FUZZY CLUSTERING FOR IMAGE SEGMENTATION USING GENERIC SHAPE INFORMATION

M. Ameer Ali
Department of ECE, East West University, Mohakhali, Dhaka-1217, Bangladesh.
Tel: +88-1918921315, Email: ameer7302002@yahoo.com

Gour C. Karmakar
Gippsland School of Information Technology, Monash University, Churchill, VIC-3842, Australia
Tel: +61-3-9902-6252, Fax: +61-3-9902-6879, Email: Gour.Karmakar@infotech.monash.edu.au

Laurence S. Dooley
Department of Communication and Systems, The Open University, Milton Keynes, United Kingdom
Tel: +44 1908 652017, Fax: +44 1908 653718, Email: L.S.Dooley@open.ac.uk

ABSTRACT

The performance of clustering algorithms for image segmentation are highly sensitive to the features used and types of objects in the image, which ultimately limits their generalization capability. This provides strong motivation to investigate integrating shape information into the clustering framework to improve the generality of these algorithms. Existing shape-based clustering techniques mainly focus on circular and elliptical clusters and so are unable to segment arbitrarily-shaped objects. To address this limitation, this paper presents a new shape-based algorithm called fuzzy clustering for image segmentation using generic shape information (FCGS), which exploits the B-spline representation of an object's shape in combination with the Gustafson-Kessel clustering algorithm. Qualitative and quantitative results for FCGS confirm its superior segmentation performance consistently compared to well-established shape-based clustering techniques, for a wide range of test images comprising various regular and arbitrary-shaped objects.

Keywords: Image Segmentation, Generic Shape, Fuzzy Clustering, B-spline

1.0   INTRODUCTION

Image segmentation is formally defined as the process of separating mutually exclusive regions (objects) of interest from other regions (objects) in an image [1]. While it plays a fundamental role in image analysis, understanding and coding with a wide diversity of applications, ranging from car assembly, airport identification, security, object recognition and second generation image coding, through to criminal investigative analysis and medical imaging [2], [3], it is also very challenging because there are an inordinate number of objects and wide variations amongst them which makes it difficult to estimate every object within a generic framework. Most real-world images possess a certain amount of ambiguity and hence the segmentation produces fuzzy regions. Fuzzy image segmentation techniques are much more adept at handling such uncertainty than classical techniques and in this context fuzzy clustering algorithms have become popular and widely applied in image segmentation [4].

The effectiveness of fuzzy clustering algorithms such as [4], [5], and [6] is very dependent on the types of features used and objects in an image. This constrains the generalisation capability of an algorithm, raising the question of which feature set produces the best segmentation results for which image type. As shape is a key perceptual attribute and humans generally use it to detect and recognise objects, this provided the motivation to examine incorporating generic shape information into the image segmentation process. There are many existing shape-based image segmentation methods, which are all based upon fuzzy clustering principles. These include the *Gustafson-Kessel* (GK) algorithm [7], and techniques that focus on separating regular geometric shapes such as *fuzzy k-ring* (FKR) [8], *fuzzy circular shell* (FCS) [9], *fuzzy c-ellipsoidal shells* (FCES) [10] and *fuzzy k-ellipse* (FKE) [11]. The FKR and FCS algorithms were originally designed to separate objects which were either ring, compact spherical or a combination of ring-shaped, while FCES and FKE respectively became extensions of these two clustering algorithms, broadening the application area of shape-based segmentation so that ring and elliptical objects (or a combination of them), could be detected and separated. However, a fundamental drawback is that most real world objects are neither ring nor elliptically shaped, so existing segmentation algorithms are limited in effectively being

applied to arbitrary shaped objects. The GK algorithm performs segmentation of objects in an image using a covariance matrix that automatically adapt to the shape of a cluster using local data distance. Even though it does not consider any explicit shape information, it may consider the shape based clustering due to having capability of adapting the shape of a cluster applying covariance matrix. Since GK does not explicitly consider shape information, and is unable to accurately segment arbitrary-shaped objects in an image. This paper presents a *fuzzy clustering for image segmentation using generic shape information* (FCGS) algorithm which explicitly incorporates generic shape information, by embedding B-spline representations of an object within the GK clustering framework. This integration of shape information specifically requires either manual or automatic determining of a set of contour points in order to create a suitable shape feature descriptor for each object. For FCGS, while in principle any clustering algorithm can be used for automatic initialization, the GK algorithm is adopted because of its inherent property of adapting to the local topological structure of a cluster.

The remainder of this paper is organized as follows: Section 2 provides a review of the key properties of existing shape-based fuzzy clustering algorithms, while Section 3 explores the underlying theory of shape contour point generation techniques. The new FCGS algorithm, including its mathematical foundations is detailed in Section 4, with the object-based segmentation performance of FCGS fully analysed in Section 5. Some conclusions are provided in Section 6.

2.0   SHAPE-BASED CLUSTERING ALGORITHMS

Before detailing the new FCGS algorithm, the merits and limitations of existing shape-based techniques are reviewed.   While *Fuzzy c-means* (FCM) [4] is undoubtedly the most popular and recognised algorithm for clustering, its performance is very sensitive to the features used and does not consider object shape information in image segmentation. To address this, a number of shape-based clustering methods have been proposed, including the FCS, FKR, FCES and FKE algorithms mentioned in Section 1, all of which are based on the iterative minimisation of an FCM-type objective function.

The FCS algorithm was designed to specifically detect and separate circular structures in an image as it uses the distance from the circular shell to the corresponding data point in its objective function. Two constraints are applied using the ratio between the distance of the data from the cluster shell and the data distance from the cluster centre. To update the cluster centre and radius of the circular shell, Newton's iteration is employed. While FCS performs well for circular-type shapes, it is ineffectual in separating other shapes.

FKR is another circular ring-shaped clustering algorithm based upon FCM. Membership functions, cluster centres and the radius of a cluster are iteratively updated based on the minimisation criteria of an objective function defined for circular ring shapes. It can detect and separate ring, spherical compact clusters and combinations of ring-shaped clusters, but again it is unsuitable for more arbitrarily-shaped objects because of its restricted mathematical model.

Both the FCES and FKE algorithms are extended versions supporting more generic-shaped object segmentation. These can detect and separate circular, elliptical and various combinations of the two geometric shapes. FCES has the same constraints as FCS employing a covariance matrix to calculate the distance between each datum and cluster centre. The matrix adapts the data distance to the shape by taking into account the orientation and scaling of the shell. Newton's iteration is again applied to update the cluster centre and radius. While FKE was originally proposed as an extension to FKR for elliptical shapes, both algorithms can detect and separate shapes having ring and elliptical features, but again fail in being able to segment arbitrary shaped objects. A clear objective for the new FCGS algorithm is therefore to be able to accurately segment generic shapes, with a key element of the strategy being to use the GK algorithm for initialization before automatically determining each object's shape feature descriptor. The GK algorithm is now to be presented in greater detail.

