Fundamenta Informaticae XX (2007) 1–17
IOS Press

Fuzzy Symmetry Based Real-Coded Genetic Clustering Technique for Automatic Pixel Classification in Remote Sensing Imagery

Sriparna Saha and Sanghamitra Bandyopadhyay

Abstract. The problem of classifying an image into different homogeneous regions is viewed as the task of clustering the pixels in the intensity space. In particular, satellite images contain landcover types some of which cover significantly large areas, while some (e.g., bridges and roads) occupy relatively much smaller regions. Automatically detecting regions or clusters of such widely varying sizes presents a challenging task. In this paper, a newly developed real-coded variable string length genetic fuzzy clustering technique with a new point symmetry distance is used for this purpose. The proposed algorithm is capable of automatically determining the number of segments present in an image. Here assignment of pixels to different clusters is done based on the point symmetry based distance rather than the Euclidean distance. The cluster centers are encoded in the chromosomes, and a newly developed fuzzy point symmetry distance based cluster validity index, *FSym*-index, is used as a measure of the validity of the corresponding partition. This validity index is able to correctly indicate presence of clusters of different sizes and shapes as long as they are internally symmetrical. The space and time complexities of the proposed algorithm are also derived. The effectiveness of the proposed technique is first demonstrated in identifying two small objects from a large background from an artificially generated image and then in identifying different landcover regions in remote sensing imagery. Results are compared with those obtained using the well known fuzzy C-means algorithm both qualitatively and quantitatively.

Keywords: cluster validity index, fuzzy clustering, symmetry, point symmetry based distance, Kd tree, variable string length genetic algorithm, remote sensing imagery.

1. Introduction

An important task in remote sensing applications is the classification of pixels in the images into homogeneous regions, each of which corresponds to some particular landcover type. This problem has often been modeled as a clustering problem [6] [13]. However since it is difficult to have apriori information about the number of clusters in satellite images, the clustering algorithms should be able to automatically

Address for correspondence: Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India, Email:{sriparna_r,sanghami}@isical.ac.in

determine this value. Moreover, in satellite images it is often the case that some regions occupy only a few pixels, while the neighboring regions are significantly large. Thus automatically detecting regions or clusters of such widely varying sizes presents a challenge in designing clustering algorithms and validity indices.

The aim of any clustering technique is to evolve a partition matrix $U(X)$ representing a possible grouping of the given data set $X = \{x_1, x_2, \ldots, x_n\}$, into a number, say c, of clusters such that patterns in the same group are similar in some sense and patterns in different groups are dissimilar in the same sense. The partition matrix $U(X)$ of size $c \times n$ may be represented as $U = [u_{ik}]$, $1 \leq i \leq c; 1 \leq k \leq n$, where u_{ik} is the membership of pattern x_k to cluster C_i ($i = 1, \ldots, c$). The measure of similarity is data dependent. It may be noted that one of the basic feature of shapes and objects is symmetry. Symmetry is considered a pre-attentive feature which enhances recognition and reconstruction of shapes and objects [2]. Almost every interesting area around us consists of some generalized form of symmetry. As symmetry is so common in the natural world, it can be assumed that some kind of symmetry exists in the clusters also. Based on this, Su and Chou have proposed a symmetry based clustering technique [18]. Here they have assigned points to a particular cluster if they present a symmetrical structure with respect to the cluster centre. A new type of non-metric distance, based on point symmetry, is proposed which is used in a *K*-means based clustering algorithm, referred to as Symmetry-based *K*-means (SBKM) algorithm. SBKM is found to provide good performance on different types of data sets where the clusters have internal symmetry. However it can be shown that SBKM will fail for some data sets, where the clusters themselves are symmetrical with respect to some intermediate point. Though this has been mentioned in a subsequent paper by Chou et al. [10] where they have suggested a modification, the modified measure has the same limitation of the previous one [18]. No experimental results have been provided in [10]. It has been shown in [4] that the PS distance proposed in [10] also has some serious drawbacks and a new PS distance (d_{ps}) is defined in [4] in order to remove these drawbacks. For reducing complexity of point symmetry distance computation, *Kd-tree* based data structure is used.

Note that, in remote sensing imagery, a pixel corresponds to an area of the land space, which may not necessarily belong to a single type of landcover. This in turn indicates that the pixels in a satellite image can be associated with a large amount of imprecision and uncertainty. Therefore, application of the principles of fuzzy set theory [9] appears to be natural and appropriate in such domains. In fuzzy partitioning of the data, the following conditions hold on the partition matrix U (representing non-degenerate clustering): $0 < \sum_{k=1}^{n} u_{ik} < n$, $\sum_{i=1}^{c} u_{ik} = 1$, and $\sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik} = n$. Fuzzy C-Means (FCM) [8] is a widely used technique that uses the principles of fuzzy sets to evolve a partition matrix $U(X)$ while minimizing the measure $\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m D^2(v_i, x_k)$ where $D(v_i, x_k)$ represents the distance from point x_k ($k = 1, \ldots, n$) to the center of ith cluster, v_i ($i = 1, \ldots, c$), and m is the weighting coefficient. However, FCM has three major limitations: it requires the *a priori* specification of the number of clusters (c), it often gets stuck at suboptimal solutions based on the initial configuration of the system and it is able to find only hyperspherical equisized clusters. In this article, we attempt to overcome the above mentioned limitations of FCM by using the search capability of genetic algorithms to automatically evolve the fuzzy partitions of a data such that some measure of goodness in terms of symmetry present in the partitions is optimized.

In this article, a fuzzy variable string length genetic point symmetry (Fuzzy-VGAPS) based clustering technique is used for the image segmentation purpose. This algorithm is capable of automatically segmenting the different landcover regions from remote sensing satellite images. Here membership values of points to different clusters are computed based on the newly developed point symmetry based

distance [4] rather than the Euclidean distance. This enables the proposed algorithm to automatically evolve the appropriate clustering of all types of clusters, both convex and non convex, which have some symmetrical structures. The chromosome encodes the centres of a number of clusters, whose value may vary. A newly developed fuzzy point symmetry based cluster validity index named *FSym*-index is utilized for computing the fitness of the chromosomes. The index is capable of detecting any type of clusters irrespective of its size, shape and convexity as long as they possess the "point symmetry" property.

The effectiveness of the proposed Fuzzy-VGAPS clustering algorithm is first demonstrated in automatically segmenting an artificially generated image having clusters of varying sizes. This image contains two small objects in the large background. Thereafter the automatic segmentation results obtained by the proposed Fuzzy-VGAPS clustering are reported for Indian Remote Sensing (IRS) satellite images of the parts of the cities of Kolkata and Mumbai and a SPOT (Systeme Probatoire d'Observation de la Terre) satellite image of Kolkata. From the ground truth available for the images, the effectiveness of the method in automatically identifying the different landcover types present in the images has been verified. The superiority of the proposed technique, as compared to the well known FCM algorithm [8], is demonstrated both quantitatively (using two well-known Euclidean distance based cluster validity indices, $\mathcal{I}$-index and XB-index) and qualitatively (i.e., visually).

2. The Existing Point Symmetry (PS)- based Distance Measures [18] [10]

Motivated by the property of point symmetry that clusters often exhibit, a PS-distance was proposed in [18] which was further modified in [10]. The modified distance is defined as follows:
Given N patterns, $\overline{x}_j, j = 1, \ldots N$, and a reference vector $\overline{c}$ (e.g., a cluster centroid), the "point symmetry distance" between a pattern $\overline{x}_j$ and the reference vector $\overline{c}$ is defined as

$$d_c(\overline{x}_j, \overline{c}) = d_s(\overline{x}_j, \overline{c}) \times d_e(\overline{x}_j, \overline{c}) \tag{1}$$

where

$$d_s(\overline{x}_j, \overline{c}) = \min_{i=1,\ldots N \text{ and } i \neq j} \left(\frac{\|(\overline{x}_j - \overline{c}) + (\overline{x}_i - \overline{c})\|}{\|(\overline{x}_j - \overline{c})\| + \|(\overline{x}_i - \overline{c})\|} \right) \tag{2}$$

and $d_e(\overline{x}_j, \overline{c})$ denotes the Euclidean distance between $\overline{x}_j$ and $\overline{c}$. The value of $\overline{x}_i$, say $\overline{x}_j^*$, for which the quantity within brackets on the right hand side of Equation 2 attains its minimum value, is referred to as the symmetrical point of $\overline{x}_j$ with respect to $\overline{c}$. Note that if $\overline{x}_j^*$ is the same as the reflected point of $\overline{x}_j$ with respect to $\overline{c}$, then the numerator on the right hand side of Equation 2 will be equal to zero, and hence $d_s(\overline{x}_j, \overline{c}) = d_c(\overline{x}_j, \overline{c}) = 0$.

