Scale Space Exploration for Mining Image Information Content

Mariana Ciucu, Patrick Heas, Mihai Datcu, James C. Tilton[*]

DLR German Aerospace Center, Remote Sensing Technology Institute,
Oberpfaffenhofen, D-82230 Wessling , Germany
Mariana.Ciucu@dlr.de
[*] NASA's Goddard Space Flight Center, Applied Information Sciences Branch,
Greenbelt, MD 20771, USA
James.C.Tilton@nasa.gov

Abstract. Images are highly complex multidimensional signals, with rich and complicated information content. For this reason they are difficult to analyze through a unique automated approach. However, a hierarchical representation is helpful for the understanding of image content. In this paper, we describe an application of a scale-space clustering algorithm (melting) for exploration of image information content. Clustering by melting considers the feature space as a thermodynamical ensemble and groups the data by minimizing the free energy, having the temperature as a scale parameter. We develop clustering by melting for multidimensional data, and propose and demonstrate a solution for the initialization of the algorithm. Due to the curse of dimensionality, for initialization of clusters we choose the initial clusters centers with an algorithm that performs a fast cluster centers estimation with low computation cost. We further analyze the information extracted by melting and propose a structure for information representation that enables exploration of image content. This structure is a tree in the scale space showing how the clusters merge. Implementation of the algorithm is through a multi-tree structure. With this structure, we can explore the image content as an information mining function, we obtain a more compact data structure, we have maximum of information in scale space because we memorize the bifurcation points and the trajectories of the centers points in the scale space. The information encoded in the tree structure enables the fast reconstruction and exploration of the data cluster structure and the investigation of hierarchical sequences of image classifications. We demonstrate the effectiveness of the approach with examples using satellite multispectral image (SPOT 4) and Synthetic Aperture Radar – SAR and Digital Elevation Models – DEM derived from SAR interferometry (SRTM).

1. INTRODUCTION

Data mining and knowledge discovery are the processes of analyzing data from different perspectives and summarizing it into useful information. Technically, data mining is the process of finding correlations or patterns of fields in large relational databases [1] .

In this article a multi-scale image information mining method is presented. A similar approaches was proposed in [3] based on an image hierarchical segmentation. The presented method is based an clustering by melting exploring the scale of the image feature space.

1.1. Clustering

Clustering is one of the most important methods in Data Mining applications. Clustering of data is a method by which large sets of data are grouped into clusters having similar behaviour, or dividing a large data set into smaller data sets based on some similarity measure.

A clustering algorithm finds *the centroid e.g. center of mass or center of gravity)* of a group of data sets and determine cluster membership. Most algorithms evaluate a distance between a point and the cluster *centroids*. The output from a clustering algorithm is a statistical description of the clusters, *the centroids* and the number of components in each cluster.

The various clustering concepts available can be grouped into two categories, by the type of structure imposed on the data [1]:

1. *Hierarchical clustering*
2. *Nonhierarchical clustering*

1. Hierarchical clustering

A hierarchical clustering is a sequence of partitions in which each partition is needed to form the subsequent partition in the sequence. These methods include those techniques where the input data are not partitioned into the desired number of classes in a single step. Instead, a series of successive fusions of data are performed until the final number of clusters is obtained. An important objective of hierarchical clustering is to provide a picture of the data that can be easy interpreted, such as a dendogram. An example of hierarchical clustering is the melting algorithm.

2. Nonhierarchical clustering (partitional clustering)

These methods include those techniques in which a desired number of clusters is assumed at the start, and a single partition is found. Points are allocated among clusters so that a particular clustering criterion is optimized. A possible criterion is the minimization of the variability within clusters, as measured by the sum of the variance of each parameter that characterizes a point. Examples of nonhierarchical clustering are K-means, and Expectation-Maximization (EM)

K-means has as an input a predefined number of clusters, and is a simple, iterative procedure. This algorithm assigns each data point to the cluster center closest to it, forming in this way k exclusive clusters of the data.

Expectation Maximization (EM) algorithm is a mixture based algorithm that assumes the data set can be modelled as a linear combination of multivariate normal distributions. The algorithm finds the distribution parameters that maximize a model quality measure, called likelihood, producing the maximum likelihood (ML) solution.

2.CLUSTERING BY MELTING AND OUR IMPLEMENTATION

The clustering by melting algorithm is based on information theory and statistical mechanics and is the only algorithm that incorporate scale and cluster independence. Using information theory and statistical mechanics, Wong [7] showed that cluster centers correspond to the local minima of a thermodynamical free energy F that depends on the data points and the scale parameter β. The algorithm is scale-space based and provides more effective clustering than other methods. The basic idea is that clusters depend on the scale one uses to examine the data.

At a very fine scale, every datum is itself a cluster, while at a very coarse scale, the whole dataset is a cluster.

