view fields (case #5 in Table V), correct results are obtained for threshold values in a very narrow range of $0.58 \leq \varepsilon_0 \leqq 1.44$ m. Therefore, the threshold value ε_0 should be determined carefully.

② The freedom of threshold is three times greater when a matching evaluation function which decreases gradually with distance is used than with a fixed degree of matching (compare #4 and #5).

③ The freedom of threshold is further doubled by confining the overlapping point search area to the overlapping view field instead of the whole area in which object points exist (compare #1 with #2, and #3 with #4).

④ There is no distinct difference between the freedoms of threshold for cases with a fixed threshold value regardless of the object point position and cases with a variable threshold dependent on the distance between the view point and the object point (compare #1 with #3, and #2 with #4). However a threshold value with which the algorithm operates effectively and normally with a short processing time, is determined by a physically clear basis without trial and error.

VI. Conclusion

A new method for integrating two adjacent small area maps into a larger area map is proposed and demonstrated experimentally, in which their common object points are automatically found by utilizing the view field information when the maps are formed. Here, the freedom of threshold, which is the ratio of the maximum and minimum threshold values providing true matching, is proposed as a measure of accuracy of the matching algorithm. This parameter is enlarged twice or three times by confining the search area for overlapping object points into the overlapping view field and by using a function which decreases gradually with distance as the matching evaluation function.

A threshold value with which the algorithm certainly operates normally with a short processing time can be determined by utilizing the characteristics of stereo vision metrology, that is, the magnitude of position errors depends on the distance between the view point and the object point.

Acknowledgment

The authors would like to thank Drs. K. Ishihara, M. Kawase and Y. Koyamada for their continuous encouragement.

References

[1] J. C. Simon, A. Checroun and C. Roche, "A method of comparing two patterns independent of possible transformations and small distortions," *Pattern Recogn.*, vol. 4, no. 1, pp. 73–81, 1972.

[2] S. Ranade and A. Rosenfeld, "Point pattern matching by relaxation," *Pattern Recogn.*, vol. 12, no. 4, pp. 269–275, 1980.

[3] H. Ogawa, "Labeled point pattern matching by Delauney triangulation and maximal cliques," *Pattern Recogn.*, vol. 19, no. 1, pp. 35–40, 1986.

[4] Z. Lin, H. Lee and T. S. Huang, "Finding 3-D point correspondences in motion estimation," *Int. Conf. on Pattern Recogn.*, pp. 303–305, 1986.

[5] S. Umeyama, "A point pattern matching algorithm," *Trans. IEICE of Japan,* D-II, J72-D-II, **2**, pp. 218–228, 1989.

[6] H. Zhang, M. Minoh and K. Ikeda, "Fast algorithm for point pattern matching by using ordered list," *Trans. Information Processing Society of Japan,*, vol. 31, no. 7, pp, 1005–1014, 1990.

[7] T. Kobayashi, N. Ohnishi and N. Sugisawa, "Recognition of partially occluded 2-D objects using knowledge of local shapes," *IEIEC Technical Digest,* IE89-113, pp. 39–46, 1989.

[8] D. J. Kriegman, E. Triendl and T. O. Binford, "Stereo vision and navigation in buildings for mobile robots," *IEEE Trans. Robotics and Automation,* , vol. 5, no. 6, pp. 792–803, 1989.

[9] R. K. Lenz and R. Y. Tsai, "Technique for calibration of the scale factor and image center for high accuracy 3D machine vision metrology," *Proc. IEEE, Int. Conf. on Robotics and Automation,* pp. 68–75, 1987.

[10] Y. Nomura, M. Sagara, H. Naruse and A. Ide, "High accuracy camera calibration for 3-D computer vision," *Proc. IECON'89,* pp. 480–485, 1989.

Approximate Clustering Via the Mountain Method

Ronald R. Yager and Dimitar P Filev

***Abstract*—We develop a simple and effective approach for approximate estimation of the cluster centers on the basis of the concept of a mountain function. We call the procedure the mountain method. It can be useful for obtaining the initial values of the clusters that are required by more complex cluster algorithms. It also can be used as a stand alone simple approximate clustering technique. The method is based upon a griding on the space, the construction of a mountain function from the data and then a destruction of the mountains to obtain the cluster centers.**

I. Introduction

The method of Fuzzy C-Means (Fuzzy ISODATA) has been the dominant approach in both theory and practical applications of fuzzy techniques to unsupervized classification for almost two decades. [1]