2.1. Limitations of the PS-distance

It is evident from Equation 1 that the PS-distance measure can be useful to detect clusters which have symmetrical shapes. But it will fail for datasets where clusters themselves are symmetrical with respect to some intermediate point. From equation 1, it can be noted that as $d_e(\overline{x}_j, \overline{c}) \approx d_e(\overline{x}_j^*, \overline{c})$, $d_c(\overline{x}_j, \overline{c}) \approx \frac{d_{symm}(\overline{x}_j, \overline{c})}{2}$, where $d_{symm}(\overline{x}_j, \overline{c}) = \|(\overline{x}_j - \overline{c}) + (\overline{x}_j^* - \overline{c})\|$. In effect, if a point $\overline{x}_j$ is almost equally symmetrical with respect to two centroids $\overline{c}_1$ and $\overline{c}_2$, it will be assigned to that cluster with respect to which it is more symmetric irrespective of the Euclidean distance between the cluster center and the particular point. This is intuitively unappealing. This is demonstrated in Figure 1. The centres of the

three clusters are denoted by $\overline{c}_1$, $\overline{c}_2$ and $\overline{c}_3$ respectively. Let us take the point $\overline{x}$. The symmetrical point of $\overline{x}$ with respect to $\overline{c}_1$ is $\overline{x}_1$ as it is the first nearest neighbor of the point $\overline{x}_1^* = (2 \times \overline{c}_1 - \overline{x})$. Let the Euclidean distance between $\overline{x}_1^*$ and $\overline{x}_1$ be d_1. So the symmetrical distance of $\overline{x}$ with respect to $\overline{c}_1$ is $d_c(\overline{x}, \overline{c}_1) = \frac{d_1}{d_e(\overline{x},\overline{c}_1)+d_e(\overline{x}_1,\overline{c}_1)} \times d_e(\overline{x}, \overline{c}_1)$. Similarly symmetrical point of $\overline{x}$ with respect to $\overline{c}_2$ is $\overline{x}_2$, and the symmetrical distance of $\overline{x}$ with respect to $\overline{c}_2$ becomes $d_c(\overline{x}, \overline{c}_2) = \frac{d_2}{d_e(\overline{x},\overline{c}_2)+d_e(\overline{x}_2,\overline{c}_2)} \times d_e(\overline{x}, \overline{c}_2)$. Let $d_2 < d_1$; Now as $d_e(\overline{x}, \overline{c}_2) \approx d_e(\overline{x}_2, \overline{c}_2)$ and $d_e(\overline{x}, \overline{c}_1) \approx d_e(\overline{x}_1, \overline{c}_2)$, therefore $d_s(\overline{x}, \overline{c}_1) \approx d_1/2$ and $d_s(\overline{x}, \overline{c}_2) \approx d_2/2$. Therefore $d_s(\overline{x}, \overline{c}_1) > d_s(\overline{x}, \overline{c}_2)$ and $\overline{x}$ is assigned to $\overline{c}_2$ even though $d_e(\overline{x}, \overline{c}_2) \gg d_e(\overline{x}, \overline{c}_1)$. This will happen for the other points also, finally resulting in merging of the three clusters. This is intuitively unappealing. From the above observations, it can be concluded that the PS-distance

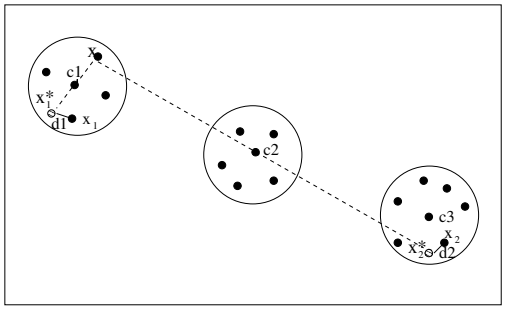


Figure 1. Example where point symmetry distance proposed by Su and Chou fails

measure [10] has two limitations:

Observation 1 : *The PS-distance measure lacks the Euclidean distance difference property.*

Here Euclidean distance difference (EDD) property is defined as follows:
Let $\overline{x}$ be a data point, $\overline{c}_1$ and $\overline{c}_2$ be two cluster centers, and θ be a distance measure. Let $\theta_1 = \theta(\overline{x}, \overline{c}_1)$, $\theta_2 = \theta(\overline{x}, \overline{c}_2)$, $d_{e_1} = d_e(\overline{x}, \overline{c}_1)$ and $d_{e_2} = d_e(\overline{x}, \overline{c}_2)$. Then θ is said to satisfy EDD property if for $\theta_1 \approx \theta_2$, point $\overline{x}$ is assigned to $\overline{c}_1$ if $d_{e1} < d_{e2}$, otherwise it is assigned to $\overline{c}_2$.

It is evident from Figure 1 and from the above discussion that in the PS-distance measure defined in Equation 1, there is no impact of the Euclidean distance. (Although a term $d_e(\overline{x}_j, \overline{c})$ is present, its effect gets almost neutralized by the denominator of the other term, $d_s(\overline{x}_j, \overline{c})$). It only measures the amount of symmetry of a particular point with respect to a particular cluster center. As a result a point might be assigned to a very far off cluster centre, if it happens to be marginally more symmetric with respect to it.

Observation 2: *The PSD measure leads to an unsatisfactory clustering result for the case of symmetrical interclusters.* If two clusters are symmetrical to each other with respect to a third cluster center, then these clusters are called "symmetrical interclusters".

In Figure 1 the first and the third clusters are "symmetrical interclusters" with respect to the middle one. As explained in the example, the three clusters get merged into one cluster since the PS-distance lacks the EDD property. This shows the limitation of the PS-distance in detecting symmetrical interclusters which is also experimentally demonstrated in this paper.

2.2. A New Definition of the Point Symmetry Distance

As discussed in Section 2, both the PS-based distances, d_s and d_c, will fail when the clusters themselves are symmetrical with respect to some intermediate point. It has been shown, in such cases the points are

assigned to the farthest cluster. In order to overcome this limitation, we describe a new PS distance [4], $d_{ps}(\overline{x},\overline{c})$ associated with point $\overline{x}$ with respect to a center $\overline{c}$. The proposed point symmetry distance is defined as follows: Let a point be $\overline{x}$. The symmetrical (reflected) point of $\overline{x}$ with respect to a particular centre $\overline{c}$ is $2\times\overline{c}-\overline{x}$. Let us denote this by $\overline{x}^*$. Let $knear$ unique nearest neighbors of $\overline{x}^*$ be at Euclidean distances of d_i, $i=1,2,\ldots knear$. Then

$$d_{ps}(\overline{x},\overline{c}) = d_{sym}(\overline{x},\overline{c})\times d_e(\overline{x},\overline{c}), \quad (3)$$

$$= \frac{\sum_{i=1}^{knear} d_i}{knear}\times d_e(\overline{x},\overline{c}), \quad (4)$$

where $d_e(\overline{x},\overline{c})$ is the Euclidean distance between the point $\overline{x}$ and $\overline{c}$. It can be seen from Equation 4 that $knear$ cannot be chosen equal to 1, since if $\overline{x}^*$ exists in the data set then $d_{ps}(\overline{x},\overline{c})=0$ and hence there will be no impact of the Euclidean distance. On the contrary, large values of $knear$ may not be suitable because it may overestimate the amount of symmetry of a point with respect to a particular cluster center. Here $knear$ is chosen equal to 2.
Note that $d_{ps}(\overline{x},\overline{c})$, which is a non-metric, is a way of measuring the amount of symmetry between a point and a cluster center, rather than the distance like any Minkowski distance.

