Detection and separation of generic-shaped objects by fuzzy clustering

Journal Item

DETECTION AND SEPARATION OF GENERIC-SHAPED OBJECTS BY FUZZY CLUSTERING

M. Ameer Ali

Department of Computer Science and Engineering

East West University, Bangladesh

Email: ameer7302002@yahoo.com

Gour C Karmakar

Gippsland School of Information Technology

Monash University, Australia

Email: Gour.Karmakar @infotech.monash.edu.au

Laurence S Dooley

Department of Communication and Systems

The Open University, Milton Keynes, United Kingdom

Email: L.S.Dooley@open.ac.uk

Received ( )
Revised ( )
Accepted ( )

ABSTRACT

Purpose - Existing shape-based fuzzy clustering algorithms are all designed to explicitly segment regular geometrically-shaped objects in an image, with the consequence that this restricts their capability to separate arbitrarily-shaped objects.
Design/Methodology/Approach – With the aim of separating arbitrary shaped objects in an image, this paper presents a new *detection and separation of generic shaped objects* (FKG) algorithm that analytically integrates arbitrary shape information into a fuzzy clustering framework, by introducing a shape constraint that preserves the original object shape during iterative scaling.
Findings- Both qualitative and numerical empirical results analysis corroborate the improved object segmentation performance achieved by the FKG strategy upon different image types and disparately shaped objects.
Originality/value- The proposed FKG algorithm can be highly used in the applications where object segmentation is necessary. Like this algorithm can be applied in MPEG-4 for real object segmentation that is already applied in synthetic object segmentation.
Key word: Fuzzy clustering, Image segmentation, B-splines, Generic shape information.

1. INTRODUCTION

Image segmentation involves the separation of mutually exclusive regions/objects of interest (see (Gonzalez and Woods, 2002)), and is integral to the image processing, coding and interpretation domains, with examples of some of the eclectic range of applications including; image analysis, robot vision, automatic car assembly, security surveillance systems, object recognition and medical imaging [1]. As there are potentially a very large number of perceptual objects in an image, with subtle variations between them, this makes generalised object-based segmentation an especially challenging task.
Fuzzy clustering techniques (see (Bezdek, 1981); (Krishnapuram and Keller, 1993); (Fan *et al.*, 2003)) have successfully been applied to image segmentation, though their performance has proven to be highly dependent on the features used and types of objects in the image. For one particular image, the *fuzzy c-means* (FCM) clustering algorithm (see (Bezdek, 1981)) may well provide the best segmentation performance when *pixel location* is used as the feature, while for a different image, either *pixel intensity* or a combination of pixel intensity and location could be more propitious choices. This raises the rhetorical question as to which features provide the best object segmentation results for a particular clustering algorithm, with the outcome ultimately restricting both the algorithm's generalisation capability and application domain. These limitations have motivated investigation of alternative strategies to characterise object-specific information in image segmentation frameworks, with *shape* being one of the most important perceptual attributes in both detecting and recognising objects.
Existing shape-based clustering techniques [2] all have their foundations embedded in fuzzy theory, with examples including; the *Gustafson-Kessel* (GK) algorithm [9], *fuzzy k-ring* (FKR) [10], *fuzzy circular shell* (FCS) [11], *fuzzy c-ellipsoidal shells* (FCES) [12] and *fuzzy elliptic-ring* (FKE) [13]. FKR and FCS were both designed to detect and separate, ring and compact spherical shapes, and their assorted combinations. Subsequently FKE and FCES were developed as the respective generalised versions of these algorithms, facilitating segmentation of ring and elliptic shaped objects and their combination. While the underlying mathematical models of both FKR and FKE have proven successful for separating particular object shapes, they are much less effective in segmenting ring and elliptical-shaped regions. In contrast, FCS and FCES are better suited for region-based separation as they include shape constraints within their objective function and use a normalized data distance to update both the membership values and other key parameters in the iterative minimisation process. Their main drawback however, is that most natural objects are neither ring nor elliptical in shape, which severely confines their capacity to segment arbitrary-shaped objects.
The GK algorithm [9] conversely does not explicitly use shape information and so is ineffectual for generic-shaped object segmentation, though it does employ a fuzzy covariance matrix in its model which automatically adapts the local data distance to the cluster shape, a property that has been exploited for shape initialisation purposes [9].
This paper generalises shape-based fuzzy clustering theory by seamlessly integrating arbitrary shape-based information into an FCM-based clustering framework. A new *detection and separation of generic shaped object* (FKG) algorithm is introduced, with shape descriptor information embedded within an analytic optimization of the FCM-based objective function. FKG is uniquely characterised by the following four distinct features: *i)* A special shape-based constraint which ensures the optimisation of the algorithm; *ii)* Provision for either manually defining or automatically generating the initial object shape contour by employing either parametric curve models or an alternative clustering technique; *iii)* Preservation of the initialised object shape during iterative scaling; and *iv)* Accurate intersection point calculation for the requisite data distances from the respective object contour. Collectively these features enable FKG to improve the overall segmentation

performance, with both qualitative and quantitative analysis conclusively confirming its superiority over all other existing shape-based clustering algorithms for many different image types and disparately shaped objects.
The remainder of this paper is organized as follows: Section 2 reviews the main properties of current shape-based fuzzy clustering algorithms, while the new FKG algorithm, including its full mathematical foundations and a complete computational complexity analysis are presented in Sections 3, 4 and 5. The object-based segmentation performance of FKG is analysed in Section 6, with some concluding comments being provided in Section 7.

2. SHAPE-BASED CLUSTERING ALGORITHMS

This section examines the various features and limitations of existing shape-based clustering algorithms. FCM [6] has become a widely adopted clustering algorithm, being either directly or indirectly applied in a broad range of applications including crucially, as the initialisation strategy for other fuzzy clustering techniques. Its performance however, is highly sensitive to the features used and the type of objects in an image, so to relax this dependency, a number of dedicated shape-based clustering methods have been developed, including the FKR, FCS, FCES and FKE algorithms (see (Man and Gath, 1994); (Dave, 1990); (Dave, 1992); (Gath and Hoory, 1995); (Babuska *et al.*, 2002)), which all have as their fundamental premise, an iterative minimisation of an FCM-type objective function.
FCS [11] detects and separates circular structures in an image by considering relevant geometric representations and using the distance from the circular shell to the corresponding data point, in its objective function. Two special constraints are applied to integrate circular shape information into the clustering framework involving the ratio of the respective data distances from the cluster shell and cluster centre. Both the cluster centre and circular shell radius are updated using Newton's iteration and while FCS performs well for circular-type shapes, it is ineffectual in segmenting arbitrary shaped objects due to its underlying non-linear mathematical model.
A similar judgement stands for FKR [10] which focuses upon ring-shaped objects. The respective membership functions, cluster centres and radii are iteratively updated based on the minimisation of an objective function defined for circular ring shapes, to separate both ring and spherical compact clusters, allied with combinations of ring-shaped clusters. Once again however, the algorithm fails to perform satisfactorily for arbitrary-shaped objects and ring-shaped regions, for the same reason delineated in Section 1.
The FCES [12] and FKE [13] algorithms generalize FCS and FKR respectively by supporting the segmentation of elliptically-shaped objects and their related geometric combinations. FCES applies the same constraints as FCS and employs a covariance matrix to calculate the distance between each datum and cluster centre, with this distance being adapted to the shape by taking cognisance of the orientation and scaling of the shell. Newton's iteration again updates the cluster centres and radii, though both FCES and FKE consistently fail to effectively segment either region-based or arbitrarily-shaped objects.
The GK algorithm in contrast, does not explicitly incorporate any particular shape information, be it circular or ellipsoidal shell. Instead it uses a fuzzy covariance matrix that automatically adapts the local data distance metric to the cluster shape [9], an attribute that gives GK superior object segmentation performance compared with existing shape-based clustering algorithms including FKR, FCES and FKE. For this reason GK was adopted in the new FKG algorithm (Section 3) to provide better object shape initialization. The formal mathematical basis of the GK algorithm, which uses Lagrangian optimization techniques to iteratively minimize an FCM-based objective function is presented in Appendix A.
In summary, existing shape-based clustering algorithms are generally not very successful in segmenting arbitrary-shaped objects because they are specifically designed for regular geometric structures. This provided the impetus to investigate seamlessly incorporating generic shape object information into fuzzy clustering paradigms.