The number of minima of F depends on the distribution of the data points and the scale parameter beta, which is the "inverse temperature." If we start with a large *beta* (low temperature) so that every data point is a cluster, then as we gradually decrease *beta* (increase the temperature), the clusters merge; and finally, at a very small *beta* (very high temperature), all data points merge to one cluster.

If clusters of several points indeed exist, the information should be present in the data itself. Data points closer to the cluster center should give more information about the clusters while those far away should give less. These different degrees of contribution can be modeled probabilistically by defining $p(x|y)$ as a contribution of data point x to a cluster center y.

The problem is to find the set of cluster center y that best suit the data points x with respect some constraints. The best solution is obtained by maximizing the entropy:

$$H = \sum_{x \in D} p(x|y) \log p(x|y) \ ,$$

where D is data space.

Suppose the cost function is $e(x) = (x - y)^2$, and maximizing the entropy with the constraint:

$$\sum_{x \in D} p(x|y) e(x) = C$$

we obtain

$$p(x|y) = \frac{\exp\left[-\beta(x-y)^2\right]}{Z}$$

where

$$Z = \sum_{x \in D} \exp\left[-\beta(x-y)^2\right]$$

To make the connection with thermodynamics, the free energy is $F = -\frac{1}{\beta}\log Z$. At equilibrium, a thermodynamic system settles into equilibrium if it has minimum free energy.

Minimum free energy is obtained if $\frac{\partial F}{\partial y} = 0$, or equivalently

$$y = \sum_{x \in D} \frac{(x-y) * \exp\left[(-\beta) * (x-y)^2\right]}{\sum_x \exp\left[(-\beta) * (x-y)^2\right]} \tag{1}$$

For a given β, the problem of clustering is mapped to the problem of finding solution for y of Eq. (1). However, for a general β, the solution cannot be found analytically. The solutions are identical to the fixed points of the following map:

$$y \xrightarrow{f} y + \sum_{x \in D} \frac{(x-y) * \exp\left[(-\beta) * (x-y)^2\right]}{\sum_x \exp\left[(-\beta) * (x-y)^2\right]} \tag{2}$$

The solutions can be computed by an iterative equation (2) .

Thus, the structure of the melting algorithm is:

1. An initial high β is chosen and every data point is set as a cluster.
2. β is decreased gradually.
3. The mapping (2) is repeated N times or until the cluster converges.
4. If two or more clusters, which previously were distinct, share the same center, the set of data associated with the new cluster is the union of those with the original clusters.
5. If more than one clusters exist, go to 2.
   Otherwise, stop.

The information obtained by melting algorithm is:

- The set of clusters as functions of temperature
- Trajectories of cluster centers as functions of temperature
- Bifurcation points
- Free energy schedule dependency of temperature
- The sequences of hierarchical image classification

This information can be used to explore the image content as an information mining function.

However, due the computational complexity, an optimal data representation is needed for:

- more compact data structure
- fast and easy access to the information

We propose a tree structure, that has a two node structure:

Node1 contains:

- a pointer to the same node structure (to *Node1)*
- a pointer to the following node structure (to *Node2*)

Node2 contains:

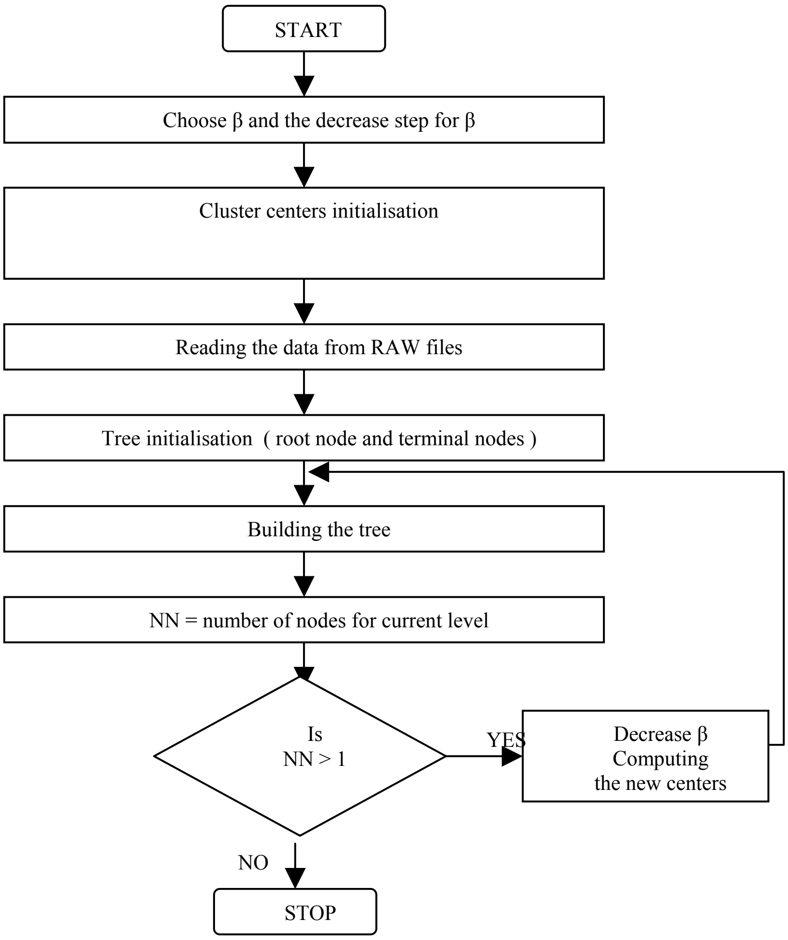
- a vector for features (in our case we have four features for four bands)
- a scalar for beta
- a scalar for index, which is for image map
- a pointer to *Node1*