The basic differences between the PS-distances in [18] and [10], and the proposed $d_{ps}(\overline{x},\overline{c})$ are as follows:

1. Instead of computing Euclidean distance between the original reflected point $\overline{x}^*=2\times\overline{c}-\overline{x}$ and its first nearest neighbor as in [18] and [10], here the average distance between $\overline{x}^*$ and its $knear$ unique nearest neighbors have been taken. Consequently this term will never be equal to 0, and the effect of $d_e(\overline{x},\overline{c})$, the Euclidean distance, will always be considered. Note that if only the nearest neighbor of $\overline{x}^*$ is considered and this happens to coincide with $\overline{x}^*$, then this term will be 0, making the distance insensitive to $d_e(\overline{x},\overline{c})$. But considering $knear$ nearest neighbors will reduce the problems discussed in Figure 1.

2. Considering the $knear$ nearest neighbors in the computation of d_{ps} makes the PS-distance more robust and noise resistant. From an intuitive point of view, if this term is less, then the likelihood that $\overline{x}$ is symmetrical with respect to $\overline{c}$ increases. This is not the case when only the first nearest neighbor is considered which could mislead the method in noisy situations.

3. In the PS-distance (in Equation 2) the denominator term is used to normalize the point symmetry distance so as to make it insensible to the Euclidean distance. But as shown earlier this will lead to lack of EDD property. As a result d_c can not identify symmetrical interclusters. Unlike this, in d_{ps} (Equation 3), no denominator term is incorporated to normalize d_{sym}.

Observation: The proposed d_{ps} measure will, in general, work well for symmetrical interclusters. Using $knear=2$, let the two nearest neighbors of the reflected point of $\overline{x}$ (in Figure 1) with respect to center $\overline{c}_1$ are at distances of d_1 and d_1^1 respectively. Then $d_{ps}(\overline{x},\overline{c}_1)=d_{sym}(\overline{x},\overline{c}_1)\times d_{e1}=\frac{d_1+d_1^1}{2}\times d_{e1}$, where d_{e1} is the Euclidean distance between $\overline{x}$ and $\overline{c}_1$. Let the two nearest neighbors of the reflected point of $\overline{x}$ with respect to center $\overline{c}_2$ be at distances of d_2 and d_2^1 respectively. Hence, $d_{ps}(\overline{x},\overline{c}_2)=d_{sym}(\overline{x},\overline{c}_2)\times d_{e2}=\frac{d_2+d_2^1}{2}\times d_{e2}$, where d_{e2} is the Euclidean distance between $\overline{x}$ and $\overline{c}_2$. Now in order to preserve the Euclidean distance difference property (EDD), i.e., to avoid merging of symmetrical

interclusters, $d_{ps}(\overline{x}, \overline{c}_1)$ should be less than $d_{ps}(\overline{x}, \overline{c}_2)$ even when $d_{sym}(\overline{x}, \overline{c}_1) \approx d_{sym}(\overline{x}, \overline{c}_2)$. Now,

$$\begin{aligned} d_{ps}(\overline{x}, \overline{c}_1) &< d_{ps}(\overline{x}, \overline{c}_2) \\ \implies \frac{d_1 + d_1^1}{2} \times d_{e1} &< \frac{d_2 + d_2^1}{2} \times d_{e2} \\ \implies \frac{d_{e1}}{d_{e2}} &< \frac{d_2 + d_2^1}{d_1 + d_1^1}. \end{aligned} \tag{5}$$

From Figure 1, it is evident that, $d_{e2} >> d_{e1}$, so $\frac{d_{e1}}{d_{e2}} << 1$. Thus even when $(d_2 + d_2^1) \approx (d_1 + d_1^1)$, the inequality in Equation 5 is satisfied. Therefore the proposed distance satisfies EDD property and avoids merging of symmetrical interclusters. The experimental results provided in [4] also support the fact that the proposed measure is robust even in the presence of symmetrical interclusters since it obeys EDD property.

It is evident that the symmetrical distance computation is very time consuming because it involves the computation of the nearest neighbors. Computation of $d_{ps}(\overline{x_i}, \overline{c})$ is of complexity $O(N)$. Hence for N points and K clusters, the complexity of assigning the points to the different clusters is $O(N^2K)$. In order to reduce the computational complexity, an approximate nearest neighbor search using the Kd-tree approach is adopted in this article.

2.3. Kd-tree Based Nearest Neighbor Computation

A K-dimensional tree, or Kd-tree is a space-partitioning data structure for organizing points in a K-dimensional space. A Kd-tree uses only those splitting planes which are perpendicular to one of the coordinate axes. In the nearest neighbor problem a set of data points in d-dimensional space is given. These points are preprocessed into a data structure, so that given any query point q, the nearest or generally k nearest points of p to q can be reported efficiently. ANN(Approximate Nearest Neighbor) is a library written in C++ [14], which supports data structures and algorithms for both exact and approximate nearest neighbor searching in arbitrarily high dimensions. In this article ANN is used to find d_is, where $i = 1, \ldots, knear$, in Equation 4 efficiently. The ANN library implements a number of different data structures, based on Kd-trees and box-decomposition trees, and employs a couple of different search strategies. ANN allows the user to specify a maximum approximation error bound, thus allowing the user to control the tradeoff between accuracy and running time.

The function performing the k-nearest neighbor search in ANN is given a query point *q*, a nonnegative integer *k*, an array of point indices, nn_{idx}, and an array of distances, $dists$. Both arrays are assumed to contain at least *k* elements. This procedure computes the *k* nearest neighbors of *q* in the point set, and stores the indices of the nearest neighbors in the array nn_{idx}. Optionally a real value $\epsilon \geq 0$ may be supplied. If so, then ith nearest neighbor is $(1 + \epsilon)$ approximation to the true ith nearest neighbor. That is, the true distance to this point may exceed the true distance to the real ith nearest neighbor of q by a factor of $(1 + \epsilon)$. If ϵ is omitted then the nearest neighbors will be computed exactly. For the purpose of this article, the exact nearest neighbor is computed; so the ϵ is set equal to 0 and $k = knear$, in this article it is $k = 2$. In order to find symmetric distance of a particular point $\overline{x}$ with respect to the centre $\overline{c}$, we have to find first $knear$ nearest neighbors of $\overline{x}^*$ which is equal to $2 \times \overline{c} - \overline{x}$. Therefore the query point q is set equal to $\overline{x}^*$. After getting the $knear$ nearest neighbors of $\overline{x}^*$ the symmetrical distance of $\overline{x}$ with respect to a centre $\overline{c}$ is calculated using equation 4.

3. Fuzzy-VGAPS Clustering: Fuzzy Variable String Length Genetic Point Symmetry Based Clustering Technique

In this section, the use of variable string length genetic algorithm using a newly developed point symmetry based distance is proposed for automatically evolving the near-optimal $K \times n$ nondegenerate fuzzy partition matrix U^*. The set $\mathcal{U}$ of all possible nondegenerate partition matrices is represented as $\mathcal{U} = \{U \in R^{K \times n} | \sum_{i=1}^{K} u_{ij} = 1,\ j = 1, \ldots n,\ 0 < \sum_{j=1}^{n} u_{ij} < n, \text{and } u_{ij} \in [0,1]\}$. Here we have considered the best partition to be the one that corresponds to the maximum value of the proposed *FSym*-index which is defined later. Here both the number of clusters as well as the appropriate fuzzy clustering of the data is evolved simultaneously using the search capability of genetic algorithms.

In GAs, the parameters of the search space are encoded in the form of strings (called *chromosomes*). A collection of such strings is called *population*. Initially a random population is created, which represents different points in the search space. An *objective/fitness* function is associated with each string that represents the degree of *goodness* of the solution encoded in the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like *crossover* and *mutation* are applied on these strings to yield a new population. The process of selection, crossover, and mutation continues for a fixed number of generations or till a termination condition is satisfied.

For the purpose of clustering, each chromosome encodes a possible partitioning of the data, the goodness of which is computed as a function of an appropriate cluster validity index. This index must be optimized in order to obtain the best partition. Since the number of clusters is considered to be variable, the string lengths of different chromosomes in the same population are allowed to vary. As a consequence, the crossover and mutation operators are suitably modified in order to tackle the concept of variable length chromosomes. The technique is described below in detail.