The GK algorithm is characterised by adapting automatically the local data distance metric to the shape of the cluster using a covariance matrix [7], [12], which is based on the iterative optimization of the following objective function:

$$J_q(S,\mu,V) = \sum_{j=1}^{n}\sum_{i=1}^{c}\left(\mu_{ij}\right)^q D_{ij}^2 \qquad (1)$$

123

$$0 \leq \mu_{ij} \leq 1; \quad i \in \{1,\ldots,c\} \; and \; j \in \{1,\ldots,n\} \tag{2}$$

$$\sum_{i=1}^{c} \mu_{ij} = 1; \quad j \in \{1,\ldots,n\} \tag{3}$$

where c and n are the number of clusters and data respectively; μ is a set of membership values μ_{ij}; V is a vector containing the values of cluster centers v_i; q is the fuzzifier $1 < q \leq \infty$ and D_{ij} is the distance norm calculated for clusters of different shapes in one data set using the following:

$$D_{ij}^2 = \left(S_j - v_i\right)^T A_i \left(S_j - v_i\right) \tag{4}$$

where A_i is the norm inducing matrix that allows the distance norm to adapt to the local topological structure of the data [12], $1 \leq i \leq c$ and $1 \leq j \leq n$. If $D_{ij} = 0$ for the case where $S_j = v_i$ or $A_i = 0$, in this case a crisp decision is necessary i.e. the jth datum must be classified into the ith cluster. So, the membership value μ_{ij} is determined as follows:

$$\text{IF } \left(D_{ij} = 0\right) \text{ THEN } \mu_{ij} = 1 \text{ maintaining } \sum_{i=1}^{c} \mu_{ij} = 1 \text{ i.e. by setting } \sum_{\substack{k=1 \\ k \neq i}}^{c} \mu_{kj} = 0 \tag{5}$$

ELSE

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{D_{ij}}{D_{kj}}\right)^{\frac{2}{q-1}}} \tag{6}$$

while the cluster centre v_i is updated as:

$$v_i = \frac{\sum_{j=1}^{n} \left(\mu_{ij}\right)^q S_j}{\sum_{j=1}^{n} \left(\mu_{ij}\right)^q} \tag{7}$$

A norm inducing matrix A_i is used to adapt to the local topological structure of the data:

$$F_i = \frac{\sum_{j=1}^{n} \left(\mu_{ij}\right)^q \left(S_j - v_i\right)^T \left(S_j - v_i\right)}{\sum_{j=1}^{n} \left(\mu_{ij}\right)^q}$$

$$A_i = \left[\rho_i \det(F_i)^{\frac{1}{n}} F_i^{-1}\right] \tag{8}$$

where F_i is the *fuzzy covariance* matrix and ρ_i the cluster volume which is usually set to 1. From a shape-based clustering perspective, while the GK algorithm does not explicitly incorporate shape information, it exhibits the useful property of locally adapting to the distance matrix (4), and as a consequence is employed in the new FCGS algorithm for initialisation purposes to generate the initial shape contour. The various steps involved in the GK algorithm [7] are given in Algorithm 1.

Algorithm 1: *The Gustafson-Kessel* (GK) *Algorithm*.
Input: The number of clusters c, max *iteration* and threshold ξ

Output: Segmented regions $\Re$.

1. Fix $q = 2$.

2. Initialize μ_{ij}.

3. FOR $l = 1, 2, \cdots, \max iteration$

    4. Update v_i using (7).
    5. Calculate the covariance matrices by (8).
    6. Calculate the distance norms using (4)
    7. Update μ_{ij} using (5) and (6).

    8. IF $\left\| \mu_{ij}^{\ l} - \mu_{ij}^{\ l-1} \right\| \leq \xi$ FOR $\forall i, j$ THEN STOP

    ENDFOR

A major consideration for any clustering algorithm, especially one being applied for object-based segmentation, is how to determine the number of clusters (objects) c either directly from image data or using *a priori* knowledge. There is no single unified definition of what exactly constitutes an object, as there typically exist a large number of objects and their definition very much depends on the user's perception and purpose of the application. In this context, the number of clusters c can either be manually provided or automatically determined from image data. In the latter case, the standard approach adopted is to use validity measures to determine an optimal number of clusters, with examples provided in [4], [9], [13], [14], [15]. There are certain situations however, particularly in object-based segmentation where such validity algorithms fail to generate the correct (optimal) number of clusters [14], [16] because they tend to focus on the homogeneous regions of interest in an image, which can contradict with the human perception of an object so degrading the overall performance of the clustering algorithm. Conversely, there are numerous applications in the manufacturing and medical imaging [3] domains whereby the number of objects to be segmented is known *a priori,* hence in this paper, for all shape based clustering algorithms, the number of clusters c is provided.

3.0   SHAPE CONTOUR POINT GENERATION TECHNIQUES

In order to define the shape feature descriptor for a particular object, it is firstly necessary to compute the contour points, which can be obtained using a parametric curve generation technique such as Bezier curves (BC) or B-splines [17], [18], [19], [20]. Being a member of the parametric curve family, B-spline is used to generate shape contour points in the FCGS algorithm, because it provides greater control flexibility as the degree of a B-spline polynomial is independent of the number of significant points and it is also possible to achieve a desired level of local control over the shape of a spline curve with shape parameters, that is unachievable using BC [18], [20]. The fundamentals of B-spline shape representation [17], [18] are now briefly examined.

3.1   B-spline Shape Representation

For a given set of *significant points* and knot vector, B-splines generate contour (curve) points, with the resulting curve always passing through the first and last significant points as illustrated in Fig. 1. The degree of the B-spline polynomial is independent of the number of significant points and hence it can handle efficiently a large number of them. The mth order B-spline function [17], [18] comprises $(m+1)$ piecewise polynomial segments of degree (m). If $(L+1)$ is the number of significant points in vector **P**, the positions of the significant points are defined as

125

$P_k : (x_k, y_k)$ where $0 \leq k \leq L$. The B-spline blending function of order m is $N_{k,m}(t)$ and the B-spline curve equation $\mathbf{P}(t)$ is:

$$\mathbf{P}(t) = \sum_{k=0}^{L} P_k N_{k,m}(t) \tag{9}$$

The B-spline blending function is:

$$N_{k,m}(t) = \left(\frac{t - t_k}{t_{k+m-1} - t_k} \right) N_{k,m-1}(t) + \left(\frac{t_{k+m} - t}{t_{k+m} - t_{k+1}} \right) N_{k+1,m-1}(t) \tag{10}$$

