

EXPERIMENTS IN FUZZY IMAGE SEGMENTATION

Dorian Cojocaru*, Răzvan Tănăsie**, Elena Bărbulescu*

University of Craiova, Faculty of Automation, Computers and Electronics,

**Mechatronics Department,*

***Software Engineering Department.*

Abstract: As a general concept, the job of the cluster analyze is the data partition into a number of groups, or clusters. Applying this partitioning operation on images, the image segmentation - a very important task for image processing - is obtained. This paper presents a fuzzy segmentation algorithm, fuzzy c-means, applied to grayscale images.

Keywords: computer vision, image processing, image segmentation, fuzzy logic.

1. INTRODUCTION

Image segmentation is an important preprocessing operation for any computer vision application. As a definition, image segmentation is a partition of an image into regions. Using the crisp sets in order to obtain a complete segmentation has, for many applications, drawbacks because the accurate edges are not always available. That's why fuzzy segmentation is recommended. The fields in which fuzzy image segmentation is most used are the following: medical applications (fuzzy segmentation for magnetic resonance images, dental X-ray image segmentation, segmentation of ventricular angiographic images), satellite image segmentation, face image segmentation, etc. There are applications in which it is needed to know if some points belong much more than others to a region, as in fuzzy logic (Cojocaru 2002) the truth is a matter of degree.

For image segmentation, depending on the type of the application, (Cocquerez, J-P., S. Philipp 1995) there are two types of methods:

- Regions extraction using pixel's similar properties
or
- Contours extraction followed by region definition as a pixel set situated inside the contour.

None of these methods is a priori better than the other. The two methods are complementary and the best way seems to be a cooperative approach of those methods.

There are many techniques (Hamid R., Tizhoosh 1997) of fuzzy image segmentation: histogram thresholding, edge based segmentation, region growing, fuzzy clustering algorithms, fuzzy rule-based approach, fuzzy integrals, measures of fuzziness and image information, fuzzy geometry, but among them the most dominant are fuzzy clustering and fuzzy rule based segmentation techniques.

Pattern recognition techniques can be classified into two categories:

- unsupervised techniques, which use sets of unclassified data points;
- supervised techniques, which use sets of points that have a known classification.

As it can be remarked, these two types of techniques are complementary. This paper first presents the unsupervised clustering, and then focuses on a fuzzy C-Means algorithm, which is the best known unsupervised fuzzy clustering algorithm. The supervised techniques are not subject of this paper.

Fuzzy clustering is the oldest fuzzy approach to image segmentation. Algorithms such as fuzzy c-means (FCM, Bezdek) and probabilistic c-means (PCM, Krishnapuram & Keller) can be used to determine the image regions. Fuzzy rule-based techniques need less computational effort than fuzzy clustering, and they can use both numerical and linguistic variables (Chang, C.-W.; Ying, H.; Hillman, G.R.; Kent, T.A. and Yen, J. 1998).

This paper focuses on fuzzy c-means algorithm and the experimental results obtained by using this algorithm.

2. UNSUPERVISED CLUSTERING

Unsupervised clustering is motivated by the need to find interesting patterns or groupings in a given set of data. Because image segmentation can be considered as a type of data clustering, it is used to perform the segmentation of images in the field of computer vision (mostly for image processing and pattern recognition). This can be done by considering each data to be described by a series of features for every pixel.

A criteria for computing the corectness of a partition makes conventional clustering algorithms determine a way to split a given set of data, in such a way that each data belongs to exactly one cluster of the partition (also called “hard partition”). This can be formulated in the following definition.

Definition: Let X be a set of data and x_i be an element of X . A partition $P=\{C_1, C_2, \dots, C_N\}$ of X is “hard” if and only if :

$$\forall x_i \in X, \exists C_j \in P \quad \text{such that } x_i \in C_j \quad (1)$$

$$\forall x_i \in X, x_i \in C_j \Rightarrow x_i \notin C_k \quad \text{where } k \neq j \quad C_j \in P \quad (2)$$

Equation (1) states that each data in the set is covered by the partition (is included in a cluster), while equation (2) makes all clusters in the partition are mutually exclusive, that is, each data in the set can be part of one and only one cluster.