3. MATHEMATICAL FOUNDATIONS FOR INCORPORATING GENERIC SHAPE INFORMATION INTO A FUZZY CLUSTERING FRAMEWORK

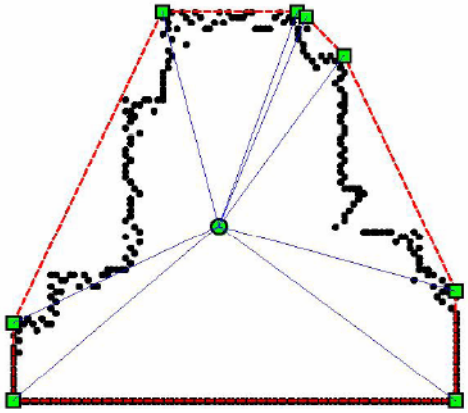
This section introduces the FKG algorithm for detecting and separating arbitrary-shaped objects in an image. In the literature, the most effective shape approximating methods, like moment invariants, Fourier descriptors and boundary signatures, represent the shape of an object by a contour (Dengsheng and Guojun, 2004). This has been the underlying rationale for all available shape-based clustering algorithms to seamlessly incorporate regular geometric contours into their segmentation frameworks.
To embed generic shape information into the segmentation model, the *contour points* (shape descriptor) for each object must either be provided in terms of significant points or automatically determined from the initialization process. These are then used to calculate the data distance from the boundary, in an analogous manner to existing shape-based clustering algorithms including FKE and FCES. These two core algorithmic components will now be examined.

3.1 Embedding Shape Information

As previously alluded, to initialise FKG, the shape contour points have to be either manually provided as a set of significant points or automatically derived from the initialisation process, with in the former case, the B-spline approximation (e.g. (Francis, 1994); (Hearn and Baker, 1994); (Zhang, 1999); (Tony, 1988)) applied to generate the respective contour points of each object. For example, the shape descriptor for the *miss america* object displayed in Figure 1(a) is given in terms of significant points (denoted by ■) [25]. The significant points are generated by (e.g. (Francis, 1994); (Hearn and Baker, 1994); (Zhang, 1999); (Tony, 1988)) and are translation, rotation and scale normalised using the window-to-viewport normalisation ((Costa *et al.*, 2001); (Foley *et al.*, 1999)) in order to find the best matching region of the initialisation process. The cluster with the largest number of pixels lying inside the polygon of a set of significant points is then designated the best matching region [25], with the contour points of this region generated by B-spline using its corresponding set of significant points. The contour points for the set of significant points shown in the Figure 1(a) example, are represented by "*" in Figure 1(b).

When initialisation is automatically performed, there is no need for translation and rotation normalisation because the same object information is used throughout the process. As highlighted in Section 2, the GK algorithm uses a fuzzy covariance matrix that helps to automatically adapt the shape of the cluster, so enabling it to approximate the original shape of objects in an image. This is the main advantage in adopting the GK algorithm for automatic object shape initialisation purposes. Note in this particular initialisation scenario, there is no need to obtain the significant points and their respective contour points, because the image is segmented by GK, and the corresponding boundary points of each segmented region are scanned to obtain the set of *contour points.* These contour points then represent the relevant object shape and are considered as the initial shape of each object.

(a) Miss America



(b) Shape description with significant points and generated points of (a)

Figure 1: (a) Miss America, (b) B-spline generated shape with its significant points.

It needs to be emphasized however that in both cases, the scaling of particular segmented regions will automatically occur during the various iterations involved in the clustering process.

A core precept of shape-based fuzzy clustering algorithms [10], [13] is that the data distance used in the objective function must be calculated from the shape of corresponding region. To calculate this distance, the corresponding point on a shape for a datum must be found and this will actually be the *intersection point* between the object contour and a line passing through the cluster centre and datum. A suitable strategy for determining this *intersection point* is now presented.

3.2 Calculating the Intersection Points

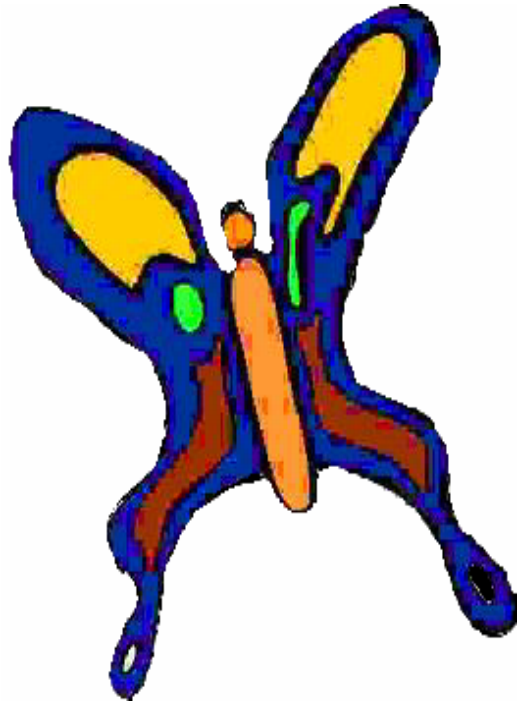
The most important consideration in any clustering–based image segmentation strategy is how best to compute the data distance d_{ij} implicit in the objective function. In FCM [6] for example, d_{ij} is calculated from the cluster centre based upon some predefined features (pixel intensity and/or location), while for FKR and FKE, it is calculated from the contour of a circle and ellipse respectively. As FKG processes arbitrary-shaped objects, d_{ij} has to be calculated from the respective contour points derived in the initialisation phase (Section 3.1). While the mathematical origins for both FKR and FKE are straightforward, no similar analytic framework exists for generic shapes, which means it is a more complex task to determine d_{ij} . To gain an intuitive insight into both this distance and the role of the shape descriptor, consider the *butterfly* shape in Figure 2(a) and its corresponding B-spline contour representation in Figure 2 (b), where v_i is assumed to be the centre of the object contour, S_j is a datum and $S_{ij}^{'}$ the corresponding *intersection point,* with d_{ij} the distance between them. The aim is to determine the point $S_{ij}^{'}$ on the initial contour intersected by the line (l_1) which joins S_j with the i^{th} cluster centre v_i as illustrated in Figure 2(b). $S_{ij}^{'}$ can be calculated using either polar or a combination of polar and Cartesian coordinates. The former does not always accurately approximate the actual intersection point because it involves the matching of two angles, namely θ_{ij} for the datum S_j and θ_{ik}^{c} for the k^{th} contour point (C_{ij}), with the minimum distance then being chosen, In many cases, it is not possible to accurately calculate the intersection point of a datum because the contour points (C_{ij}) are mainly discontinuous as evinced in Figure 1(b). The corollary is that it is not feasible to compute the exact distance d_{ij} for that datum, which causes improper scaling of the initial shape of a region during the clustering iterations, and can ultimately lead to poor segmentation performance.

The combined polar and Cartesian coordinate approach in contrast, involves the following two processing steps: *i)* Find two points on the contour of the curve that are closest to and lie on opposite sides of the line l_1 and; *ii)* As the shape descriptor linearly interpolates between two consecutive shape contour points (C_{ij}), the intersection of these points and l_1 gives $S_{ij}^{'}$. To locate the two consecutive shape contour points (C_{ij}), the difference in angle $\Delta\theta_{ijk}$ needs to be calculated, where $\Delta\theta_{ijk} = \theta_{ij} \sim \theta_{ik}^{c}$ for $1 \le k \le N_i^{''}$, $1 \le i \le c$ and $1 \le j \le n$, with $N_i^{''}$ being the number of contour points (C_{ij})in the i^{th} cluster. The two shape contour points with minimum $\Delta\theta_{ijk}$ lying either side of the line l_1 will be the respective consecutive shape contour points (C_{ij}). These two points together with l_1 are then be used to calculate the accurate intersection point of the j^{th} datum S_j. In this scenario, the calculation of the intersection point is determined from the intersection of two straight lines. If more than one point is found, the point having the smallest d_{ij} will be the exact intersection point, a decision that resolves any problem arising of handling concave and convex object shapes.

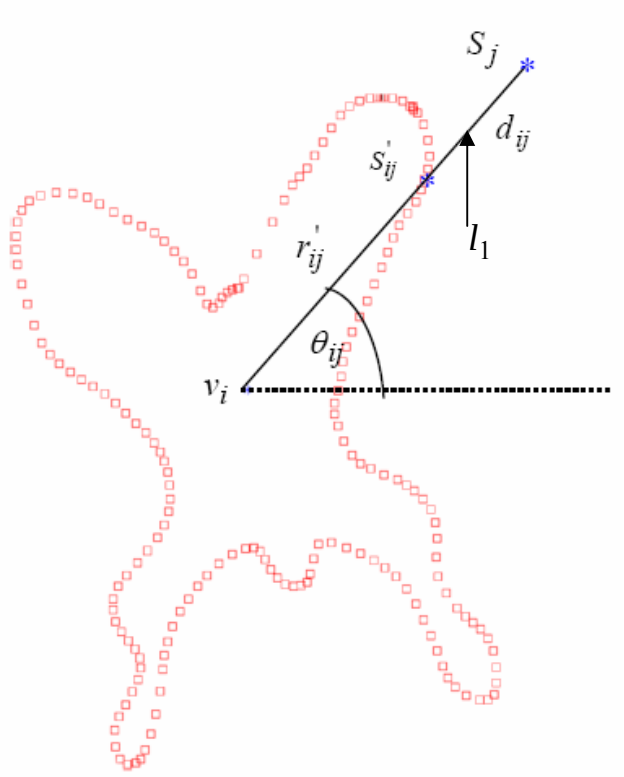
The various steps to locate the *intersection point* are detailed in **Algorithm 1,** where (r_{ij}, θ_{ij}) and $(r_{ik}^{c}, \theta_{ik}^{c})$ respectively represent the polar coordinate of datum S_{ij} and corresponding k^{th} contour point in the i^{th} cluster. Firstly convert all the contour points (C_{ij}) into its corresponding polar form (**Step 1 of Algorithm 1**) while it also requires same conversion for all data points to polar form like $(r_{ik}^{c}, \theta_{ik}^{c})$ in Step 2. After conversion to polar form of both contour points (C_{ij}) and data points S_j, a datum is taken and then calculate angle differences $(\Delta\theta_{ijk})$with all the contour points (C_{ij}) (Step 3) and

then find two contour points having minimum angle difference $(\Delta\theta_{ijk})$ and lying both side of the line (l_1) in Step 4 of Algorithm 1. Now, find out the intersection point $S_{ij}^{'}$ between the lines l_1 and the line passing through the contour points found in Step 4, which is the respective intersection point $S_{ij}^{'}$ (Step 5). Once all the intersection points are found, the algorithm terminates (Step 6).

