Fuzzy Clustering and Images Reduction *

A. F. Gómez-Skarmeta[1], J. Piernas[1] and M. Delgado[2]

[1] Dept. Informática y Sistemas. Universidad de Murcia, Spain
[2] Dept. Ciencias de la Computación e I.A. Universidad de Granada, Spain

Abstract. In this paper we present an efficient method for estimating the significant points of a gray level image by means of a fuzzy clustering algorithm. This method can be used to reduce the resolution of the image so it can be transmited and later reconstructed with the greatest reliability. We will show how using less than $0,01$ of the original information it is possible to reconstruct an image with a considerable level of detail.

1 Introduction

Clustering of numerical data forms the basis of many classification and system modeling techniques. The purpose of clustering is to distill natural groupings of data from a large data set, producing a concise representation of the original information. Fuzzy clustering in particular has shown excellents results, because it has a no restrictive interpretation of the membership of the data to the different clusters. Within the fuzzy clustering algorithms, the Fuzzy C-means [1] has been widely studied and applied in different environments [3] [4].

In our case we are interested in the reduction of a gray level image replacing it by its more significant points, so that in a later moment we can reconstruct the original image with the greatest precision, or in other words with the less information loss. The number of points must be as small as possible in relation with the points of the original image, but trying to preserve the more information as possible of the original image, in order that the later reconstruction allows its regeneration it with great reliability.

In order to obtain these significant points, we use the Mountain Clustering Algorithm. This is a fuzzy clustering algorithm developed by Yager and Filev [7] which can be used for estimating by means of a simple and effective algorithm, the number and location of fuzzy cluster centers . Their method is based in griding the data space and computing a score value for each grid point based on its distance to the actual data points; a grid point with many data points nearby will have a high potential value. The grid point with the highest potential value is chosen as the first cluster center. Once a centroid is detected the potential of all grid points is reduced according to their distance from the cluster center. This process is repeated until the potential of all grid points fall below a threshold.

* This work has been partially supported by CICYT project TIC95-1019

This method is specially suitable for our problem, because we work with retinal or polar images. Images which griding try to simulate the eyes vision using a radial griding over a circular image, having in this way a great resolution near the central point of focus and with a smaller resolution as we move away from the focus. This is an important difference with other applications of the Mountain Clustering Algorithm [2]. In this context a natural alternative is to reduce the number of points in each radial axis or radii, searching for the most representative points corresponding to the original data set, producing in this way a concise representation of the original information. As we have indicated, this is the main objective of clustering in general and fuzzy clustering in particular. Furthermore, as the representation we have adopted in the image treatment is similar in comparison with the Mountain Clustering technique, this latter is an excellent candidate as a clustering algorithm and an alternative to other proposals in the literature [5].

Regardless of other possible applications of this technique as image compression or image storage, in our case it is part of a vision component of an autonomous system which main objective is to capture images and send them to a remote controller where the whole image is reconstructed to be processed, or just a partial zone where some object of interest could be located based on some predictions is reconstructed.

In the next section we present the basic ideas behind the Mountain Clustering Algorithm. Then in section 3 we show how we apply this technique in our image reduction/reconstruction method using the centroids obtained by the Mountain Clustering algorithm. In section 4 we discuss the experimental results obtained with our method, and finally in section 5 we indicate some future trends.

2 Mountain Clustering Algorithm Overview

In the following we shall briefly explain the basis ideas of the Mountain Clustering method (MC). For simplicity, we shall focus on two dimensional space, but the generalisation of the result is straightforward.

The MC can be seen to be a three step process. In the first step we discretize the object space and in doing so generate the potential cluster centers. The second step uses the observed data, the objects to be clustered, to construct the mountain function. The third step generates the cluster centers by a iterative destruction of the mountain function.