The index is necessary for this structure because if two clusters centers have the same value we put in the next level of the tree the same index. With this index, we can obtain the sequences of images classification, as we can see in Section 4, in figure 4, 11. With this structure we can make fast and easy the plot of clusters centers versus temperature, as we can see in figures 5-8, 12, 13. Thus, is only necessary to cross the tree from the terminal nodes to the root node, for each terminal node, with a recursive function. In our algorithm each level of tree corresponds to each temperature, and for this consideration, we can reconstruct the information of image from one temperature to another.

The tree contains the complete information about the image in scale space, because we don't record only the bifurcation points, but also the trace of all the center points in the scale space.

The tree structure is a multi tree, which has a multi –tree to the left and a multi –tree to the right. The tree is built from the terminal nodes to the root, because we wish that all the computations be done during the building of tree.

The flowchart of this algorithm, which contains the melting algorithm and the tree structure, follows:


START
Choose β and the decrease step for β
Cluster centers initialisation
Reading the data from RAW files
Tree initialisation (root node and terminal nodes)
Building the tree
NN = number of nodes for current level
Is
NN > 1
YES
Decrease β
Computing
the new centers
NO
STOP

3.1. Computational problem and dimensionality aspects

The generalization of the algorithm for the multidimensional case raises two problems:

- *the computational complexity*

The computational complexity is :

$$O\left(\left(n \times d \times n_i \times n_\beta\right)\right) + \log 2\left(n_t\right),$$

where :

n is number of points

d is the dimensions for the feature space

n_i, n_β is number of iterations

$\log 2(n_t)$ is the tree complexity, where n_t is number of nodes from tree,

$$n_t = 2^{(n_\beta \div 1)} - 1$$

The solution for this is to split the computation into two steps:

1. off-line – generating the tree information structure
2. on-line – analyzing and exploring of image content stored in the tree information

- *the curse of dimensionality at algorithm initialization*

We can deal with this in many ways. For example:

1. choosing the initial clusters centers randomly. However, in this case we can lose much information about data;
2. choosing the initial cluster centers with another algorithm, such as the "Fast cluster centers estimation," which will be discussed in the next section.

The second way is better than first, because we don't lose information and with this we have a low computational cost, because we begin only with few data points as a cluster and not with all data points.

3.1.1. Fast cluster centers estimation

Numerical gradient estimation methods may be used in order to reduce the computational demands of a class of multidimensional clustering algorithms, or may be used in a direct way to make an initial exploration of large data sets by evaluating the number of existing clusters.

3.1.1.1. Description of the Merging Gradient Estimation algorithm

This algorithm is presented in Fox [5].

Assuming that clusters are regions of relatively high point density within the data space, which is to say that the rate of change of points occurrence with respect to distance travelled in all directions of the space is relatively high – i.e. higher than the rate occurrence which would be encountered if all the points were uniformly distributed over all the space since this represents the maximum entropy case in which any cluster exists. Furthermore clusters centers may then be considered as local maxima of such gradients. However this local maxima of the gradient, i.e. marginal density, has to exhibit a value greater than the marginal density that would occur if all the points were evenly distributed. As a example the upper right graph of figure 1 shows the density of points repartition in a two dimensional space and the marginal densities on the two axes of synthetic Gaussian data.

The computational procedure is as follows:
First, of the N dimensional Gaussian data X of n elements is read.

$$X^i = \left(x_1^i, x_2^i, \ldots, x_n^i\right) \; ; i = 1,\ldots,n \qquad \textbf{(3)}$$

The next step is to sort the data for each of the N dimensions into ascending numerical order since travelling sequentially through sorted vectors corresponds to travelling along the different dimension axes.

$$S^m = \left(s_1^m, s_2^m, \ldots, s_N^m\right) \; ; m = 1,\ldots,n \qquad \textbf{(4)}$$

$$S^m = \mathrm{sort}\left(s_1^i, s_2^i, \ldots, s_N^i\right) \; ; i = 1,\ldots,n \qquad \textbf{(5)}$$