Having determined the *intersection point* of every datum, d_{ij} can now be calculated and the FKG objective function formulated, as detailed in the following section.



(a)



(b)

Figure 2: (a) Original *butterfly* object, (b) Contour points (C_{ij}) of the *butterfly* object for the given set of significant points in Figure 1(a) generated by B-spline. Example of an intersection point S'_{ij} denoted by "*" between the contour and line l_1 joining datum S_j and the cluster centre v_i.

Algorithm 1: *Determining the intersection point between a datum and its corresponding cluster centre.*

Precondition: cluster contour points (C_{ij}), cluster centre v_i and datum S_j.

Post condition: Intersection point S'_{ij}.

1. Convert all contour points into polar form $(r^c_{ik}, \theta^c_{ik})$ with respect to the corresponding cluster centre v_i.
2. Convert all data points into polar form (r_{ij}, θ_{ij}) with respect to the corresponding cluster centre v_i.
3. Calculate $\Delta\theta_{ijk} = \theta_{ij} \sim \theta^c_{ik}$ for $1 \le k \le N''_i$, where N''_i is the number of contour points in the i^{th} cluster
4. Use the minimum $\Delta\theta_{ijk}$ value to locate the two points lying either side of l_1 for S_j.
5. Calculate intersection point S'_{ij} between a line joining the two selected contour points and l_1. If more than one point is found, the point with the shortest d_{ij} is the intersection point.
6. STOP

3.3 Integrating Shape Constraints

Existing shape-based algorithms like FKR and FCS do not require shape constraints in their paradigms, because they respectively consider regular circular and elliptic geometric shapes, which can be analytically expressed by standard equations that both preserve and scale the respective shapes during the clustering process. In contrast, no mathematical representation exists for generically shaped objects, so in order to incorporate arbitrary-shape information into the clustering framework, an appropriate shape constraint must be introduced. To address this requirement, the shape constraint integrated into the FKG algorithm is defined as:

$$r_{ij} \Big/ \sum_{t=1}^{n} r_{it} = k_{ij} \tag{1}$$

where r_{ij} is the distance between the *intersection point* $S_{ij}^{'}$ and the i^{th} cluster centre v_i and k_{ij} is a constant for the j^{th} datum of the i^{th} cluster. The notable feature of (1) is that it preserves the initial shape during iterative scaling, irrespective of the size of the initial contour, as will now be proven in Lemma 1.

Lemma 1: The arbitrary shape constraint $r_{ij} = k_{ij} * \sum r_{ij}$ always preserves the original shape during iterative scaling, regardless of size of the initial shape contour.

A formal proof of this Lemma is provided in Appendix B.

The objective function and its associated constraint for the FKG algorithm are now formally defined:

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

subject to $\sum_{i=1}^{c} \mu_{ij} = 1$ and $r_{ij} \Big/ \sum_{t=1}^{n} r_{it} = k_{ij}$ (3)

and

$$0 \le \mu_{ij} \le 1; \quad i \in \{1, \ldots, c\} \text{ and } j \in \{1, \ldots, n\} \tag{4}$$

where S is the dataset containing $\lfloor S_j \rfloor$, μ is the set of membership values μ_{ij}, V is a vector containing the cluster centre values v_i, q is the fuzzifier $1 < q \le \infty$, and distance $d_{ij} = d(S_j, v_i) - r_{ij} = D_{ij}^{'}$. $d(S_j, v_i)$ is the Euclidian distance between datum S_j and v_i, while *c* and *n* represent the number of clusters and data respectively.

For iterative optimization of the objective function, (2) and (3) are minimised by Lagrangian techniques. In calculating the membership function, if $d_{ij} = 0$ a crisp decision is mandated and the j^{th} data is classified into the i^{th} cluster, otherwise the membership value is updated based upon the distance d_{ij} as (see Appendix C for formal derivation):

IF $d_{ij} = 0$ THEN $\mu_{ij} = 1$ maintaining $\sum_{i=1}^{c} \mu_{ij} = 1$ (5)

ELSE $\mu_{ij} = 1 \Big/ \sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{q-1}}$ (6)

To scale the initial shape during the iterative minimization of the objective function (2), the contour radius r_{ij} as proven in Appendix C, is analytically derived as:

$$r_{ij} = D_{ij}^{'} - \frac{k_{ij} \sum_{t=1}^{n} D_{it}^{'} - D_{ij}^{'}}{k_{ij} \sum_{t=1}^{n} \frac{1 - k_{it}}{\mu_{it}^q} - \frac{1 - k_{ij}}{\mu_{ij}^q}} \left(\frac{1 - k_{ij}}{\mu_{ij}^q} \right) \tag{7}$$

When data are further away from the contour of a cluster, their corresponding membership values μ_{ij} are close to zero, which forces the second term in (7) towards zero and as a consequence, leads to faster convergence and an over-scaling of the initial shape. To mitigate the impact of over-scaling, r_{ij} is updated as follows (Ameer *et al.*, 2006):

$$r_{ij}(new) = \lambda\, r_{ij} + (1-\lambda) r_{ij}^{0} \tag{8}$$

where r_{ij}^{0} and r_{ij} are the initial and current values respectively, while λ is an empirically selected constant derived from the data which is a trade-off between the current and initial cluster sizes.

The i^{th} cluster centre v_i is given by (see Appendix C):

$$v_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{q} \left\{ S_j - D_{ij}^{'} . \frac{S_{ij}^{'} - v_i}{r_{ij}} + S_{ij}^{'} - r_{ij} \frac{S_j - v_i}{D_{ij}^{'}} \right\}}{2 \sum_{j=1}^{n} \mu_{ij}^{q}} \tag{9}$$

where for any image, the 2-D data and cluster centre are respectively given by $S_j = \begin{bmatrix} S_{j1} \\ S_{j2} \end{bmatrix}$ and $v_i = \begin{bmatrix} v_{i1} \\ v_{i2} \end{bmatrix}$.

The complete FKG algorithm is formalised in **Algorithm 2**, with the object shape initialised in STEP 1 by the GK algorithm, as highlighted in Section 3.1. The intersection points for each datum are calculated in STEP 2, with r_{ij} and k_{ij} being initialised in STEP 3. The membership values μ_{ij}, contour radii r_{ij} and cluster centres v_i are then iteratively updated in STEPS 4 and 5 until either the maximum number of iterations (*maxits*) is fulfilled or a specified threshold (ξ) is exceeded.

A major consideration for any fuzzy clustering algorithm, especially one designed for object-based segmentation applications, is how to determine the optimal number of clusters (objects), either directly from the image data or using *a priori* knowledge. This issue is considered in the next section.

Algorithm 2: *Detection and separation of generic shaped object using fuzzy clustering* (FKG).

Precondition: Dataset *X*, number of clusters *c*, *maxits* and threshold ξ.

Post condition: Final segmented regions *R*

1. Initialise the object shape (Section 3.1).
2. Find *intersection point* using **Algorithm 1**.
3. Calculate the initial r_{ij} and k_{ij} values from (3).
4. FOR *l = 1, 2, 3,…maxits*
   5. Update μ_{ij}, r_{ij} and v_i using (5) to (9)
   6. IF $\left\| \mu_{ij}^{\,l} - \mu_{ij}^{\,l-1} \right\| < \xi$ for $\nabla i, j$ THEN STOP

4. DETERMINING THE NUMBER OF CLUSTERS

As there is no single unified definition of what exactly constitutes an object, there typically exists a large number of objects with their definition very much depending upon user perception and the purpose of the application. In this context, the cluster number *c* can either be manually provided or automatically determined from image data, with the standard approach adopted in the latter case being to use *validity measures* to find the optimal *c* [6]; [11] (Jain and Dubes, 1988); (Chong *et al.*, 2002); (Yen and Langari, 1999)). There are certain situations however, particularly in object-based

segmentation, where validity algorithms fail to generate the correct (optimal) number of clusters (e.g. (Chong *et al.,* 2002); (Dave, 1992)) because they focus on homogeneous regions of interests, which are often contrary to the human perception of an object, and this ultimately degrades the overall clustering performance. Conversely, there are numerous applications in the manufacturing and medical imaging (Pal and Pal, 1993) domains for example, where the number of objects to be segmented is known *a priori*, and so in this paper for all shape-based fuzzy clustering algorithms, c is provided.