According to [2]–[5] if our data, $\{x_1, \ldots, x_n\} \subset R^s$, are n points in the s-dimensional space, the Fuzzy C-Means (FCM) defines a soft clustering into $q < n$ clusters, characterized by cluster centers:

$$x_i^* = \frac{\sum_{k=1}^{n} v_{ik}^* x_k}{\sum_{k=1}^{n} v_{ik}^*}, i = (1, q)$$

and membership grades in fuzzy subsets focused at these cluster centers:

$$v_{ik}^* = \frac{\frac{1}{|x_k - x_i^*|^m}}{\sum_{j=1}^{q} \frac{1}{|x_k - x_j^*|^m}}, i = (1, q), k = (1, n) \text{ and } m \geq 1$$

In the above $|.|$ denotes any inner product norm induced on R^s (in particular for $m = 2$ we get the Euclidean norm). If for some k and i, $|x_k - x_i^*| = 0$, a singularity occurs, then v_{ik}^* for all i are any nonnegative numbers such that:

$$\sum_{i=1}^{q} v_{ik}^* = 1 \text{ and } v_{ik}^* = 0 \text{ if } |x_k - x_i^*| \neq 0$$

Such a partitioning provides a necessary condition for the local minimization of the FCM objective function J_m:

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{q} v_{ik}^m |x_k - x_i|^m \quad (m \geq 1)$$

Manuscript received November 6, 1992; revised March 23, 1993 and September 18, 1993.

The authorsd are with the Machine Intelligence Institute, Iona College, New Rochelle, NY 10801 USA.

IEEE Log Number 9402289.

The FCM approach is essentially an iterative technique which starting with cluster centers generates membership grades which are used to induce new cluster centers.. The process continues until it stabilizes.

One difficulty with this approach is the estimation of the initial values of the cluster centers. This fact along with the local minimizing property of the FCM algorithm can sometimes complicate the process of clustering.

In this paper we suggest a new heuristic algorithm for the determination of the cluster centers. This algorithm can be used in two modes. In one mode this approach can serve as a tool to obtain the initial estimation of the cluster centers that are used in the fuzzy c-means algorithm. In this mode the aim of this approach is to provide a tool for supporting the FCM method. However, in situations where only approximate, not too exact, values of the cluster centers are needed, this approach can act as a stand alone clustering algorithm. The method introduced here is called the **Mountain Method (MM)** and its spirit is based upon what a human does in visually forming clusters. As we shall subsequently see this approach could be particularly useful in an interactive mode. The idea of this approach is rather straightforward. The first step is to form a discretetization of the object space R^S by forming a grid on R^S(see Fig. 1). The intersection of the grid lines, which occurs at what we shall call the vertex or node points, provides our desired discretetization. We shall denote the finite subset of R^S consisting of the vertices as V. The set of points in V constitute our candidates for cluster centers. Thus we see the degree approximation of our final centers is very sensitive to the fineness of our griding. The finer the griding the less approximate but, more calculations are needed. It should be noted that the griding need not be uniform throughout the space R^S; in different parts of the space we can have a different density of griding. This facility for variable griding can be among other things be used to reflect information we have about the objects to be clustered. The second step is the introduction of the data and construction of what we call the mountain function. The mountain function which we denote as M is defined on V and is constructed as follows. For each piece of data, x_i, we add an amount to the M value at each point v in V. The amount added depends on the distance of v from x_i; the closer the two the more added. In this way after all the data points have been considered we have a function on R^S, actually V, which looks like a mountain range reflecting the distribution of the data. The next step is the selection of the cluster centers. This is accomplished by the destruction of the mountains. We find the point in V, c_1, which has the greatest value for M, the peak of the mountain range, this becomes our first cluster center. For all points v in V we subtract from their M value a quantity dependent upon its distance from c_1 and the value $M(c_1)$. The effect of this subtraction is to reduce the mountains. We next look for the new peak. This becomes our next cluster center, c_2. We now use c_2 and its value to further reduce our mountain function. We continue in this manner until the mountain function is virtually destroyed.

The introduction of the griding allows us to approximate a continuous optimization problem by a finite one. The destruction of the mountain function essentially allows us to eliminate the effects of already identified cluster centers when searching for new centers.