Where $N_{k,1}(t) = \begin{cases} 1 & \text{if } t_k \leq t \leq t_{k+1} \\ 0 & \text{otherwise} \end{cases}$ (11)

where $[t_0, t_1, \cdots]$ represents the knot point sequence. Fig. 1 shows an example of a 2nd order B-spline curve representation with $0 \leq t \leq 3$ for 4 significant points and a uniform knot sequence.
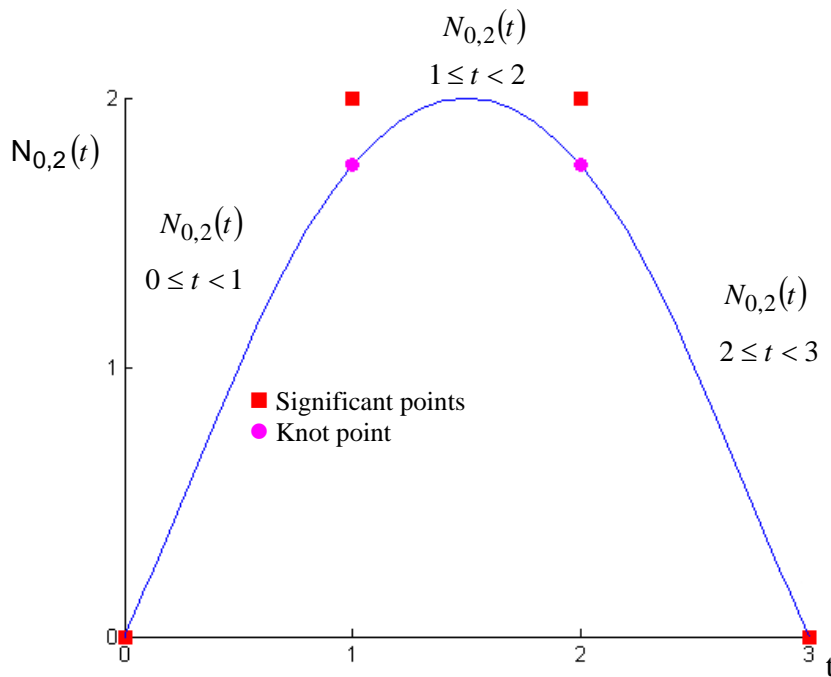

Fig. 1: B-spline approximation for 4 significant points.

4.0   INTEGRATING GENERIC SHAPE INFORMATION

This section formally presents the *fuzzy clustering for image segmentation using generic shape information* (FCGS) algorithm. In order to embed shape information seamlessly into the image segmentation process, it is necessary to generate a shape feature descriptor for each identified object. The set of significant points for a shape is generated by scanning the boundary points of the respective initial segmented regions using the convex hull, before a B-spline approximation is applied to derive the requisite shape contour points. These points are used to measure the data distance from the shape contour adopting the same strategy employed by other shaped-based clustering algorithms such as FKR, FKE, FCS and FCES (Section 2.0). To calculate this distance, the point on the shape for each datum that is closest to the corresponding datum must be found. This point is the *intersection point* between the shape and

a line from the data point passing through the cluster centre. The technique used to find the location of this intersection point on the B-spline shape descriptor is elaborated further in the following section.

4.1 *Locating the Intersection Points*

In any segmentation strategy, the most important consideration is how to derive the distance d_{ij} of a datum for the objective function. For example, in FCM, d_{ij} is calculated from the cluster centre in order to segment objects in an image based on some predefined features, while for FKR and FKE, d_{ij} is respectively calculated from the contour of the circle and ellipse. Since arbitrary shapes are considered in the new FCGS algorithm, d_{ij} is calculated from the respective contour (*intersection*) points of that particular shape. To provide an intuitive insight of this distance and the role of the B-spline approximation, consider the *butterfly* object in Fig. 2 (a) and its corresponding B-spline representation Fig. 2 (b), where v_i is assumed to be the centre of the initial object shape produced by a B-spline. S_j is a datum with its corresponding intersection point $S_{ij}^{'}$, so d_{ij} is the distance between S_j and $S_{ij}^{'}$. The challenge now is how to calculate the intersection point $S_{ij}^{'}$ between the initial contour and the line $\left(l_{ij} \right)$ that connects the datum S_j with the centre of the ith cluster v_i (Fig. 2 (b)). In the Cartesian coordinate system, the intersection point can be calculated using the following heuristic approach: i) Find the two successive points on the ith shape contour that are closest and lie opposite sides of the line l_{ij} i.e. if one closest point is in '+ve' side of l_{ij}, the other will be in '-ve' side; ii) Determine a point that lies between these two points and also on the boundary of the curve and l_{ij}; this is *intersection point* $S_{ij}^{'}$. Locating these two contour and intersection points is however, computationally expensive so a more efficient approach is to employ polar coordinates to compute $S_{ij}^{'}$, where an arbitrary point is represented by (r, θ), with r being the distance between the centre v_i and the respective point and θ the angle between line l_{ij} and the horizontal line passing through the cluster centre v_i as illustrated in Fig. 2 (b). S_j and $S_{ij}^{'}$ are now denoted as $\left(r_{ij}, \theta_{ij} \right)$ and $\left(r_{ij}^{'}, \theta_{ij}^{'} \right)$ respectively, where it is assumed each datum S_j and its corresponding contour point $S_{ij}^{'}$ do not necessarily make exactly the same angle with the cluster centre v_i, though often $\theta_{ij} = \theta_{ij}^{'}$. Also, if more than one point is found for a particular angle, the closest intersection point to the data is always considered. The various steps involved in determining each intersection point on the shape contour are formalized in Algorithm 2.
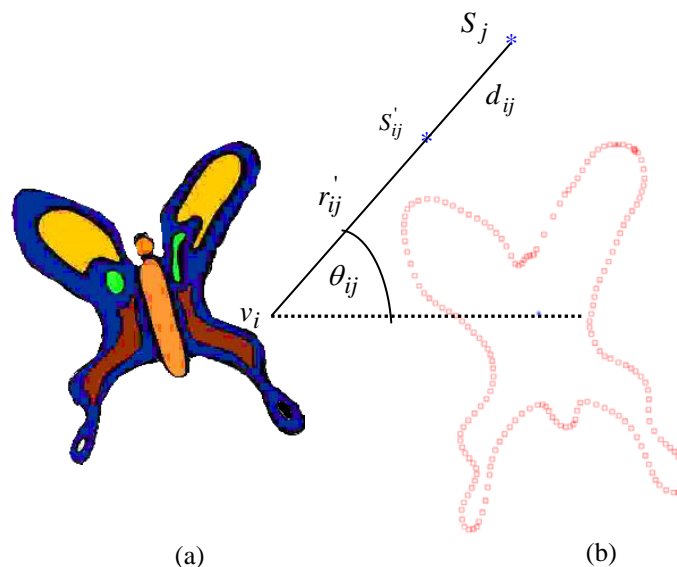

(a) (b)

Fig. 2: (a) Original *butterfly* object; (b) Example of the *intersection point* between datum S_j and the B-spline shape contour of (a) along line l_{ij}.