5. COMPUTATIONAL TIME COMPLEXITY

A detailed computational time complexity analysis for the FKG algorithm is now presented.

Assumption 1: For object-based image segmentation, $n >> c$ and $n >> p'$ where n is the number of data points, c the number of clusters and p' the data dimension**,** so both $O(c)$ and $O(p')$ will be constant.
A stepwise computational complexity for **Algorithm** 1 is firstly derived, which calculates the intersection point for every datum in each cluster.

5.1 Time Complexity for Algorithm 1

Step 1: To convert $N_i^{''}$ contour points for the i^{th} cluster from Cartesian to polar coordinates takes $O(N_i^{''})$ time , so for *c* clusters, this takes $O(cm')$, where m' is the average number of contour points i.e., $m' = \frac{1}{c}\sum_{i=1}^{c} N_i^{''}$.
Using **Assumption 1**, for object-based segmentation *c* will be small, so $O(c')$ is constant and the time taken $O(m')$.

Step 2: Takes $O(n)$ time to convert n data points from Cartesian to polar coordinates.

Step 3: The angle difference $\Delta\theta_{ijk} = \theta_{ij} \sim \theta_{ik}^{'}$ for a particular j^{th} data point in the i^{th} cluster needs to calculate the difference between it and the angles of all contour points $\theta_{jk}^{'}$ where $1 \le k \le N_i^{''}$. Thus for a given *c* and m' average contour points, this takes $O(cm') = O(m')$ time and for *n* data points $O(nm')$.

Step 4: To find the two contour points on either side of the connecting line l_1 i.e., the line between a datum and cluster centre v_i requires $O(m')$ and so for *n* data takes $O(nm')$.

Step 5: To find the intersection point of two lines takes $O(1)$ time for a single datum, and $O(n)$ time to find the intersection point for n data points.

In summary therefore, for *n* data and *c* cluster contours, all the *intersection points* can be computed in $O(nm')$ time.

5.2 Time Complexity for the FKG Algorithm

A full stepwise complexity analysis for the FKG algorithm is now delineated.

Step 1: The shape initialisation technique (Section 3.1) can be computed in $O(n)$ time.
Step 2: For *n* data points, **Algorithm 1** needs $O(nm')$ time (see Section 5.1), while the contour radius $r_i^{''}$ can be initialised in $O(n)$ time, so the total time for this step is $O(nm')$.

Step 3: To initialise K_{ij} requires $O(n)$ time.

Step 4: The membership values are updated in $O(n)$ time, while similar times are incurred to update the contour radii and cluster centre values, so the total execution time for this step is $O(n)$.

This means the overall computational complexity of the FKG algorithm is $O(nm')$, i.e., it is dependent on the average number of cluster contour points m'. For most real world images, if pixels are uniformly distributed among all clusters, the minimum number of contour points will be $O\left(\sqrt{\frac{n}{c}}\right)$ i.e., $O(m') = O\left(\sqrt{n}\right)$ (Karmakar, 2002) so in the best case scenario, the time complexity will be $O\left(n^{3/2}\right)$. In contrast, for the worst case, particularly where objects comprise only the contour so $m' \cong n$, the overall computational complexity of FKG becomes $O(n^2)$.

6. EXPERIMENTAL RESULTS

To analyse the segmentation performance of the new FKG algorithm, the results were both qualitatively and numerically compared with five other shape-based clustering algorithms, namely FKR, FKE, GK, FCS, and FCES. While both the FKE and FCES algorithms are the generalized versions of FKR and FCS respectively, all algorithms were considered in the evaluation in order to rigorously test the performance of FKG. Various natural and synthetic gray-scale images together with medical images were randomly selected for analysis having multiple regions (objects) and comprising diverse features and shapes[1]. When *pixel location* alone is used as a feature in FCM, it arbitrarily divides the image into a given number of clusters (Ameer *et al.,* 2005b). For this reason, in order to segment foreground objects into perceptually meaningful regions, all background pixels were manually set to zero, with any zero-valued foreground object pixels being replaced by 1 to avoid the possibility of foreground pixels merging with the background, without impacting upon visual perception. All the algorithms are implemented using Matlab 6.1 and the background of the images are separated using Paintshop-Pro software while Pentium 4 personal computer with 1GB RAM is also used to perform the experiments.
To quantitatively appraise the performance of all the fuzzy clustering algorithms, the objective segmentation evaluation method, *discrepancy based on the number of misclassified pixels* (Karmakar, 2002) was used. The two errors, namely Type I, $errorI_i$ and Type II, $errorII_i$ are formally defined as:

$$errorI_i = \frac{\left(\sum_{j=1}^{c} M_{ji} - M_{ii}\right)}{\sum_{j=1}^{c} M_{ji}} \times 100 \tag{10}$$

$$errorII_i = \frac{\left(\sum_{j=1}^{c} M_{ij} - M_{ii}\right)}{\left(\sum_{i=1}^{c}\sum_{j=1}^{c} M_{ij} - \sum_{j=1}^{c} M_{ji}\right)} \times 100 \tag{11}$$

where M is the confusion matrix. 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 misclassified into the i^{th} region. Representative samples of the manually segmented reference regions together with the original images are shown in

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

Figure 3 (a)-(b) to Figure 5 (a)-(b), where to improve the visual interpretation of the results, both the reference and segmented regions are displayed in different colours rather than their original gray-scale intensities.

Before a detailed analysis of the results, an outline is given on the various initialisation strategies used for each clustering method. The GK, FCS and FCES algorithms were all initialised using random membership values μ_{ij}, while FKR used a *fuzzy k-means* (FKM) (see (Karmakar, 2002)) algorithm and FKE employed the same initialisation approach as in [13], namely 10 iterations of FKM followed by 10 iterations of FKR. For FKG, while in principle any clustering algorithm can be used, the GK algorithm was selected for automatic shape initialisation because as highlighted in Section 2, it has consistently proven to give superior results for object-based segmentation compared with FKR, FCS, FKE and FCES. In the FKG experiments, $\lambda = 0.1$ was chosen in (8) to give a higher priority to the initial shape size, the rationale being that as the initial shape has been automatically determined, the ensuing clustering process will not alter it very much and as a consequence, it will not improve the quality of the segmentation. To achieve better performance it is therefore essential to downscale the cluster in each iteration, and for this reason, the initial cluster size was heuristically downscaled by 25%.

The first set of results relate to the *X-ray* image in Figure 3 (a) which has two objects (regions), namely the *femur* (R_1) and *tibia* (R_2) with both regions being arbitrarily shaped. The segmentation results for all the various shape-based algorithms are displayed in Figure 3 (c)-(h). If Figure 3 (c) is compared with the manually segmented reference regions in Figure 3 (b), it is visually apparent that for FKR, a large number of pixels in R_1 have been misclassified into R_2 and vice versa because neither region is circular in shape. Similar judgments can be made concerning the results for the FKE, FCS and FCES algorithms in Figure 3 (d), (f) and (g) respectively, while the GK algorithm failed to effectively segment R_1 and R_2 because as alluded earlier, it does not explicitly consider specific shape information. In contrast, the results for the FKG algorithm (Figure 3 (h)) reveal it correctly classified both R_1 and R_2 with a minimal number of misclassified pixels vindicating the incorporation of generic shape information. The corresponding numerical results for R_1 in Table 1 confirm this improved performance with FKG producing the lowest overall average error of 3.3% compared with the next best average error of 6.2% for both FCS and FCES. This error is directly attributable to the erroneous initial object shape generated by the GK algorithm as evidenced in Figure 3(e).

