Sponsored Research (RESPOND) at SAC

Format for Final Completion Report under RESPOND Scheme

A. GENERAL INFORMATION

1. Title of Research Project : Optimal Cluster Detection Over High Resolution Satellite Imagery and Its application

2. Name & Designation of PI : Dr. D K Bhattacharyya, Professor, Dept of CSE,

   Name & address of the institution : Tezpur University, Napaam, Tezpur, Assam.

3. Other Investigator(s) with address(es) : Nil

4. Period of the Project : 2008-2010

5. Total grant approved by ISRO : Rs. 4.91 Lakhs

6. Total amount spent during the period of the project :

Amount approved By ISRO (Rupees)	Amount Spent (Rupees)	Remarks if any
4,91,000.00	3,38,013.00	Cheque received in June, 2009

7. Name and address of the granted University/Institution : Tezpur University, Dept of Computer Science & Engg.

   Napaam, Tezpur, Pin-784028, Assam, India.

8. Names and present addresses of Research Fellows who worked in the Project : N/A

9. Names and present addresses of Research Associates other research Staff recruited for the project : N/A

10. Names of Research staff (in 8 & 9 above) completing Ph.D under the scheme : N/A

]

11. Abstract of the Report

Extraction of hidden information from huge datasets is a challenging task in data mining. Clustering is the process of division of a data set into subsets or clusters, so that the similarity of points in each partition is as high as possible, while points in different partitions are dissimilar. From the aspect of geometry, clustering is the process of identifying dense regions from sparsely populated regions. Clustering is very effective in discovering hidden patterns of data sets and is therefore an important research topic.

A high dimensional satellite image is a remotely sensed image of the earth's surface which is a collection of huge amount of information in terms of number of pixel data. In a high dimensional satellite image, each pixel represents an area on the earth's surface. Multi-spectral images are the main type of images acquired by remote sensing. It is a technology originally developed for space-based imaging which can capture light from frequencies beyond the visible range of light, such as infrared.

This can allow extraction of additional information that the human eye fails to capture with its receptors for red, green and blue. A multi-spectral satellite image is a digital image of multiple bands where each band represents a particular wavelength of light. Segmentation or clustering a multi-spectral satellite image is a process of discovering a finite number of non-overlapping and meaning regions or clusters in an image data space which has been a complex problem for a long time. Remotely sensed satellite images mainly consists of objects (regions) such as vegetation, water bodies, concrete structures, open spaces, habitation etc. which are separated due to their different reflectance characteristics, leading to wide variety of clusters of different sizes, shapes and densities.

There are two fundamental properties of the pixel value:

(i) Discontinuity that uses discontinuities between gray level regions to detect isolated points and contours within an image, and

(ii) The similarity that uses decision criteria to separate clusters in an image based on the similarity of the pixel values.

Based on these two fundamental properties, several image segmentation methods have been developed. Clustering approaches are based on the second property. Due to the presence of huge amount of data in satellite images, a good clustering algorithm is of utmost need which can quickly and qualitatively detect clusters.

A good clustering technique for satellite images should have

(i) Minimal number of input parameters,

(ii) Detecting arbitrary shaped clusters, and

(iii) Good efficiency on large datasets.

There are several popular tools for automatic classification (supervised as well as unsupervised) of satellite images with lower or moderate resolution, however, in the context of high resolution images such as IRS P6 LISS IV, IKONOS or CARTOSTAT, with variable density coverage over large land area, most of these solutions exhibit poor performance. They have been found inadequate and inaccurate in the estimation of the clusters qualitatively.

In the proposed study, it is aimed to develop a robust clustering algorithm which can detect optimal clusters at any orientation of any shape over multi-spectral, multi-resolution satellite data. Major clustering techniques have been classified into partitional, hierarchical, density based, grid based and model based. Among these techniques, the density-based approach is famous for its capability of discovering arbitrary shaped clusters of good quality even in noisy datasets. Grid-based clustering approach is well known for its fast processing time especially for large and dense datasets. Here, it is aimed to develop a grid-density based optimal cluster detection algorithm, which can handle any multi-spectral, multi-resolution satellite dataset with better clustering quality. The method will exploit adaptive grid formation as well as triangulation technique to cluster the satellite data. It is also aimed to design and implement a distributed version of the proposed algorithm to handle large, high resolution satellite data.

In this report, we present three techniques for clustering medium and high resolution multi-spectral satellite image data that can handle voluminous high resolution satellite image data effectively and qualitatively, are:

(i) A Grid-density based Technique for Clustering Satellite Image (SATCLUS),

(ii) Clustering Satellite Data Using a Grid-Density based Technique (GDSDC), and

(iii) DisClus: A Distributed Clustering Technique over High Resolution Satellite Data.

We have also tested the above techniques over several large multi-spectral satellite image data with medium and high resolution and experimental results are reported to establish their capability in handling large satellite image data and for determining all the clusters present effectively and dynamically. Moreover, work is going on for developing a real-life CBIR (Content Based Image Retrieval) application to illustrate the usefulness of the clustering techniques.

B. SCIENTIFIC/TECHNICAL INFORMATION:

1. Introduction – Background and Objectives –

One of the major applications of mining high resolution images (such as those found in satellite image domain) is to find clusters of spatial objects which are close to each other. Most traditional clustering algorithms try to discover clusters of arbitrary densities, shapes and sizes. Very few clustering algorithms show preferable efficiency when clustering such datasets. This is also because small clusters with small number of points in a local area are possible to be missed by a global density threshold. Some clustering algorithms that can cluster on multi-density datasets are Chameleon [1], SNN [2] (shared nearest neighbor), the multi-stage density-isoline algorithm [3] and so on. Chameleon [1] can handle multi-density datasets, but for large and high resolution datasets, the time complexity is too high. SNN [2] algorithm can find clusters of varying shapes,

sizes and densities and can deal multi-density dataset. The disadvantage of SNN is that the degree of precision is low on the multi-density clustering and finding outliers. The multi-stage density-isoline algorithm [3] clusters datasets by the multi-stage way and the idea of density-isoline. The disadvantage of the algorithm is that each cluster cannot be separated efficiently. DGCL [4] is based on density-grid based clustering approach. But, since it uses a uniform density threshold it causes the low density clusters to be lost.

Also, since high resolution satellite images (such as IRS P6 LISS IV, IKONOS or CARTOSTAT, with variable density) can be characterized by high user interpretability, rich information content, sharpness, accuracy, high image clarity and integrity, it is essential to integrate these feature information together to improve classification accuracy. However, existing pixel-based classifiers have proven ineffective at extracting information from these new data sources. It demands for substantial involvement of human interpretation during classification, which causes the process to be expensive from both time and cost point of view. A robust optimal clustering technique is expected to relieve current classification/ clustering methods from those bottlenecks. The algorithm should be capable to detect multiple objects of any shape and orientation over multi-spectral, multi-resolution satellite data.

Also, to handle massive datasets with high dimensionality, a parallel implementation of the DBSCAN algorithm based on low cost distributed memory multi-computers can be found in the literature. Here, a centrally located dataset is spatially divided into nearly equal partitions with minimum overlap. Each such partition is sent to one of the processors for parallel clustering. The clustering results of the partitions are then collected by the central processor in an orderly manner and they are merged together to obtain the final clustering. The algorithm is scalable both in terms of speedup and scale-up and significantly reduces the computation time. In [5], a parallel version of the k-means algorithm was proposed based on shared nothing architecture. This algorithm was designed based on the Single Program Multiple Data (SPMD) model having several processors, each having its own local memory, connected together with a communication network. Another parallel version of DBSCAN, called PDBSCAN [6], also uses a shared-nothing architecture with multiple computers interconnected through a network. Here, as a data structure, the dR*-tree was introduced which is a distributed spatial index structure in which the data is spread among multiple computers and the indexes of the data are replicated on every computer. The master distributes the entire dataset to every slave. Each slave locally clusters the replicated data and the interference between computers is minimized due to local access of data. The slave-to-slave and master-to-slaves communication is done via message passing. The master manages the task of dynamic load balancing and merges the result produced by the slaves. PDBSCAN offers nearly linear speedup and has excellent scale-up and size-up behavior.

However, based on our survey, it has been strongly felt that-

- Almost all the existing clustering algorithms have proven ineffective at appropriate classification of high resolution satellite data sets containing high dimensional objects. They require input parameters, determination of which is very difficult. Moreover, the algorithms are highly sensitive to those parameters.
- Density based approach is most suitable for quality cluster detection in terms of shapes and in the context of variable density dataset.
- Grid based approach is suitable for fast processing of massive datasets.
- None of the techniques discussed above, is appropriate in detecting objects or clusters in different orientations optimally over multi-spectral, multi-resolution datasets.

Hence, this work is proposed.

Objectives :

Following objectives are aimed to achieve:

- to carry out an exhaustive study on the existing clustering techniques available for multi-spectral medium and high resolution satellite data; also to analyze their behaviour experimentally in the context of real-life data;
- to develop an enhanced clustering technique which can detect clusters of any shape qualitatively over those real-life satellite data;
- to develop a distributed version of the clustering technique to handle high resolution massive satellite data in parallel;
- to evaluate the performance of the proposed sequential as well as the distributed clustering technique(s) in comparison to its other counterparts;
- to establish the usefulness of the clustering technique, a real-life CBIR (Content Based Image Retrieval) application will also be considered;

3. Instrumentation – Development of equipment/hardware:

The estimated budget for the minor equipment is reported in Table 1. The estimated budget summary and approved budget is presented in Table 2 and Table 3.