II. THE CONCEPT OF A MOUNTAIN FUNCTION

Let us assume a collection of n data points $\{x_1, \ldots, x_n\}$ in the s dimensional space R^s. We shall denote by x_{kj} the j-th coordinate of the k-th point, where $k = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, s$. We shall without loss of generality, restrict the s dimensional space R^s to an s dimensional hypercube $I_1 \times \cdots \times I_s$ where the intervals $I_j, j = 1, 2, \ldots, s$ are defined by the ranges of the coordinates x_{kj},

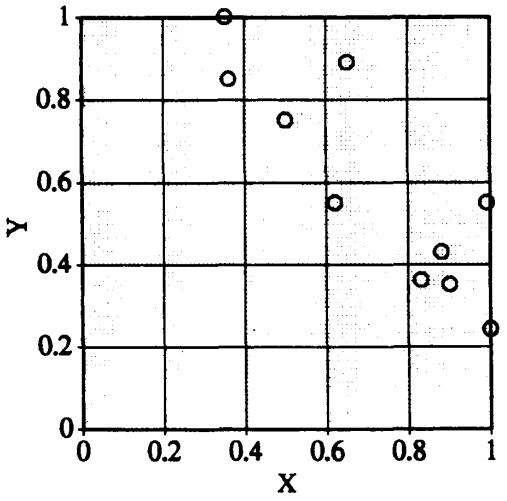

Fig. 1. Example 1: Given data set.

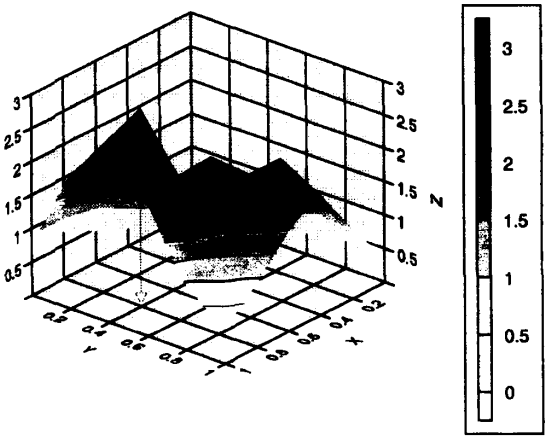

Fig. 2. Example 1: The mountain function.

i.e.:

$$I_j = [\min_k(x_{kj}), \max_k(x_{kj})]$$

Evidently the hypercube contains all the points of the data set $\{x_1, \ldots, x_n\}$. Further we shall discretize each of the intervals I_j into r_j equidistant points. Such a discretization forms an s dimensional grid in the hypercube with nodes $N_{(i_1,\ldots,i_s)}$, where indices $i_1, \ldots, i_s$ take values from the sets $[1, \ldots, r_1], \ldots, [1, \ldots, r_s]$. We shall denote the equidistant coordinates of the grid nodes by $X_1^{(j)}, \ldots, X_{r_j}^{(j)}$, where points quantize the interval $I_j$and $j = (1, s)$. The grid discretizes the space restricted by the hypercube. We emphasize the significance of this discretization: the grid nodes are our potential cluster centers. On one hand the coarser the discretetization, fewer nodes, the less calculations are required but also the coarser the final cluster center values. We emphasis the fact that the process of obtaining the set of potential cluster centers, the nodes, need not be based upon a uniform griding of the space, we can use a variable griding of the space. More generally any technique which provides a representative selection of points from the space can be used. In the

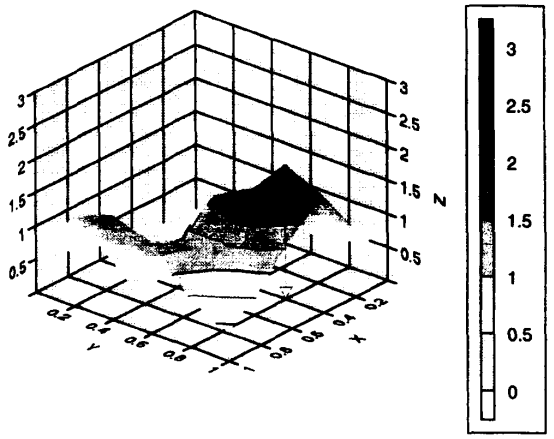

Fig. 3. The mountain function after removing the first center.