Define the vector *C* representing the cumulative sum of points encountered as one move along any of the sorted vectors $s_{j.}$

$$C_i = i \; ; i = 1,\ldots,N\text{-}1 \qquad \textbf{(6)}$$

The marginal density estimates in each direction may be then e interpreted as the gradient of the N graphs generated by plotting C versus s_j the figure 1 (upper left and lower right graphs). This exhibits the repartition of a Gaussian synthetic data for two dimensions of the feature space the marginal densities on two axes of this space and also the step functions *C* versus s_j. However, to compute the gradients presented as well in these graphs a numerical differentiation from discretely sampled data is required. A simple but fast technique is applied here. It begins by filtering the sorted vectors s_j in order to smooth out the raw data C versus s_j curves. Hence, we obtain:

$$f_j^m = \frac{1}{2\mathrm{h}+1} \sum_{r=\mathrm{m}-\mathrm{h}}^{\mathrm{r}=\mathrm{m}+\mathrm{h}} s_j^r \qquad \textbf{(7)}$$

The smoothing window used here is a parameter that determinates the scale of Gaussian structures we will detect. The next step is the computation of the gradient estimates g_j . It may then be obtained from the smoothed *C* versus g_j curves according to the constructions.

$$g_j^m = \frac{2\mathrm{h}}{f_j^{m+h} - f_j^{m-h}} \qquad \textbf{(8)}$$

$$g_{mean,j} = \frac{\mathrm{n\text{ - }1}}{f_j^n - f_j^1} \qquad (\mathbf{9})$$

The second equation computes the average point density that would exist if the data was uniformly distributed in all the space. The edges may be computed for the filtering and for the gradient estimates by the use of descending spans. Then all local maxima of the gradient estimates, which are above the average marginal density value, have to be extracted. The final step is to select only the maxima that correspond to an existing data value in the n different dimensions. Of course, the correspondence to the original data has to be saved. These maxima correspond to the approximated centers of the clusters.

3.1.1.2 Algorithm optimisation

In order to reduce the computational time of a "classical" sorting procedure, a sorting routine of complexity N*n (number of dimension by number of data points) has been developed. The idea is to scan the data only once and to sort, each data point for each dimension, in his associated dynamic collections itemized by his value. Then, for each dimension, the collections are concatenated by order of crescent value to constitute the N different sorted vectors.

A last change is applied here in order to avoid centers of similar value. This can happen when irregularity remain after smoothing the data. The extra centers are simply removed.

Finally, this algorithm has complexity N*n, what in time computation, constitute an advantage on for example the K-Means algorithm which has complexity N*n*K, where K is the number of cluster. Furthermore, the algorithm doesn't need to have a fixed number of clusters as an input.

Taking into account the main quality of the algorithm, which is the low computational cost, the results shows a good efficiency versus time consumed.

We tested this algorithm initially on 4 dimensional synthetic data composed of uniform distributed noise, and 3 Gaussian structures of different mean only in two dimensions in order to simplify the interpretation of the results. One of them has a larger variance.

The algorithm performance in finding the correct number of Gaussian structures with their precise center values in a reasonable amount of time consumption depends on the smoothing parameter discussed previously. This parameter influences the regularity of the gradient function and consequently the number of maxima detected. Moreover, if we use a large smoothing window to detect only the relevant Gaussian structures, the lost of precision on the centers value will make it impossible to find the correspondence of maxima between the different dimensions. On one hand, we will obtain, by a small smoothing window, a good detection of all the clusters

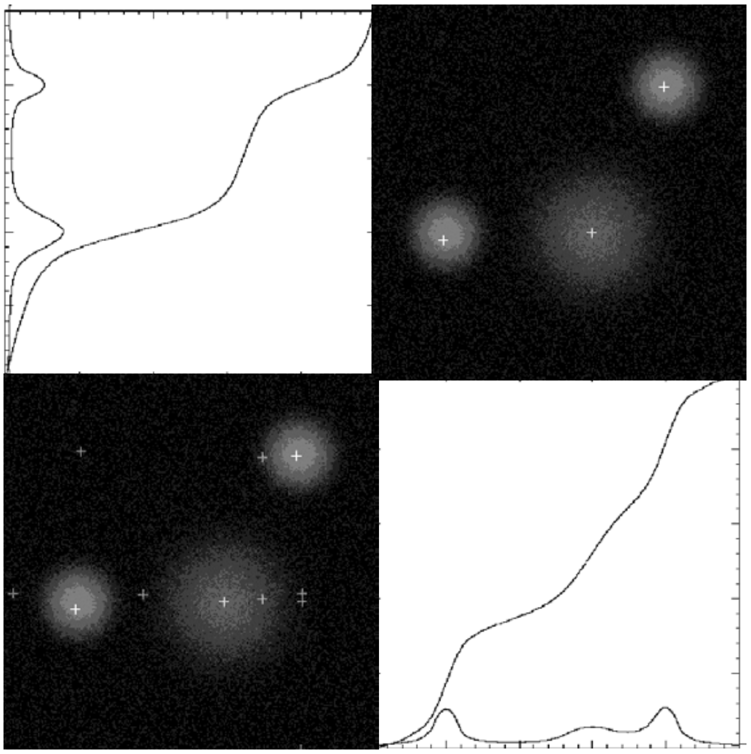


Fig. 1. Merging Gradient Algorithm on synthetic data set.