The second condition is not suited for real world cases, because, in this type of clustering problems, some data elements (points) may partially belong to more than one cluster.

Another type of partitioning considers “soft partitions” of a given set of data using a different criteria which considers that a data can partially belong to multiple clusters. This can be formulated in the following definition.

Definition: Let X be a set a data, and x_i be an element of X . A partition $P=\{C_1, C_2, \dots, C_N\}$ of X is soft if and only if:

$$\forall x_i \in X, \forall C_j \in P \quad 0 \leq u_{ij} \leq 1 \quad (3)$$

$$\forall x_i \in X, \exists C_j \in P \quad \text{such that } u_{ij} > 0 \quad (4)$$

where u_{ij} denotes the degree to which x_i belongs to cluster C_j .

A important type of this kind of clustering (with “soft partitions”) is one for which the sum of the membership degree of a data element (point) x for all clusters is one:

$$\sum_j u_{ij} = 1 \quad (5)$$

A “soft partition” that satisfies equation (5) is called a constrained soft partition. The fuzzy c-means algorithm determines a constrained soft partition.

The definition of compact well-separated clusters is necessary to continue. This is presented below.

Definition: A partition $P=\{C_1, C_2, \dots, C_k\}$ of the dataset X has compact well-separated clusters if and only if any two points in a cluster are closer than the distance between two points in different clusters.

$$\forall x, y \in C_p \quad d(x, y) < d(z, w) \quad (6)$$

where

$$z \in C_q, w \in C_r \quad (7)$$

and d denotes a distance measure.

Suppose that a set of data contains c compact well-separated clusters. Then, the goals of c-means algorithm are:

- 1) To find the centers of the clusters;
- 2) To determine to which cluster each data element (point) in the set is a part of.

The second goal can be achieved based on the following: a data element (point) is aprt of the cluster whose center is closest to:

$$x_i \in C_j \quad \text{if } \|x_i - c_j\| < \|x_i - c_k\| \quad k = 1..c, k \neq j \quad (8)$$

where c_j denotes the center of cluster C_j .

To fulfil the condition imposed by the first goal, a criterion that can be used to search for the cluster centers must be established. Such a criteria is the sum of the distance between points in each cluster and their center. This criterion is used because a set of real cluster centers will give an optimal solution.

Based on these observations, the c-means algorithm tries to find the clusters centers C than minimize J (give an optimal solution). However, the optimal

solution also depends on the partition P , which is determined by the cluster centers C . Therefore, the c-means algorithm iterates in order to find the real cluster centers:

- 1) Compute the current partition based on the current cluster.
- 2) Modify the current cluster centers using a gradient decent method to minimize the J function to optimize the solution.

The iteration stops when the difference between the cluster centers in iteration k and $k+1$ is smaller than an imposed value. This means the function saturated and converged to the local minimum, yhus obtaining the optimal solution.

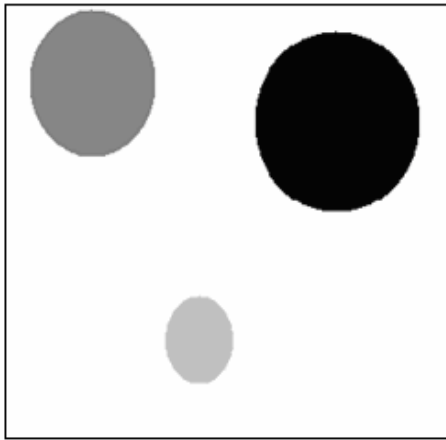


Fig. 1. An Example of compact well separated clusters.