First Step

Assume the data consist of a set of q points (x_k, y_k) in the $\Re^2$ space. We restrict ourselves to the rectangular subspace $X \times Y$ of $\Re^2$ containing the data points. The first step in the MC is to form a discretization of $X \times Y$ space by griding X and Y with r_1 and r_2, respectively, equidistant lines (although this is not obligatory). The intersection of these grid lines, called nodes, form out set of potential cluster centers. We shall denote this set as N, and an element in N as N_{ij} and with (X_i, Y_j) indicating the node obtained by the intersection of the grid lines passing through the lines at X_i and at Y_j.

As we shall subsequently see the purpose of this discretization is to turn the continuos optimisation problem of finding the centers into a finite one.

Second Step

In this step we shall construct the mountain function M, which is defined on the space N of potential cluster centers:

$$M : N \longrightarrow \Re$$

The mountain function M is constructed from the observed data by adding an amount to each node in N proportional to that nodes distances from the data point. More formally for each point N_{ij}, (X_i, Y_j), in N

$$M(N_{ij}) = \sum_{k=1}^{q} e^{-\alpha \times d(N_{ij}, O_k)}$$

where O_k is the kth data point (x_k, y_k), α is a constant and $d(N_{ij}, O_k)$ is a measure of distance between N_{ij} and O_k typically, but not necessarily, measured as

$$d(N_{ij}, O_k) = (X_i - x_k)^2 + (Y_j - y_k)^2$$

Obviously the closer a data point is to a node the more it contributes to the score at that node. It is evident from the construction of the mountain function that its value are approximations of the density of the data points in the neighbourhood of each node. The higher the mountain function value at a node the larger is its potential to be a cluster center.

Third Step

The third step in the MC is to use the mountain function to generate the cluster center. Let the node N_1^* be the grid point with maximal total score, the peak of the mountain function. We shall denote its score by $M_1^* = Max_{ij}[M(N_{ij})]$. If there are more than one maxima we select randomly one of them. We designate this node as the first cluster center and indicate its coordinates by $N_1^* = (x_1^*, y_1^*)$. We next must look for the next cluster center, so we must eliminate the effect of the cluster center just identified because usually this peak is surrounded by a number of grid points that also have high scores. This means that we must revise the mountain function for all the other nodes. The process of revision can be seen as a destruction of the mountain function, and is realised by subtracting from the total score of each node a value that is inversely proportional to the distance of the node to the just identified cluster center, as well as being proportional to the score at this just identified cluster center. More specifically, we form a revised mountain function $\hat{M}_2$ also defined on N such that

$$\hat{M}_2(N_{ij}) = \hat{M}_1(N_{ij}) - M_1^*(N_{ij}) * e^{-\beta \times d(N_1^*, N_{ij})}$$

where $\hat{M}_1$ is the original mountain function M, β is a positive constant, N_1^* and M_1^* are the location of and score at the just identified cluster center and $d(N_1^*, N_{ij})$ is a distance measure.

We now use the revised mountain function $\hat{M}_2$, to find the next cluster center by finding the location N_2^* , and score M_2^* , of its maximal value. N_2^* becomes our second cluster center. We then revise our mountain function to obtain $\hat{M}_3$ as

$$\hat{M}_3\left(N_{ij}\right) = \hat{M}_3\left(N_{ij}\right) - M_3^*\left(N_{ij}\right) * e^{-\beta \times d\left(N_2^*, N_{ij}\right)}$$

and so on.

This process, that can be seen as a destruction of the mountain function, ends when the score of the last found cluster center is less than a constant δ. This means that there are only very few points around the last cluster center and it can be omitted.

The main advantage of this method is that it does not require a predefined number of clusters. It determines the first m cluster centers that satisfy the stopping rule, starting from the most important ones which are characterised with maximal value of the mountain functions at nodes $N_1^*, N_2^*, ..., N_m^*$ with coordinates $\left(x_1^*, y_1^*\right), \left(x_2^*, y_2^*\right), ..., \left(x_m^*, y_m^*\right)$.