3.1. String Representation and Population Initialization

In Fuzzy-VGAPS clustering, the chromosomes are made up of real numbers which represent the coordinates of the centers of the partitions. If chromosome i encodes the centers of K_i clusters in d dimensional space then its length l_i is taken to be $d * K_i$. For example, in three dimensional space, the chromosome $<$ 12.3 1.4 5.6 22.1 0.01 10.2 0.0 5.3 15.3 13.2 10.2 7.5 $>$ encodes 4 cluster centers, (12.3, 1.4, 5.6), (22.1, 0.01, 10.2), (0.0, 5.3, 15.3) and (13.2, 10.2, 7.5). Each center is considered to be indivisible. Each string i in the population initially encodes the centers of a number, K_i, of clusters, such that $K_i = (rand() \bmod K^*) + 2$. Here, $rand()$ is a function returning an integer, and K^* is a soft estimate of the upper bound of the number of clusters. The number of clusters will therefore range from two to $K^* + 1$. The K_i centers encoded in a chromosome are randomly selected distinct points from the data set. The selected points are distributed randomly in the chromosome.

Thereafter five iterations of the *K*-means algorithm is executed with the set of centers encoded in each chromosome. The resultant centers are used to replace the centers in the corresponding chromosomes. This makes the centers separated initially.

3.2. Fitness Computation

This is composed of two steps. Firstly membership values of n points to different clusters are computed by using the newly developed point symmetry based distance d_{ps}. Next, the *FSym*-index is computed and used as a measure of the fitness of the chromosome.

3.2.1. Computing the Membership Values

For each point $\overline{x}_j, j = 1, 2, \ldots n$, the membership values to K different clusters are calculated in the following way. Find the cluster center nearest to $\overline{x_j}$ in the symmetrical sense. That is, we find the cluster center *k* that is nearest to the input pattern $\overline{x}_j$ using the minimum-value criterion:

$$k = Argmin_{i=1,\ldots K} d_{ps}(\overline{x}_j, \overline{c}_i) \tag{6}$$

where the point symmetry based distance $d_{ps}(\overline{x}_j, \overline{c}_i)$ is computed by Equation 4. Here, $\overline{c}_i$ denotes the center of the ith cluster. If the corresponding $d_{sym}(\overline{x}_j, \overline{c}_k)$ is smaller than a pre-specified parameter θ, then we update the membership u_{ij} using the following criterion:

$$u_{ij} = 1, \text{ if } i = k$$

$$u_{ij} = 0, \text{ if } i \neq k.$$

Otherwise, we update the membership u_{ij} by using following rule which corresponds to the normal Fuzzy C-Means [8] algorithm:

$$u_{ij} = \frac{1}{\sum_{i=1}^{K} (\frac{d_{ij}}{d_{kj}})^{\frac{2}{m-1}}} \tag{7}$$

where $m \in (1, \infty)$ is a weighting exponent called the fuzzifier. Here we have chosen $m = 2$. d_{ij} represents the Euclidean distance from a pattern $\overline{x}_j$ to the cluster center $\overline{c}_i$. The value of θ is kept equal to the maximum nearest neighbor distance among all the points in the data set. It may be noted that if a point is indeed symmetric with respect to some cluster centre then the symmetrical distance computed in the above way will be small, and can be bounded as follows. Let d_{NN}^{max} be the maximum nearest neighbor distance in the data set. That is $d_{NN}^{max} = \max_{i=1,\ldots N} d_{NN}(\overline{x}_i)$, where $d_{NN}(\overline{x}_i)$ is the nearest neighbor distance of $\overline{x}_i$. Assuming that reflected point of $\overline{x}$ with respect to the cluster centre $\overline{c}$ lies within the data space, it may be noted that $d_1 \leq \frac{d_{NN}^{max}}{2}$ and $d_2 \leq \frac{3 \times d_{NN}^{max}}{2}$ resulting in $\frac{d_1 + d_2}{2} \leq d_{NN}^{max}$. Ideally, a point $\overline{x}$ is exactly symmetrical with respect to some $\overline{c}$ if $d_1 = 0$. However considering the uncertainty of the location of a point as the sphere of radius d_{NN}^{max} around $\overline{x}$, we have kept the threshold θ equals to d_{NN}^{max}. Thus the computation of θ is automatic and does not require user intervention.

3.2.2. Updating the Centers

The centers encoded in a chromosome are updated using the following equation as in FCM

$$\overline{c}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \overline{x}_j}{\sum_{j=1}^{n} (u_{ij})^m}, \quad 1 \leq i \leq K.$$

3.2.3. Fitness Calculation

The fitness of a chromosome indicates the degree of goodness of the solution it represents. In this paper, we have proposed a fuzzy symmetry based cluster validity index, *FSym*-index which measures the goodness of the partition in terms of "symmetry" and separation present in the clusters. The fitness of a chromosome is computed using the *FSym*-index. Let K cluster centres be denoted by $\overline{c}_i$ where $1 \leq i \leq K$ and $U(X) = [u_{ij}]_{K \times n}$ is a partition matrix for the data. Then *FSym*-index is defined as follows:

$$FSym(K) = (\frac{1}{K} \times \frac{1}{\mathcal{E}_K} \times D_K), \tag{8}$$

where K is the number of clusters. Here,

$$\mathcal{E}_K = \sum_{i=1}^{K} E_i \tag{9}$$

such that

$$E_i = \sum_{j=1}^{n} (u_{ij} \times d_{ps}(\overline{x}_j, \overline{c}_i)) \tag{10}$$

and

$$D_K = max_{i,j=1}^{K} \|\overline{c}_i - \overline{c}_j\|. \tag{11}$$

D_K is the maximum Euclidean distance between two cluster centers among all centers. $d_{ps}(\overline{x_j}, \overline{c_i})$ is computed by Equation 4.

The objective is to maximize the *FSym*-index in order to obtain the actual number of clusters and to achieve proper clustering. As formulated in Equation 8, *FSym* is a composition of three factors, these are $1/K$, $1/\mathcal{E}_K$ and D_K. The first factor increases as K decreases; as *FSym* needs to be maximized for optimal clustering, so it will prefer to decrease the value of K. The second factor is the within cluster total symmetrical distance. For clusters which have good symmetrical structure, E_i value is less. This, in turn, indicates that formation of more number of clusters, which are symmetrical in shape, would be encouraged. Finally the third factor, D_K, measuring the maximum separation between a pair of clusters, increases with the value of K. As these three factors are complementary in nature, so they are expected to compete and balance each other critically for determining the proper partitioning.

The use of D_K, as the measure of separation, requires further elaboration. Instead of using the maximum separation between two clusters, several other alternatives could have been used. For example, if D_K was the sum of pairwise inter cluster distances in a K-cluster structure, then it would increase largely with increase in the value of K. This might lead to the formation of maximum possible number of clusters equal to the number of elements in the data set. If D_K was the average inter cluster distance then it would decrease at each step with K, instead of being increased. So, this will only leave us with the minimum possible number of clusters. The minimum distance between two clusters may be another choice for D_K. However, this measure would also decrease significantly with increase in the number of clusters. So this would lead to a structure where the loosely connected sub-structures remain as they were, where in fact a separation was expected. Thus maximum separability may not be attained. In contrast, if we consider the maximum inter cluster separation then we see that this tends to increase significantly until we reach the maximum separation among compact clusters and then it becomes almost constant.

The upper bound of this value, which is equal to the maximum separation between two points, is only attainable when we have two extreme data elements as two single element clusters. But the terminating condition is reached well before this situation. This is the reason why we try to improve the maximum distance between two maximally separated clusters.

The fitness function for chromosome j is defined as $FSym_j$ i.e., the $FSym$ index computed for the chromosome. The objective of GA is to maximize this fitness function.

3.3. Selection

Conventional proportional selection is applied on the population of strings. Here, a string receives a number of copies that is proportional to its fitness in the population. We have used roulette wheel strategy for implementing the proportional selection scheme.