Table 1: BUDGET ESTIMATES (MINOR EQUIPMENT)

Generic name of equipment & accessories with make & model	**Imported/ Indigenous**	**Estimated Cost (in Rs)**	**Spare time available for use by others (in %)**
Graphics WS with 21" LCD color monitor (IBM MPRO with intel CPU) and software	Indigenous	1,60,000	35
Laptop with Intel core 2 Duo processor with mobile Centrino technology	Imported	60,000	25
Laser Printer	Imported	40,000	25
Power Units (2 KVA online UPS)	Indigenous	30,000	25
Total		**2,90,000**	

Table 2: BUDGET ESTIMATES: SUMMARY

Items	1st Year	2nd Year	3rd Year	Total
A (i) Recurring consumables	5,000	10,000	15,000	30,000
(ii)Research Associate (01)	1,80,000	1,80,000	1,80,000	5,40,000
B Minor Items of Equipment				2,90,000
C Travel expenditure	20,000	25,000	30,000	75,000
Grand Total	**(A+B+C)**			**9,35,000**

Table 3: REVISED BUDGET APPROVED BY ISRO, DEPT OF SPACE (Rs. Lakhs)

S.No	Items	1st Year	2nd Year	Total
1	Salary (SRF-1)	1.80	1.80	3.60
2	Equipment (Workstation/Printer/Matlab Windows software/ power unit)	2.30	----	2.30
3	Consumables	0.10	0.10	0.20
4	Travel	0.25	0.25	0.50
	Sub Total	**4.45**	**2.15**	**6.60**
	Institute Overhead chgs.	0.45	0.22	0.67
	Annual Total	4.90	2.37	**7.27**
Grand Total Rs. 7.27 Lakhs (Rs. Seven lakhs twenty seven thousand only)				

4. Techniques and methods employed for the investigation including details of development of the software

During the stipulated period of time, three methods were developed (two work on stand-alone mode, whereas the third one operates on distributed mode) for optimal cluster detection over satellite imagery. Next we discuss each of these methods and their results in detail.

A. A Grid-density based Technique for Clustering Satellite Image (SATCLUS)

The aim of our clustering algorithm is to discover clusters over satellite image datasets. In SATCLUS, we regard each pixel data as a point in the space. The image data space is divided into grid cells and the grid cells whose HSI values with respect to neighboring cells (see Figure 1) are similar are merged. Once merging of grid cells according to HSI values terminates, a rough cluster is obtained. The border cells in a cluster are found and clustering proceeds at the pixel level using a partitioning algorithm to obtain the finer clustering of the dataset. We introduce some definitions, which are used in the proposed technique. The basis of the definitions has been taken from [7].

32	32	32	33	32	33	32	32	32	32
33	32	115	114	33	33	222	32	32	32
33	32	115	114	112	225	223	222	32	32
33	32	114	113	112	223	224	223	223	33
32	112	113	112	113	222	223	225	223	33
32	114	112	113	112	222	225	222	222	32
32	113	114	112	33	223	224	225	223	32
32	32	113	33	33	222	225	223	32	33
32	32	33	33	33	32	223	32	32	32
32	32	32	32	33	33	33	33	32	32

Fig. A.1. An example image with 5 × 5 grids and the hue values for corresponding pixels.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0

Fig. A.2. A 0-1 matrix obtained from the difference value w.r.t. to seed.

0	0	0	1/4	0
0	0	2/4	4/4	1/4
0	0	2/4	4/4	2/4
0	0	2/4	4/4	1/4
0	0	0	1/4	0

Fig. A.3. Population-object ratio of each grid cell.

Definition 1: *Density of a cell is the number of pixels within a particular grid cell.*

Definition 2: *Difference value of a pixel is the distance between the HSI values of a pixel w.r.t. the seed pixel, if it is within the range of certain threshold θ, then the difference value is considered 1 else 0.*

The distance may be any Euclidean or Manhattan distance. Euclidean distance between two pixels x_1 and x_2 is given as,

$$dist(x_1, x_2) = \sqrt{(h_1 - h_2)^2 + (s_1 - s_2)^2 + (i_1 - i_2)^2}$$

where h_j , s_j and i_j are the hue, saturation and intensity values of the j^{th} pixel, and

Mahalanobis distance of the pixel x to the mean μ of the distribution is given as

$$D = (x - \mu) \sum^{-1} (x - \mu)$$

where $\sum$ is the covariance matrix of the distribution.

However, the performance of SATCLUS is superior when we use Mahalanobis distance.

Definition 3: *Population count of each Grid cell can be defined as the number of ones in each grid cell.*

Definition 4: *Population-object ratio is defined as the ratio of the population count and cell density of a grid cell.*

$$Population\_object\_ratio = \frac{population\_count}{cell\_density}$$

Definition 5: *Confidence of a cell: If the difference of the population-object ratio of the current cell and one of its neighbors is greater than or equal to some threshold α (user input) then α is*

the confidence between them. Confidence plays an important role in cluster formation. For two cells p and q to be merged into the same cluster the following condition should be satisfied:

$$\alpha \leq | P_o(p) - P_o(q) |$$

- where Po represents the population-object ratio of that particular cell.

Definition 6: *A cell p is reachable from a cell q if p is a neighbor cell of q and cell p satisfies the confidence condition w.r.t. cell q.*

Definition 7: *A rough cluster is defined to be the set of points belonging to the set of reachable cells. A rough cluster C w.r.t. α is a non-empty subset satisfying the following condition, ; ∀p, q: if p ∈ C and q is reachable from p w.r.t. α, then q ∈ C, where p and q are cells.*

Definition 8: *A cell p is a border cell if it is part of a rough cluster C_i and at least one of its neighbors is part of another rough cluster C_j.*

Definition 9: *Noise is simply the set of points belonging to the cells not belonging to any of its clusters. Let $C_1, C_2, \ldots\ldots C_k$ be the clusters w.r.t. α, then*

$$Noise = \{no\_p \mid p \in n \times n, \forall i : no\_p \notin C_i\}$$

where no_p is the set of points in cell p and Ci (i = 1,... ,k).

A.1 Confidence of a cell

The confidence for a given set of cells reflects the general trend of that set. If the information of one cell is abnormal from the others it will not be included in the set. Similarly, each cell has a confidence with each of its neighbor cells. If the confidence of a current cell with one of its neighbor cell does not satisfy the confidence condition than that neighbor cell is not included into the local cluster area. On the contrary, if it satisfies the condition, then we treat the neighbor cell as a part of the local cluster area and merge the cell with the cluster area to form the rough cluster. This method has the ability to recognize the local clusters in the data space in the presence of embedded clusters also.

In light of the above definitions, following lemmas are trivial.

Lemma 1: *Let C_1 and C_2 be two rough clusters w.r.t α and let p be any cell in C_1 and q be any cell in C_2. Then, for a cell r, if r is reachable from p w.r.t α, then r is not reachable from q w.r.t α, i.e. r ∈C1 and r ∉C2.*

Lemma 2: *Let C be a set of clusters w.r.t α and let p be a cell corresponding to noise points. Then, ∀p: p is not reachable from any cell in C i.e. p ∉C.*

A.2 Procedure of SATCLUS

The algorithm starts with dividing the image space into $n \times n$ non-overlapping square grid cells, where n is a user input, and maps the image pixels to each cell. It then calculates the density of each cell. It then converts the RGB values of each pixel to its corresponding HSI values. The algorithm uses the cell information (density) of the grid structure and clusters the data points according to their surrounding cells. The clustering process is divided into two phases. In the first phase a rough clustering of the image space is obtained and the second phase deals with cluster smoothening for quality cluster identification.

A.2.1 Phase I: Rough Clustering

The maximum hue value is selected and an arbitrary pixel with this hue value becomes the seed for cluster initiation. An example is shown in Figure A.1 where the shaded pixel is the seed and each grid cell contains 4 pixels. The difference of the HSI values of the remaining pixels with this seed is calculated. If the difference value is less than some threshold θ, then that corresponding pixels difference value becomes 1 else 0. The image is thus converted into a 0-1 matrix as shown in Figure A.2. The population count of each grid cell is computed and the corresponding population-object-ratio calculated. The clustering process now starts with the grid-cell having the highest population-object ratio value as shown by the shaded grid cell in Figure A.3. The remaining cells are then clustered iteratively in order of their population-object ratio values, thereby building new clusters or merging with existing clusters. The cells adjacent to a cluster can only be merged. A neighbor search is conducted, starting at the highest population-object-ratio value grid-cell and inspecting adjacent cells. If a neighbor cell is found which satisfies the density confidence condition of a cell, then the neighbor cell is merged with the current cell and the search proceeds recursively with this neighbor cell. This search is similar to a graph traversal where the nodes represent the cells and an edge between two nodes exists if the respective cells are adjacent and satisfies the confidence condition of a cell.

The process of cell merging stops when no more cells satisfy the confidence condition of a cell. The process then starts the next cluster formation from the set of unclassified cells with the maximum hue pixel value. The process continues recursively merging neighboring cells that satisfy the confidence condition of a cell. This process of merging cells and selecting seeds is repeated until all the useful cells have been classified. The classified cells represent the rough clusters and finally the pixels receive the cluster id of the respective cells.

Figure A.4 shows the result of clustering an example image. The rough clusters obtained are grainy in nature. This is a drawback of a grid based algorithm. To obtain clusters with smooth borders, the border cells are detected and re-clustered using a partitioning based approach.

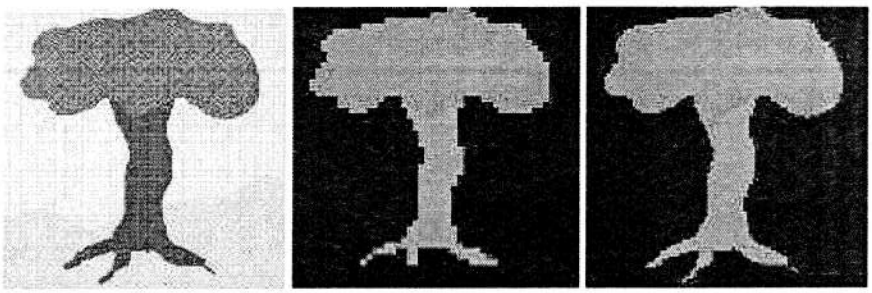


Fig. A.4. a. An example image with its grid structure; b. The four rough clusters formed and c. The final clusters

A.2.2 Phase II: Cluster Smoothening

In the cluster smoothening step, the number of clusters is already known which is given by the number of rough clusters so formed. The border cells are detected according to definition 8. Once the border cells have been found, clustering process starts at the pixel level. Suppose, k number of rough clusters have been obtained in the first phase of clustering. The pixels in only the border cells are checked for their re-assignment to clusters for quality cluster detection. The k rough clusters will each have one seed pixel. Now, let x be a pixel in a border cell. The distance of x with each of the k seeds is calculated and x will be assigned to that cluster with which it has the

least distance w.r.t. the seed. This process is repeated for all pixels belonging to border cells. The final clusters obtained for the image of Figure A.4a. are shown in Figure A.4c.
The execution of the rough clustering algorithm includes the following steps:

Phase I:
Input: Dataset D, Cell width and threshold θ.