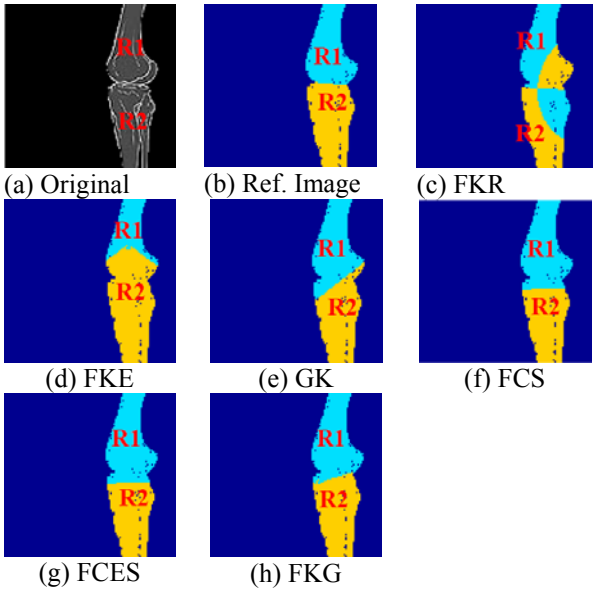


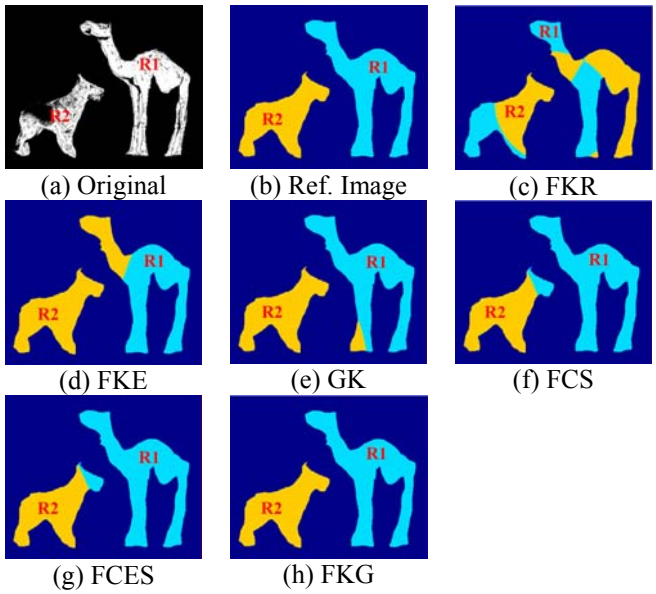
Figure 3: (a) Original *X-ray* image, (b) Manually segmented reference of (a). Figures (c)-(h) the segmented results of (a) using FKR, FKE, GK, FCS, FCES, and FKG respectively.

Table 1: Percentage error of segmentation results in Figure 3 for region R_1

Algorithm	Error		
	Type I	***Type II***	***Mean***

FKR	31	37.4	34.2
FKE	43.8	0	21.9
GK	5.2	10.3	7.7
FCS	0	12.4	6.2
FCES	0	12.4	6.2
FKG	**0.4**	**6.2**	**3.3**

A second series of experiments were performed upon the *dog* image in Figure 4 (a), which has two distinctly asymmetric-shaped objects; the *camel* (R_1) and *dog* (R_2). The corresponding results for FKR, FKE, GK, FCS, FCES, and FKG are evinced in Figure 4 (c)-(h) respectively. If the segmented results produced by FKR in Figure 4 (c) are compared with the reference image in Figure 4 (b), it is clear many pixels from R_2 have been misclassified into R_1 and vice versa, which is not surprising given that neither object is circular in shape. Similarly, FKE also generated a high number of misclassified pixels for R_2 in Figure 4 (d), while interestingly the GK algorithm produced a lower misclassification than FKR, FKE, FCS and FCES because it adapts to the local topological structure of the cluster shape (Figure 4 (e)), though some misclassified pixels from R_1 in R_2 remain. In contrast, both R_1 and R_2 have been correctly classified by FKG with Table 2 corroborating its superior segmentation performance, with a minimum average error of just 0.01% compared with the next best error of 2.75% for GK, which was used to initialize the new FKG algorithm.



(a) Original (b) Ref. Image (c) FKR

(d) FKE (e) GK (f) FCS

(g) FCES (h) FKG

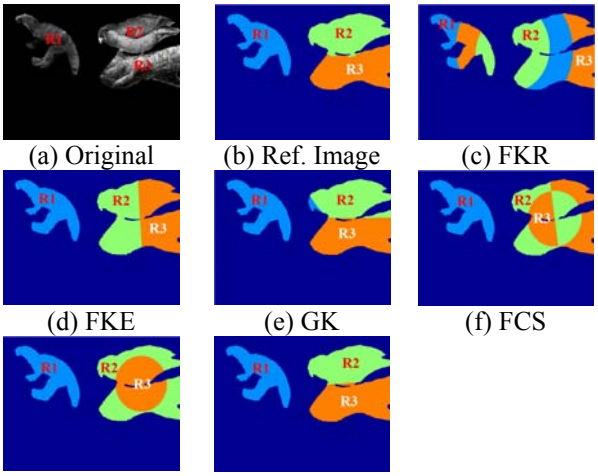
Figure 4: (a) Original *dog* image, (b) Manually segmented reference of (a). Figures (c)-(h) the segmented results of (a) using FKR, FKE, GK, FCS, FCES, and FKG respectively.

Table 2: Percentage error of segmentation results in Figure 4 for R_1 region

Algorithm	Error		
	Type I	***Type II***	***Mean***
FKR	52.3	38.9	45.6
FKE	23.7	0	11.8
GK	5.5	0	2.75
FCS	0	10.5	5.25
FCES	0	9	4.5
FKG	**0**	**0.02**	**0.01**

The final test image (*bird*) used in the experiments (Figure 5 (a)) contained three regions, all of which were arbitrarily shaped, namely *reptile* (R_1), *bird* (R_2) and *tree branch* (R_3), with R_2 and R_3 being connected. The corresponding results for the FKR, FKE, GK, FCS, FCES and FKG algorithms are displayed in Figure 5 (c)-(h) respectively. As with the other

test images, because all objects are arbitrarily shaped, there is an ensuing large number of misclassified pixels between R_1, R_2 and R_3 for FKE as confirmed in Figure 5 (c). The results for FKE, FCS and FCES in Figure 5 (d), (f) and (g) respectively, are also characterized by high pixel misclassifications, though for these algorithms this is especially noticeable between the connected two regions R_2 and R_3. Conversely, the FKG algorithm (Figure 5 (h)) has successfully separated R_1 and generated significantly fewer misclassified pixels for both R_2 and R_3 compared with the other algorithms. These perceptual judgments are substantiated by the quantitative results in Table 3 which confirm FKG again had the lowest mean error of 1% compared with 46.8% for FKR and 3.02% for GK.



(a) Original (b) Ref. Image (c) FKR

(d) FKE (e) GK (f) FCS

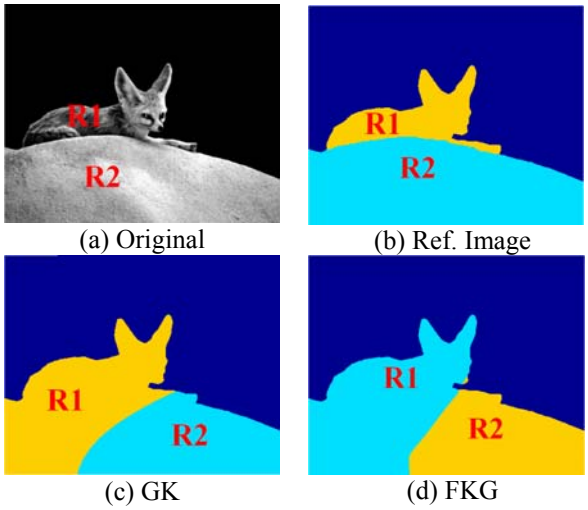
(g) FCES (h) FKG

Figure 5: (a) Original *bird* image, (b) Manually segmented reference of (a). Figures (c)-(h) the segmented results of (a) using FKR, FKE, GK, FCS, FCES, and FKG respectively.

Table 3: Percentage errors for the segmentation results in Figure 5

Algorithm	Error						Mean
	Type I			***Type II***			
	R1	R2	R3	R1	R2	R3	
FKR	65.7	59.4	62.8	35.8	27.5	29.8	46.8
FKE	0	47.6	46.5	0	27.1	26.6	24.6
GK	0	7	4.7	1.8	2.7	1.9	3.02
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
FKG	**0**	**3.9**	**0**	**0**	**0**	**2.2**	**1**

To rigorously assess the performance of the FKG algorithm, experiments were conducted upon 185 randomly selected images containing multiple objects with circular, elliptic and arbitrary shaped structures of differing orientations and size. With the object shape descriptor being generated by GK initialization (Section 3.1), FKG produced the best segmentation performance for 80 images while as shown in Figure 8, FKR, FKE, GK, FCS, and FCES provided better segmentation results for only 2, 17, 4, 6, and 21 respectively. FKG also produced analogous results for 29 and 27 images respectively to GK and the other algorithms considered, while FKR, FKE, FCS, and FCES exhibited similar results for only 6, 17, 16, and 19 images respectively. Since these latter algorithms are specifically tailored for regular geometric objects, they provided either comparable or slightly improved segmented results compared with FKG for such objects. As highlighted in Section 2, while FKG adopts the same philosophy as the FKR, FCS, FKE and FCES algorithms (see (Man and Gath, 1994); (Dave, 1990); (Dave, 1992); (Gath and Hoory, 1995)), it does explicitly consider contour-based shape information and not the enclosed region that defines a particular object, with the consequence that in only 4 images did FKG fail to improve upon the GK initialization. An example is illustrated in Figure 6 for the *cat* image, where the degraded FKG segmentation performance is directly due to the poor initial segmented regions produced by GK leading to poor initial shapes and subsequent improper scaling of these shapes.