3.4. Crossover

For the purpose of crossover, the cluster centers are considered to be indivisible, i.e., the crossover points can only lie in between two cluster centers. The crossover operation, applied stochastically, must ensure that information exchange takes place in such a way that both the offsprings encode the centers of at least two clusters. For this, the operator is defined as follows [13]: Let parent chromosomes P_1 and P_2 encode M_1 and M_2 cluster centers respectively. τ_1, the crossover point in P_1, is generated as τ_1=rand() mod M_1. Let τ_2 be the crossover point in P_2, and it may vary in between [LB(τ_2),UB(τ_2)], where LB(τ_2) and UB(τ_2) indicate the lower and upper bounds of the range of τ_2 respectively. LB(τ_2) and UB(τ_2) are given by

$$\mathrm{L}B(\tau_2) = \min[2, \max[0, 2 - (M_1 - \tau_1)]] \tag{12}$$

and

$$\mathrm{U}B(\tau_2) = [M_2 - \max[0, 2 - \tau_1]]. \tag{13}$$

Therefore τ_2 is given by

$$\begin{aligned} \tau_2 &= \mathrm{L}B(\tau_2) + rand()\mathrm{m}od(\mathrm{U}B(\tau_2) - \mathrm{L}B(\tau_2)) \ \text{ if}(UB(\tau_2) \geq LB(\tau_2)), \\ &= 0 \ \ \mathrm{o}therwise. \end{aligned}$$

It can be verified by some simple calculations that if the crossover points τ_1 and τ_2 are chosen according to the above rules, then none of the offspring generated would have less than two clusters.

Crossover probability is selected adaptively as in [17]. The expressions for crossover probabilities are computed as follows. Let f_{max} be the maximum fitness value of the current population, $\overline{f}$ be the average fitness value of the population and $f^{'}$ be the larger of the fitness values of the solutions to be crossed. Then the probability of crossover, μ_c, is calculated as:
$\mu_c = k_1 \times \frac{(f_{max} - f^{'})}{(f_{max} - \overline{f})}$, if $f^{'} > \overline{f}$,
$\mu_c = k_3$, if $f^{'} \leq \overline{f}$.
Here, as in [17], the values of k_1 and k_3 are kept equal to 1.0. Note that, when $f_{max} = \overline{f}$, then $f^{'} = f_{max}$ and μ_c will be equal to k_3. The value of μ_c is increased when the better of the two chromosomes to be crossed is itself quite poor. In contrast when it is a good solution, μ_c is low so as to reduce the likelihood of disrupting a good solution by crossover.

3.5. Mutation

Mutation is applied on each chromosome with probability μ_m. Mutation is of three types.

1. Each valid position (i.e., which is not '#') in a chromosome is mutated with probability μ_m in the following way. The valid position is replaced with a random variable drawn from a Laplacian distribution, $p(\epsilon) \propto e^{-\frac{|\epsilon-\mu|}{\delta}}$, where the scaling factor δ sets the magnitude of perturbation. Here μ is the value at the position which is to be perturbed. The scaling factor δ is chosen equal to 0.5. The old value at the position is replaced with the newly generated value.

2. One randomly generated valid position is removed and replaced by '#'.

3. One randomly chosen invalid position is replaced by randomly chosen point from the data set.

Any one of the above mentioned types of mutation is applied randomly on a particular chromosome if it is selected for mutation.

The mutation probability is also selected adaptively for each chromosome as in [17]. The expression for mutation probability, μ_m, is given below:
$\mu_m = k_2 \times \frac{(f_{max}-f)}{(f_{max}-\overline{f})}$ if $f > \overline{f}$,
$\mu_m = k_4$ if $f \leq \overline{f}$.
Here, values of k_2 and k_4 are kept equal to 0.5. This adaptive mutation helps GA to avoid getting stuck at local optimum. When GA converges to a local optimum, i.e., when $f_{max} - \overline{f}$ decreases, μ_c and μ_m both will be increased. As a result GA will come out of local optimum.

3.6. Termination

In this paper, we have executed the algorithm for a fixed number of generations. Moreover, the elitist model of GAs has been used, where the best string seen so far is stored in a location outside the population. The best string of the last generation provides the solution to the clustering problem.

4. Complexity Analysis of Fuzzy-VGAPS Clustering Technique

Below we have analyzed both the time and space complexities of the proposed Fuzzy-VGAPS clustering technique.

4.1. Time Complexity Analysis

1. As discussed above Kd-tree data structure has been used in order to find the nearest neighbor of a particular point. The construction of Kd-tree requires $O(NlogN)$ time and $O(N)$ space [1].

2. Initialization of GA needs $O(Popsize \times stringlength)$ time where $Popsize$ and $stringlength$ indicate the population size and the length of each chromosome in the GA, respectively. Note that $stringlength$ is O($K^* \times d$) where d is the dimension of the data set and K^* is the soft estimate of the upper bound of the number of clusters.

3. Fitness Computation is composed of 3 steps.

(a) In order to find membership values of each point to all cluster centers minimum symmetrical distance of that point with respect to all clusters have to be calculated. For this purpose the Kd-tree based nearest neighbor search is used. If the points are roughly uniformly distributed, then the expected case complexity is $O(c^d + logN)$, where c is a constant depending on dimension and the point distribution. This is $O(logN)$ if the dimension d is a constant [7]. Friedman et al. [11] also reported $O(logN)$ expected time for finding the nearest neighbor. So in order to find minimal symmetrical distance of a particular point, $O(K^*logN)$ time is needed.
Thus total complexity of computing membership values of N points to K^* clusters is $O(K^*NlogN)$.

(b) For updating the centres total complexity is $O(K^*)$.

(c) Total complexity for computing the fitness values is $O(N \times K^*)$.

So the fitness evaluation has total complexity=$O(Popsize \times K^*NlogN)$.

4. Selection step of the GA requires $O(Popsize \times stringlength)$ time.

5. Mutation and Crossover require $O(Popsize \times stringlength)$ time each.

Thus summing up the above complexities, total time complexity becomes $O(K^*NlogN \times Popsize)$ per generation. For maximum $Maxgen$ number of generations total complexity becomes $O(K^*NlogN \times Popsize \times Maxgen)$.

4.2. Space Complexity Analysis

The major space requirement of Fuzzy-VGAPS clustering is due to its population. Thus, the total space complexity of Fuzzy-VGAPS clustering is $O(Popsize \times Stringlength)$, i.e., $O(Popsize \times d \times K^*)$.

5. Results

This section provides a description of the image data set and the experimental results obtained after application of the above mentioned genetic fuzzy clustering technique for segmenting one artificially generated image and three remote sensing satellite images of the parts of the cities of Kolkata and Mumbai. The three satellite images are of size 512×512, i.e., the size of the data set to be clustered in all the images is 262144. For these multispectral satellite images, the feature vector is composed of the intensity values at different bands of the image. Note that in case of the satellite images, the processing is done in the intensity space of the pixels, where symmetry does exist. The parameters of the algorithm are as follows: population size is equal to 10, and the weighting coefficient m = 2.0. The algorithm is executed for a maximum of 10 generations. The crossover and mutation probabilities are chosen adaptively. The K^* is kept equal to 16. For the purpose of comparison, Fuzzy C-means (FCM) [8] clustering is also executed on these artificially generated and three real-life images.

5.1. Simulated Circle Image2 (SCI2)

The effectiveness of the proposed Fuzzy-VGAPS clustering is first shown in identifying small segments along with much larger ones from an image, where there is a significant overlap among them. For that

very purpose, an artificial image of size 256×256, shown in Figure 2(a), is generated. There are two small circles of radius 20 each, centered at (113,128) and (170,128), respectively. The pixels of these two small circles take gray values randomly in the range [160-170] and [65-75], respectively. The background pixels take values randomly in the range [70-166]. Figure 2(b) shows the segmented image using Fuzzy-VGAPS clustering optimizing *FSym*-index, when 3 clusters were automatically found. We have calculated *Minkowski Score* [5] (MS) of the segmented image provided by the Fuzzy-VGAPS clustering algorithm. This is a measure of the quality of a solution given the true clustering. Smaller value of MS indicates better segmentation. The MS value corresponding to the partitioning provided by Fuzzy-VGAPS clustering is 0.177026. FCM-clustering [8] is not able to find out the proper clustering from this data set. Figure 2(c) shows the corresponding segmented image for $K = 3$. MS value in this case is 0.806444. This shows that Fuzzy-VGAPS clustering is able to find out the proper partitioning from images having segments of widely varying sizes, where FCM-clustering is found to fail.