(1) Create the grid structure.
(2) Compute the density of each cell.
(3) Convert the RGB values of each pixel into their HSI values.
(4) Identify the maximum hue value as the seed.
(5) Calculate each pixels difference value w.r.t. the seed.
(6) The population count of each grid cell is computed and the corresponding population-object ratio calculated.
(7) Traverse the neighbor cells starting from the grid-cell having the highest population-object ratio value.
(8) Merge the cells and assign *cluster_id*.
(9) Repeat steps 5 through 9 till all cells are classified.

The algorithm for the cluster smoothening is given below:

Phase II:
Input: q border cells, k seeds corresponding to the k rough clusters obtained from *Phase* I.

(1) Start with an arbitrary border pixel x.
(2) Find the distance of x to each of the k seeds.
(3) Assign x to the cluster to which it has minimum distance w.r.t. the seed.
(4) Repeat steps 1 to 3 till all border pixels have been reassigned.

A.3 Complexity Analysis

The partitioning of the dataset into $n \times n$ non-overlapping cells results in a complexity of $O(n \times n)$.

In *Phase* I, the expansion of the grid-cells to form clusters results in $O(p)$ time complexity, where p is the number of cells in a cluster so formed and $p \ll n \times n$ in the average case. If the number of clusters obtained is k then the overall time complexity for the clustering will be $O(k \times p)$. Therefore, total time complexity for the rough clustering will be $O(n \times n) + O(k \times p)$.

In *Phase* II, the identification of the q border cells require $O(q)$ times where $q \ll n \times n$. The assignment of r pixels to k clusters requires $O(k \times r)$ times, where r is the total number of pixels in q border cells. Therefore, total time complexity for *Phase* II will be $O(q) + O(k \times r)$.

Overall time complexity will be $O(n \times n) + O(k \times p) + O(i) + O(k \times r)$. Thus, $O(k \times r)$ dominates the overall time complexity.

A.4 Performance Evaluation

To implement and evaluate the technique in terms of quality of clustering, we used the following environment and datasets.

A.4.1 Environment Used

The algorithm is implemented using Java in Windows environment with Pentium IV processor with 1 GHz speed and 256 MB RAM. To smooth out any variation, each experiment was carried out for several times and the average result was taken.

A.4.A.2 Dataset Used

To test the performance of SATCLUS, we used several real-life satellite image datasets with medium as well as high resolution. A brief description of each dataset is given below:

a) *Dataset* I: This dataset is a Landsat MSS image as shown in figure 5a. Landsat Multi Spectral Scanner (MSS) was a sensor on-board Landsats 1 through 5 and acquired images of the earth nearly continuously from July 1972-Oct 1992, with an 18-day repeat cycle for Landsats 1 through 3 and a 16-day repeat cycle for Landsat 4 and 5. Landsat MSS image data consist of 4 spectral bands although the specific band designations change between Landsats 1-3 and Landsats 4-5. The resolution for all bands is of 79 m, and approximate size is 170 km North-South by 185 km East-West.

The clusters obtained by SATCLUS from the image data of figure A.5 are shown in figure A.6.

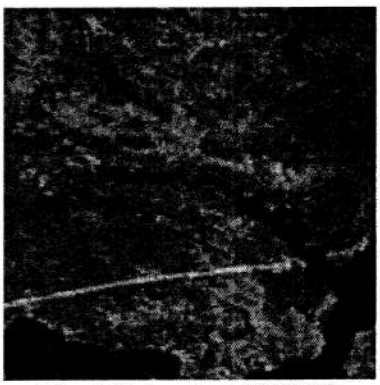
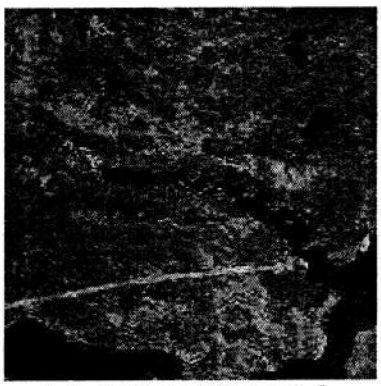


Fig. A.5. Landsat-MSS Fig.



A.6. Output of Figure A.5

b) *Dataset* II: This dataset was obtained from Indian Remote Sensing Satellite which is a circular sun synchronous satellite, rotating around the earth at the rate of 14 orbits/day, at an altitude of 904 km and a repeat cycle of 22 days. This satellite has two sensors LISS (Linear Imaging Self Scanner)-I and LISS-II. LISS-I has a spatial resolution of 72.5 m ×72.5 m while LISS-II has 36.25 m ×36.25 m. The IRS-1A image of Kolkata used in this work was taken by LISS-II sensor in the wavelength range 0.45μm - 0.86μm. The full spectrum range is decomposed into four spectral bands namely blue band of wavelength 0.45μm-0.52μm, green band of wavelength 0.52μm - 0.59μm, red band of wavelength 0.62μm - 0.68μm and near-infrared (NIR) band of wavelength 0.77μm - 0.86μm. Figure A.7 shows an area around Kolkata in the NIR band. There is a prominent black stretch across the image which is the river *Hoogly*. The prominent light patch at the bottom right

corner is the *Salt Lake stadium* and the black patches nearby are the fisheries. Two parallel lines at the upper right hand side of the image correspond to the airport runway in the *Dumdum* airport. Other than these there are several water bodies, roads, open spaces, etc. in the image.

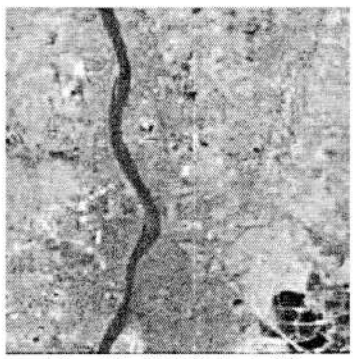
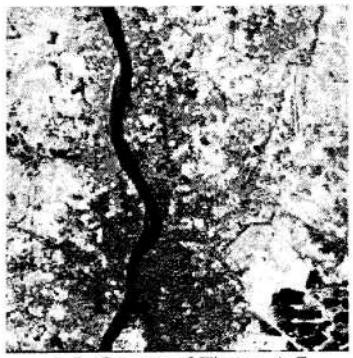


Fig. A.7. IRS Kolkata Fig.



A.8. Output of Figure A.7

SATCLUS automatically detects four clusters for this data as observed in figure A.8. From our ground knowledge, we can infer that these four clusters correspond to the classes: Water Bodies (black color), Habitation and City area (deep grey color), Open space (light grey color) and Vegetation (white color). The river Hoogly, stadium, fisheries, city area as well as the airport run- way is distinctly discernible in the output image. The predominance of city area on both sides of the river, particularly at the bottom part of the image is also correctly classified which corresponds to the central part of Kolkata city.

Figure A.9 shows the Calcutta image partitioned using FCM algorithm. From the figure, it can be noted that the river Hoogly and the city area has not been correctly classified. In fact, those have been classified as belonging to the same class. Another misclassification is that the whole Salt Lake city has been put into one class. Although some portions have been correctly identified such as canals, the Dumdum airport runway, fisheries, etc. still there is a significant amount of confusion in the FCM clustering result.

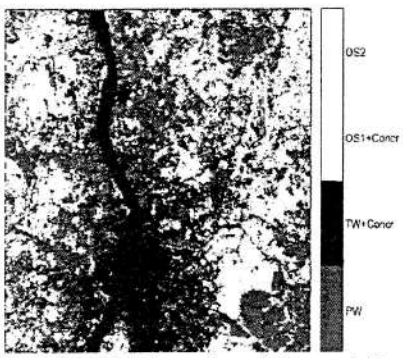


Fig. A.9. FCM clustering of Figure A.7

c) *Dataset* III: This dataset was acquired from the Cartosat-1 remote sensing satellite using the panchromatic (PAN) cameras that takes black and white stereoscopic pictures of the earth in the visible region of the electromagnetic spectrum as shown in figure 7a. The swath covered by these high resolution PAN cameras is 30 km and their spatial resolution is 2.5 m and wavelength of 0.5 – 0.85μm. Figure 7a shows the Cartosat-1 image of a plain built up area in Sonari, Assam. Some characteristic regions in the image are the river *Brahmaputra* shown in black color and spirally cutting across the middle of the image, roads, agricultural land, human settlements, etc. The SATCLUS clustering algorithm automatically detects the clusters (figure A.11) corresponding to river, road, agricultural land, water bodies and human settlements.

Fig. A.10. Cartosat-1 image of Sonari, Assam

Fig. A.11. Output of Figure A.10

d) *Dataset* IV *and* V: These two datasets show two different views of the Borapani area of the state of Meghalaya obtained from the IRS P6 LISS IV sensor that has a spatial resolution of 5.8m and the full spectral range is divided into four spectral: red, green blue and infrared. Figure 8a shows the IRS P6 LISS IV image of Borapani, Meghalaya. The characteristic regions in this image are the Deep water (Deep Blue color), Wetlands (light blue color), Vegetation (Red and Pink colors) and Open spaces (White color). Figure B.9a shows the IRS P6 LISS IV another image of Borapani, Meghalaya. The characteristic regions in this image are the water (dark color), Wetlands (light Yellowish-Green color), Vegetation (Violet colors) and Open spaces (light green color).

