Novel Circular-Shift Invariant Clustering

Dimitrios Charalampidis

University of New Orleans, Electrical Engineering Department, 2000 Lakeshore Dr.
New Orleans, Louisiana, 70148, United States

Abstract. Several important pattern recognition applications are based on feature extraction and vector clustering. Directional patterns may be represented by rotation-variant directional vectors, formed from M features uniformly extracted in M directions. It is often required that pattern recognition algorithms are invariant under pattern rotation or, equivalently, invariant under circular shifts of such directional vectors. This paper introduces a K-means based algorithm (Circular K-means) to cluster vectors in a circular-shift invariant manner. Thus, the algorithm is appropriate for rotation invariant pattern recognition applications. An efficient Fourier domain implementation of the proposed technique is presented to reduce computational complexity. An index-based approach is proposed to estimate the correct number of clusters in the dataset. Experiments illustrate the superiority of CK-means for clustering directional vectors, compared to the alternative approach that uses the original K-means and rotation-invariant vectors transformed from rotation-variant ones.

1 Introduction

Texture analysis and object recognition have attracted great interest due to their large number of applications, including medicine, remote sensing [13], and industry. The term *object* may either be used to describe high level structures such as a vehicles or buildings, or low level image components such as edges or junctions. Both texture analysis and object recognition may require that the image be segmented into several regions. In general, the first step in the segmentation process is feature extraction. More specifically, multiple features are extracted from different image regions to form vectors representing those regions. Vector clustering [1]-[7] is usually the second step in the process. Similar vectors may correspond to similar regions, therefore clustering results in a segmented image. Generally, it is desirable that the segmentation process is invariant under image rotations, rotation invariant features may be needed. Such invariance is usually achieved by transforming rotation variant vectors into rotation invariant vectors. In this approach, M features, $\{f_m,\ m = 0, 1,\ldots, M-1\}$, may be uniformly extracted from M directions defined by the angle $\theta_m = 360^o m/M$ to form a rotational variant M-dimensional vector $\boldsymbol{F}_d = [f_0, \ldots, f_{M-1}]$. Traditionally, this feature vector may be transformed into a rotational invariant vector [8]-[11]. The problem with transforming rotation variant to rotation invariant feature vectors is that such a transformation results in

Charalampidis D. (2005).
Novel Circular-Shift Invariant Clustering.
In *Proceedings of the 5th International Workshop on Pattern Recognition in Information Systems*, pages 33-42
DOI: 10.5220/0002568500330042
Copyright © SciTePress

some loss of information. For instance, consider the Discrete Fourier Transform (DFT) coefficient magnitudes |DFT{$\boldsymbol{F}_d$}| of vector $\boldsymbol{F}_d$ defined above. These coefficients are invariant under image rotation by increments of $360^o/M$, since such a rotation causes a circular shift of vector $\boldsymbol{F}_d$. Some preprocessing [8] may achieve invariance of the DFT magnitude coefficients under any rotation. However, useful DFT phase information is ignored.

An impractical solution to the problem of information loss would be to examine all possible circular shifts for all vectors, and determine the best vector-grouping case regardless of the shift. On the other hand, rotation invariance can be effectively achieved by clustering the original feature vectors $\boldsymbol{F}_d$ using a circular-shift invariant clustering algorithm that causes no information loss. This paper introduces an algorithm, namely Circular K-means (CK-means), for clustering vectors with directional information, such as vector $\boldsymbol{F}_d$ in a circular invariant manner. Furthermore, an efficient Fourier domain representation of CK-means is presented to reduce computational complexity. An index based approach is proposed for estimating the correct number of clusters (CNC). The performance of CK-means has been tested on textural images.

The paper is organized as follows: In Section 2, the CK-means clustering algorithm is introduced. Section 3 presents examples to demonstrate the effectiveness of CK-means. Finally, Section 4 closes with some concluding remarks.

2 Circular-Shift Invariant K-means

First, the distance measure used by the technique and the algorithmic steps are presented. Then, the computational complexity of the algorithm is discussed.