5.2. IRS Image of Kolkata

The data used here was acquired from Indian Remote Sensing Satellite (IRS-1A) using the *LISS-II* sensor that has a resolution of 36.25m×36.25m. The image is contained in four spectral bands namely, blue band of wavelength 0.45 - 0.52 μm, green band of wavelength 0.52 - 0.59 μm, red band of wavelength 0.62 - 0.68 μm, and near infra red band of wavelength 0.77 - 0.86 μm. Thus, here feature vector of each image pixel composed of four intensity values at different bands. The distribution of the pixels in the first 3 feature space of this image is shown in Figure 4. It can be easily seen from the Figure 4 that the entire data can be partitioned into several hyperspherical clusters where symmetry does exist.

Fig. 3 shows the Kolkata image in the near infra red band. Some characteristic regions in the image are the river *Hooghly* cutting across the middle of the image, several fisheries observed towards the lower-right portion, a township, *SaltLake*, to the upper-left hand side of the fisheries. This township is bounded on the top by a canal. Two parallel lines observed towards the upper right hand side of the image correspond to the airstrips in the *Dumdum* airport. Other than these, there are several water bodies, roads etc. in the image. From our ground knowledge, we know that the image has four clusters [13] and these four clusters correspond to the classes turbid water, pond water, concrete and open space.

The Fuzzy-VGAPS clustering technique automatically provides four clusters for this data (Fig. 5). It may be noted that the water class has been differentiated into turbid water (the *Hooghly*) and pond water (fisheries etc.) because of a difference in their spectral properties. The canal bounding *SaltLake* from the upper portion has also been correctly classified as pond water. The airstrips of *Dumdum* airport has again been classified correctly as belonging to the class concrete. Fig. 6 shows the Kolkata image partitioned in 4 clusters using FCM algorithm [8]. But the segmentation result is unsatisfactory from the human visualization judgement. As can be seen, the river *Hooghly* as well as the city region has been incorrectly classified as belonging to the same class. Therefore we can conclude that although some regions, viz., fisheries, canal bounding *SaltLake*, parts of the airstrip etc., have been correctly identified, a significant amount of confusion is evident in the FCM clustering result.

In order to validate the segmentation result obtained by Fuzzy-VGAPS clustering quantitatively, here two well-known Euclidean distance based cluster validity indices, namely $\mathcal{I}$ index [12] and XB-index [19] values are also computed. These are provided in Table 1. Smaller value of XB-index and larger value of $\mathcal{I}$ index correspond to good clustering. The values again show that the segmentation provided by Fuzzy-VGAPS clustering is much better than that of FCM-clustering.

5.3. IRS Image of Mumbai

As for the Kolkata image, the IRS image of Mumbai was also obtained using the LISS-II sensor. It is available in four bands, viz., blue, green, red and near infra-red. Fig. 7 shows the *IRS* image of a part of the city of Mumbai in the near infra red band. As can be seen, the elongated city area is surrounded on three sides by the Arabian sea. Towards the bottom right of the image, there are several islands, including the well known *Elephanta island*. The dockyard is situated on the south eastern part of Mumbai, which can be seen as a set of three finger like structure. This image has been classified into seven clusters [13].

The result of the application of the proposed clustering technique on the Mumbai image is shown in Fig. 8. The method automatically yielded six clusters. From the result it can be seen that the large water body of Arabian sea has been distinguished into one single class which is desirable. The islands, dockyard have mostly been correctly identified in the image. The results obtained, for both the Kolkata and Mumbai images are quite encouraging, since the technique has managed to automatically discriminate the classes without any sort of *a priori* knowledge about the data or the number of clusters.

Fig. 9 demonstrates the Mumbai image clustered using the FCM technique when $K = 6$ is given *a priori*. Again the result is unsatisfactory from the human visualization judgement. It is very difficult to clearly distinguish the Arabian sea from this segmented image (in Figure 9). A significant amount of confusion is evident in the FCM clustering result. In order to show it quantitatively, $\mathcal{I}$ index [12] and XB-index [19] values are again calculated. These are provided in Table 1. Larger value of $\mathcal{I}$ index and smaller value of XB-index correspond to good partitioning. The values again show that the segmentation provided by Fuzzy-VGAPS clustering is much better than that of FCM-clustering.

5.4. SPOT image of Kolkata

The French satellites SPOT (Systems Probataire d'Observation de la Terre) [16], launched in 1986 and 1990, carry two imaging devices that consist of a linear array of charge coupled device (CCD) detectors. Two imaging modes are possible, the multispectral and panchromatic modes. The 512×512 *SPOT* image of a part of the city of Kolkata is available in three bands in the multispectral mode. These bands are:
Band 1 - green band of wavelength 0.50 - 0.59 μm
Band 2 - red band of wavelength 0.61 - 0.68 μm
Band 3 - near infra red band of wavelength 0.79 - 0.89 μm.
Thus, here feature vector of each image pixel composed of three intensity values at different bands. The distribution of the pixels in the feature space of this image is shown in Figure 11. It can be easily seen from the Figure 11 that the entire data can be partitioned into several hyperspherical clusters where symmetry does exist.

Some important landcovers of Kolkata are present in the image. Most of these can be identified, from a knowledge about the area, more easily in the near infra-red band of the input image (Fig. 10). These are the following: The prominent black stretch across the figure is the river *Hooghly*. Portions of a bridge (referred to as the *second bridge*), which was under construction when the picture was taken, protrude into the *Hooghly* near its bend around the center of the image. There are two distinct black, elongated patches below the river, on the left side of the image. These are water bodies, the one to the left being *Garden Reach lake* and the one to the right being *Khidirpore dockyard*. Just to the right of these water bodies, there is a very thin line, starting from the right bank of the river, and going to the bottom edge of the picture. This is a canal called the *Talis nala*. Above the *Talis nala*, on the right side of the picture,

there is a triangular patch, the *race course*. On the top, right hand side of the image, there is a thin line, stretching from the top edge, and ending on the middle, left edge. This is the *Beleghata canal* with a road by its side. There are several roads on the right side of the image, near the middle and top portions. These are not very obvious from the images. A bridge cuts the river near the top of the image. This is referred to as the *first bridge*.

Fuzzy-VGAPS clustering technique automatically partitions this data in six different clusters (shown in Figure 12). As identified in [15] the above satellite image has seven classes namely, *turbid water, concrete, pure water, vegetation, habitation, open space* and *roads* (including bridges). The partitioning provided by the Fuzzy-VGAPS clustering is able to separate almost all the regions well. The Khidirpore dockyard and Talis nala have been identified properly by the proposed method (shown in Figure 12). The bridge is also correctly identified by the proposed algorithm. This again shows that the proposed Fuzzy-VGAPS clustering algorithm is able to detect clusters of widely varying sizes as long as they possess the 'symmetry' property. The segmentation result obtained by Fuzzy C-means algorithm on this image for $K = 6$ is shown in Figure 13. It can be seen from Figure 13 that FCM algorithm is not able to detect the bridge.

Again, in order to validate the segmentation result obtained by Fuzzy-VGAPS clustering quantitatively, two well-known Euclidean distance based cluster validity indices, namely $\mathcal{I}$ index [12] and XB-index [19] values are computed. These are provided in Table 1. As mentioned earlier, smaller value of XB-index and larger value of $\mathcal{I}$ index correspond to good clustering. The values again show that the segmentation provided by Fuzzy-VGAPS clustering is much better than that of FCM-clustering.