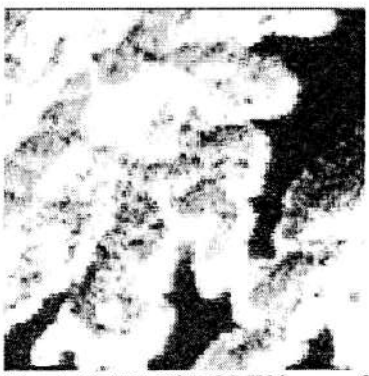


Fig. A.12. IRS P6 LISS IV image of Borapani, Meghalaya

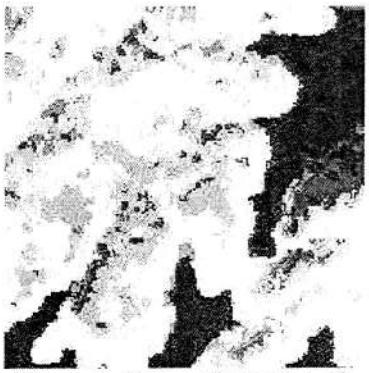


Fig. A.13. Output of Figure A.12

Executing SATCLUS on the images shown in figure A.12 resulted in the detection of the above four classes as shown in figure A.13.

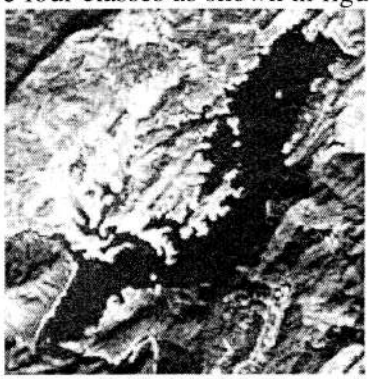


Fig. A.14. IRS P6 LISS IV image of Borapani - another view, Meghalaya

Fig. A.15. Output of Figure A.14 (False coloring is done for better visibility of the clusters.)

Figure A.14 shows another IRS P6 LISS IV image of Borapani, Meghalaya. The characteristic regions in this image are the water (dark color), Wetlands (light Yellowish-Green color), Vegetation (Violet colors) and Open spaces (light green color). SATCLUS clustered the image into five classes as shown in figure A.15. The resulting image classified the regions as deep water (dark blue), wetland (sky blue), vegetation (pink), open spaces (white) and pond water (black). It can be seen that the water body at the left hand top corner of the image has been detected which corresponds well to the ground information available.

From the experimental results given above, we can conclude that the technique is highly capable of detecting clusters qualitatively. The clustering results of the remote sensing images obtained above have also been evaluated quantitatively using an index β as in [8].

$$\beta = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n_i}(X_{ij}-\bar{X})^T(X_{ij}-\bar{X})}{\sum_{i=1}^{c}\sum_{j=1}^{n_i}(X_{ij}-\bar{X}_i)^T(X_{ij}-\bar{X}_i)}$$

Let n_i be the number of pixels in the i^{th} cluster ($i = 1, \cdots, c$). Let X_{ij} be the vector (of size 3×1) of the HSI values of the j^{th} pixel ($j = 1, \cdots, n_i$) for all the images in cluster i, and $\overline{X_i}$ the mean of n_i HSI values of the i^{th} cluster. Then, β is defined as [8]: where n is the size of the image and $\overline{X}$ is the mean HSI value of the image. It may be noted that X_{ij}, $\overline{X}$, and $\overline{X_i}$ are all 3×1 vectors. The above measure is the ratio of the total variation and within-cluster variation and is widely used for feature selection and cluster analysis [9].

Table A.1: Comparison of β value and CPU time for different clustering algorithms

Method	k-means	Astrahan's	Mitra's	SATCLUS
beta	5.30	7.02	9.88	17.82
CPU time (in hrs)	0.11	0.71	0.75	0.08

For a given image and c (number of clusters) value, the higher the homogeneity within the segmented regions, the higher the β value. The proposed method has the highest β as can be seen in Table A.1.

B. Clustering Satellite Data Using a Grid-Density based Technique

The aim of this clustering algorithm is to handle mixed pixels usually present in the border regions of the clusters efficiently which leads to the discovery of quality clusters over satellite image datasets. In GDSDC, we regard each pixel data as a point in the space. The image data space is divided into equal sized non-overlapping grid cells of width w_{grid} which is an input parameter. The grid cells whose HSI values with respect to neighboring cells (see Figure B.1) are similar are merged. Once merging of grid cells according to HSI values terminates, a rough cluster is obtained. The border cells in a cluster are found and clustering proceeds at the pixel level using a fuzzy approach which helps in smoothening of the rough clusters and handling the mixed pixels. We introduce some definitions, which are used in the proposed technique.

The distance measure may be either Euclidean or Manhattan distance. Euclidean distance between two pixels x_1 and x_2 is given as,

$$dist(x_1, x_2) = \sqrt{(h_1 - h_2)^2 + (s_1 - s_2)^2 + (i_1 - i_2)^2} \quad (1)$$

where h_j , s_j and i_j are the hue, saturation and intensity values of the j^{th} pixel, and

Mahalanobis distance of the pixel x to the mean μ of the distribution is given as

$$D = (x - \mu) \sum^{-1} (x - \mu)$$

where $\sum$ is the covariance matrix of the distribution.

However, the performance of SATCLUS is found to be superior when we use Mahalanobis distance.

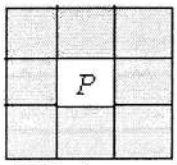


Fig. B.1. The white cell P is the current cell and all the gray cells are its neighbors

The basis of the definitions which are used in GDSDC has been taken from [7].

Definition 1. *Density of a cell is the number of pixels within a particular grid cell of width* w_{grid}.

Definition 2. *Difference value of a pixel w.r.t. the seed pixel is the distance between the HSI values of both the pixels, if it is within the range of certain threshold θ, then the difference value is considered 1 else 0.*

Definition 3. *Population count of each Grid cell can be defined as the number of ones in each grid cell.*

Definition 4. *Population-object ratio is defined as the ratio of the population count and cell density of a grid cell.*

$$Population\_object\_ratio = \frac{population\_count}{cell\_density} \quad (2)$$

Definition 5. *Confidence of a cell: If the difference of the population-object ratio of the current cell and one of its neighbors is greater than or equal to some threshold α (user input) then α is the confidence between them. Confidence plays an important role in cluster formation. For two cells p and q to be merged into the same cluster the following condition should be satisfied:*

$$\alpha \leq | P_o(p) - P_o(q) |$$

where P_o represents the population-object ratio of that particular cell.

Definition 6. *The α-neighborhood of a cell n_i in the dataset G_s is denoted by $N_\theta(n_i)$ is defined by,* $N_\alpha(n_i) = \{n_i \in G_s \mid \text{dist}(n_i, n_j) \leq \alpha\}$

where G_s is the set of grid cells and $n_j \in G_s$.

Definition 7. *A cell p is reachable from a cell q if p is a neighbor cell of q and cell p satisfies the confidence condition w.r.t. cell q.*

Definition 8. *A cluster is defined to be the set of points belonging to the set of reachable cells. A cluster C w.r.t. α is a non-empty subset of cells satisfying the following condition, $\forall p, q$: if $p \in C$ and q is reachable from p w.r.t. α, then $q \in C$, where p and q are cells.*

Definition 9. *A cell p is a border cell if it is part of a cluster C_i and at least one of its neighbors is part of another cluster C_j.*

Definition 10. *Let C be a set of clusters over the dataset G_s w.r.t. α. A noise cell is defined as a cell which is not reachable from any of the cells belonging to any of the clusters, $C_i \in C$. In other words,*

$$Noise = \{n_j \in G_s \mid \forall j: n_j \notin C_i \mid\}$$

Also, a cell n_i is said to be a noise cell if it does not satisfy the confidence condition with any other cell $n_j \in G_s$ and $| N_\alpha (n_i) | = \varphi$.

B.1 confidence of a cell

The confidence for a given set of cells reflects the general trend of that set. If the information of one cell is abnormal from the others it will not be included in the set. Similarly, each cell has a confidence with each of its neighbor cells. If the confidence of a current cell with one of its neighbor cell does not satisfy the confidence condition than that neighbor cell is not included into the local cluster area. On the contrary, if it satisfies the condition, then we treat the neighbor cell as a part of the local cluster area and merge the cell with the cluster area to form the rough cluster. This method has the ability to recognize the local clusters in the data space in the presence of embedded clusters also.

In light of the above definitions, following lemmas are trivial.

Lemma 1. *Let C_1 and C_2 be two clusters w.r.t α and let p be any cell in C_1 and q be any cell in C_2. Then, for a cell r, if r is reachable from p w.r.t α, then r is not reachable from q w.r.t α, i.e. if $r \in C_1$, $r \notin C_2$.*

Lemma 2. *Let C be a set of clusters w.r.t α and let n_i be a cell corresponding to noise points. Then, $\forall n_i$: n_i is not reachable from any cell in C i.e. $n_i \notin C$.*

B.2 Procedure of GDSDC

The algorithm starts with dividing the image space into $n \times n$ non-overlapping grid cells, where n is a user input, and maps the image pixels to each cell. It then calculates the density of each cell and converts the RGB values of each pixel to its corresponding HSI values. The algorithm uses the cell information (density) of the grid structure and clusters the data points according to their surrounding cells.

The clustering process is divided into two phases. In the first phase a rough clustering of the image space is obtained and the second phase deals with the smoothening of the cluster borders using a fuzzy membership function for quality cluster identification.