2.1 Development of the Distance Measure

In the following, vectors and matrices are lowercase and uppercase, respectively. The vector or matrix superscripts specify 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 here is based on Mahalanobis distance (MD). The commonly used Euclidean distance is a special case of MD. The square of the MD between a vector $\mathbf{x}_j^N$ and a centroid $\mathbf{m}_l^N$ is defined as

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

or

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

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

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

where

$$\boldsymbol{\delta}_k = [\underbrace{0..0}_{k}010...0]^{\mathrm{T}} \tag{4}$$

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

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

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

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

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

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

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

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

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

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

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

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

where

$$\boldsymbol{\Phi}_j^{NN} = [\Im\{\mathbf{x}_j^N\} \ldots . \Im\{\mathbf{x}_j^N\}] \tag{11}$$

$$\mathbf{Q}_l^{NN} = \Im_v^{-1}\left\{\Im_h\{\mathbf{K}_l^{NN}\}\right\} \tag{12}$$

$$\mathbf{P}_{j,l}^{NN} = \boldsymbol{\Phi}_j^{NN} \circ \mathbf{Q}_l^{NN} \circ (\boldsymbol{\Phi}_j^{NN})^{\mathrm{H}}, \tag{13}$$

where superscript H denotes Hermitian transpose, $\mathbf{F}^{NN}$ is the $N \times N$ FT 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 FT operators applied on $\mathbf{X}^{NN}$ row- and column-wise.

Equation (10) uses the property [8] 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 a diagonal matrix representing, respectively, the FT and the conjugate FT coefficients of $\mathbf{x}_j^N$. Another property used in equation (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 $\mathbf{m}_l^N$ is:

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

and is circular-shift invariant. Based on the previous discussion, the *j*-th pattern is assigned to the cluster that provides the minimum distance measure:

$$D_j = \min_l\{D_{j,l}\} \tag{15}$$

2.2 CK-means

Essentially, CK-means employs a circular-shift invariant distance measure to assign each vector to a cluster. Then, each one of the vectors associated to a particular cluster is shifted by the shift that minimizes its distance from this cluster. Finally, the centroids and covariance matrices are updated (using the shifted vectors) as in the traditional K-means algorithm. Next, the steps of the CK-means are presented:

a. INITIALIZATION (Iteration 0): Calculate $\mathbf{\Phi}_j^{NN}$ (equation (11)) only once since it is iteration independent. Also, initialize the centroids and covariance matrices.

b. Iteration t:

1. Calculate $\mathbf{P}_{j,l}^{NN}$, $\mathbf{a}_l^N$ and b_l, for l = 1, …, L, once, in the beginning of each iteration, since they remain unchanged in a single iteration.
2. For each vector $\mathbf{x}_j^N$ and *l*-th cluster, calculate $D_{j,l}$ as in equation (14).
3. If the centroid that provides the minimum $D_{j,l}$ is l_o, 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 that minimizes D_{j,l_o} in (14).
4. Update the cluster centers and covariance matrices:

$$\mathbf{m}_l^N(t) = \frac{1}{J_l}\sum_{j_l}(\mathbf{x}_{j_l}^N * \boldsymbol{\delta}_{k_{j,l_o}}) \tag{16}$$

$$\mathbf{C}_l^{NN}(t) = \frac{1}{J_l}\sum_{j_l}(\mathbf{x}_{j_l}^N * \boldsymbol{\delta}_{k_{j,l_o}} - \mathbf{m}_l^N(t))(\mathbf{x}_{j_l}^N * \boldsymbol{\delta}_{k_{j,l_o}} - \mathbf{m}_l^N(t))^{\mathrm{T}} \tag{17}$$

where J_l is the total number of vectors associated to the *l*-th cluster, and j_l identifies vectors associated to the *l*-th cluster.

5. Stop if the measure $D_{sum} = \sum_{j=1}^{J} D_j$ does not decrease more than a specified threshold. Otherwise, go to step b.1.