but with many centers belonging to the same Gaussian and other unsignificant centers resulting from noise. The bottom left plot in figure 1 illustrates this effect. On the other hand, we will obtain, by a large smoothing window (which means a greater time consumption), single center detection for each Gaussian structure. However, some structures, as Gaussian of greater variance or lower density, may not be detected and we will loose precision on the center's value. Currently, this parameter is estimated heuristically. However, a correct estimation of this parameter could be performed.

The inability of finding a good estimate of the number of clusters when the structures are too different has little consequence when this algorithm is used only to initialise a more complex clustering algorithm such as "Melting" algorithm.

3.1.1.3. Enhanced algorithm for estimation of number of clusters

This fast center algorithm estimator may also be used to explore large data sets by estimating directly the number of Gaussian structures existing in the data and their center's value. We assume the data to be a mixture of Gaussians. The problem, to be solved, is to detect Gaussian structures with different variances, densities, regularities, with only one maximum associated with each one of them.

3.1.1.3.1. Removing the centers which migrate

A way to face this problem is to observe the evolution of the centers value given by the merging gradient estimator algorithm, while we compute their new value.

To compute them, we first create classes associated to each center value. Each class regroups the smoothed data that present a minimum distance to each center value. The new center's values are calculated as the gravity center of each class.

Let's suppose we have detected all the structures with at least one center associated by an appropriate smoothing window. We will observe after the computation of the new center values a fast migration of unsignificant centers or centers which share the same Gaussian structure and divide it into more than one class.

These "extra centers" will move to the barycenter of the "unclustered mass".

Therefore, the idea is to keep updating the centers, by removing those that migrate farther than a fixed limit, while we iterate the procedure describe above.

This procedure will end when no remaining center migrates farther than this limit.

The choice of the migration limit depends on the topology of the smoothed data. We choose here a heuristic migration limit. However, an estimation of this parameter can be computed to optimise this choice.

3.1.1.3.2. Injection of an attractor

To enhance the migration phenomena, uniformed distributed noise can be injected in the feature space to favour as equally as possible the removal of the "extra centers". The quantity of noise-injected must be adjusted so that it attracts only the "extra centers'" This noise mustn't drown or modify significantly any of the structures detected (i.e. its density must be much lower). The quantity of noise injected constitutes another parameter that has to be estimated. Here the estimation was again only heuristic. Performing the enhanced algorithm on the synthetic data set, the number and center values of Gaussian structures were correctly estimated. The upper right plot of figure 1 shows that the unsignificant centers detected previously where effeciently removed.

4. EXPERIMENTAL RESULTS

4.1. Merging gradient algorithm applied on a SPOT image

We applied the preceding algorithm on a sample 256*256 of a 4 Bands Spot4 image from a region near Bucharest-Romania. The original image color representation is presented in figure 2a. The repartition of the multispectral data in the feature space is illustrated in figure 3.

Three different center estimations have been computed leading to 142, 18 and 4 cluster centers. The classification resulting of these clustering are presented in respectively figure 2b, c and d.

The tuning of the parameters leads to different results : a over-estimation of the number of clusters in the first case. However, the 142 centers detected, which are ploted in the left plots of figure 3, will be used to initialise the melting procedure. The classification with 4 classes is a sub-estimation of the number of clusters due to a too large smoothing window. The classification with 18 classes is a good fast number of clusters estimation. It is presented in the right plots of figure 3.

The computation was for the example with 142 classes performed in 47 sec on a "300 MHz SUN, UltraSPARC-II".and the K-means algorithm was computed with the same conditions and last 2,35 sec.