3 Image Reduction/Reconstruction

3.1 Mountain Function Alternatives

The first question to address in the applicability of the Mountain Clustering Algorithm is the determination of the possible centroid candidate points. As we have indicate in the introduction we are working with a retinal or polar vision where we transform a rectangular image obtained from a camara, in a circular one where the new image points are distributed in an equidistant way around concentric circumferences with different radii from a focal point.

Although in this image treatment there are areas in the original image that are not considered and which information is lost as can be seen in figure 1, it has shown good behaviour in machine vision though it let simplificate the original image and concentrate the vision in the objects around the focal point in a similar way as human beings do.

In our case we are going to trace r radii and s concentric circumferences, having in this way $r * s + 1$ possible centroid candidates including the focal point or centre of the image. Along each radial axis or radii, we will obtain C centroids using the Mountain Clustering Algorithm, thus finally the original image will be replaced by a collection of $C * r + 1$ centroids that will concentrate the more important information representing the image and that will permit us later its reconstruction.

Once we have established the centroid candidates, the next step is to define the mountain function to be aplied to each of them. In this point a critical decision appears that will determine the success of our method. What we mean by a centroid?. There are different alternatives:

- A centroid will represent a weighted average of the different gray levels of the points around it.

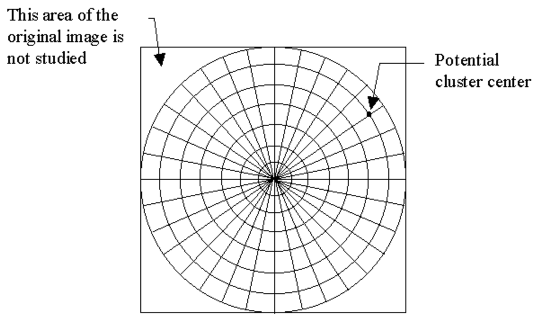


Fig. 1. Griding applied to the Image

- A centroid candidate will be more candidate as its gray level is equal or similar to the gray levels of the points that surround it.
- A centroid candidate will be more candidate as its gray level is different to the gray levels of the points that surround it.

The first alternative leads to consider as centroids only the points that are in light areas of the images, reducing the possibilities of the points in darker areas. The mountain function value for each candidate is calculated by means of the gray level sum of all the points around the candidate, weighting each gray level by a distance exponential function.

The second alternative indicate us that the centroids are those candidate points whose gray level is much similar to those points around it. The mountain function value for each candidate point is calculate as the inverse of the neighbourhood points' gray level sum, weighting each gray level by means of a distance exponential function to the candidate.

The third alternative places the centroids in those zones where a uniform color does not exist. The mountain function value is calculate in a similar way as in the second alternative, but without consider the inverse of the sum.

Within the different alternatives, the third one has shown the best results. The reasons are clear: from an image we are interested in the points that represent the most significant information, in order to use them in the later reconstruction. This reconstruction will be better as we can recognize the objects that are present in the image. Thus, the points that contain the most representative information will be those that are in the border between objects, small areas that are characterized by significant changes in the gray level of the points. In this way, the third alternative for the mountain function is the more adequate in order to represent this situation.

In our case, the mountain function is defined using the expression:

$$M(N_{ij}) = \sum_k |(Gray(N_{ij}) - Gray(O_k))| * e^{-\alpha \times d(N_{ij}, O_k)}$$

where N_{ij} is a centroid candidate, $Gray(\cdot)$ is a function that returns the gray level of the original image for each point, α is a constant, $d(\cdot)$ is the euclidian distance and O_k is a point in the neighbourhood of N_{ij}. We must remark that in our method, during the mountain function evaluation in each point, we are not going to consider all the possible image points as it is done in the original Mountain Clustering Algorithm. Instead we only take into acccount the original image points that are close to each centroid candidate, within certain context using ideas from [4] [6].

Once we have obtained the c possible centroid candidate in each radii, and once we have calculated the mountain function value in each of these points, what remains is to apply the Mountain Clustering Algorithm as indicated in section 2, in order to find the C centroids over each of the radii.