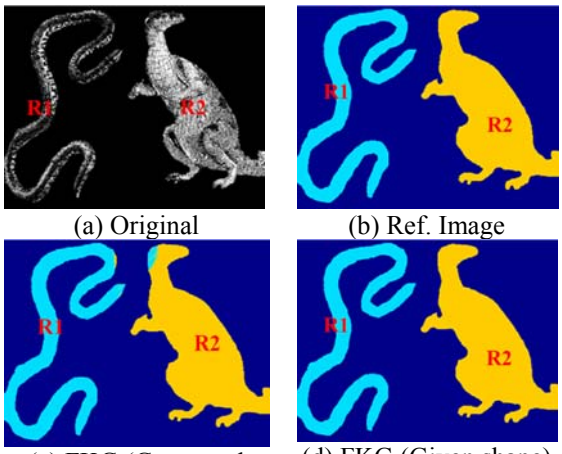


(a) Original (b) Ref. Image

(c) GK (d) FKG

Figure 6: (a) Original *cat* image, (b) Manually segmented reference of (a). Figures (c)-(d) the segmented results of (a) using GK and FKG respectively.

Another example is given in **Figure 7** for the *snake* image containing two arbitrary shaped objects, where it clearly visible that the segmentation results of FKG for given shape information is better than that of for automated generated shape information.



(a) Original (b) Ref. Image

(c) FKG (Generated Points) (d) FKG (Given shape)

Figure 7: (a) Original *snake* image, (b) Manually segmented reference of (a). Figures (c)-(d) the segmented results of (a) using FKG for generated contour points and given contour points respectively.

Overall, to assess the results validate the enhanced performance of the FKG algorithm in being able to segment arbitrary-shaped objects, with the consistent superior segmentation results for most images justifying the strategy of seamlessly integrating generic shape information into a clustering framework.

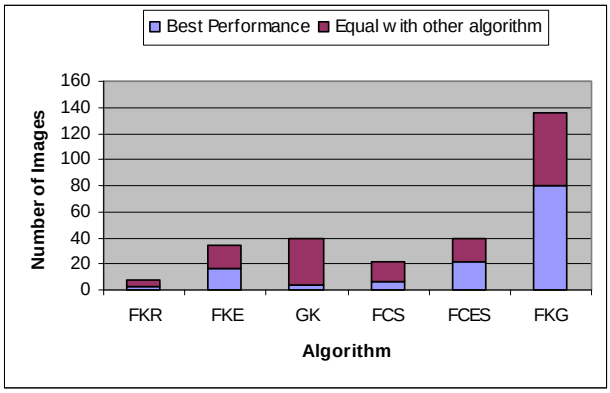


Figure 8: The best segmentation results for six different fuzzy clustering algorithms when arbitrary shape contours are automatically determined from the initialization.

7. CONCLUSION

This paper has presented a new shape-based fuzzy image segmentation algorithm called *detection and separation of generic shaped objects* (FKG) that seamlessly incorporates generic shape information, and introduces a shape constraint to preserve the original object shape and ensure its optimization during subsequent iterative scaling. A thorough qualitative and quantitative analysis has been conducted to compare the performance with existing shape-based algorithms using a myriad of images comprising multiple objects having different shapes, with FKG consistently providing better segmentation results by virtue of integrating generic shape information into an FCM-based fuzzy clustering framework. The FKG algorithm can further be extended to implement it in MPEG-4 for real images, which has already been implemented for synthetic images.

8. REFERENCES

[1] Gonzalez R. C. and Woods R. E. (2002). *Digital Image Processing*, New Jersey: Prentice Hall Inc.

[2] Hoppner, F., *et al.* (1999). *Fuzzy Cluster Analysis: methods for classification, data analysis, and image recognition*, New York: John Wiley & Sons, Ltd.

[3] Pham D. L. and Prince J. L. (1999). An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities, *Pattern Recognition Letters*, 20, 57-68.

[4] Gath I. and Geva A. C. (1989). Unsupervised optimal fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(7), 773-781.

[5] Pal N. R. and Pal S. K. (1993). A review on image segmentation techniques, *Pattern Recognition*, 26(9), 1277-1294.

[6] Bezdek J.C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithm*, New York: Plenum Press.

[7] Krishnapuram R. and Keller J. M. (1993). A possibilistic approach to clustering, *International Journal of Fuzzy Systems*, 2(2), 98-110.

[8] Jiu-Lun Fan, Wen-Zhi Zhen, and Wei-Xin Xie (2003). Suppressed fuzzy *c*-means clustering algorithm, *Pattern Recognition Letters*, 24, 1607-1612.

[9] Gustafson D. E. and Kessel W. C. (1979). Fuzzy clustering with a fuzzy covariance matrix, *In Proceedings of IEEE CDC*, 761-766.

[10] Yael Man and Isak Gath (1994). Detection and separation of ring-shaped clusters using fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16 (8), 855-86.

[11] Dave R.N. (1990). Fuzzy shell-clustering and applications to circle detection in digital images, *International Journal on General System*, 16 (4), 343-355.

[12] Dave R.N. (1992). Generalized Fuzzy c-Shell clustering and detection of circular and elliptical boundaries, *Pattern Recognition*, 25 (7), 713-721.

[13] Isak Gath and Dan Hoory (1995). Fuzzy clustering of elliptic ring-shaped clusters, *Pattern Recognition Letters*, 16, 727-741.
[14] Babuska R., Van der Veen P. J. and Kaymak U. (2002). Improved covariance estimation for Gustafson-Kessel clustering, *IEEE international Conference on Fuzzy Systems*, 2, 1081-1085.
[15] Dengsheng Zhang and Guojun Lu (2004). Review of shape representation and description techniques, *Pattern Recognition*, 37, 1-19.
[16] Ameer Ali M., Karmakar G., and Dooley L.S. (2005). Fuzzy image segmentation of generic shaped clusters, *IEEE International Conference on Image Processing*, 2, 1202-1205.
[17] Jain A. K. and Dubes R. C. (1988). *Algorithms for Clustering Data*, Prentice Hall Inc., New Jersey.
[18] Chong, A., Gedeon, T.D. and Koczy, L.T (2002). A hybrid approaches for solving the cluster validity problem, *International Conference on Digital Signal Processing*, 2, 1207-1210.
[19] Yen J. and Langari R. (1999), *Fuzzy logic*, New Jersey : Prentice Hall Inc.
[20] Dave R. N. (1992). Boundary detection through fuzzy clustering, *IEEE International Conference on Fuzzy Systems*, 127-134.
[21] Francis S. Hill (1994). *Computer Graphics*, New Jersey: Prentice Hall Inc.
[22] Hearn D. and Baker M. P. (1994). *Computer Graphics*, New Jersey: Prentice Hall Inc., 1994.
[23] Jiwen Zhang (1999). C-Bezier curves and surfaces, *IDEAL International Conference on Graphical Models and Image Processing*, 2-15.
[24] Tony D. Derose (1988). Geometric continuity, shape parameters, and geometric constructions for catmull-rom splines, *ACM Transaction on Graphics*, 7(1), 1-41.
[25] Ahmed R., Dooley L. S., and Karmakar G. C. (2007). Probabilistic spatio-temporal video object segmentation using *a priori* shape descriptor, *IEEE International Conference on Acoustics, Speech & Signal Processing*, 1, 1081-1084.
[26] L. da F. Costa, and R. M. Cesar Jr. (2001). *Shape Analysis and Classification: Theory and Practice*, 331-419, CRC Press LLC.
[27] Foley J. D. *et al.* (1999). *Computer Graphics: Principles and Practice*, Addison-Wesley Pub. Co. Inc.
[28] Ameer Ali M., Dooley L. S., and Karmakar G. C. (2006). Object based segmentation using fuzzy clustering, *IEEE International Conference on Acoustics, Speech & Signal Processing*, 2, 105-108.
[29] Ameer Ali M., Dooley L. S., and Karmakar G. C. (2005). Automatic feature set selection for merging image segmentation results using fuzzy clustering, *International Conference on Computer and Information Technology*, 337-342.
[30] Karmakar G. C. (2002). *An Integrated Fuzzy Rule-Based Image Segmentation Framework*, PhD, Thesis Dissertation, Monash University, Australia.
[31] Watanabe N and Imaizumi T. (2002). Fuzzy *k*-Means Clustering with Crisp Regions, *IEEE International Conference on Fuzzy Systems*, 1528-1531.

BIOGRAPHY

Dr. Ameer Ali has 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 Computer Science and Engineering, East West University, Dhaka, Bangladesh. He has more than 28 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.

GOUR C. KARMAKAR received the B.Sc. Eng. degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 1993 and Masters and Ph.D. degrees in Information Technology from the Faculty of Information Technology, Monash University, in 1999 and 2003, respectively. He also successfully completed the Graduate Certificate in Higher Education, Higher Education Development Unit, Monash University, Australia in Semester 1, 2005. He was with the Computer Division, Bangladesh Open University from 1994 to 1998. He is currently a Senior Lecturer at the Gippsland School of Information Technology, Monash University, Australia. He has published over 67 peer-reviewed research publications including twelve international peer reviewed reputed journal papers and was jointly awarded three best paper awards. His research interest includes image and video processing, mobile and wireless sensor networks.