4.2 *Mathematical Foundations*

Having obtained the intersection points (Section 4.1), d_{ij} is now calculated to be used in the objective function. As mentioned in Section 2.0, the GK algorithm is employed because of its property that the cluster adapts to the local topological structure of a shape, so the objective function used in the FCGS algorithm is based upon Eq. (1):

$$J_q(\mu, V) = \sum_{i=1}^{c} \sum_{j=1}^{n} (\mu_{ij})^q d_{ij}^2 \tag{12}$$

$$\text{subject to } \sum_{i=1}^{c} \mu_{ij} = 1 \text{ and } d_{ij} = D_{ij} - r_{ij}^{'} \tag{13}$$

where $r_{ij}^{'}$ is the Euclidean distance between datum S_j and the corresponding intersection point $S_{ij}^{'}$ on the shape contour and the other parameters are defined in Section 2. If $d_{ij} = 0$, the corresponding j^{th} datum must be classified into the ith cluster and in this case, a crisp decision is necessary i.e., the membership value $\mu_{ij} = 1$ in Eq. (14), otherwise it will be updated according to the distance d_{ij} which is derived in Eq. (15) using Eq. (12).

$$\text{IF } \left(d_{ij} = 0\right) \text{ THEN update } \mu_{ij} \text{ using (5)} \tag{14}$$

$$\text{ELSE } \quad \mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{q-1}}} \tag{15}$$

Algorithm 2: *Determine the intersection point between a datum and its corresponding cluster centre.*

Input: cluster contour points, cluster centre v_i and data S_j.

Output: The final intersection point $S_{ij}^{'}$.

1. Convert contour points to polar form $\left(r_{ij}^{'}, \theta_{ij}^{'}\right)$ with respect to corresponding cluster centre v_i.

2. Convert data points into polar form $\left(r_{ij}, \theta_{ij}\right)$ with respect to corresponding cluster centre v_i.

3. Calculate $\left(\Delta\theta_{ijk} = \left|\theta_{ij} - \theta_{ik}^{'}\right|\right)$ where $1 \leq k \leq N_i^{''}$ and $N_i^{''}$ is the number of contour points of the ith cluster.

4. Intersection point $S_{ij}^{'}$ is the point with minimum $\Delta\theta_{ijk}$.

5. If more than one point have minimum $\Delta\theta_{ijk}$, the closest point of datum S_j will be the intersection point $S_{ij}^{'}$.

During the iterative process, it needs to update the cluster shape and the main goal of this proposed method is to preserve the original (initial) shape during iterative scaling. For example if the initial shape is a circle, then it will remain circle during iterative scaling. To achieve this goal, the shape i.e. every contour point is updated by the same amount by multiplying all the contour radii by a scaling factor $(Ratio)$. To define the scaling factor $(Ratio)$, the

average radius of a cluster shape is exploited by considering the minimization of the objective function based on a circular shape [21]. The average radius i.e. circular radius r_i used in this process is expressed as:

$$r_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^q D_{ij}}{\sum_{j=1}^{n} (\mu_{ij})^q} \qquad (16)$$

Now, the scaling factor $(Ratio)$ is defined as in Eq. (17), i.e. the ratio of the current (r_i) to the previous radius (r_{i-1}) of the circular shape:

$$Ratio = \frac{r_i}{r_{i-1}} \qquad (17)$$

To update the ith cluster centre v_i, the objective function in Eq. (12) is optimized together with (13) with respect to v_i using Lagrangian multiplier and hence can be derived as follows:

$$v_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^q \begin{pmatrix} f_x \\ f_y \end{pmatrix}}{\left\{ 2 \sum_{j=1}^{n} (\mu_{ij})^q \right\}} \qquad (18)$$

where f_x and f_y are the X- and Y-component of the ith cluster centre v_i respectively and are as follows:

$$f_x = S_{j1} - D_{ij} \frac{S'_{ij1} - v_{i1}}{r'_{ij}} + S'_{ij1} - r'_{ij} \frac{S_{j1} - v_{i1}}{D_{ij}}$$

$$f_y = S_{j2} - D_{ij} \frac{S'_{ij2} - v_{i2}}{r'_{ij}} + S'_{ij2} - r'_{ij} \frac{S_{j2} - v_{i2}}{D_{ij}}$$

For an image, the 2-D data and cluster centre are $S_{jl} = \begin{bmatrix} S_{j1} \\ S_{j2} \end{bmatrix}$ and $v_{il} = \begin{bmatrix} v_{i1} \\ v_{i2} \end{bmatrix}$ respectively for $l = 1, 2$.

The complete FCGS algorithm is presented in Algorithm 3, where the shape contours are generated using B-spline based on a set of significant points (Step 1). Since the shape contours are generated using B-spline, there is a low probability that the initial shapes may overlap with each other that may degrade the segmentation performance i.e. increase the misclassification error of the new FCGS algorithm due to having overlapping pixels. To manage this eventuality, if an overlap of the shape contours is detected, then the initial segmented results of the GK algorithm are considered as the final segmentation results (Step 2). Otherwise, the average cluster radius is considered (Step 4) and the shape is scaled (Step 5), with the intersection points determined using Algorithm 2 in Step 6, and the membership values μ_{ij} and cluster centre v_i iteratively updated in Steps 7 and 8 respectively.

Algorithm 3: *Fuzzy clustering for image segmentation using generic shape information* (FCGS).
Input: The number of clusters c, the initial values of cluster centre v_i, initial μ_{ij}, the significant points P, knot vector, number of points m representing the shape contour, max *iteration*

129

and the threshold ξ.

Output: Final membership values μ_{ij}.

1. Generate m points on the contour of the shape using B-spline for significant points P.
2. IF NOT overlaps between contours THEN
   3. FOR $l = 1, 2, 3, \cdots, \max iteration$
      4. Update r_i by (16)
      5. Scale contour by *Ratio* using (17).
      6. Find intersection point by **Algorithm 2**
      7. Update μ_{ij} using (14) and (15).

      8. Update v_i using (18).

      9. IF $\left\| \mu_{ij}{}^l - \mu_{ij}{}^{l-1} \right\| \leq \xi$ FOR $\forall i, j$ THEN
         STOP
         ENDIF
   ENDFOR
ENDIF

5.0   EXPERIMENTAL RESULTS

In evaluating the segmentation performance of the new FCGS algorithm, empirical results were both qualitatively and numerically compared with five other shape-based clustering algorithms, namely FKR, FKE, GK, FCS and FCES. While the FKE and FCES algorithms are extensions of FKR and FCS respectively, each algorithm was analyzed in order to rigorously assess the performance improvement achieved using FCGS. Different natural and synthetic gray-scale images[1] as well as medical images and frames from the *football* video sequence were randomly selected for analysis. Fuzzy clustering algorithm arbitrarily divides the image into a given number of clusters when only pixel location is used as feature in segmentation process as image is rectangular in size that is detailed in [21] and the main goal of the proposed FCGS algorithm is to segment only foreground objects. For these reasons, the background of each image is removed so that only foreground objects are separated. This necessitated the manual setting of all background pixels to zero, with zero-valued foreground object pixels being replaced by 1 to avoid the possibility of foreground pixels merging with the background, while not impacting upon visual perception.

