is mostly off by one cluster. On the other hand, the FFT-magnitude and the MODF-5 approaches are correct for a significantly less number of times.

TABLE 2
Average Percentage of Correct Clustering for the Proposed, the FFT-Magnitudes, and MODF-5 Approaches, Considering Only Combinations that Include Texture 5

		Average PCC		
ANC	CR	Proposed	FFT-magn	MODF-5
2	1|1	98.2%	90.0%	87.5%
2	2|1	98.8%	90.0%	88.0%
3	1|1|1	97.6%	90.1%	83.7%
3	4|3|2	98.9%	88.6%	83.5%
4	1|1|1|1	96.8%	89.5%	82.4%
4	4|2|1|1	98.0%	88.8%	83.9%

ANC: Actual Number of Clusters, CR: Cluster Ratios.

TABLE 3
Statistics from the "Number of Correct Clusters" Experiment for (a) the Proposed SMCK-Means Clustering Technique, (b) the Original K-Means Using the Vectors' FFT Magnitudes, and (c) the Original K-Means Using the MODF-5 Features

ANC	CR	Average NC	Correct NC	NC = ANC+1	NC = ANC-1	NC=ANC±*h*, *h*>1
2	1|1	2.00	100.0%	0.0%	0.0%	0.0%
2	2|1	2.00	100.0%	0.0%	0.0%	0.0%
3	1|1|1	3.12	83.9%	8.9%	3.6%	3.6%
3	4|3|2	3.20	75.0%	14.3%	5.4%	5.4%
4	1|1|1|1	4.20	71.4%	12.9%	8.6%	7.1%
4	4|2|1|1	3.94	58.6%	11.4%	24.3%	5.7%

(a)

ANC	CR	Average NC	Correct NC	NC = ANC+1	NC = ANC-1	NC=ANC±*h*,*h*>*1*
2	1|1	2.42	71.4%	21.4%	0.0%	7.1%
2	2|1	2.57	53.6%	39.3%	0.0%	7.1%
3	1|1|1	3.34	48.2%	25.0%	17.9%	8.9%
3	4|3|2	3.04	75.0%	8.9%	12.5%	3.6%
4	1|1|1|1	3.97	64.3%	5.7%	14.3%	15.7%
4	4|2|1|1	3.33	20.0%	14.3%	50.0%	15.7%

(b)

ANC	CR	Average NC	Correct NC	NC = ANC+1	NC = ANC-1	NC=ANC±*h*,*h*>*1*
2	1|1	2.25	75.0%	25.0%	0.0%	0.0%
2	2|1	2.25	75.0%	25.0%	0.0%	0.0%
3	1|1|1	2.64	46.4%	8.9%	44.6%	0.0%
3	4|3|2	2.55	33.9%	10.7%	55.4%	0.0%
4	1|1|1|1	2.94	21.4%	5.7%	34.3%	38.6%
4	4|2|1|1	2.89	21.4%	5.7%	28.6%	44.3%

(c)

ANC: Actual Number of Clusters, NC: Number of Clusters (Estimated), CR: Cluster Ratios.

The difference in performance between the proposed technique, the FFT magnitude features, and the MODF-5 features appears to be more significant in Table 3 than in Table 1. Nevertheless, the performance differences shown in Table 1 are also significant considering the high PCCs presented. Moreover, it is expected that detecting the correct number of clusters could be a more challenging task than performing clustering for a forced number of clusters. Furthermore, the number of clusters in the cases of FFT and MODF-5 features is usually off by one cluster. Therefore, the performance of these features is acceptable, yet not as accurate as the proposed technique. It is possible that some other split and merge technique would be more appropriate depending on the particular clustering algorithm and features used. On the other hand, one of the goals of this paper is to introduce a rotational invariant technique that performs well for an unknown number of clusters, which is not necessarily straightforward. The comparisons in Table 3 are presented to show that the proposed technique is successful in performing this task.

5 Conclusions

This paper presents an algorithm based on K-means, namely, SMCK-means, for circular invariant clustering of vectors. In general, one of the problems associated to the need for circular invariant clustering is that most feature vectors extracted from the images or objects under consideration are not circular invariant. Thus, a feature vector transformation which provides the desired invariance characteristic is required. In most cases, such a transformation ignores some vector information. In this paper, it is shown that eliminating such information is crucial.

On the other hand, the proposed technique performs clustering in a circular invariant manner without eliminating any information from the original feature vectors, with the exception of the circular shift. The proposed clustering algorithm is robust in terms of PCC and in terms of estimating the correct number of clusters. The proposed split and merge approach contributes significantly to that effect.

Multidirectional feature extraction can be applied to most applications that require directional representation of an object or texture, such as texture analysis, object shape extraction, junction shape/type extraction, and filter steerability. In this sense, the proposed approach is a general technique since it deals with such directional vectors. Additionally, more general forms of directional vector representation can be handled by the proposed technique. For instance, consider that S features have been extracted in each one of the M directions. Then, instead of the directional vector $\boldsymbol{F}_d$, the feature information is represented by an $M \times S$ array $\boldsymbol{A}_d$. Array $\boldsymbol{A}_d$ can be reformed to an

$M \cdot N$ column vector $\boldsymbol{F}'_d$ containing each row of $\boldsymbol{A}_d$. Based on this representation, the proposed algorithm can be used for clustering. Nevertheless, the distance of (17), should not be minimized for all possible shifts of vector $\boldsymbol{F}'_d$, but only for shifts equal to multiples of S.