6. Discussion and Conclusions

In this article, classification of satellite images into different landcover regions is modeled as the task of clustering the pixels in the intensity space. Consequently an unsupervised genetic variable string length fuzzy clustering technique using the point symmetry based distance (Fuzzy-VGAPS clustering) has been used for classifying the image. The assignment of pixels to different landcover types are done based on this newly proposed point symmetry based distance rather than the Euclidean distance. In Fuzzy-VGAPS clustering, cluster centers are encoded in the chromosome whose value may vary. As a result, the proposed Fuzzy-VGAPS is capable of automatically detecting the number of segments present in an image. A newly developed fuzzy symmetry based cluster validity index (*FSym*-index) is used as a measure of the fitness of the chromosomes. The proposed method is able to detect any kind of segments present in an image irrespective of their shapes, sizes and convexities as long as they possess the point symmetry property. As a result, Fuzzy-VGAPS is also able to detect small overlapping clusters from a larger one where most of the existing clustering algorithms fail. Note that the well known FCM algorithm is a standard and popular fuzzy clustering technique when the number of clusters is known *a priori*. In case of the satellite images, the processing is done in the intensity space of the pixels, where symmetry does exist. Superiority of the proposed Fuzzy-VGAPS clustering technique over the widely used FCM algorithm is established for one artificially generated, two IRS images of Kolkata and Mumbai and one SPOT image of the part of the city of Kolkata both quantitatively and qualitatively.

The proposed way of segmenting the satellite multispectral images can be extended in many ways. Here the feature vector for a multispectral image is created by taking the intensity values at different bands of the image. But the algorithms taking spatial information in the feature vector are often found to

Table 1. $\mathcal{I}$ index and XB-index values of the segmented Mumbai and Kolkata satellite images provided by Fuzzy-VGAPS clustering and FCM-clustering

Index	Kolkata IRS		Mumbai IRS		Kolkata SPOT	
	Fuzzy-VGAPS	FCM	Fuzzy-VGAPS	FCM	Fuzzy-VGAPS	FCM
$\mathcal{I}$ index	27.836112	5.707293	254.783111	23.057593	120.314	76.2081
XB index	0.982644	23.666127	1.641962	4.674314	1.1521	12.2275

be more effective [3]. Future work includes incorporation of spatial information in the feature vector of each pixel for multispectral satellite image segmentation by Fuzzy-VGAPS clustering technique.

References

[1] Anderberg, M. R.: *Computational Geometry: Algorithms and Applications*, Springer, 2000.

[2] Attneave, F.: Symmetry Information and Memory for Pattern, *Am. J. Psychology*, **68**, 1995, 209–222.

[3] Bandyopadhyay, S.: Satellite Image Classification Using Genetically Guided Fuzzy Clustering with Spatial Information, *International Journal of Remote Sensing*, **26**(3), 2005, 579–593.

[4] Bandyopadhyay, S., Saha, S.: GAPS: A Clustering Method Using A New Point Symmetry Based Distance Measure, *Pattern Recog.*, Accepted (March, 2007), URL: http://dx.doi.org/10.1016/j.patcog.2007.03.026.

[5] Ben-Hur, A., Guyon, I.: *Detecting Stable Clusters using Principal Component Analysis in Methods in Molecular Biology*, Humana press, 2003.

[6] Bensaid, A. M., Hall, L. O., Bezdek, J. C., Clarke, L. P., Silbiger, M. L., Arrington, J. A., Murtagh, R. F.: Validity-Guided (Re)Clustering with Applications to Image Segmentation, *IEEE Transactions Fuzzy Systs.*, **4**(2), 1996, 112–123.

[7] Bentley, J. L., Weide, B. W., Yao, A. C.: Optimal expected-time algorithms for closest point problems, *ACM Transactions on Mathematical Software*, **6**(4), 1980, 563–580.

[8] Bezdek, J. C.: *Fuzzy Mathematics in Pattern Classification*, Ph.D. Thesis, 1973.

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

[10] Chou, C. H., Su, M. C., Lai, E.: Symmetry as A new Measure for Cluster Validity, in: *2nd WSEAS Int. Conf. on Scientific Computation and Soft Computing*, Crete, Greece, 2002, 209–213.

[11] Friedman, J. H., Bently, J. L., Finkel, R. A.: An Algorithm for finding best matches in logarithmic expected time, *ACM Transactions on Mathematical Software*, **3**(3), 1977, 209–226.

[12] Maulik, U., Bandyopadhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(12), 2002, 1650–1654.

[13] Maulik, U., Bandyopadhyay, S.: Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification, *IEEE Transactions Geoscience and Remote Sensing*, **41**(5), 2003, 1075– 1081.

[14] Mount, D. M., Arya, S.: ANN: A Library for Approximate Nearest Neighbor Searching, 2005, Http://www.cs.umd.edu/∼mount/ANN.

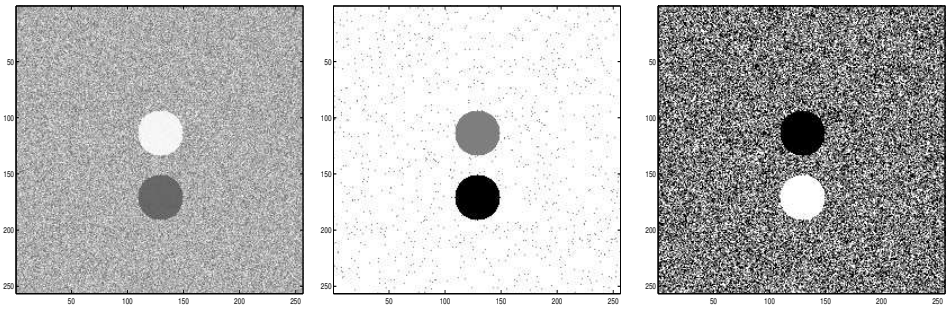
[15] Pal, S. K., Bandyopadhyay, S., Murthy, C. A.: Genetic Classifiers for Remotely Sensed Images : Comparison with Standard Methods, *International Journal of Remote Sensing*, **22**, 2001, 2545–2569.

[16] Richards, J. A.: *Remote Sensing Digital Image Analysis : An Introduction*, Springer-Verlag, New York, 1993.

[17] Srinivas, M., Patnaik, L.: Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms, *IEEE Transactions on Systems, Man and Cybernatics*, **24**(4), April, 1994, 656–667.

[18] Su, M.-C., Chou, C.-H.: A Modified Version of the K-means Algorithm with a Distance Based on Cluster Symmetry, *IEEE Transactions Pattern Analysis and Machine Intelligence*, **23**(6), 2001, 674–680.

[19] Xie, X. L., Beni, G.: A Validity Measure for Fuzzy Clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 1991, 841–847.



(a) (b) (c)

Figure 2. (a) SCI2; (b) Automatic segmented SCI2 obtained by Fuzzy-VGAPS clustering (c) Segmented SCI2 obtained by FCM-clustering for $K = 3$

Figure 3. IRS Image of Kolkata in the Near Infra Red Band with Histogram Equalization