To quantitatively appraise the performance of all the fuzzy clustering algorithms, the objective segmentation evaluation method, *discrepancy based on the number of misclassified pixels* [21] was used. The two errors, namely Type I, $errorI_i$ and Type II, $errorII_i$ are formally defined as follows:

$$errorI_i = \frac{\left(\sum\limits_{j=1}^{\Re} M_{ji} - M_{ii} \right)}{\sum\limits_{j=1}^{\Re} M_{ji}} \times 100 \tag{19}$$

$$errorII_i = \frac{\left(\sum\limits_{j=1}^{\Re} M_{ij} - M_{ii} \right)}{\left(\sum\limits_{i=1}^{\Re} \sum\limits_{j=1}^{\Re} M_{ij} - \sum\limits_{j=1}^{\Re} M_{ji} \right)} \times 100 \tag{20}$$

---

[1] Obtained from IMSI (Master Photo Collection, San Rafael, CA 94901-5506, USA.) and the Internet.

where the confusion matrix M_{ij} is a $\Re$ by $\Re$ square matrix and denotes the number of jth region pixels classified as region i by the segmentation process. Type I, $errorI_i$ is the percentage error of all i^{th} region pixels that are misclassified in other regions, while Type II, $errorII_i$ is the percentage error of all region pixels that are misclassified into the ith region. Representative samples of the manually segmented reference regions together with the original images are shown in Fig. 3 (a)-(b), Fig. 4 (a)-(b), Fig. 5 (a)-(b) and Fig. 6 (a)-(b). To afford an improved visual interpretation of the segmentation performance, both the reference and segmented regions are displayed in different colours rather than their original gray-scale intensities.

Before providing the detailed results analysis, some comments upon the initialisation strategies used for each clustering method in the experiments are detailed in this section. GK, FCS and FCES algorithms were initialised using random membership values μ. FKR used a *fuzzy k-means* (FKM) [22] algorithm, while for the FKE algorithm the same initialisation approach in [11] was used, namely 10 iterations of FKM followed by 10 iterations of FKR. For the proposed FCGS algorithm, while any clustering algorithm can be used for initialization with B-spline, the GK algorithm (Algorithm 1) was selected because it consistently provided the highest percentage of superior results for all the test images used in the experiments in comparison to FKR, FCS, FKE and FCES. Finally, for the reasons discussed in Section 2, during the initialisation of each experiment, the number of clusters was manually defined for each algorithm. Please note that if GK totally fails to segment the objects, the FCGS algorithm will certainly improve the segmentation performance of GK but it will also unable to segment the objects well.

The first test image (Fig. 3 (a)) was selected because it comprised both a circular (*sun*) and arbitrary shaped (*tree branch*) object, represented by regions R_1 and R_2 respectively. The segmentation results for the various algorithms are shown in Fig. 3(c)-(h).

(a) Original

(b) Ref. Image

(c) FKR

(d) FKE

(e) GK
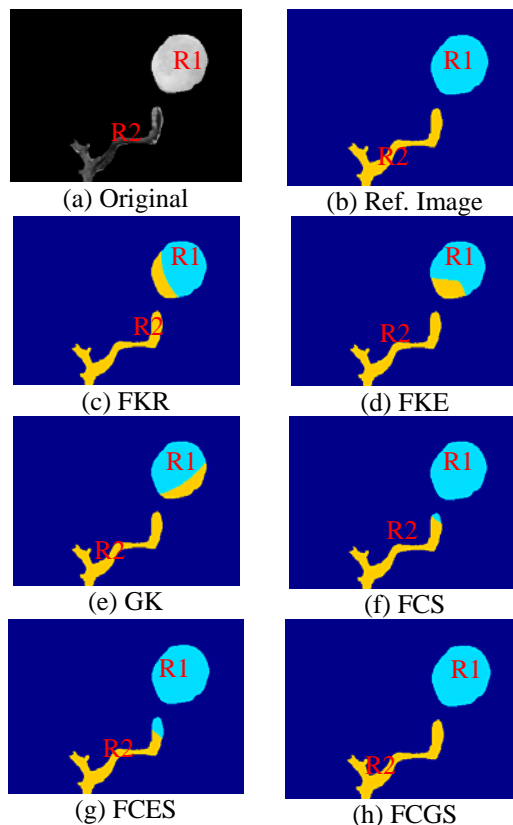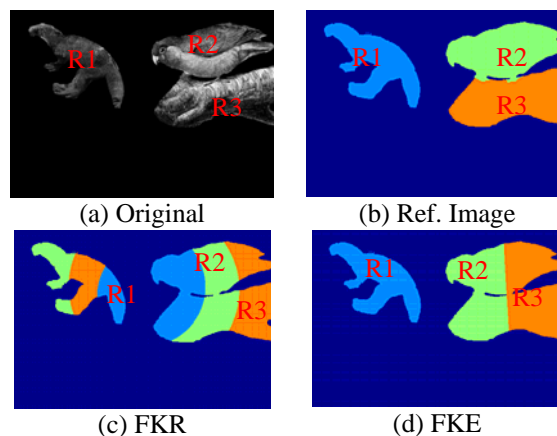
(f) FCS

(g) FCES

(h) FCGS

Fig. 3: (a) Original *sun* image, (b) Manually segmented reference of (a). Fig. (c) – (h) the segmented results using FKR, FKE, GK, FCS, FCES, and FCGS respectively.

131

If the results in Fig. 3(c)-(e) are subjectively compared with the manually segmented reference regions in Fig. 3(b), it is readily apparent a large number of pixels R_1 have been misclassified in R_2 by the FKR, FKE and GK algorithms respectively, while there are also misclassified pixels from R_2 in R_1 for the FCS and FCES algorithms in Fig. 3(f) and 3(g) respectively. Since R_1 is circular, both the FCS and FCES algorithms are able to completely segment region R_1, while being unable to segment R_2 (*tree branch*) as this is neither a circular nor elliptical object compared with R_1 which possesses a distinctly circular form. The reason behind this is that the FCS and FCES algorithms are designed for segmenting specific shape-based objects such as circular and elliptical objects respectively. In contrast, the new FCGS algorithm provided ideal segmentation with no misclassified pixels (Fig. 3(h)), validating the strategy of embedding shape information into the GK framework. The numerical results for the *Type I* and *Type II* errors for region R_1 for all five segmentation algorithms in Table 1 endorse this observation. Note that since there are only two segmented regions in this particular image, the results for R_2 will be the inverse of those presented for R_1 in Table 1.