B.2.1 Phase I: Rough Clustering

Phase I start with selecting the maximum hue value and an arbitrary unclassified pixel with this hue value becomes the seed for cluster initiation. An example is shown in Figure A.1 where the shaded pixel is the seed and each grid cell contains 4 pixels. The difference of the HSI values of the remaining pixels with this seed is calculated. If the difference value of a particular pixel is less than some threshold θ, then the value of that corresponding pixels becomes 1 else 0. The image is thus converted into a 0-1 matrix as shown in Figure A.2. The population count of each grid cell is computed and the corresponding population-object ratio calculated. The clustering process now starts with the grid-cell having the highest population-object ratio value as shown by the shaded grid cell in Figure A.3. The remaining cells are then clustered iteratively in order of their population-object ratio values, thereby building new clusters or merging with existing clusters. The cells adjacent to a cluster can only be merged. Starting with the grid cell having highest population-object-ratio, a neighbor search is conducted. If a neighbor cell is found which satisfies the density confidence condition of a cell, then the neighbor cell is merged with the current cell and the search proceeds recursively with this neighbor cell. This search is similar to a graph traversal where the nodes represent the cells and an edge between two nodes exists if the respective cells are adjacent and satisfies the confidence condition of a cell.

The process of cell merging stops when no more cells satisfy the confidence condition of a cell. The process then starts the formation of the next cluster from the set of unclassified cells initiating with the pixel having the highest hue value. The process continues recursively merging neighboring cells that satisfy the confidence condition of a cell. This process of merging cells and selecting seeds is repeated until all the useful cells have been classified. The classified cells represent the rough clusters and finally the pixels belonging to those cells are assigned the respective cluster id.

Figure B.1 shows the result of clustering an example image. The rough clusters obtained are grainy in nature. This is a drawback of a grid based algorithm. To obtain clusters with smooth

borders, the border cells are detected and re-clustered using a fuzzy membership based approach [10].

4.2.2 Phase II: Cluster Smoothening using Fuzzy Membership Function

In this phase, the output of the previous phase i.e. rough clusters are accepted as input. It initially detects the border cells according to Definition 9. Once the border cells have been found, clustering process starts at the pixel level using a fuzzy membership function [10] as described below. The border pixels for Figure B.1. (a) is shown in Figure B.1. (c). Suppose, k number of rough clusters have been obtained in the first phase of clustering. The pixels only in the border cells are checked for their re-assignment to clusters for improving the quality of clusters. Each of the k rough clusters detected will have one seed pixel. Now, let x_j be a pixel in a border cell p of cluster C_i. Since the 8-neighborhood of the cell p may have other clusters C_j where $j = 1, 2,...n_k$, then these clusters will be the neighbor clusters of cluster C_i and x_j can be assigned to any of the neighboring clusters. The membership of x_j with each of the clusters present in the 8-neighborhood of cell p is calculated using equation 3 and x_j will be assigned to that cluster for which its membership has the least value w.r.t. the seed. This process is repeated for all pixels belonging to border cells. Fuzzy membership function [10], is given by,

$$u_{c_i x_j} = \frac{1}{\sum_{l=1}^{nk} \left[\frac{d(c_i, x_j)}{d(c_l, x_j)} \right]^{\frac{2}{m-1}}}$$

where x_j is a border pixel, n_k is the number of clusters detected in the neighborhood of the cell p to which x_j belongs, u is the fuzzy membership matrix such that u_{ij} □ [0, 1] is the membership degree of x_j to cluster i. $c = \{c_1, c_2,..., c_{nk}\}$ is the set of rough clusters found in the neighborhood of cell p. c_i is the current cluster for which the membership of x_j is to be determined and c_l are the clusters present in the 8-neighborhood of cell p, d is a distance measure (Euclidean or Manhattan distance) between a seed of a rough cluster and a border pixel. The factor m is called fuzziness and is usually equal to 2 [10].

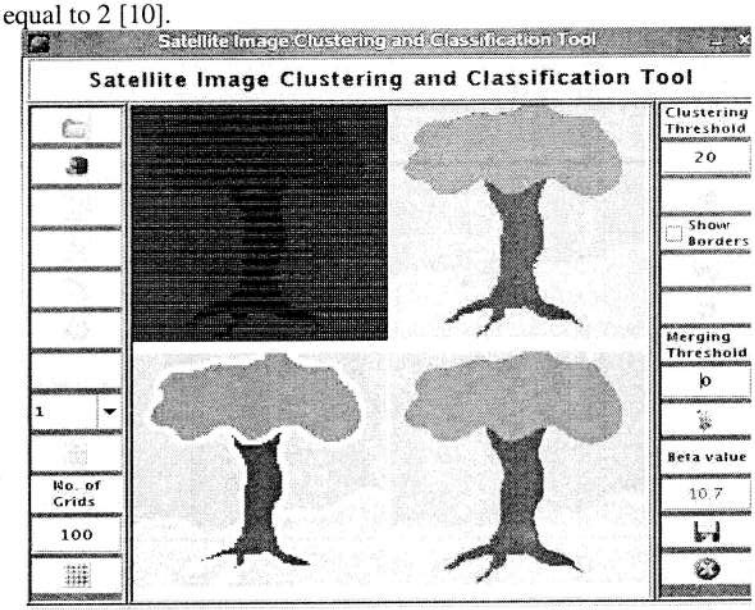


Fig. B.2. GDSDC Interface: a) original image with grid structure; b) Four rough clusters (top-right); c) Border pixels in white (bottom-left) and d) Final clusters(Output of *phase* II)

The membership for each of the border pixels is computed and assigned to that cluster for which it has the lowest value. It can be easily seen that Figure B.1. (d) is more qualitative than Figure B.1. (b). The execution of the rough clustering algorithm includes the following steps:

Phase I:

1) Create the grid structure.
2) Compute the density of each cell.
3) Convert the RGB values of each pixel into their HSI values.
4) Identify the maximum hue value as the seed.
5) Calculate each pixels difference value w.r.t. the seed.
6) The population count of each grid cell is computed and the corresponding population-object ratio calculated.
7) Traverse the neighbor cells starting from the grid cell having the highest population-object ratio value.
8) Merge the cells and assign cluster id.
9) Repeat steps 5 through 9 till all cells are classified.

The algorithm for the cluster smoothening is given below:

Phase II:
Input: q border cells, k seeds corresponding to the k rough clusters obtained from *phase* I.

1) Start with an arbitrary border pixel x_j .
2) Find the membership of x_j to each of the k seeds.
3) Assign x_j to the cluster which minimizes the fuzzy membership function.
4) Repeat steps 1 to 3 till all border pixels have been reassigned.

4.3 Complexity Analysis

Phase I of this technique executes in two steps: partitioning the dataset and formation of the rough clusters. The partitioning of the dataset into $n \times n$ non-overlapping grid cells results in a complexity of $O(n \times n)$. During the step of rough cluster formation, the expansion of the grid-cells to form clusters results in $O(p)$ time complexity, where p is the number of cells in a cluster so formed in the average case. If the number of clusters obtained is k then the overall time complexity for rough clustering will be $O(k \times p)$. Since, $O(n \times n) \approx O(k \times p)$, therefore, total time complexity for this *phase* will be $O(n \times n)$.

In *Phase* II, the identification of the q border cells require $O(q)$ times where $q \square (n \times n)$. If r be the total number of pixels in q border cells, then assignment of r pixels to k clusters requires $O(k \times r)$ times. As, $O(q) \square O(k \times r)$, therefore, total time complexity for *phase* II will be $O(k \times r)$. Hence, the overall time complexity will be $O(n \times n) + O(k \times r)$.

4.B.4 Performance Evaluation

To implement and evaluate the technique in terms of quality of clustering, we used the environment and datasets already discussed in the previous technique along with some more satellite images. Experimental results of GDSDC are reported below:

The clustering results of the remote sensing images obtained above have also been evaluated quantitatively using above mentioned β index [11].

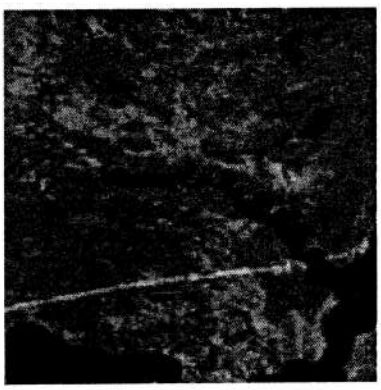
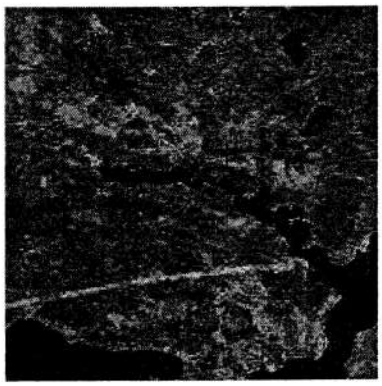
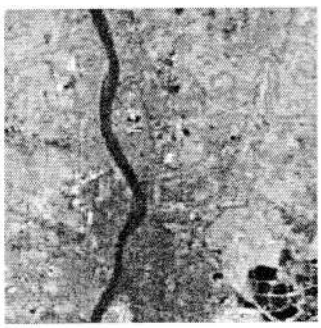
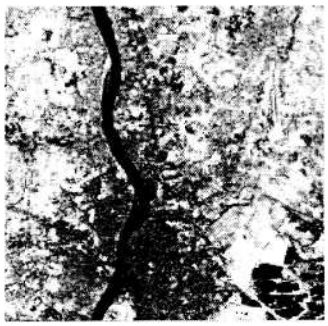


Fig. B.3. a) Landsat-MSS



b) GDSDC Result





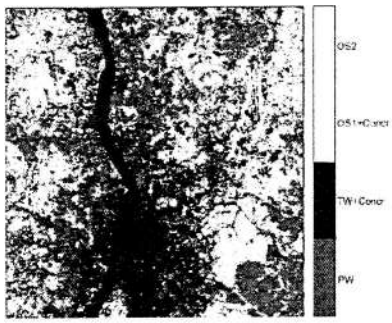


Fig. B.4. a) IRS Kolkata b) GDSDC Result c) FCM Result

Fig. B.5. a) Cartosat-1 of Sonari b) GDSDC Result

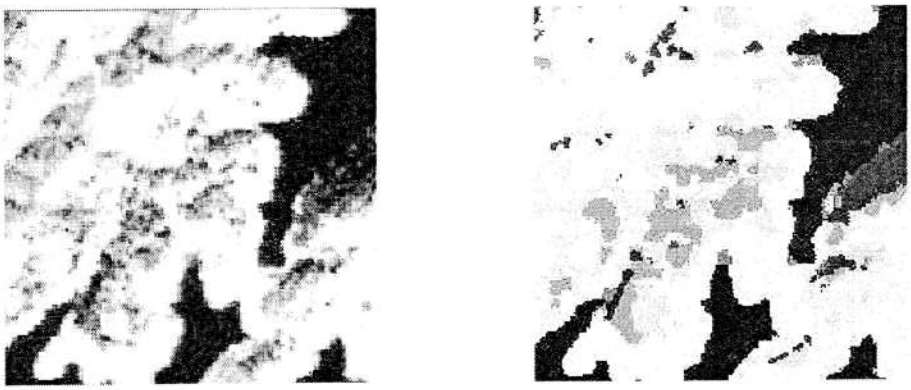


Fig. B.6. a) IRS of Borapani b) GDSDC Result

Table B.1 Comparison of beta value and CPU time for different clustering algorithms