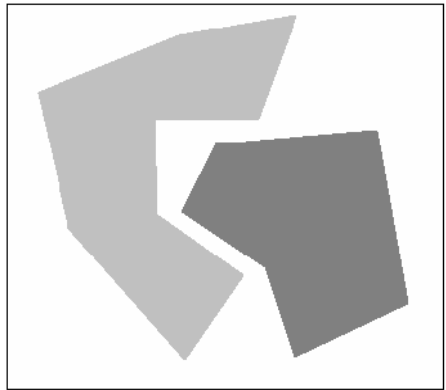


Fig. 2. An example of two clusters that are not compact and well separated.

3. FUZZY C-MEANS CLUSTERING ALGORITHM

Fuzzy C-means is an algorithm based on one of the oldest segmentation methods witch allows data to have membership of multiple clusters, each to varying degrees. This method, used in pattern recognition, was developed in 1973 by Dunn and improved by Bezdek in 1981. The algorithm is based on minimization of the following function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2 \quad (9)$$

$$1 \leq m < \infty$$

Where:

- m is any real number greater than 1,
- u_{ij} is the degree of membership of x_i in the cluster j ,
- x_i is the i -th of d -dimensional measured data,
- c_j is the d -dimension center of the cluster,
- $\|*\|$ is any norm expressing the similarity between any measured data and the center.

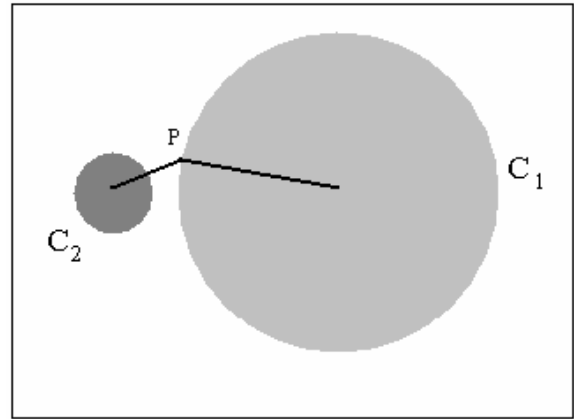


Fig. 3. Two clusters that are compact, but not well separated.

This algorithm (Bezdek, J. C. 1981) realizes an iterative optimization of the J_m function, updating membership u_{ij} and the cluster centers c_j using the following formulas:

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (10)$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{j=1}^N u_{ij}^m} \quad (11)$$

The minimization of J_m is achieved only when the u_{ij} function saturates, that is, the stop criterion is given by the equation:

$$\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon \quad (12)$$

Where ε is a number between 0 and 1, and k is the iteration step.

Fuzzy c-means algorithm has the following steps (Yen J. and Langari R. 2000):

1. Consider a set of n data points to be clustered, x_i .
2. Assume that the number of clusters, c , is known, $2 \leq c < n$.
3. Choose an appropriate level of cluster fuzziness, $m \in \mathbb{R} > 1$.
4. Initialize the $(n \times c)$ sized membership matrix U to random values such as $u_{ij} \in [0,1]$ and

$$\sum_{j=1}^c u_{ij} = 1.$$

5. Calculate the cluster centers c_j using (11) for $j = 1 \dots c$.
6. Calculate the distance measures $d_{ij} = \|x_i^{(j)} - c_j\|$, for all clusters $j = 1 \dots c$ and data points $i = 1 \dots n$.
7. Update the fuzzy membership matrix U according to d_{ij} .

$$\text{If } d_{ij} > 0 \text{ then } u_{ij} = \left[\sum_{k=1}^c \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}} \right]^{-1}$$

If $d_{ij} = 0$ then the data point x_j coincides with the cluster center c_j , and so full membership can be set $u_{ij} = 1$.

8. Repeat from step 5 until the change in U is less than a given tolerance, ϵ .

This algorithm's fuzzy behaviour is given by the membership function, which links the data to each cluster. Some conditions must be respected in building the matrix U (conditions from step 4.).

Matrix U factors represent the degree of membership between the centers of the clusters and the data.