LAURENCE S. DOOLEY received his B.Sc.(Hons), M.Sc. and Ph.D. degrees in Electrical and Electronic Engineering from the University of Wales/Cymru (Swansea) in 1981, 1983 and 1987, respectively. He is currently *Professor of Information and Communication Technologies* in the Department of Communication and Systems at The Open University, UK, where his research interests include: *next generation* multimedia technologies, cognitive radio networks, MANETs, 4G security, and technology transfer and commercialisation strategies for small-to-medium enterprises. He has edited one book and published more than 200 peer-reviewed scientific journals, book chapters; monographs and conference papers, with 3 international best paper awards and/or nominations, including the IEEE Communications Society sponsored Outstanding Paper Prize at the *2006 International Conference on Next Generation Wireless Systems*. He has successfully supervised 18 PhD/Masters research students together with being the recipient of significant public and private sector funding to support his multifaceted research. He is a Chartered Engineer, a Fellow of the British Computer Society and a Senior Member of the IEEE, as well as being a Vice President of the Welsh Crawshay's Rugby Football Club.

APPENDIX A

THE GUSTAFSON-KESSEL (GK) ALGORITHM

Let *c* and *n* be the number of clusters and data respectively. S is the dataset containing $\lfloor S_j \rfloor$, μ is a set of membership values, V is a vector containing the cluster centre values 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:

$$D_{ij}^2 = (S_j - v_i)^T A_i (S_j - v_i) \tag{A.1}$$

where A_i is the norm inducing matrix, which allows the distance norm to adapt to the local topological structure of the data (see (Babuska *et al.*, 2002)). The GK algorithm iteratively optimizes the following objective function to derive μ_{ij} and v_i:

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

for $0 \leq \mu_{ij} \leq 1; \quad i \in \{1, \ldots, c\}$ and $j \in \{1, \ldots, n\}$ (A.3)

and $\sum_{i=1}^{c} \mu_{ij} = 1; \quad j \in \{1, \ldots, n\}$ (A.4)

For $D_{ij} = 0$, a crisp decision is mandated and the j^{th} datum is classified into the i^{th} cluster, otherwise the membership value μ_{ij} is updated using D_{ij} so:

IF $(D_{ij} = 0)$ THEN

$$\mu_{ij} = 1 \text{ maintaining } \sum_{i=1}^{c} \mu_{ij} = 1 \tag{A.5}$$

ELSE

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

While the cluster centre v_i is updated by:

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

and A_i adapts the local topological structure of the cluster shape as follows:

$$F_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^q (S_j - v_i)^T (S_j - v_i)}{\sum_{j=1}^{n} (\mu_{ij})^q} \tag{A.8}$$

$$A_i = \left| \rho_i \det(F_i)^{1/P'} F_i^{-1} \right| \tag{A.9}$$

where F_i is the fuzzy covariance matrix, P' is the data dimension, and ρ_i is the cluster volume, which is usually set equal to unity.

APPENDIX B

PROOF OF LEMMA 1

Proof: The shape constraint $r_{ij} = k_{ij} * \sum r_{ij}$ will not deform an object shape because k_{ij} is defined as the ratio of the normalized radius with respect to the sum of all radii $\sum r_{ij}$, so it is constant for every iteration. This therefore represents the shape irrespective of its size, so for example, in the t^{th} and $(t+1)^{th}$ iterations, the following holds:

$$k_{ij}(t) = \frac{r_{ij}(t)}{\sum r_{ij}(t)} = \frac{r_{ij}(t+1)}{\sum r_{ij}(t+1)} = k_{ij}(t+1)$$

$$\Rightarrow \frac{\sum r_{ij}(t)}{r_{ij}(t)} = \frac{\sum r_{ij}(t+1)}{r_{ij}(t+1)}$$

$$\Rightarrow \frac{r_{ij}(t) + \sum_{k\neq i, m\neq j} r_{km}(t)}{r_{ij}(t)} = \frac{r_{ij}(t+1) + \sum_{k\neq i, m\neq j} r_{km}(t+1)}{r_{ij}(t+1)}$$

$$\Rightarrow 1 + \frac{\sum_{k\neq i, m\neq j} r_{km}(t)}{r_{ij}(t)} = 1 + \frac{\sum_{k\neq i, m\neq j} r_{km}(t+1)}{r_{ij}(t+1)}$$

$$\Rightarrow \frac{\sum_{k\neq i, m\neq j} r_{km}(t)}{r_{ij}(t)} = \frac{\sum_{k\neq i, m\neq j} r_{km}(t+1)}{r_{ij}(t+1)}$$

$$\Rightarrow \sum_{k\neq i,\ m\neq j} r_{km}(t+1) = \frac{r_{ij}(t+1)}{r_{ij}(t)} \sum_{k\neq i,\ m\neq j} r_{km}(t) \qquad \text{(B.1)}$$

So every $r_{km}(t+1)$ in the current iteration is updated by the ratio $\frac{r_{ij}(t+1)}{r_{ij}(t)}$ and the respective $r_{km}(t)$ from the previous iteration. Since this ratio is constant for all radii, it either scales up or down by the same amount and so the scaling does not distort the original shape of the region. The shape constraint $r_{ij} = k_{ij} * \sum r_{ij}$ thus ensures the original object shape is preserved. ■

APPENDIX C

SOME FORMAL FKG ALGORITHM PROOFS

The objective function of the proposed FKG algorithm is defined as follows:

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

subject to:

$$0<\sum_{j=1}^{n}\mu_{ij}<n;\qquad 1\le j\le n \tag{C.2}$$

$$\sum_{i=1}^{c}\mu_{ij}=1;\qquad 1\le i\le c \tag{C.3}$$

$$r_{ij}=K_{ij}\sum_{t=1}^{n}r_{it} \tag{C.4}$$

where $d_{ij}=d(S_j,v_i)-r_{ij}=D_{ij}^{'}-r_{ij}$, $r_{ij}=d(S_{ij}^{'},v_i)$, $d(S_j,v_i)=D_{ij}^{'}=\|S_j-v_i\|$, with all other parameters being defined as in **Appendix A**.

Applying Lagrangian multiplier and using (C.1) to (C.4), (C.1) can be derived as follows:

$$\begin{aligned}J^{'}=J_q(\mu,V)&=\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^q d_{ij}^2+\lambda_1\left(\sum_{i=1}^{c}\mu_{ij}-1\right)\\&+\lambda_2\left(r_{ij}-K_{ij}\sum_{t=1}^{n}r_{it}\right)\end{aligned} \tag{C.5}$$

To derive μ_{ij}, (C.5) is differentiated with respect to μ_{ij} and set equal to 0

$$\frac{dJ^{'}}{d\mu_{ij}}=\frac{d}{d\mu_{ij}}\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^q d_{ij}^2+\lambda_1\frac{d}{d\mu_{ij}}\left(\sum_{i=1}^{c}\mu_{ij}-1\right)=0$$

$$\Rightarrow q\mu_{ij}^{q-1}d_{ij}^2+\lambda_1=0$$

$$\Rightarrow -\frac{\lambda_1}{q}=\mu_{ij}^{q-1}.d_{ij}^2$$

$$\Rightarrow\left(-\frac{\lambda_1}{q}\right)^{\frac{1}{q-1}}=\mu_{ij}\,.\,d_{ij}^{\frac{2}{q-1}}$$

$$\Rightarrow\mu_{ij}=\left(-\frac{\lambda_1}{q}\right)^{\frac{1}{q-1}}.\left(\frac{1}{d_{ij}}\right)^{\frac{2}{(q-1)}} \tag{C.6}$$

Again from (C.3):

$$\sum_{k=1}^{c} \mu_{kj} = 1$$

$$\Rightarrow \sum_{k=1}^{c} \left(-\frac{\lambda_1}{q} \right)^{\frac{1}{q-1}} \cdot \left(\frac{1}{d_{kj}} \right)^{\frac{2}{q-1}} = 1$$

$$\Rightarrow \left(\frac{-\lambda_1}{q} \right)^{\frac{1}{q-1}} \cdot \sum_{k=1}^{c} \left(\frac{1}{d_{kj}} \right)^{\frac{2}{q-1}} = 1$$

$$\Rightarrow \mu_{ij} \cdot d_{ij}^{\frac{2}{q-1}} \cdot \sum_{k=1}^{c} \left(\frac{1}{d_{kj}} \right)^{\frac{2}{q-1}} = 1$$

$$\Rightarrow \mu_{ij} \cdot \sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{q-1}} = 1$$

$$\Rightarrow \mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{q-1}}} \qquad \text{(C.7)}$$

To calculate r_{ij}, (C.5) is differentiated with respect to r_{ij}:

$$\frac{dJ'}{dr_{ij}} = \frac{d}{dr_{ij}}\left\{\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{q}d_{ij}^{2} + \lambda_2\left(r_{ij} - K_{ij}\sum_{t=1}^{n}r_{it}\right)\right\}$$

$$= \mu_{ij}^{q}\,2d_{ij}\,\frac{d}{dr_{ij}}\left(d_{ij}\right) + \lambda_2\left(1 - K_{ij}\right) \tag{C.8}$$