Method	[19]	[6]	[7]	[27]	GDSDC
beta	5.30	7.02	9.88	3.84578	12.63
CPU time (in hrs)	0.11	0.71	0.75	unknown	0.09

C. DisClus: A Distributed Clustering Technique over High Resolution Satellite Data

The method is divided into three phases. In the first phase, the whole dataset is partitioned into regions with marked overlappings at an initiator node and sent to each of the nodes available for clustering. The second phase is executed in each of the nodes. In this phase, the clustering of the data on the partitions is performed using SCLUST or GDSDC at each node. Then, in the third phase, the nodes transmit the cluster results back to the initiator node where the results are merged to get the final result.

C.1 The Proposed DisClus

The proposed architecture adopts a shared-nothing architecture. It considers a system having k-nodes where the entire dataset D is located in any of the nodes (say node 1). Node 1 (also called initiator node) executes a fast partitioning technique to generate the k initial overlapped partitions. The partitions are then distributed among k nodes (including it) for cluster detection. Finally, the local cluster results are received from the nodes at the initiator node (node 1) and a merger module is used to obtain the final cluster results.

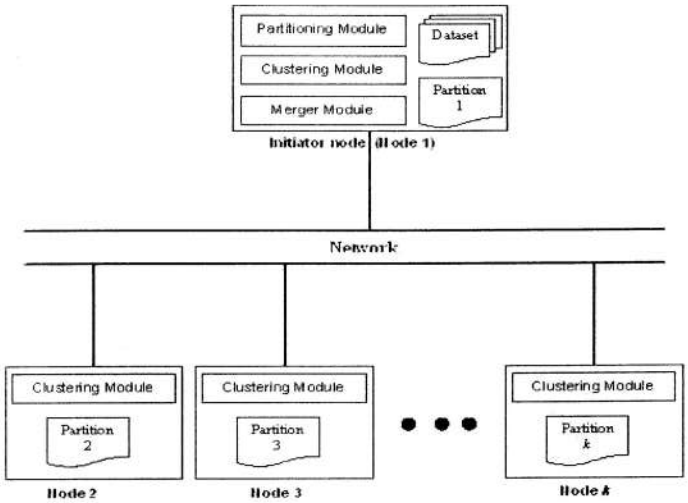


Fig. C.1.Overal architecture of DisClus

Basically the technique works in three phases and the output of each phase becomes the input of the subsequent phase.

***Phase* I**

Phase I of the architecture is executed in one of the nodes (node 1 or initiator node). The dataset is spatially divided into equal sized grids based on the (x, y) coordinates. Initially, the data space is divided into $n \times n$ non-overlapping square grid cells, where n is a user input, and maps the data points to each cell. It then calculates the density of each cell. The grid mesh is then partitioned with some overlap between adjacent partitions and distributed over k available computers (nodes). No subsequent movement of data between partitions will take place.

Assuming, the grid mesh D contains the set of $n \times n$ objects say, $D = O_0, O_1, O_2, ..., O(n \times n)-1$. Suppose, $O_j = (a_{0j}, a_{1j}, a_{2j}, ..., a_{(d-1)j}; d_n)$ represents a grid cell with d real-valued attributes a_i, $i=0,..,d-1$ and density d_n. The i^{th} attribute value of object O_j is drawn from domain a_j. If there are k clients, the grid mesh D is partitioned into k subsets $D_0, D_1, ..., D_{(k-1)}$ ordered in sequence. We refer the clients by the corresponding partition D_j that it receives for processing.

$$D = D_0 \cup D_1 \cup D_2 \cup ..... \cup D_{k-1}$$

$$D_i \cap D_j \neq \phi \text{ for } i, j = 0, ..., (n \times n) - 1$$

$$= \phi \text{ for } | i, j \geq 2, i = 0, ..., (n \times n) - 1, j = 0, ..., (n \times n) - 1$$

The partially overlapped partitions are shown in Fig. C.2. An overlap of one grid cell occurs between two adjacent partitions. The overlapped regions are much smaller than the partitions. The grid cells in the overlapped regions are locally clustered in both the adjacent partitions. Thus they provide the information for merging together the local clustering results of two adjacent partitions. The overlapped width should be at least one cell width because adjacent cells are neighbors. The grid mesh D is partitioned in this manner based on the values of a selected

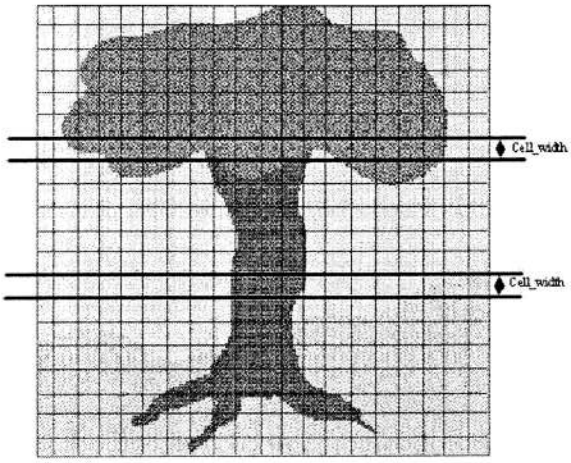


Fig. C.2. Overlapped spatial partitioning of a data set

attribute of the data objects say a_s. The values of a_s have a range of [min_a_s, max_a_s]. We need to select (k + 1) constants in the given range. Let c_i, i = 1, 2, ..., k+1 represent the constants such that $c_i = min\_a_s$, c_{k+1} = max as and $c_i < c_{i+1}$. Therefore the overlapped region can be represented as,

$$D_i = \{\exists j(O_j \in D) \mid c_i - cell\_width \leq a_{sj} \leq c_{i+1}\}, i = 2, \cdots, k-1$$

$$D_i = \{\exists j(O_j \in D) \mid c_i \leq a_{sj} \leq c_{i+1} + cell\_width\}, i = 1$$
$$D_i = \{\exists j(O_j \in D) \mid c_i - cell\_width \leq a_{sj} \leq c_{i+1}\}, i = k$$

Load Balancing

Partition D_i is sent to processor P_i, i=1, 2,....., k for concurrent clustering. Since no data movement takes place after the partitions are created, care has been taken so that each processor eceives nearly equal number of data objects for processing. We assume that the processing speeds of the processors are equal. The range of as is divided into intervals of width of one cell width and the frequencies of data in each interval is counted. The load balancing is done in a manner similar to [13] which ensures that each partition gets number of objects nearly equal to N/k.

***Phase* II**

This *phase* is executed in each of the k nodes and plays the actual role of clustering. Here, each node executes SCLUST/GDSDC over the partition of data received from the *initiator node*. For the partition D_i in node i, the grid cells in it will be assigned *cluster_id* according to the clusters formed in that partition.

The cluster expansion based on grid cells reduces the computation time as data points are not considered for cluster expansion, only the density information of each cell is used. Moreover, the information of the marked cells used during merging process of *phase* III saves the cost of merging to a great extent. Finally, *phase* II transmits the cluster objects to the initiator node along with the *cluster_ids*.

***Phase* III**

In *phase* III, the cluster results received from the k nodes undergo a simplified, yet faster merging procedure to obtain the final clusters. Since the *phase* II process in a node may yield more than one cluster, so there are always possibilities for merging during *phase* III operation. The Merger module works as follows:

1. Join the partitions received from the k nodes according to their overlapping marked cells.
2. Consider the marked grid cells (overlapping cells) of the candidate partitions.
3. If any of the marked grid cells is identified by different *cluster_id*s by different partitions (say *l*, *m*), then assign the smallest of the *cluster_id*s (say *l*) to that cell.
4. Assign all those cells having the same *cluster_id* as that of the replaced *cluster_id* (*m*) with *l*.

C.2 Complexity Analysis

Since the proposed technique is executed in three *phases* and each *phase* is independent of each other, therefore, the total complexity will be the sum of the complexities due to these three *phases*. The first *phase*, divides the dataset of N points into $n \times n$ cells which are partitioned into k overlapped partitions with a total of $((k-1)\times n)$ overlapped cells. Therefore, this *phase* results in a complexity of $O(n \times n)$ approximately, where $n << N$. After partitioning, $(N+(k-1)\times t)$ points will be transmitted to k nodes, where t is the average number of points present in an overlapped region, results in a complexity of $O((N+(k-1)\times t)$. The second phase results in a complexity of $O(((n\times n)/k + n)+(c\times b))$ [1], where c is the number of clusters detected locally and b is the number of border points obtained in a partition in a node. The clustered points are retransmitted to the *initiator node* with a transmission cost of $O((N + (k-1)\times t)$. The third *phase* is responsible for merging of the clusters resulting in atmost $O(N + k\times t)$ time. Thus, the overall time complexity of DisClus will be $O(n\times n)+(N+(k-1)\times t)+O(((n\times n)/k+n)+(c\times b))+(N+(k-1)\times t)+((N + k \times t))$.