It is important to understand the importance of m . The smaller m is, the crisper the algorithm ($m=1$ represents a crisp algorithm), while the greater m , the fuzzier the clusters are (smaller values for the membership function, hence fuzzier clusters). It is recommended to start with $m=2$, and after a series of tests, decide whether it is necessary to choose a different value.

For $n = 5$ and $c = 3$, the matrix could have the following form:

$$U = \begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.7 \\ 0.6 & 0.1 & 0.3 \\ 0.1 & 0.8 & 0.1 \end{bmatrix} \quad (13)$$

4. IMPLEMENTATION

The algorithm was implemented using Microsoft Visual C++ from the package Microsoft Visual Studio .NET. It was integrated in a already implemented project that deals with fuzzy image processing. A new function, Image Segmentation, was added to the existing menu. The application is designed to work on 8 bits grayscale images.

The user has the possibility, using this application, to open and work with .bmp grayscale images on 8 bits. No function is allowed before an image is loaded. Once an image loaded, the user can operate on the image, using his own parameters.

For the image segmentation function, the default

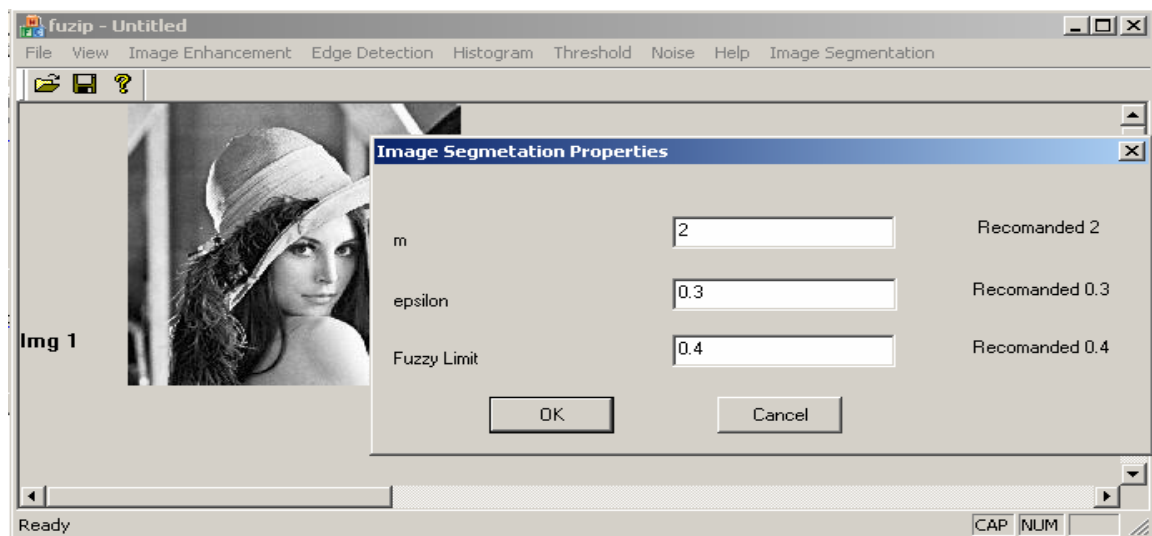


Fig. 4. User interface example.

values for parameters are the following:

- $m = 2$;
- $\varepsilon = 0.3$;
- Fuzzy limit = 0.4.

Those values can be modified by the user on the Properties menu. Those parameters can be chosen according to the theoretical requirements of the application or to user's experience (heuristics).

5. EXPERIMENTAL RESULTS

Fuzzy c-means algorithm was tested on various images. Each image is a bitmap with 256 gray levels. The examples presented are on two different types of images. Figure 5 and figure 8 are the original images, and the other images are segmented images in different clusters. It's not difficult to observe that the algorithm works better on images with clear forms or with a high contrast. Those results are obtained for a U matrix randomly generated by a matrix generator, who acts from the rules imposed to that matrix.