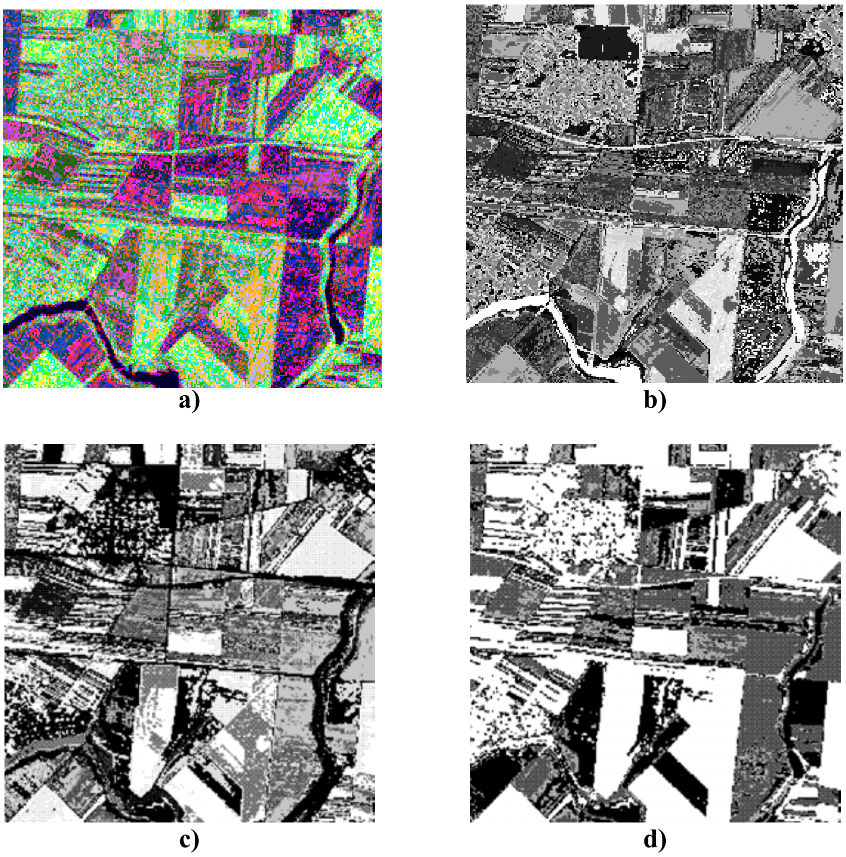


Fig. 2. a) Original image (band 1, 2 and 3), classification with: b) 142 classes, c) 18 classes, d) 4 classes

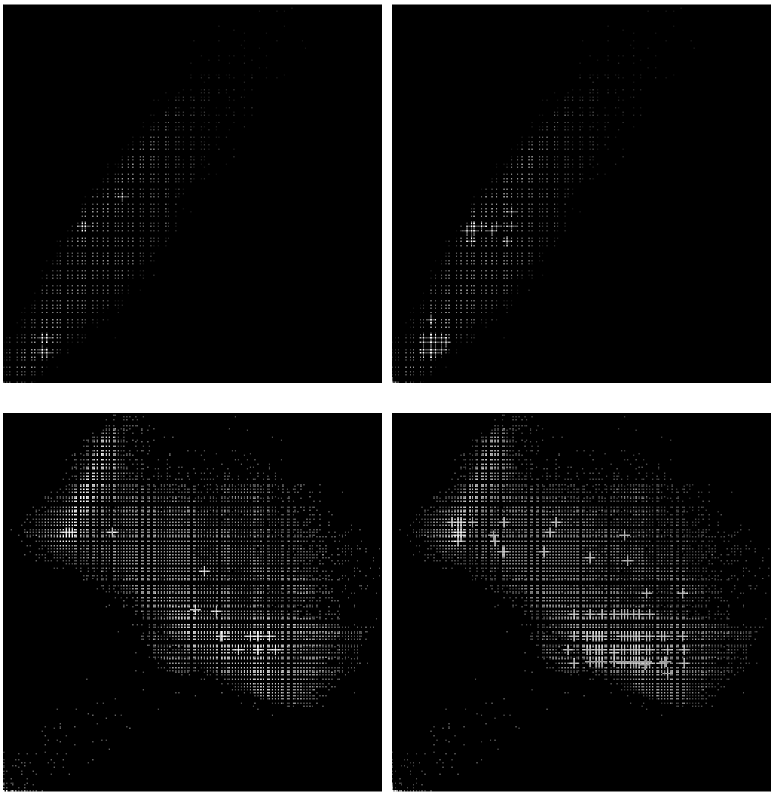


Fig. 3. center location, for classification with 18 classes (up) and with 142 classes (down), in feature space: band1-2 (left), bands3-4 (right)

4.2. Clustering by melting for mining the image content

Images are high complexity multidimensional signal with rich information content. The information extraction in terms of image classification is not an easy task. The results depend on the used model – algorithm. Thus many times uncertitudes remains unsolved.

In this paragraph we demonstrate the use of clustering by melting as an alternative solution for understanding images as a "data – mining" concept.

With propose structure we obtain a sequences of hierarchical image, so we have more information of classification than only with one image. We can see what clusters merge together, how many clusters we have at each temperature and we can choose what is the good number of clusters.

In the classical solution, when we need the initial number of clusters we can lose clusters, because we don 't know the best number of clusters or we can have many clusters without points.

The sequences of hierarchical image classification in figure 4 are for bifurcation points in figures 5 - 6 and in figure 11 for figures 12,13.

Trajectories list the clustering one after another. Cutting a trajectory at any level defines a clustering and identifies clusters.