C.3 Performance Evaluation

The algorithm was tested over several real-life satellite images as shown in Table C.1. The Dataset 1 is shown in figure C.1 (a). The clusters obtained from the image of Fig. C.1 are shown in Fig. C.1b. Figure C.2 (a) shows Dataset 2. There is a prominent black stretch across the image which is the river *Hoogly*. The prominent light patch at the bottom right corner is the *Salt Lake stadium* and the black patches nearby are the fisheries. Two parallel lines at the upper right hand side of the image correspond to the airport runway in the *Dumdum* airport. Other than these there are several water bodies, roads, open spaces, etc. in the image.

Table C.1. Results of the clustering algorithm over several multi-spectral satellite images

Serial No.	Dataset	Spectral Bands	Resolution	Clusters Detected
1	Landsat MSS	4	79 m	4 clusters
2	IRS LISS II image of Kolkata, West Bengal	4	36.25 m	4 clusters
3	Cartosat-I image of Sonari, Assam	4	2.5 m	5 clusters
4	IRS P6 LISS IV image of Borapani, Meghalaya	4	5.8 m	5 clusters

DisClus automatically detects four clusters for this data as observed in Fig. C.3b. From our ground knowledge, we can infer that these four clusters correspond to the classes: Water Bodies (black color), Habitation and City area (deep grey color), Open space (light grey color) and Vegetation (white color). The river *Hoogly*, stadium, fisheries, city area as well as the airport runway is distinctly discernible in the output image. The predominance of city area on both sides of the river, particularly at the bottom part of the image is also correctly classified which corresponds to the central part of Kolkata city. Figure C.2c shows the Kolkata image partitioned using FCM algorithm. It can be seen from the result that the river Hoogly and the city area has not been properly classified. These two objects have been classified as belonging to the same class. Similarly, the whole Salt Lake city as a whole has been put into one class. However, some portions such as canals, the Dumdum airport runway, fisheries, etc. have been classified properly.

Figure C.1(a) Landsat-MSS

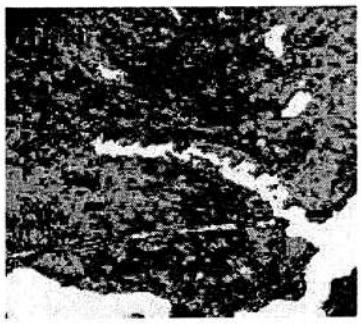


Figure C.1 (b) DisClus output of C.1(a) (False coloring is done for better visibility of the clusters.)

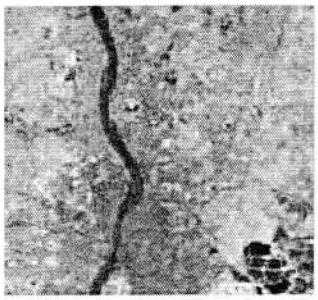


Figure C.2(a) IRS-Kolkata

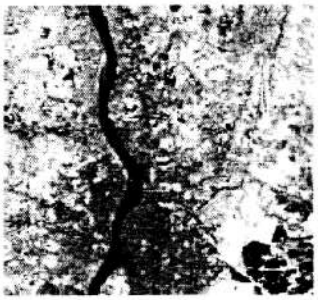


Figure C.2 (b) DisClus output

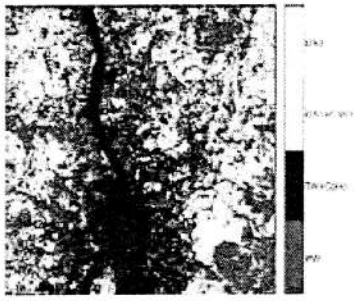


Figure C.2 (c) FCM output

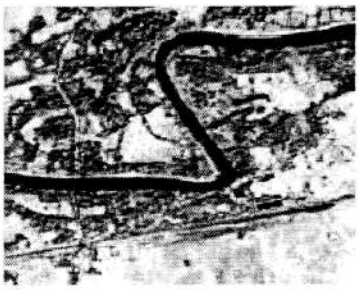


Figure C.3 (a) Cartosat-1 of Sonari

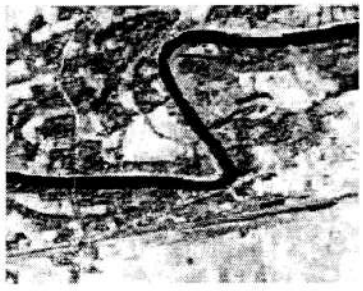


Figure C.3 (b) DisClus output of C.3 (a)

The fourth dataset used in this work shows a view of the Borapani area of the state of Meghalaya (Figure C.4(a)). The characteristic regions in this image are the Deep water (Deep Blue color), Wetlands (light blue color), Vegetation (Red and Pink colors) and Open spaces (White color). *DisClus* clustered the image into five classes as shown in Fig. C.4b. The resulting image classified the regions as: deep water (dark blue), wetland (sky blue), vegetation (pink), open spaces (white) and pond water (black). It can be seen that the water body at the left hand top corner of the image has been detected which corresponds well to the ground information available.

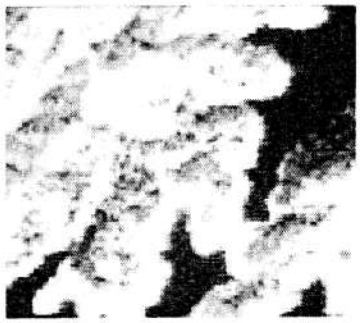


Figure C.4(a) IRS of Borapani

Figure C.4 (b) DisClus output of C.4(a)

From the experimental results given above, we can conclude that the technique is highly capable of detecting clusters of all shapes.

C.4 Performance and Scalability Analysis

In our implementation environment, there is no inter-processor communication except for a single processor communicating with each of the remaining processors. Each processor has the same specification i.e. PIV with 1 GHz speed and 128 MB RAM and the processors are connected through Ethernet LAN of speed 10/100 Mbps. To smooth out any variation, each experiment was carried out for several times and the average results were taken and each reported data point is to be interpreted as an average over five measurements. Our algorithm was implemented in JAVA in Linux environment.

Parallel Execution Time: T(k), the parallel execution time of a program is the time required to run the program on k nodes in parallel. When k = 1, T(1) denotes the sequential run time of a program on a single processor. Our experiments reveal that the execution time decreases significantly with the increase in the number of processors.

Speedup: Speedup is a measure of relative performance between a multiprocessor system and a single processor system, defined as, $S(k) = T(1)/T(k)$. On experimenting it has been found that the speedup factor increases with the increase in the number of processors. Fig. 12 (a) shows relative speedup curves for two data sets with points $N = 8\times10^5$ and 6×10^5. The number of dimensions and the number of clusters are fixed for both the data sets. The solid line represents "ideal" linear relative speedup. For each data set, a dotted line connects observed relative speedups, which is a sub-linear type.

Efficiency: The efficiency of a program on n processors, i.e. E(k) is defined as the ratio of speedup achieved and the number of processors used to achieve it. E(k) = S(k)/k = T(1)/k.T(k). In case of the proposed technique we observed that too many processors do not ensure the efficiency.

Scale-up: The scale-up characteristic of the proposed technique has been found to be satisfactory with the increase in the number of processors as can be seen from Fig. 12 (b). Here the number of data points is scaled by the number of processors while dimensions and number of clusters are held constant.

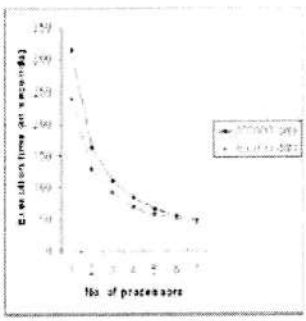


Figure C.5(a) Execution Time

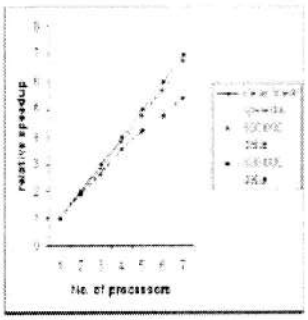


Figure C.5 (b) Relative Speedup Curve

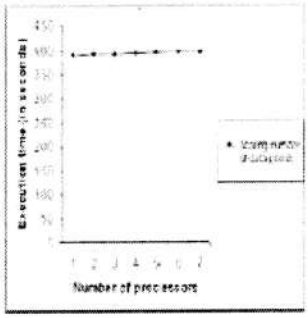


Figure C.5 (c) Scale-up Curve

While comparing to DBSCAN, OPTICS, EnDBSCAN, GDLC and Density-isoline, the proposed DisClus requires the number of grid cells, i.e. n and threshold α as input parameters. However, from our experiments it has been observed that the threshold α does not vary significantly with different datasets. DisClus can effectively detect clusters of all shapes, sizes and densities.

C.4 Comparison of DisClus with its counterparts

The results of clustering the remote sensing images have been evaluated quantitatively using the β index [8]. The proposed method has the highest β value as can be seen in Table C.1.

Table C.1. Comparison of beta value and CPU time for different clustering algorithms

Method	k-means	Astrahan's	Mitra's	DisClus
beta	5.30	7.02	9.88	17.82

DisClus is also compared w.r.t. some of the pioneering algorithms in terms of time complexity and the result is shown in Table C.2.

Table C.2. Comparison of DisClus with its counterparts

Algorithms	*No. of parameters*	Structure	*Complexity* (Approximate)
k-means	1 (N)	Spherical	$O(N)$
FCM	1 (N)	Non-Convex	$O(N)$
DBSCAN	2 ($MinPts$, ε)	Arbitrary	$O(NlogN)$ using $R*$ tree
OPTICS	3 ($MinPts$, ε, ε')	Arbitrary	$O(NlogN)$ using $R*$ tree
DisClus	2 (n, α)	Arbitrary	$O(N)$

D. Application of the Techniques in the Retrieval of Relevant Objects

Image Retrieval based on contents is a fast growing technology and is the field that deals with application of computer vision for retrieval of images from digital libraries. In this section, we report a retrieval technique built based on the clustering results reported in the preceding sections.