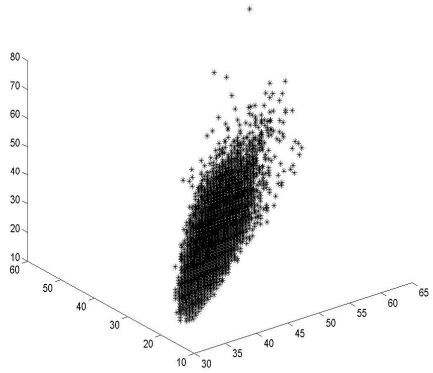


Figure 4. Data Distribution of IRS Image of Kolkata in the First 3 Feature Space

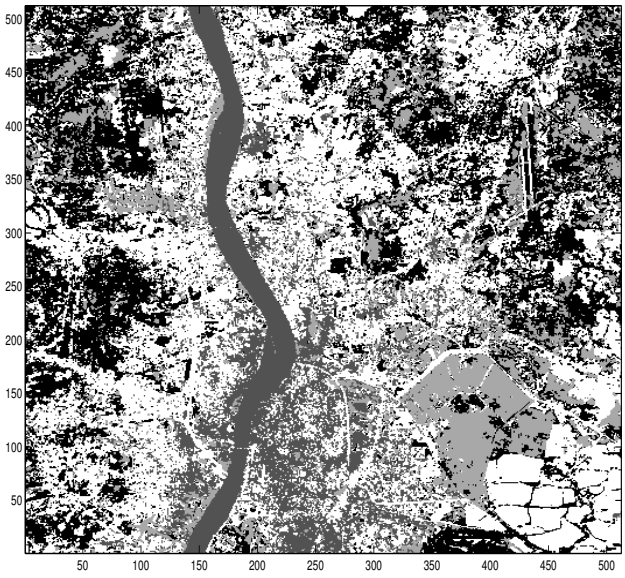


Figure 5. Clustered IRS Image of Kolkata Using Fuzzy-VGAPS Clustering

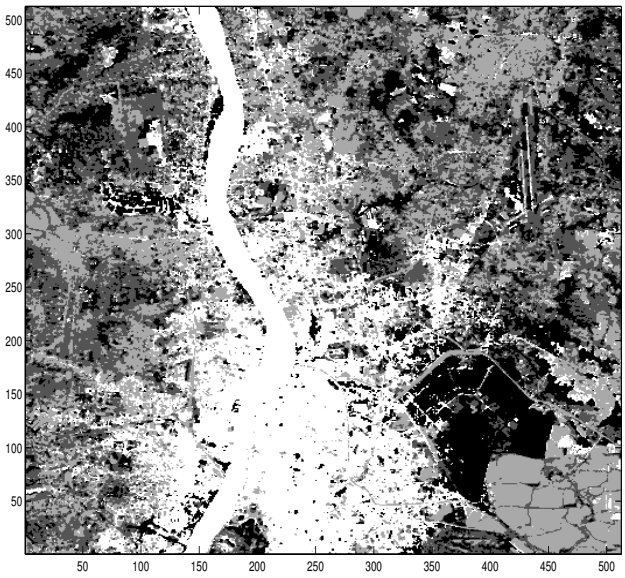


Figure 6. Clustered IRS Image of Kolkata Using FCM Clustering

Figure 7. IRS Image of Mumbai in the Near Infra Red Band with Histogram Equalization

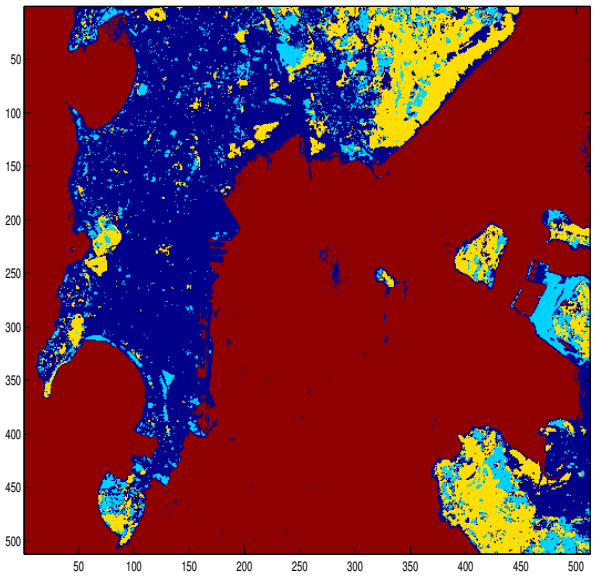


Figure 8. Clustered IRS Image of Mumbai Using Fuzzy-VGAPS Clustering

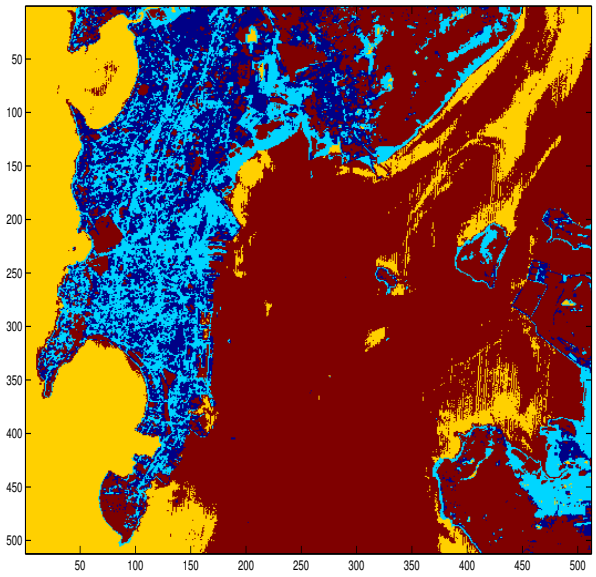


Figure 9. Clustered IRS Image of Mumbai Using FCM Clustering

Figure 10. SPOT Image of Kolkata in the Near Infra Red Band with Histogram Equalization

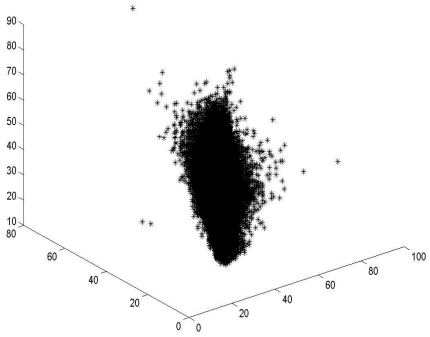


Figure 11. Data distribution of SPOT image of Kolkata in the Feature Space

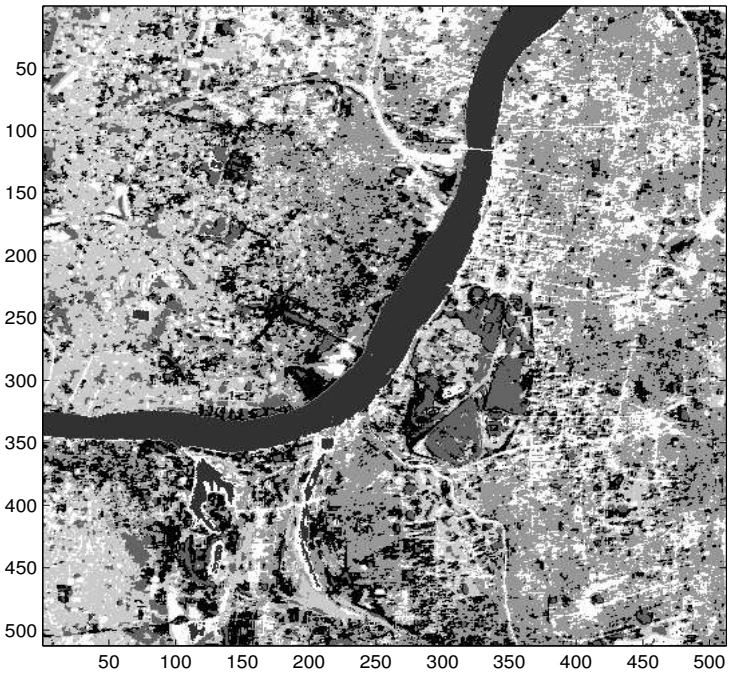


Figure 12. Clustered SPOT Image of Kolkata Using Fuzzy-VGAPS Clustering

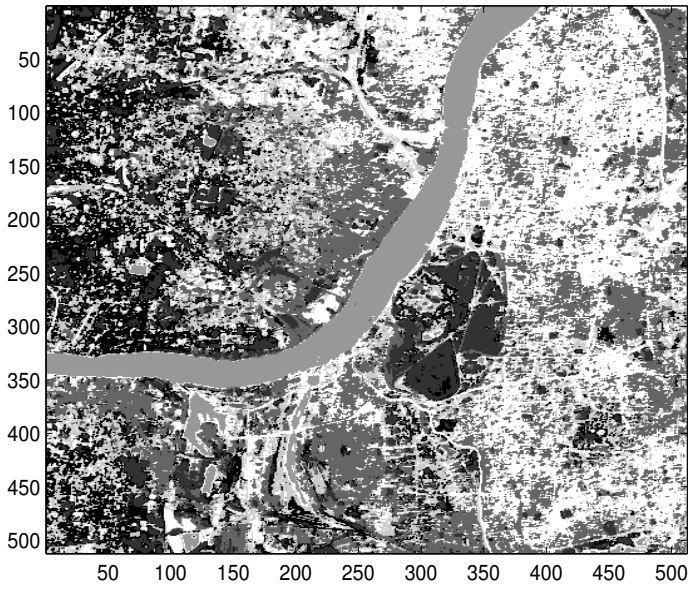


Figure 13. Clustered SPOT Image of Kolkata Using FCM Clustering