It can be shown that the proposed algorithm converges since D_{sum} is reduced in each iteration: Step b.3 minimizes equation (14) for each individual pattern with respect to cluster and circular shift. Furthermore, similarly to the traditional K-means, step b.4 reduces the total distance D_{sum} between vectors and corresponding cluster centers.

2.3 FFT-based Implementation and Computational Complexity

The computational complexity of the distance defined in equation (15) can be reduced using a Fast Fourier Transform (FFT) approach. A Radix-2 FFT requires vector lengths N equal to a power of 2. If this condition is not satisfied, a modification can be applied to equation (10), so that the $\Im_h^{-1}$ operation can be performed with an Inverse FFT. Consider the N-point FT of a vector $\mathbf{v}^N$: $\mathbf{f}_v^N = \left(\mathbf{F}^{NN}\mathbf{v}^N\right)^{\mathrm{T}}$. Vector $\mathbf{v}^N$ can be interpolated by appropriate zero-insertion in $\mathbf{f}_v^N$, followed by the Inverse FT. Zero-insertion can be defined by the column-wise operator $\Delta^M\{\}$:

$$\Delta^M\left\{\mathbf{f}_v^N\right\} = \begin{cases} [f_v(1), \ \ldots \ f_v(\frac{N+1}{2}), \ \mathbf{0}^{M-N}, \ f_v(\frac{N+1}{2}+1), \ \ldots, \ f_v(N)]^{\mathrm{T}}, & N: odd \\ [f_v(1), \ \ldots \ f_v(\frac{N}{2})/2, \ \mathbf{0}^{M-N-1}, \ f_v(\frac{N}{2})^*/2, \ \ldots, \ f_v(N)]^{\mathrm{T}}, & N: even \end{cases} \quad (18)$$

where M = 2^{μ}, and $\mathbf{0}^N$ represents an all zeros column vector of size N. Thus, if the inverse FT is applied on vector $\Delta^M\left\{\mathbf{f}_v^N\right\}$, the result is a size N vector $\left(\mathbf{F}^{MM}\Delta^M\left\{\mathbf{f}_v^N\right\}\right)^{\mathrm{T}}$ which is an interpolated version of $\mathbf{v}^N$. Similarly to equation (18), each column of $\mathbf{P}_{j,l}^{NN}$ in equation (13) can be zero-inserted to obtain the $M \times N$ matrix:

$$\overline{\mathbf{P}}_{j,l}^{MN} = \Delta^M\left\{\mathbf{P}_{j,l}^{NN}\right\}, \quad (19)$$

Then, the $\Im_h^{-1}$ operation in equation (10) can be implemented using an Inverse FFT. Matrix $\Im_h^{-1}\left\{\overline{\mathbf{P}}_{j,l}^{MN}\right\}$ is also zero-inserted row-wise, to obtain

$$\overline{\overline{\mathbf{P}}}_{j,l}^{MM} = \Delta^M\left\{\left\{\Im_h^{-1}\left\{\overline{\mathbf{P}}_{j,l}^{MN}\right\}^{\mathrm{T}}\right\}^{\mathrm{T}}\right\} \quad (20)$$

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

$$\mathbf{r}_{j,l}^M = \Im^{-1}\left\{\Delta^M\left\{\Im\{\mathbf{a}_l^N\} \circ \Im\{\mathbf{x}_j^N\}^*\right\}\right\} \quad (21)$$

Based on the previous discussion, equation (10) can be expressed as

$$\mathbf{e}_{j,l}^N = diag\left\{\Im_v\left\{\overline{\overline{\mathbf{P}}}_{j,l}^{MM}\right\}\right\} \quad (22)$$

Considering that there is a reasonably large number of iterations, step a. of the algorithmic description in Section 2.2 does not require significant additional processing time. Similarly, step b.1 is only performed once for every iteration.