The proposed technique employs an index, of seven dimensional feature vectors < *Cluster_Id, r, g, b,* $\varphi_1, \varphi_2, \varphi_3$>, to represent the cluster profiles of the objects present in an image, termed as *spatial cluster index* as well as the object of interest, termed as *spatial object index*. Here, *r, g, b* are mean of the red, green and blue components of the pixels of a cluster and $\varphi_1, \varphi_2, \varphi_3$ are the *Silhouette moments* of the respective cluster. *Cluster_Id* represents the profile id of a cluster. The details about the *Silhouette moments* used in this work can be found in [14]. The spatial cluster indices and spatial object indices are stored in a spatial cluster object tree (Fig. D.1) which is a variant of B-tree. The root node termed as *Cluster_Object Root Node*, of the tree points to *k* independent tree structures, where *k* is the number of parameters in the index. If a new parameter is to be accommodated (i.e., for a *k*+1 dimensional index), the root node has to be updated by insertion of a new pointer and accordingly an associated tree structure will have to be generated. Each of the parameter trees will maintain the parameter key value (i.e., say, *r, g, b,* $\varphi_1, \varphi_2, \varphi_3$), along with a pointer to a list of Image IDs (i.e., PIDs).

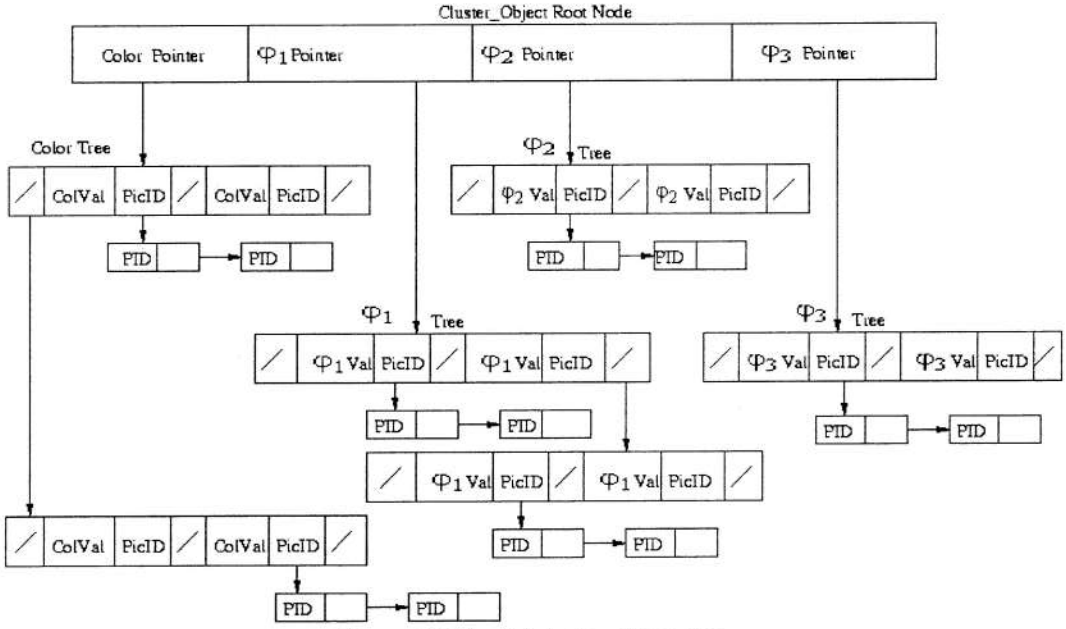


Figure D.1. Spatial cluster_object tree

D.1 Matching Engine

The matching engine has the facility for searching images in the image database based on the cluster and object indices using the *cluster_object tree*[14] as shown in Figure D.1. For global search, the matching engine fetches all those images from the image database whose cluster

indices matches with all the cluster indices of the query image. For object level search, at first the objects present in the query image are determined from the set of clusters present in the query image where the objects may consist of either a single cluster or a combination of clusters. Then, the matching engine queries the database for those images which contains the desired cluster or the combination of clusters. Moreover, the matching engine has the facility for searching in the image database for clusters and objects using both the conjunction 'AND' and disjunction 'OR'.

For example, the query image as depicted in Figure A.4 (a) may have a query that "Find all images that have cluster (1 AND 2) OR 3; i.e. find all those images which have either both the clusters 1 and 2 or cluster 3. For this query, the indices <*cluster_id* ^{1}, $r1^1$, g_1^1, b_1^1, φ_1^1,φ_2^1, φ_3^1>, <*cluster_id* ^{2}, $r1^2$, g_1^2, b_1^2, φ_1^2,φ_2^2, φ_3^2>, and <*cluster_id* ^{3}, $r1^3$, g_1^3, b_1^3, φ_1^3,φ_2^3, φ_3^3>, of the query image will be utilized and the search will initiate from cluster 1. At first it will search in the tree that matches (r^1, g^1, $b^{1)}$, and a Picture Id list will be fetched. Next it will take up φ_1^1 value and search will be done in the φ^1 tree and will fetch another Picture Id list. Similarly, the matching engine will fetch another two Picture Id lists for φ_2^1 and φ_3^1 value by searching in the φ_2 and φ_3 tree respectively. After that all the four Picture Id lists will be concatenated to form a single Picture Id List (PID^1) which matches the first color index <*cluster_id* ^{1}, $r1^1$, g_1^1, b_1^1, φ_1^1,φ_2^1, φ_3^1> of the query. Thus another two Picture Id lists (PID^2 and PID^3) for the indices <*cluster_id* ^{2}, $r1^2$, g_1^2, b_1^2, φ_1^2,φ_2^2, φ_3^2>, and <*cluster_id* ^{3}, $r1^3$, g_1^3, b_1^3, φ_1^3,φ_2^3, φ_3^3> will be generated. Now, according to the query the Picture Id lists PID^1, PID^2 and PID^3 will be combined together according to (PID^1 and PID^2) OR PID^3 to produce the final set of Picture Id list that corresponds to the images in the image database which satisfy the given query. This type of query can also be applied to object level search and combination of object and cluster level search.

D.2 Performance Evaluation

We are implementing the proposed technique using Java in Windows environment with Pentium IV processor with 1 GHz speed and 256 MB RAM. To evaluate the performance of the proposed technique, we considered a set of synthetic images (50 images where 10 are distinct and rest others are variations of each distinct image). To smooth out any variation, each experiment was carried out for several times and the average result was taken. The results (in terms of precision and recall) have been found satisfactory for the synthetic datasets. However, work is going on for further enhancement of the technique to establish its effectiveness over large collection of satellite images.

8. Discussions including future plans for the investigation

Following are the future plans:

- The stand-alone techniques reported above can be enhanced to handle hyperspectral satellite images.
- The mixed pixel handling approach used in the GDSDC can be further explored for obtaining more fine tuned results.
- A rough fuzzy classification approach can be developed to handle large high resolution satellite image data.

9. References

[1] Ertoz, L., Steinbach, M., Kumar, V. "Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data". SIAM International Conference on Data Mining (SDM '03).

[2] Karypis, G., Han, & Kumar, V. "CHAMELEON: A hierarchical clustering algorithm using dynamic modelling". IEEE Computer, 32(8), pp 68-75, 1999.

[3] Zhao Yan-chang, Song Mei, Xie Fan, Song Jun-de. "Clustering Datasets Containing Clusters of Various Densities". Journal of Beijing University of Posts and Telecommunications, 26(2) :42-47, 2003

[4] Kim, H. S., Gao, S., Xia, Y., Kim, G. B., & Bae, H. Y. "DGCL: An Efficient Density and Grid Based Clustering Algorithm for Large Spatial Database". Advances in Web-Age Information Management (WAIM 2006), pp. 362-371.

[5] Dhilon, I. S., & Modha, D. S. "A Data-Clustering Algorithm on Distributed Memory Multiprocessors". In the International Conference on Knowledge Discovery and Data Mining (SIGKDD 99), 1999.

[6] Xu, X., Jager, J., & Kriegel, H. P. "A Fast Parallel Clustering Algorithm for Large Spatial Databases". Data Mining and Knowledge Discovery, 3, 263-290, Kluwer Academic Publisher, 1999.

[7] S. Sarmah, R. Das, and D. K. Bhattacharyya. A distributed algorithm for intrinsic cluster detection over large spatial data. International Journal of Computer Science, 3(4):246–256, 2008.

[8] S. Pal, A. Ghosh, and B. U. Shankar. Segmentation with remotely sensed images with fuzzy thresholding and quantitative evaluation. International Journal of Remote Sensing, 21(11):2269–2300, 2000.

[9] P. Mitra, C. A. Murthy, and S. K. Pal. Density-based multiscale data condensation. IEEE Transactions on Pattern Analysis and Machine intelligence, 24(6), June 2002.

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

[11] S.K. Pal, A. Ghosh, and B. U. Shankar. Segmentation with remotely sensed images with fuzzy thresholding and quantitative evaluation. *International Journal of Remote Sensing*, 21(11):2269–2300, 2000.

[12] G. Karypis, J. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modelling. *IEEE Computer*, 32(8):68–75, 1999.

[13] B. Borah, D.K. Bhattacharyya, and R. K. Das. A parallel density-based data clustering technique on distributed memory multicomputers. In *Proceedings of the ADCOM*, 2004.

[14] P. J. Dutta, D. K. Bhattacharyya, J. K. Kalita, M. Dutta. Clustering Approach to Content Based Image Retrieval. In the Proceedings of the Geometric Modeling and Imaging- New Trends (GMAI'06), 2006.

9. List of Publications

[1] A Grid-density based Technique for Clustering Satellite Image, S. Sarmah, D. K. Bhattacharyya Pattern Recognition Letters, 2010 (under review)

[2] Clustering Satellite Data Using a Grid-Density based Technique, Sauravjyoti sarmah, Dhruba Kumar Bhattacharyya, IEEE Transaction on Knowledge and Data Engineering, 2010 (under review).

[3] DisClus: A Distributed Clustering Technique over High Resolution Satellite Data, Sauravjyoti Sarmah, Dhruba Kumar Bhattacharyya, ICDCN 2010: 353-364

10. Acknowledgement

I am very much grateful to Dr B Gopala Krishna of SAC for his continuous support, guidance and co-operation while executing the project.