Input
1. Beta and step for beta
2. Original image
3. Center of clusters (initial configuration)
4. Tree structure

Output
1. Sequences of images classification
2. Graphics of bifurcation points

4.2.1. Example of mining multispectral data.

We apply the mining method for the exploration of the information content of a multispectral image of agriculture fields.

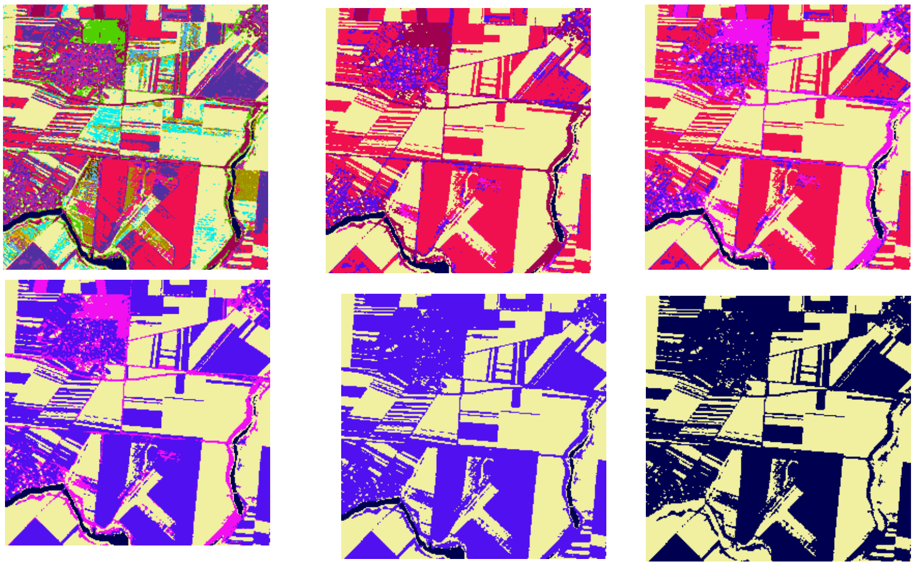


Fig. 4. Figures contains labeled images at initial $\beta = 500$ with decremental step $\Delta\beta$ =1.05

The trajectories of the cluster centers, in one dimensional projection, are presented in figures 5 – 8.

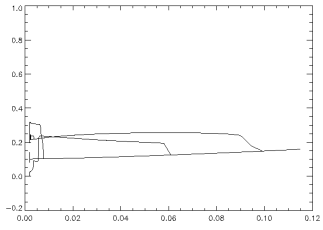


Fig. 5. *y1*

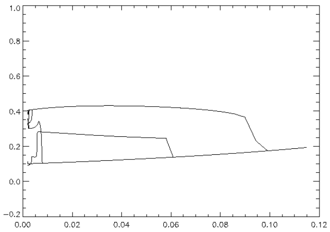


Fig. 6. *y2*

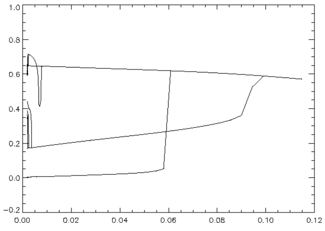


Fig. 7. *y3*

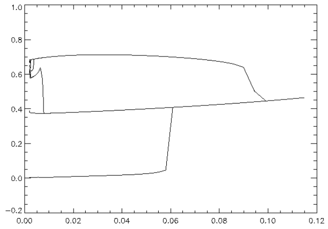


Fig. 8. *y4*

Components of the trajectories of the cluster centers versus scale

4.2.2.Example of fusion of SAR image and DEM.

The paragraph presents the application of the proposed algorithm for understanding of a scene imaged by the SRTM X-SAR sensor.

The melting algorithm was applied on the pair SAR-Synthetic Aperture Radar image and DEM – Digital Elevation Model, thus a data fusion is performed.