Now $$\frac{d}{dr_{ij}}\left(d_{ij}\right) = \frac{d}{dr_{ij}}\left(D_{ij}' - r_{ij}\right) = \left(-1\right) \tag{C.9}$$

So from (B.8) and (C.9), and setting (C.8) =0 gives:

$$\frac{dJ'}{dr_{ij}} = \left(-1\right)\mu_{ij}^{q}.2d_{ij} + \lambda_2\left(1 - K_{ij}\right) = 0$$

$$\Rightarrow \left(-1\right)\mu_{ij}^{q}.2\left(D_{ij}' - r_{ij}\right) + \lambda_2\left(1 - K_{ij}\right) = 0$$

$$\Rightarrow -2\mu_{ij}^{q}D_{ij}' + 2\mu_{ij}^{q}r_{ij} + \lambda_2\left(1 - K_{ij}\right) = 0$$

$$\Rightarrow 2\mu_{ij}^{q}D_{ij}' - 2\mu_{ij}^{q}r_{ij} - \lambda_2\left(1 - K_{ij}\right) = 0$$

$$\Rightarrow r_{ij} = D_{ij}' - \frac{\lambda_2}{2}.\frac{1 - K_{ij}}{\mu_{ij}^{q}} \tag{C.10}$$

Again, using (C.4) and (C.10),

$$r_{ij} = K_{ij}\sum_{t=1}^{n}r_{it} = K_{ij}\sum_{t=1}^{n}\left\{D_{it}' - \frac{\lambda_2}{2}.\frac{1 - K_{it}}{\mu_{it}^{q}}\right\}$$

$$= K_{ij}\sum_{t=1}^{n}D_{it}' - \frac{\lambda_2}{2}.K_{ij}\sum_{t=1}^{n}\frac{1 - K_{it}}{\mu_{it}^{q}}$$

$$\Rightarrow K_{ij}\sum_{t=1}^{n}D_{it}' - \frac{\lambda_2}{2}.K_{ij}\sum_{t=1}^{n}\frac{1 - K_{it}}{\mu_{it}^{q}} = r_{ij}$$

$$\Rightarrow K_{ij}\sum_{t=1}^{n}D_{it}' - \frac{\lambda_2}{2}.K_{ij}\sum_{t=1}^{n}\frac{1 - K_{it}}{\mu_{it}^{q}} = D_{ij}' - \frac{\lambda_2}{2}.\frac{1 - K_{ij}}{\mu_{ij}^{q}}$$

$$\Rightarrow K_{ij}\sum_{t=1}^{n}D_{it}' - D_{ij}' = \frac{\lambda_2}{2}.\left\{K_{ij}\sum_{t=1}^{n}\frac{1 - K_{it}}{\mu_{it}^{q}} - \frac{1 - K_{ij}}{\mu_{ij}^{q}}\right\}$$

$$\Rightarrow \frac{\lambda_2}{2} = \frac{K_{ij}\sum_{t=1}^{n} D_{it}^{'} - D_{ij}^{'}}{K_{ij}\sum_{t=1}^{n}\frac{1-K_{it}}{\mu_{it}^{q}} - \frac{1-K_{ij}}{\mu_{ij}^{q}}} \tag{C.11}$$

Using (C.11) and (C.10) can be expressed as:

$$\therefore r_{ij} = D_{ij}^{'} - \frac{K_{ij}\sum_{t=1}^{n} D_{it}^{'} - D_{ij}^{'}}{k_{ij}\sum_{t=1}^{n}\frac{1-K_{it}}{\mu_{it}^{q}} - \frac{1-K_{ij}}{\mu_{ij}^{q}}} \cdot \frac{1-K_{ij}}{\mu_{ij}^{q}} \tag{C.12}$$

To obtain v_i, (C.5) is differentiated with respect to v_i:

$$\frac{dJ^{'}}{dv_i} = \frac{d}{dv_i}\left\{\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{q} d_{ij}^{2}\right\}$$

$$= \sum_{j=1}^{n}\mu_{ij}^{q} \cdot \frac{d}{dv_i}(d_{ij}^{2})$$

$$= \sum_{j=1}^{n}\mu_{ij}^{q} .2.d_{ij} \cdot \frac{d}{dv_i} d_{ij}$$

$$= 2\sum_{j=1}^{n}\mu_{ij}^{q} .d_{ij} \cdot \frac{d}{dv_i}\left\{d(S_j, v_i) - r_{ij}\right\}$$

$$= 2\sum_{j=1}^{n}\mu_{ij}^{q} d_{ij}\left\{\frac{d}{dv_i} d(S_j, v_i) - \frac{d}{dv_i} r_{ij}\right\} \tag{C.13}$$

If datum S and cluster center V are 2D, i.e., $\begin{bmatrix} S_{j1} \\ S_{j2} \end{bmatrix}$ and $\begin{bmatrix} v_{i1} \\ v_{i2} \end{bmatrix}$ respectively, then

$$\frac{d}{dv_i} d(S_j, v_i) = \frac{d}{dv_i}\left\{\sqrt{(S_{j1} - v_{i1})^2 + (S_{j2} - v_{i2})^2}\right\}$$

$= \frac{1}{2\sqrt{(S_{j1} - v_{i1})^2 + (S_{j2} - v_{i2})^2}} .2(S_{j1} - v_{i1}).(-1)$ when the X-coordinate is considered and:

$$\frac{d}{dv_i}\left\{\sqrt{(S_{j1} - v_{i1})^2 + (S_{j2} - v_{i2})^2}\right\}$$

$= \frac{1}{2\sqrt{(S_{j1} - v_{i1})^2 + (S_{j2} - v_{i2})^2}} .2(S_{j2} - v_{i2}).(-1)$ while this considers only the Y-coordinate. So,

$$\therefore \frac{d}{dv_i} d(S_j, v_i) = \frac{2(S_j - v_i)(-1)}{2\sqrt{(S_{j1} - v_{i1})^2 + (S_{j2} - v_{i2})^2}}$$

$$= \frac{(-1)(S_j - v_i)}{d(S_j, v_i)} = \frac{(-1)(S_j - v_i)}{D_{ij}^{'}} \tag{C.14}$$

Similarly,

$$\frac{d}{dv_i} r_{ij} = \frac{(-1)(S_{ij}^{'} - v_i)}{r_{ij}} \tag{C.15}$$

Using (C.13):

$$\therefore \frac{dJ^{'}}{dv_i} = 2\sum_{j=1}^{n} \mu_{ij}^{q} d_{ij} \left\{ \frac{d}{dv_i} d(S_j, v_i) - \frac{d}{dv_i} r_{ij} \right\}$$

$$= 2\sum_{j=1}^{n} \mu_{ij}^{q} . d_{ij} \left\{ \frac{(-1)(S_j - v_i)}{D_{ij}^{'}} - \frac{(-1)(S_{ij}^{'} - v_i)}{r_{ij}} \right\}$$

$$= 2\sum_{j=1}^{n} \mu_{ij}^{q} \left(D_{ij}^{'} - r_{ij}\right)(-1) \left\{ \frac{S_j - v_i}{D_{ij}^{'}} - \frac{S_{ij}^{'} - v_i}{r_{ij}} \right\}$$

$$= (-1)2\sum_{j=1}^{n} \mu_{ij}^{q} \left\{ S_j - v_i - D_{ij}^{'} \frac{S_{ij}^{'} - v_i}{r_{ij}} \right.$$

$$\left. - r_{ij} \frac{S_j - v_i}{D_{ij}^{'}} + S_{ij}^{'} - v_i \right\}$$

$$= (-1)2\sum_{j=1}^{n} \mu_{ij}^{q} \left\{ S_j - 2v_i - D_{ij}^{'} \frac{S_{ij}^{'} - v_i}{r_{ij}} \right.$$

$$\left. - r_{ij} \frac{S_j - v_i}{D_{ij}^{'}} + S_{ij}^{'} \right\} \tag{C.16}$$

For optimization, (C.16) is set to 0:

$$(-1)2\sum_{j=1}^{n} \mu_{ij}^{q} \left\{ S_j - 2v_i - D_{ij}^{'} \frac{S_{ij}^{'} - v_i}{r_{ij}} \right.$$

$$\left. - r_{ij} \frac{S_j - v_i}{D_{ij}^{'}} + S_{ij}^{'} \right\} = 0$$

$$=> 2v_i \sum_{j=1}^{n} \mu_{ij}^{q} = \sum_{j=1}^{n} \mu_{ij}^{q} \left\{ S_j - D_{ij}^{'} \frac{S_{ij}^{'} - v_i}{r_{ij}} \right.$$

$$\left. + S_{ij}^{'} - r_{ij} \frac{S_j - v_i}{D_{ij}^{'}} \right\}$$

$$=> v_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{q} \left\{ S_j - D_{ij}^{'} . \frac{S_{ij}^{'} - v_i}{r_{ij}} + S_{ij}^{'} - r_{ij} \frac{S_j - v_i}{D_{ij}^{'}} \right\}}{2 \sum_{j=1}^{n} \mu_{ij}^{q}} \qquad \text{(C.17)}$$