Step b.2. requires calculation of equations (21) and (22). Equation (21) requires N multiplications for the element-wise vector product and $M \cdot \log_2 M$ operations for the Inverse FFT. Equation (22) requires calculation of equations (13) and (20), which require, respectively, $2N$ multiplications for the element-wise matrix products, and $N \cdot M \cdot \log_2(M)$ operations. The FT in equation (22) is needed only for the diagonal elements of the resulted matrix, thus it requires approximately M^2 operations. All above calculations are required for each iteration, and for all combinations of J vectors and L clusters. The computational complexity for step b.4 is $O\{J \cdot M^2\}$. Thus, the complexity is $O\{ J \cdot L \cdot N \cdot M \cdot \log_2(M)\}$. For Euclidean distance, it can be shown that the complexity is $O\{ J \cdot L \cdot M \cdot \log_2 M\}$. The complexity of the traditional circular variant K-means algorithm is $O\{J \cdot L \cdot N^2\}$ for Mahalanobis and $O\{ J \cdot L \cdot N\}$ for Euclidean distance.

2.4 Estimating the Actual Number of Clusters (ANC)

In this paper, the technique for estimating the ANC introduced in [9] 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*) given by:

$$CIVRC_L = \frac{(J-L)BCD_L}{(L-1)WCD_L} \tag{23}$$

where BCD_L is the "between clusters distance", 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 defined as:

$$WCD_L = \sum_{l=1}^{L} J_l \sum_{j=1}^{N} diag\left\{ \mathbf{C}_l^{NN} \right\} \quad \Big| \text{ for } L \text{ clusters} \tag{24}$$

$$BCD_L = TD - WCD_L \tag{25}$$

where

$$TD = J \sum_{j=1}^{N} diag\left\{ \mathbf{C}_1^{NN} \right\} \quad \Big| \text{ for one cluster} \tag{26}$$

and J_l is the number of vectors associated to the l-th cluster. Considering different numbers of clusters L, the estimated number of clusters is the one that maximizes $CIVRC_L$. TD measures the "extent" of the vector set in N-dimensions. Since $CIVRC_L$ is proportional to $BCD_L / WCD_L = TD/WCD_L - 1$, TD acts as a normalization factor.

3 Experimental Results

The algorithm is evaluated on a texture clustering problem. Textural energy features are extracted from eight different textures of size 512 × 512. A directional exponential filter $g(i,m)$ is chosen to extract energy features in several orientations:

$$g(i,m) = -i \cdot \exp\left\{-\frac{i^2}{s_i^2} - \frac{m^2}{s_m^2}\right\} \tag{27}$$

where s_i^2 and s_m^2 are the horizontal and vertical variances of the Gaussian envelope. The oriented texture energy at location (*i*,*m*) is defined as:

$$E^{\varphi}(i,m) = \frac{1}{W_E^2} \sum_{m'=-W_E/2}^{W_E/2} \sum_{i'=-W_E/2}^{W_E/2} \left| I_{g^{\varphi}}(i+i', m+m') \right| \tag{28}$$

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

$$E_n^{\varphi}(i,m) = E^{\varphi}(i,m) / E^T(i,m) \tag{29}$$

where

$$E^T(i,m) = \sum_{\varphi} E^{\varphi}(i,m) \tag{30}$$

is the total energy over all directions. Energy vectors are extracted from texture blocks of size $W_E \times W_E$ (here $W_E = 64$). Each vector consists of the energy calculated uniformly in 32 orientations in the interval (θ, $180^o+\theta$), where θ specifies the starting orientation in the interval. In order to evaluate the circular-shift invariant performance of the algorithm, feature vectors should be extracted from several rotated versions of the same texture. Equivalently, several feature vectors can be extracted from the same block considering different starting orientations θ.

Fig. 1(a) shows texture samples from the textures used in the experiments. Fig. 1(b) shows polar plots of eight feature vectors, one vector sample from each of the eight textures. In general, oriented energy feature vectors are periodic with period 360^o. In this particular case where the filters used are anti-symmetric, feature extraction results in feature vectors with period 180^o.