Currently, the proposed method deals only with one aspect of pattern invariance. Similarly to rotation, scaling of an object or texture affects the feature extraction process. The plan for future work is to extent the proposed technique to handle clustering of data in both rotation and scale invariant manner.

REFERENCES

[1] A.K. Jain and R.C. Dubes, *Algorithms for Clustering.* Englewood Cliffs, N.J.: Prentice Hall, 1988.

[2] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[3] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum, 1981.

[4] J. Hartigan, *Clustering Algorithms.* New York: Wiley, 1975.

[5] J. Tou and R. Gonzalez, *Pattern Recognition Principles.* Reading, Mass.: Addison-Wesley, 1974.

[6] E. Ruspini, "A New Approach to Clustering," *Information Control,* vol. 15, no. 1, pp. 22-32, July 1969.

[7] M.C. Su and C.-H. Chou, "A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, June 2001.

[8] T. Kohonen, "The Neural Phonetic Typewriter," *Computer,* vol. 27, no. 3, pp. 11-12, Mar. 1988.

[9] T. Kohonen, *Self-Organization and Associative Memory,* third ed. New York, Berlin: Springer-Verlag, 1989.

[10] G.A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing,* vol. 37, pp. 54-115, 1987.

[11] D. Charalampidis, T. Kasparis, and M. Georgiopoulos, "Classification of Noisy Signals Using Fuzzy ARTMAP Neural Networks," *IEEE Trans. Neural Networks,* vol. 12, no. 5, pp. 1023-1036, Sept. 2001.

[12] H. Arof and F. Deravi, "Circular Neighborhood and 1-D DFT Features for Texture Classification and Segmentation," *IEE Proc. Vision Image and Signal Processing,* vol. 145, no. 3, pp. 167-172, June 1998.

[13] D. Charalampidis and T. Kasparis, "Wavelet-Based Rotational Invariant Segmentation and Classification of Textural Images Using Directional Roughness Features," *IEEE Trans. Image Processing,* vol. 11, no. 8, pp. 825-836, Aug. 2002.

[14] G.M. Haley and B.S. Manjunath, "Rotation-Invariant Texture Classification Using a Complete Space-Frequency Model," *IEEE Trans. Image Processing,* vol. 8, no. 2, pp. 255-269, Feb. 1999.

[15] F.S. Cohen, Z. Fan, and M.A. Patel, "Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 2, pp. 192-201, Feb. 1991.

[16] T. Kasparis, D. Charalampidis, J. Rolland, and M. Georgiopoulos, "Segmentation of Textured Images Using Fractals and Image Filtering," *Pattern Recognition,* vol. 34, no. 10, pp. 1963-1973, Oct. 2001.

[17] G.W. Milligan and M.C. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Phychometrica,* pp. 159-179, 1985.

[18] R.M. Gray, "Toeplitz and Circulant Matrices: A Review," http://www-ee.stanford.edu/~gray/toeplitz.html, Aug. 2005.

[19] S.Z. Selim and M.A. Ismail, "K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 1, 1984.

[20] W.T. Freeman and E.H. Adelson, "The Design and Use of Steerable Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 9, pp. 891-906, Sept. 1991.

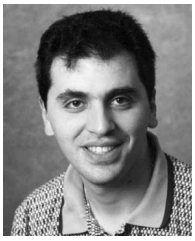
[21] D. Charalampidis, T. Kasparis, and L. Jones, "Multifractal and Intensity Measures for the Removal of Non-Precipitation Echoes from Weather Radars," *IEEE Trans. Geoscience and Remote Sensing,* vol. 40, no. 5, pp. 1121-1131, May 2002.

[22] J. Mao and A.K. Jain, "Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models," *Pattern Recognition,* vol. 25, no. 2, pp. 173-188, 1992.

[23] M. Georgiopoulos, I. Dagher, G.L. Heileman, and G. Bebis, "Properties of Learning of a Fuzzy ART Variant," *Neural Networks,* vol. 12, no. 6, pp. 837-850, 1999.

[24] H. Greenspan, S. Belongie, R. Goodman, and P. Perona, "Rotation Invariant Texture Recognition Using a Steerable Pyramid," *Proc. 12th IAPR Int'l Conf. Pattern Recognition—Conf. B: Computer Vision & Image Processing,* vol. 2, pp. 162-167, 1994.

[25] P.S. Bradley and U.M. Fayyad, "Refining Initial Points for K-Means Clustering," *Proc. 15th Int'l Conf. Machine Learning,* pp. 91-99, 1998.



Dimitrios Charalampidis (S'99-M'01) received the Diploma degree in electrical engineering and computer technology from the University of Patras, Greece, in 1996, and the MS and PhD degrees in electrical engineering from the University of Central Florida, Orlando, in 1998 and 2001, respectively. Since 2001, he has been an assistant professor in the Electrical Engineering Department at the University of New Orleans, Louisiana. His research interests include image processing, pattern recognition, digital signal processing, and neural networks. He is a member of the IEEE.