Fig. 5. Original image 1

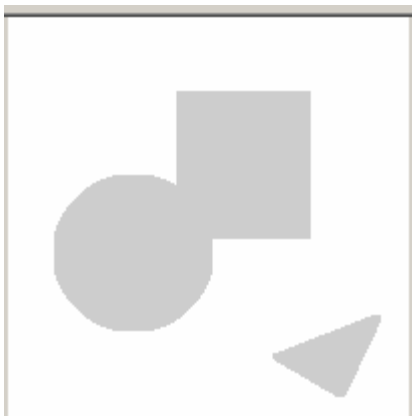


Fig. 6. Segmented image 1 in cluster 1

Those results are obtained for a U matrix randomly generated by a matrix generator, who acts on the rules imposed to that matrix. The used values for parameters are the default ones $m = 2$; $\varepsilon = 0.3$; Fuzzy limit = 0.4, except for one experiment, $m = 2$, $\varepsilon = 0.3$, fuzzy limit = 0.



Fig. 7. Segmented image 1 in cluster 2



Fig. 8. Original image 2



Fig. 9. Segmented image 2 in cluster 1

It's not difficult to observe that the algorithm works better on images with clear forms, or with a high contrast.

6. CONCLUSIONS

Image processing field, and even more fuzzy image processing, is a very large field, and for best results, user's experience is essential. As you can see, one algorithm has not the same performance on all types of image and the chosen parameters could have a good or bad influence on the results. So the algorithm should be personalized on the image characteristics.



Fig. 10. Segmented image 2 in cluster2



Fig. 11. Segmented binarized image 2 in cluster3

That's why a good knowledge of the application field is very important.

Fuzzy c-means algorithm works better on images with high contrast, or clear forms. Of course, better results can be obtained with different values of the parameters for example $\varepsilon = 0.01$ in stead of $\varepsilon = 0.3$. But better results have their own price, in this case the computing effort and elapsed time. Another parameter that can be modified, fuzzyness coefficient $m = 2$. A large m produces smaller memberships u_{ij} , and also fuzzier clusters. For $m=1$, you can obtain a crisp segmentation.

A further developpement of this application means adding new tools (like fuzzy thresholding) or improving the existent ones.

7. REFERENCES

- Bezdek, J. C. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York
- Bezdek, J. C. and S. Pal (eds.), (1991), *Fuzzy Models for Pattern Recognition*, IEEE Press.
- Chang, C.-W.; Ying, H.; Hillman, G.R.; Kent, T.A. and Yen, J. (1998), *A rule-based fuzzy segmentation system with automatic generation of membership functions for pathological brain MR images*. *Computers and Biomedical Research*, <http://gopher.cs.tamu.edu/faculty/yen/publications/index.html>.
- Cocquerez, J-P., S. Philipp, (1995), *Analyse d'images: filtrage et segmentation*, ed. Masson, Paris.
- Cojocaru, D., (2002), *Achiziția, prelucrarea și recunoașterea imaginilor*, Editura Universitaria, Craiova.
- Cuevas, E., D. Zaldivar, R. Rojas, (2004), *Fuzzy segmentation applied to face segmentation*, Intelligent Bipedal Robot Project, Technical Report B-04-09.
- Hamid R. Tizhoosh, (1997), *Fuzzy Image Processing: Introduction in Theory and Practice*, Springer-Verlag.
- Pham D.L and Prince J.L., (1999), *An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneities*, *Pattern Recognition Letters* vol. 20, no.1, pp.57-68.
- Tolias Y.A and Panas S.M., (1998), *On applying spatial constraints in fuzzy image clustering using a fuzzy rule-based system*, *IEEE Signal Processing Letters*, vol.5, no.10, pp. 245-247.
- Yen J. and Langari R., (2000), *Fuzzy logic, intelligence, control and information*, Prentice Hall, New York.
- Zhang D.Q and Chen S.C., (2002), *Fuzzy clustering using kernel methods*, *Proceedings of the International Conference on Control and Automation, Xiamen, China*.