3.1 Percentage of Correct Clustering

In this work, the clustering performance is measured using the *Percentage of Correct Clustering* (PCC) [14] defined as follows. Let $l = 1, 2, \ldots, L$, be the index in a set of L known labels, and $l' = 1, 2, \ldots, L'$ be the index that identifies the l'-th cluster out of L' clusters. A cluster l' is labeled "l", if the number of vectors labeled l, contained in l', is larger than the number of vectors, also contained in l', labeled with any other single label. Then, the PCC for cluster l' is defined as:

$$\mathrm{PCC}_{l'} = 100\ J_l / J_{l'} \tag{31}$$

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

$$\mathrm{PCC} = \frac{100}{J} \sum_{l'} \mathrm{PCC}_{l'}\ J_{l'} \tag{32}$$

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. The average PCC is found in each case. Moreover, clustering using all eight textures is performed.

Since only the PCC is examined here, the number of clusters is assumed to be equal to the ANC, which is equivalent to the number of textures from which feature vectors have been obtained. 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 feature vectors [8], and the original K-means using the maximum orientation-differences [9]. A maximum orientation-difference feature is defined as the maximum value considering all vector element differences for a given angular distance. In this example, angular distances of $m \cdot 18^{o}$, m = 1, 2, …, 5 are used. Small angular distances do not add significant information, since they are expected to be small. Thus, a larger number of angular distances does not necessarily add to the clustering performance.

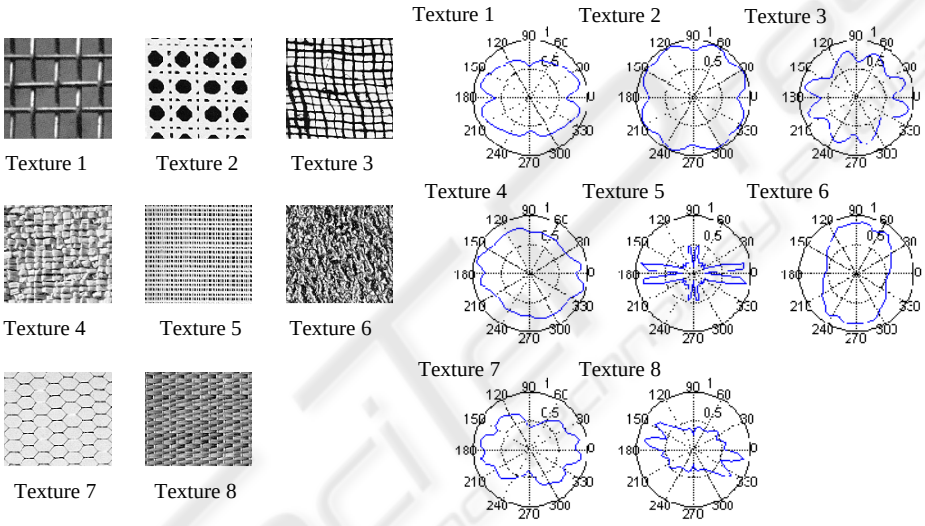

Fig. 1. (a) Samples from the textures used for the clustering experiment. (b) Polar plots of energy extracted in multiple orientations from a 64×64 block from each texture of Fig. 1(a).

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 4 parts, another has 3 parts and the third has 1 part. Similarly, 1|1|1 indicates that the number of vectors per cluster is the same for all three clusters. Table 1 illustrates that CK-means provides accurate clustering results. More specifically, the proposed technique provides 3-5% or 2-9% higher PCC than the original K-means, where the FFT magnitudes of the original vectors or the MODF - 5 features are used.

Table 1. Average percentage of correct clustering for three approaches.

		Average PCC		
ANC	CR	Proposed	FFT-magn.	MODF - 5
2	1|1	99.6%	96.7%	96.0%
2	2|1	99.7%	96.7%	94.3%
3	1|1|1	99.1%	96.3%	90.6%
3	4|3|2	99.6%	95.7%	90.1%
4	1|1|1|1	98.4%	94.7%	86.3%
4	4|2|1|1	98.8%	93.4%	91.6%
8	1|1|…|1	93.8%	90.6%	84.4%

3.2 Estimating the Actual Number of Clusters

This experiment is performed for the same set of eight textures, and the results are presented in Table 2. Table 2 presents the average number of clusters, and the percentage of times each approach found: the correct ANC, one cluster different than the ANC, and more than one clusters different than the ANC. Table 2 illustrates that the proposed technique is mostly 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 approach is correct for a significantly less number of times, while it is frequently off by more than one cluster.