Table 1: Percentage errors for R_1 in the Fig. 3 (a) segmentation

Algorithm	Errors		
	Type I	*Type II*	Mean
FKR	21.2	0	10.6
FKE	20.4	0	10.2
GK	22.5	0	11.2
FCS	0	3.8	1.9
FCES	0	12	6
FCGS	**0**	**0**	**0**

The second test image in Fig. 4(a) contains three regions which are all arbitrarily shaped, namely the *reptile* (R_1), *bird* (R_2) and *tree branch* (R_3), where R_2 and R_3 are connected, with the corresponding results for the FKR, FKE, GK, FCS, FCES and FCGS algorithms being displayed in Fig. 4(c)-(h) respectively. In FKR, as all three objects are arbitrarily shaped, not surprisingly there are a large number of misclassified pixels occurring between R_1, R_2 and R_3 as shown in Fig. 4(c). The results for FKE, FCS and FCES in Fig. 4(d), (f) and (g) respectively, are again characterised by large numbers of misclassified pixels, though in these cases only between R_2 and R_3, i.e. the two connected regions, while the results for GK in Fig. 4(e) show both R_1 and R_2 contain misclassified pixels. Conversely, the FCGS algorithm (Fig. 4(h)) successfully separated R_1 and generated significantly fewer misclassified pixels compared with other algorithms, especially the GK algorithm, so verifying the benefit of embedding generic shape information within the segmentation framework.

(a) Original

(b) Ref. Image

(c) FKR

(d) FKE

(e) GK
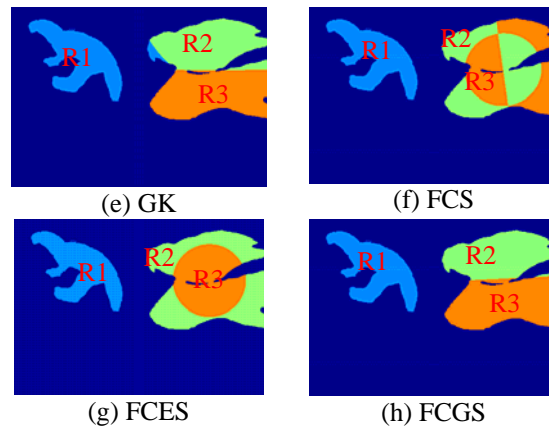
(f) FCS

(g) FCES

(h) FCGS

Fig. 4: (a) Original *bird* image, (b) Manually segmented reference of (a). Fig. (c) – (h) the segmented results of (a) using FKR, FKE, GK, FCS, FCES, and FCGS respectively.

The numerical results in Table 2 support the perceptual judgements of the performance of the various segmentation algorithms, with an overall lowest mean error of 1.7% being achieved by FCGS, contrasting with the 49.6% for FKR and 3.1% for GK.

Table 2: Percentage errors for Fig. 4 (a) segmentation

Algorithm	Errors						Mean
	Type I			*Type II*			
	R1	R2	R3	R1	R2	R3	
FKR	73.58	62.7	62.9	34.2	34.4	29.8	49.6
FKE	0	47.6	46.5	0	27.1	26.6	24.6
GK	0	7.1	4.7	1.8	2.8	1.9	3.1
FCS	0	52.2	47.7	0	27.7	29.2	26.1
FCES	0	47.8	57.2	0	33.3	26.8	27.5
FCGS	**0**	**5.2**	**1.4**	**0**	**0.8**	2.9	**1.7**

The third test image used to analyze the performance of the FCGS algorithm was an X-ray of the anterior-posterior view of the knee-joint [23] in Fig. 5(a) comprising two arbitrary shape objects, namely the femur (R_1) and tibia (R_2), with the corresponding reference regions given in Fig. 5(b). Fig. 5(c)-(h) show the segmented results for FKR, FKE, GK, FCS, FCES and FCGS respectively. Comparing the results with the reference image in Fig. 5 (b), it is clear many pixels of R_2 have been misclassified into R_1 (Fig. 5(c)-(d), (f)-(g)) because the shapes of both R_1 and R_2 are neither circular nor elliptical, so FKR, FKE, FCS and FCES all arbitrarily segmented the objects. GK also generated a considerable number of misclassified pixels in both R_1 and R_2 while in comparison, the segmentation for both regions in FCGS was superior (Fig. 5(h)) despite the generation of an erroneous shape from the initialization using GK (Fig. 5(e)). This inaccurate initial shape is the reason for the *Type I* error results in Table 3 for R_1 being 23.2% for GK compared to the perfect results for both FCS and FCES. The FCGS algorithm subsequently reduced this *Type I* error to just 0.26% and also produced considerable lower *Type II* errors than either FCS or FCES, thereby providing overall superior segmented results with a minimum average error of 2.8% compared to 43.3% and 4.2% for FKR and FCES respectively.
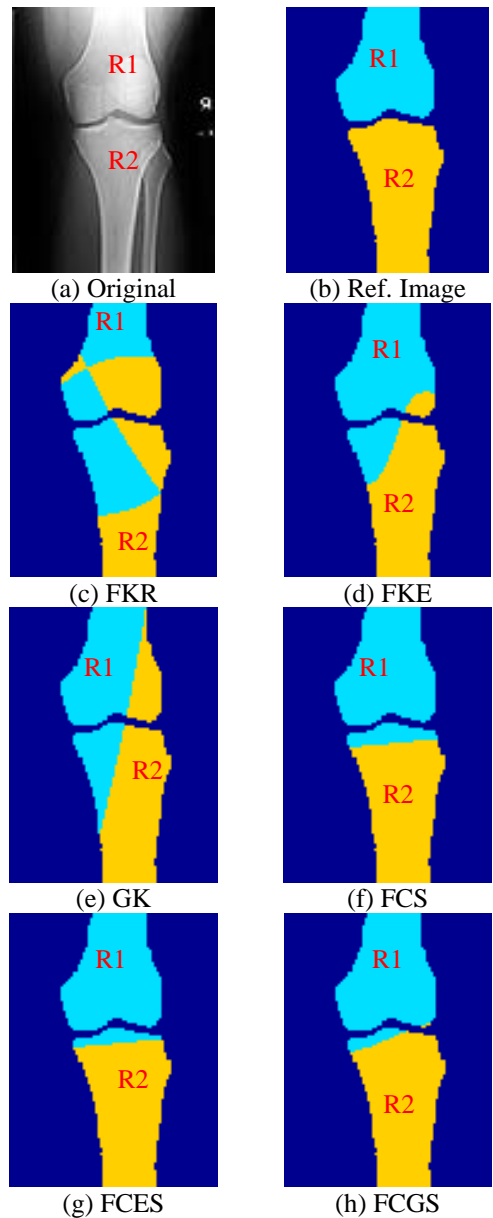
Fig. 5: (a) Original *X-ray* image, (b) Manually segmented reference of (a). Fig. (c) – (h) the segmented results of (a) using FKR, FKE, GK, FCS, FCES, and FCGS respectively.