3.2 Image Reconstruction from the Centroids

Once we have found the C centroids over each of the radii, we need a method to reconstruct the original image starting from this reduced number of points. We must remark that we have obtained excellent results for real images of $450 * 350$ pixels, using 18 centroids for each of the 50 radius. In other words, with our method we can reconstruct the image by means of only $900 (= 18 * 50)$ centroids representing just the $0,0057$ of the $157.500 (= 450 * 350)$ points of the real image.

In order to reconstruct the original image we must proceed in the following way. We can consider the gray level of each point as the return value of a two variables function, x and y which represents the coordinates of the point. The graphical representation of this function will be a "colour mountain" where the hills represent the light zones and the valleys the dark ones. The centroids represent the points of this colour montain that will let us reconstruct it with minor information loss.

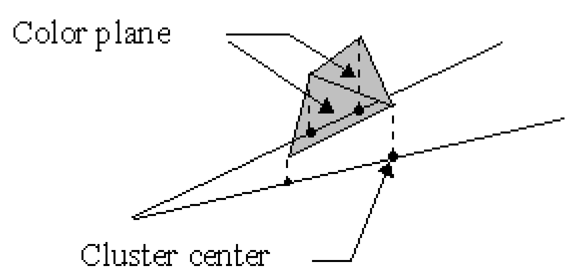


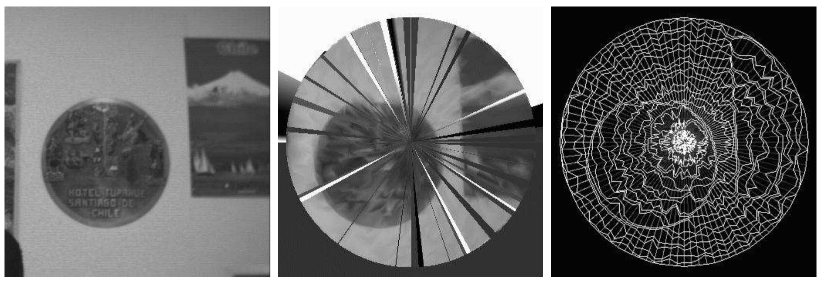
Fig. 2. Colour Plane

As we can see in figure 2, the centroids of each radius are paired with the centroids of the next radius, so a quadrilateral is formed by the two centroids of one radii and the two centroids of the other one. This quadrilateral will be divided in two components with three points each and two of them in common. The objective of this operation is to characterize by means of the three points just indicated the "colour plane", which equation lets us infer the corresponding colour to any point within the triangle formed by the three points. Figure 2 show these ideas.

In this way and using all the quadrilaterals that the centroids generate, we can reconstruct the original image.

4 Experimental Results

Experiments have been realized with real images and synthetic ones and several interesting conclusions have been obtained.



a **b** **c**

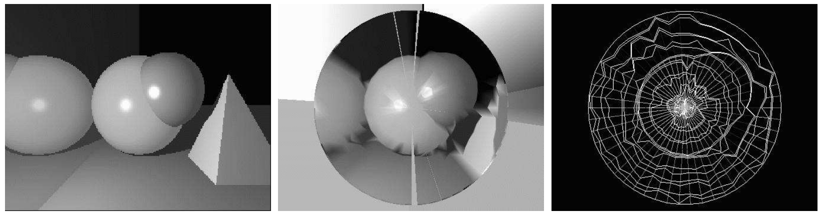
Figure 3: Original Image / Reconstructed Image / Clusters Center and Quadrilaterals

In figures 3.a and 3.b[3] we can see the real image and its reconstruction after applying our methods with 2101 cluster centers in all. As it can be noted and in spite of the low number of cluster centers, the results are enough satisfactory noticing clearly all the objects in the image although, evidently, without the same resolution. An interesting event that is shown in Figure 3.c, where quadrilaterals constituted by the cluster centers for the above image are shown up, is that the borders of the objects in the real image can been identified. That is, the cluster centers have adapt theirselves to recognize those areas in which important changes of color take place and certainly among these areas are the object borders because, otherwise, the human eye would not recognize the border either.