Table 2. Statistics from the "Number of Correct Clusters" experiment for the proposed CK-means clustering technique, and the original K-means using the vectors' FFT magnitudes.

		Proposed Algorithm				Original K-means			
ANC	CR	Average NC	Correct NC	NC = ANC±1	NC=ANC±*h* *h*>1	Average NC	Correct NC	NC= ANC±1	NC=ANC±*h* *h>1*
2	1|1	2.00	100.0%	0.0%	0.0%	3.20	46.7%	13.3%	40.0%
2	2|1	2.00	100.0%	0.0%	0.0%	3.40	40.0%	13.3%	40.0%
3	1|1|1	2.89	78.6%	21.5%	0.0%	4.54	32.1%	25.0%	42.9%
3	4|3|2	2.88	57.1%	41.1%	1.8%	4.43	33.9%	23.2%	42.9%
4	1|1|1|1	3.87	61.6%	35.7%	2.9%	4.69	60.0%	27.2%	24.3%
4	4|2|1|1	3.61	42.9%	22.8%	14.3%	6.24	12.9%	24.3%	62.9%

4 Conclusions

This paper presents an algorithm based on K-means, namely CK-means, for circular invariant vector clustering. 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 CK-means performs clustering in a circular invariant manner without eliminating information from the original feature vectors, other than the circular shift. Furthermore, CK-means is robust in terms of PCC and in terms of estimating the Actual Number Clusters. Furthermore, the computational complexity of CK-means is not significantly higher than that of K-means.

References

1. Jain A.K. and Dubes R.C.: Algorithms for Clustering. Englewood Cliffs, N.J. Prentice Hall, (1988)
2. Duda R.O. and Hart P.E.: Pattern Classification and Scene Analysis. New York, Wiley, (1973)
3. Bezdek J.: Pattern Recognition with Fuzzy Objective Function Algorithms. New York, Plenum, (1981)
4. Hartigan J.: Clustering Algorithms. New York, Wiley, (1975)
5. Tou J. and Gonzalez R.: Pattern Recognition Principles. Reading, Mass., Addison-Wesley, (1974)
6. Ruspini E.: A New Approach to Clustering. Information Control, Vol. 15, No. 1, (1969) 22-32
7. Su M.C. and Chou C.-H.: A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 6, June (2001)
8. Arof H. and Deravi F.: Circular Neighborhood and 1-D DFT Features for Texture Classification and Segmentation," IEE Proceedings, Vision Image and Signal Processing, Vol. 145, No. 3, (1998) 167-172
9. Charalampidis D. and Kasparis T.: Wavelet-based Rotational Invariant Segmentation and Classification of Textural Images Using Directional Roughness Features. IEEE Transactions on Image Processing, Vol. 11, No. 8, (2002) 825-836
10. Haley G.M. and Manjunath B.S.: Rotation-invariant Texture Classification Using a Complete Space-Frequency Model. IEEE Trans. on Image Processing, Vol. 8, No. 2, (1999) 255-269
11. Cohen F.S., Fan Z., and Patel M.A.: Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 2, (1991) 192-202
12. Gray R.M.: Toeplitz and Circulant Matrices: A Review. Web Address: http://ee.stanford.edu /~gray/toeplitz.pdf
13. Charalampidis D., Kasparis T., and Jones L.: Multifractal and Intensity Measures for the Removal of Non-precipitation Echoes from Weather Radars. IEEE Transactions on Geoscience and Remote Sensing, Vol. 40, No. 5, (2002) 1121-1131
14. Georgiopoulos M., Dagher I., Heileman G.L. and Bebis G.: Properties of learning of a Fuzzy ART Variant. Neural Networks, Vol. 12, No. 6, (1999) 837-850