Table 3: Percentage errors for R_1 in Fig. 5 (a) segmentation

Algorithm	Error		
	Type I	*Type II*	Mean
FKR	42	44.6	43.3
FKE	6.4	19.6	13
GK	23.2	24.4	23.8
FCS	0	12.7	6.4
FCES	0	8.4	4.2
FCGS	**0.26**	**5.5**	**2.8**

(a) Original   (b) Ref. Image

(c) FKR   (d) FKE

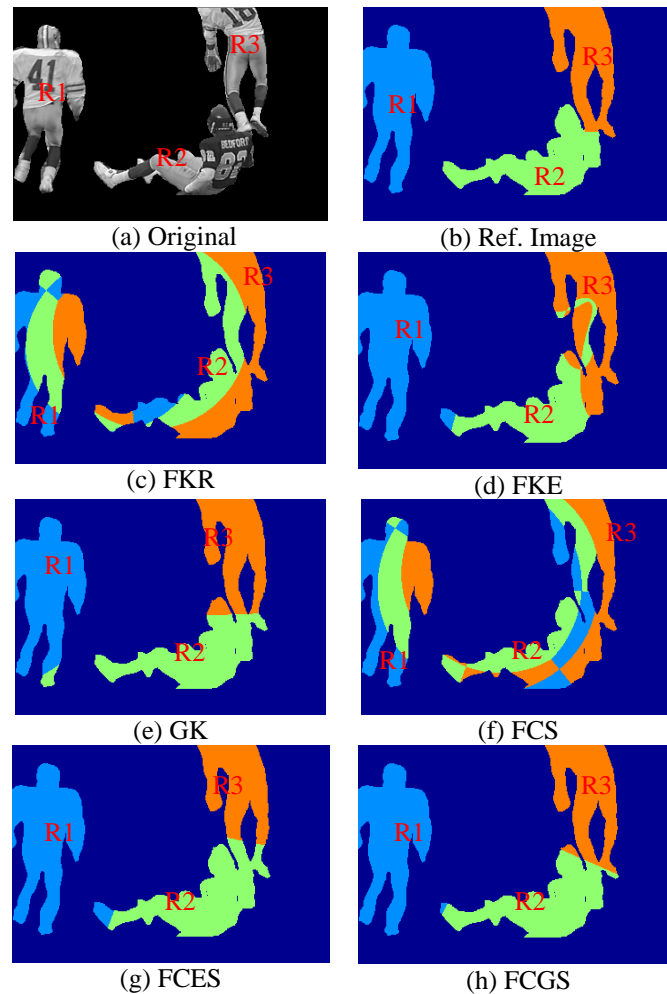(e) GK   (f) FCS

(g) FCES   (h) FCGS

Fig. 6: (a) Original *football* image, (b) Manually segmented reference of (a). Fig. (c) – (h) the segmented results of (a) using FKR, FKE, GK, FCS, FCES, and FCGS respectively.

The final image analysed was a frame (Fig. 6(a)) taken from the popular *football* video test sequence used by researchers in the field of motion estimation and video coding. This frame contains three arbitrary-shaped objects R_1 (*left player*), R_2 (*player on the ground)* and R_3 (*right player*), each having a different shape and orientation, with some occlusion between R_2 and R_3. The results for the FKR, FKE, GK, FCS, FCES and FCGS algorithms are respectively shown in Fig. 6(c)-(h). If the results of FKR and FCS shown in Fig. 6 (c) and (f) are visually compared with the reference image in Fig. 6(b), it can be seen that there is considerable pixel misclassification between all three regions, with the main reasons being over scaling during iterations, improper initialization and the absence of any circularly shaped object in the image. A similar pixel misclassification trend is observed for both FKE and FCES in Fig. 6(d) and Fig. 6(g) respectively. The GK algorithm misclassified a number of pixels from R_2 into R_3 as well as from R_3 into R_2 due to the occlusion and also from R_1 into R_2 as shown in Fig. 6(e). FCGS in contrast generated significantly lower numbers of misclassified pixels (Fig. 6 (h)), so vindicating the strategy of incorporating shape feature information into the segmentation process, with Table 4 showing the average error for the proposed FCGS algorithm (1.6%) is consistently the lowest. This particular example also highlights the potential of the FCGS algorithm to be extended into multimedia applications such as video object segmentation, with for instance, it being applied to segment video object planes (VOP) for object-based MPEG-4 coding [24].

Table 4: Percentage errors for Fig. 6 (a) segmentation.

135

Algorithm	Errors						Mean
	Type I			*Type II*			
	R1	R2	R3	R1	R2	R3	
FKR	71.2	58	44.4	6.4	44	35.8	43.3
FKE	0	19.4	9	1.9	3.8	7.2	6.9
GK	3	5.8	6.6	**0**	4.5	2.6	3.8
FCS	63.7	53.6	53.3	22.6	34.6	28.4	42.7
FCES	0	5.2	15.1	2.7	6.4	**0**	4.9
FCGS	0	**2.6**	**3.8**	0.4	**1.6**	0.8	**1.6**

To assess the generalization capability of the proposed FCGS algorithm, experiments were conducted upon 185 randomly selected images containing multiple objects having circular, elliptic and arbitrary shapes with different orientations. While the shape descriptor was generated from the GK initialization (Section 2), the FCGS algorithm produced best segmentation performance for 81 images while FKR, FKE, GK, FCS and FCES respectively yielded better segmentation results for only 3, 17, 12, 11, and 18 (see Fig. 7).
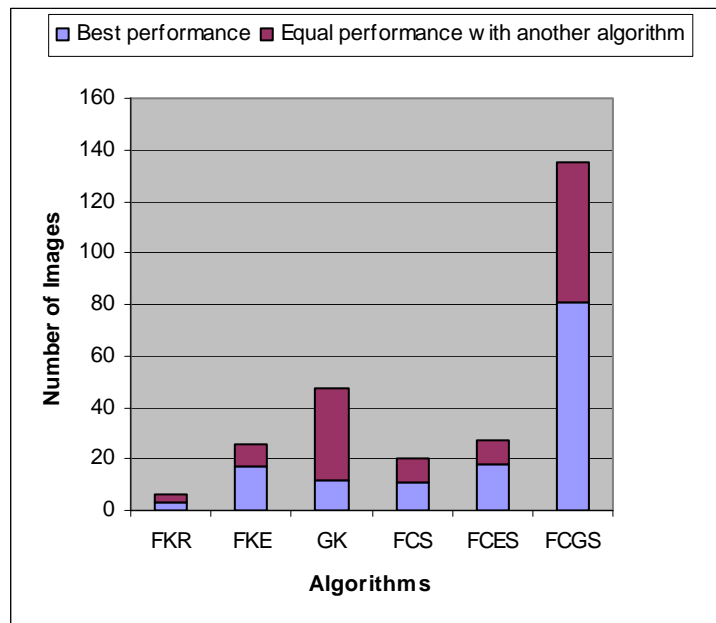