It is important to emphasize that in spite of the image darkness, we have obtained very similar results to the above ones with different iluminations. Hence we can conclude that our method is not much sensitive to image ligthing, being this a very interesting property.

Figures 4.a 4.b and 4.c show the application of our method on a synthetic image. In this case, we can note some anomalies in the reconstructed image that distort it. These anomalies arise because two adjacent radii cross very different zones, so cluster centers do not uniformely distribute on each radius, but tends

[3] The reconstructed image has a little noise due to the reproduction process



a **b** **c**

Figure 4: Original Image / Reconstructed Image / Clusters Center and Quadrilaterals

to concentrate on certain radii parts. Moreover as the number of cluster centers over both radius is the same and they match theirselves correlatively (the first with the first, the second with the second, and so on), it is posible that some quadrilaterals constituted by the cluster centers get longer longitudinally following radius direction, giving rise to the appearance of that "peaks" in the reconstructed image.

There are several posibilities that can help us to alleviate these anomalies. One solution is to increase the number of radii. Another one is to draw a new radius between two existing radii when the cluster center behaviour is enough different over both. One additional solution is to restrict cluster centers mobility on a radii. In order to achieve this, we can divide the radius in several parts (like having concentric rings) and fixing the number of cluster centers for each part. In this way, we achieve an uniforme distribution of cluster centers along each radius.

5 Conclusion and Future Trends

In this paper we have shown how the use of a fuzzy clustering technique like the Mountain Clustering Algorithm allows the reduction of the information needed to be maintained/transmited, relative to a gray level image preserving the more information about it, so later it can be reconstructed with greater reliability. The examples presented have shown that the method obtain good results either in the case of real images with adequate or slight ilumination, or in the case of synthetic images.

Although the results obtained indicate the good performance of the method proposed, there are some points that need further improving. Specifically the mountain function evaluation is a computational expensive process. In order to improve its evaluation it could be interesting to consider just a random collection of points in the centroids' neighbourhood. Another important aspect to reconsider is the way the centroid candidates are elected. Instead of assigning them to fixed positions over the radius, we could give them more movility, so they could move to more significant zones in relation to the final reconstruction of the image, although with an increase in the complexity of the algorithm.

References

1. Bezdek, J.C., "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenun, New York, 1981.
2. Chiu, L.S., "A Cluster Estimation Method with Extension to Fuzzy Model Identification", *Proceedings of the 3rd IEEE Int. Conference on Fuzzy Systems*, Orlando, USA, pp. 1240-1245, 1994.
3. Delgado, M.; Gómez-Skarmeta, A.F.; Vila, M.A., "Hierarchical Clustering to Validate Fuzzy Clustering", *Proceedings of the 4th IEEE Int. Conf. on Fuzzy Systems*, pp. 1807-1812, 1995.
4. Delgado, M.; Gómez-Skarmeta, A.F.; Martín, F., "Generating Fuzzy Rules using Clustering Based Approach", *Proc. Third European Congress on Fuzzy and Intelligent Technologies and Soft. Computing*, Aachen, Germany, 810-814, 1995.
5. Djirar, S.; Banks, S.P.; Williams, K., "Dynamical Maps Applied to Image Compression", *Proc. Sixth Int. Conference Information Procesing and Management of Uncertainty in Knowledge-Based Systems IPMU'96*, Granada, Spain, pp. 641-645, 1996.
6. Karayiannis, N.B., "MECA: Maximun Entropy Clustering Algorithm", *Proceedings of the 3rd IEEE Int. Conference on Fuzzy Systems*, Orlando, USA, pp. 630-635, 1994.
7. Yager, R.R.; Filev, D.P., "Generation of fuzzy rules by mountain Clustering", *Tech. Report* MII-1318, 1993.