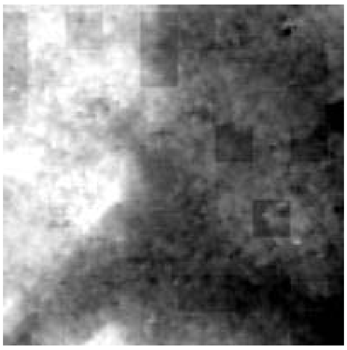


Fig. 9. SRTM - DEM

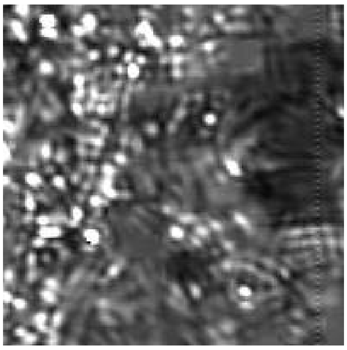


Fig. 10. SRTM , X-SAR image

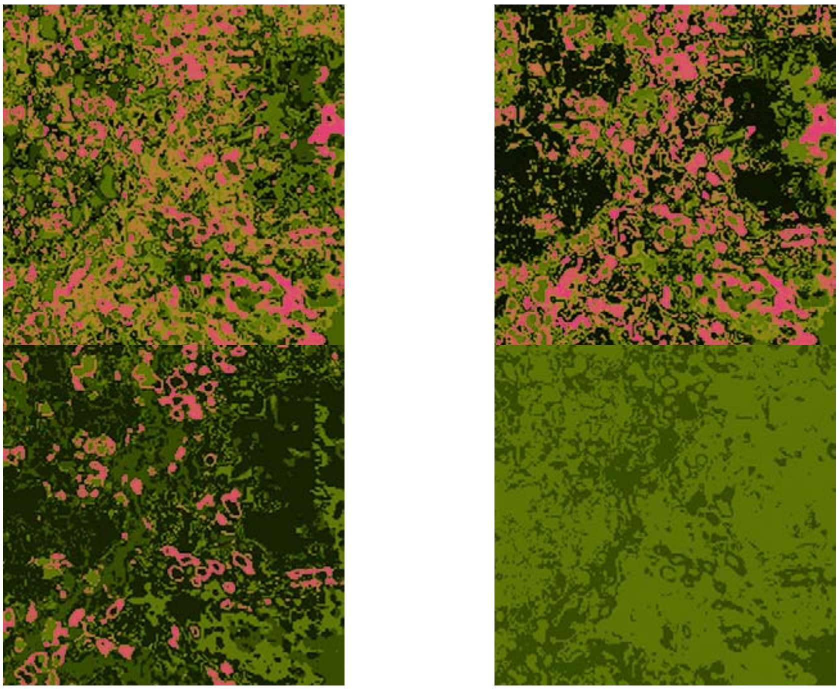


Fig. 11. figures contains labeled images at initial beta=2000 with decremental step $\Delta\beta$ =1.05

The trajectories of the cluster centers , in one dimensional projection, are presented in figures 12, 13.

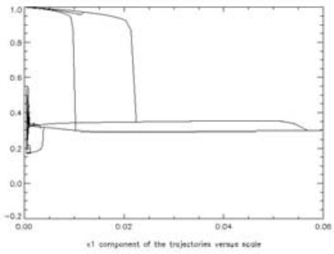


Fig. 12. y1

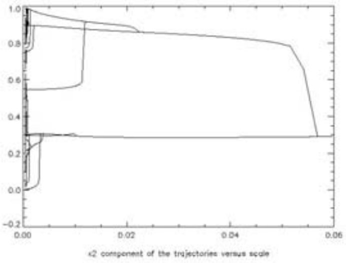


Fig. 13. y2

Component of the trajectories of the cluster centers versus scale

The sequence of classification of the pair SAR image and DEM makes evident various types of urban and non-urban areas.

5. CONCLUSIONS

The article is presents an enhancement of the algorithm for clustering by melting and proposes its is use for image information mining.

In our application, the implementation of the melting algorithm is a multi-tree structure and with it we can access easily and in a fast way the information, thus, we can rebuild the image information content at any temperature. Therefore, we can visualize the clusters of image and we can choose the best number of clusters for images.

With fast cluster center estimation algorithm we reduce the computational cost which allows us to start the melting procedure with the appropriate number of clusters according to this computation cost.

The multi-tree structure offers the possibility of accelerating the procedure by adjusting the error allowing cluster centers to merge together.

ACKNOWLEDGMENTS

The SPOT image was provided by the Centre National d'Etudes Spatiales, CNES.

REFERENCES

1. Anil K. Jain, Richard C. Dubes, "Algoritms for Clustering Data", *Michigan State University*

2. "Digital Patern Recognition", *Communication and Cybernetics*

3. James C. Tilton and William T. Lawrence, "Interactive Analysis of Hierarchical Image Segmentation," *Proceedings of the 2000 International Geoscience and Remote Sensing Symposium* (IGARSS '00), Honolulu, HI, Jul. 24-28, 2000.

4. M. Schröder, H. Rehrauer, K. Seidel and M. Datcu, "Interactiv Learning and Probabilistic Retrieval in Remote Sensing Image Archives", IEEE Trans. on Geoscience and Remote Sensing, pp. 2288--2298, 2000

5. P.D.Fox, "On Merging Gradient Estimation with Mean-Tracking Techniques for Cluster Identification",1997

6. Richard O. Duda, Peter E. Hart, David G. Stork, "Patern Recognition"

7. Yiu-fai Wong and Edward C. Posner, , "A new Clustering Algorithm Applicable to Multispectral and Polarimetric SAR Images", *IEEE Transactions on Geoscience and Remote Sensing ,* vol. 31, no. 3, May 1993.