Fig. 7: The best segmentation results for six different fuzzy
clustering algorithms when the arbitrary shape contour is determined

In addition, FCGS provided similar results for 35 and 19 images respectively compared to GK and the other four algorithms considered, while FKR, FKE, FCS and FCES exhibited similar results for only 3, 9, 9, and 9 images respectively. As these four algorithms were designed specifically for regularly shaped geometric objects (ring and elliptic) they as expected, generated either comparable or better results compared with FCGS for such objects, so accounting for improved segmentation performance in 3, 17, 11, and 18 images respectively. As highlighted in Section 4.1, while FCGS adopts the same principle used in the FKR, FCS, FKE and FCES algorithms [8]-[11], it considers shape information based upon a contour, and not the region of an object. As a consequence, for 12 images, the FCGS algorithm is unable to improve further upon the initialization produced by GK. To recap, as previous shape-based clustering algorithms have focused upon only objects with specific geometric clusters, the proposed FCGS algorithm's performance in handling arbitrary shaped objects and providing consistently superior segmentation results for most images, justifies the strategy of seamlessly integrating generic shape information into the clustering framework.

6.0 CONCLUSIONS

This paper has presented a new shape-based image segmentation algorithm called *fuzzy clustering for image segmentation using generic shape information* (FCGS) which seamlessly integrates generic shape information into the *Gustafson-Kessel* (GK) clustering framework. Both qualitative and quantitative analysis were conducted to

compare the performance against existing shape-based algorithms including *fuzzy k-ring* (FKR), *fuzzy circular shell* (FCS), *fuzzy c-ellipsoidal shells* (FCES) and *fuzzy k-ellipse* (FKE), as well as the original GK algorithm, for many images consisting of multiple objects having different shapes and orientations. With the initialization automatically performed using a GK algorithm and B-spline combination, FCGS provided superior segmentation performance for many differently shaped objects, so broadening the application base of this innovative clustering framework into domain areas such as medical imaging, security systems, and any image processing application where arbitrary shaped object segmentation is required.

REFERENCES

[1] Gonzalez R. C. and Woods R. E., *Digital Image Processing*, Prentice Hall Inc., New Jersey, 2002.
[2] Gath I. and Geva A. B., Unsupervised optimal fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(7), 1989, 773-781.
[3] Pal N. R. and Pal S. K., A review on image segmentation techniques, *Pattern Recognition*, 26(9), 1993, 1277-1294.
[4] Bezdek J. C., *Pattern Recognition with Fuzzy Objective Function Algorithm*, Plenum Press, New York, 1981.
[5] Krishnapuram R. and Keller J. M., A possibilistic approach to clustering, *International Journal of Fuzzy Systems*, 2(2), 1993, 98-110.
[6] Fan J. L., Zhen W. Z., and Xie W. X., Suppressed fuzzy c-means clustering algorithm, *Pattern Recognition Letters*, 24, 2003, 1607-1612.
[7] Gustafson D. E. and Kessel W. C., Fuzzy clustering with a fuzzy covariance matrix, *In Proc. IEEE Conference on Decision Control*, 1979, 761-766.
[8] Man Y. and Gath I., Detection and separation of ring-shaped clusters using fuzzy clustering, *IEEE Transactions on Pattern Analysis and machine Intelligence*, 16 (8), 1994, 855-861.
[9] Dave R. N., Fuzzy shell-clustering and applications to circle detection in digital images, *International Journal on General Systems*, 16 (4), 1990, 343-355.
[10] Dave R. N., Generalized fuzzy c-shell clustering and detection of circular and elliptical boundaries, *International Journal on Pattern Recognition*, 25 (7), 1992, 713-721.
[11] Gath I. and Hoory D., Fuzzy clustering of elliptic ring-shaped clusters, *IEEE Transactions on Pattern Recognition Letters*, 16, 1995, 727-741.
[12] Babuska R., Van der Veen P. J. and Kaymak U., Improved covariance estimation for Gustafson-Kessel clustering, *IEEE International Conference on Fuzzy Systems*, 2, 2002, 1081-1085.
[13] Jain A. K. and Dubes R. C., *Algorithms for clustering data*, Prentice Hall Inc., New Jersey, 1988.
[14] Chong A., Gedeon T. D., and Koczy, L. T, A hybrid approaches for solving the cluster validity problem, *International Con. On Digital Signal Processing*, 2, 2002, 1207-1210.
[15] Yen J. and Langari R., *Fuzzy logic*, Prentice Hall Inc., New Jersey, 1999.
[16] Dave R. N., Boundary detection through fuzzy clustering, *IEEE International Conference on Fuzzy Systems*, 1992, 127-134.
[17] Francis S. H., *Computer Graphics*, Prentice Hall Inc., New Jersey, 1994.
[18] Hearn D. and Baker M. P., *Computer Graphics*, Prentice Hall Inc., New Jersey, 1994.
[19] Zhang J., C-Bezier curves and surfaces, *IDEAL International Con. on Graphical Models and Image Processing*, 1999, 2-15.
[20] Tony D. D., Geometric continuity, shape parameters, and geometric constructions for catmull-rom splines, *ACM Transaction on Graphics*, 7(1), 1988, 1-41.
[21] Ali M. A., Karmakar G. C., and Dooley L. S., Fuzzy image segmentation using shape information, *IEEE International Con. On Multimedia and Expo* (ICME'05), 2005.
[22] Watanabe N. and Imaizumi T., Fuzzy k-means clustering with crisp regions, *IEEE International Con. On Fuzzy Systems*, 2002, 1528-1531.
[23] Medical X-ray Images, http://www.med.yale.edu/, last visited: November 12, 2005.
[24] Yun Q. S. and Sun H., *Image and Video Compression for Multimedia Engineering*, New Jersey, 2008.

BIOGRAPHY

Dr. Ameer Ali was born in Dhaka, Bangladesh in 1977. He completed his B. Sc. in Computer Science and Engineering in 2001 from Bangladesh University of Engineering and Technology, Dhaka and PhD in IT in 2006 from Monash University, Australia. Currently, he is working as an Assistant Professor in the Department of Electronics and Communication Engineering, East West University, Dhaka, Bangladesh. He has more than 20 published articles in both reputed international journals and conferences. His research interests are image processing, segmentation, fuzzy clustering, telemedicine, vendor selection using fuzzy techniques, and networking.