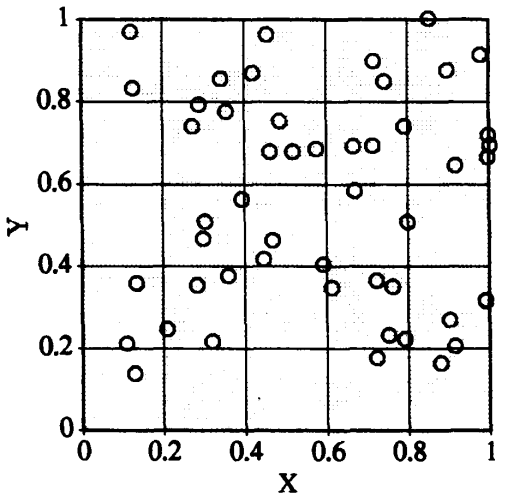

Fig. 4. Example 2: Original data set.

following we shall use the simpler notation of N_i to indicate a node with the implicit fact that i equals some tuple of the form $(i_1, \ldots, i_s)$.

We shall look at any of the grid nodes as a possible candidate to become a cluster center. Let $d(x_k, N_i)$ be the distance from the data point x_k to the grid nodes N_i. Consider the simple case of two dimensional data, $S = 2$. We assume a discretization of I_1 into 3 and I_2 into 4 points. In this case we can express $x_k = (x_{k1}, x_{k2})$. Furthermore our discretetization gives us

$$I_1 = \{X_1^{(1)}, X_2^{(1)}, X_3^{(1)}\},$$
$$\text{and } I_2 = \{X_1^{(2)}, X_2^{(2)}, X_3^{(2)}, X_4^{(2)}\}.$$

In the case where $i = (1,3)$ the distance is defined as follows:

$$d(x_k, N_i) = (|x_{k1} - X_1^{(1)}|p + |x_{k2} - X_3^{(2)}|p)^{\frac{1}{p}}$$

TABLE I
GIVEN POINTS IN THE R^2 SPACE

k	x	y
1	0.36	0.85
2	0.65	0.89
3	0.62	0.55
4	0.50	0.75
5	0.35	1.00
6	0.90	0.35
7	1.00	0.24
8	0.99	0.55
9	0.83	0.36
10	0.88	0.43

TABLE II
NUMERICAL VALUES OF THE MOUNTAIN FUNCTION

		X					
		0.00	**0.20**	**0.40**	**0.60**	**0.80**	**1.00**
	0.00	0.06	0.12	0.24	0.44	0.68	0.72
	0.20	0.11	0.23	0.48	0.96	1.71	1.97
Y	**0.40**	0.19	0.42	0.88	1.64	2.94	2.52
	0.60	0.30	0.71	1.41	2.19	2.12	1.97
	0.80	0.40	1.07	2.15	2.13	1.46	0.94
	1.00	0.38	1.02	1.79	1.45	0.92	0.49

TABLE III
NUMERICAL VALUES OF THE MOUNTAIN FUNCTION AFTER REMOVING THE FIRST CENTER

		X					
		0.00	**0.20**	**0.40**	**0.60**	**0.80**	**1.00**
	0.00	0.04	0.07	0.11	0.20	0.36	0.48
	0.20	0.08	0.15	0.24	0.35	0.75	1.37
Y	**0.40**	0.15	0.32	0.55	0.68	0.00	1.56
	0.60	0.27	0.63	1.17	1.59	1.17	1.37
	0.80	0.38	1.02	2.01	1.89	1.15	0.70
	1.00	0.37	1.00	1.74	1.36	0.82	0.41

In particular we can consider $p = 1$ or $p = 2$ (Euclidean distance).

As we discussed in the introduction, we construct a mountain function defined over the set grid nodes by adding, for each data, an amount to each node proportional to its distance from the data point.

In the following we shall use the distance measure $d(x_k, N_i)$ to score the contribution of the data point x_k to the value of the the mountain function at the grid node N_i. We shall assign higher scores to the nodes that are closer to a data point. The following function provides the value our mountain function at the vertex point N_i

$$M(N_i) = \sum_{k=1}^{n} e^{-\alpha\ d(x_k, N_i)} \qquad (I)$$

In the above α is a positive constant.

It is evident from the form of function (I) that the closer a grid node N_i is to the data point x_k the higher is the score supplied by the exponential term of (I); it is maximal for a data point that coincides with the grid node; it is decreasing exponentially for the farther grid nodes. We can look at the values of function $M(N_i)$ as heights of a mountain range, having as a base the grid. We call the function $M(N_i)$ the **mountain function**. The value of the mountain function can be seen to be closely related to the *density* of data points in the neighborhood of the node. As such it can be seen to represent the potential ability of any of the grid nodes to be a cluster center. The higher the mountain function value the larger is this potential ability.

Therefore the mountain function value can be used as indicators of the clustering of the the hypercube from the data.

In the preceding we have suggested the use of an exponential type function in the construction of the mountain function, it is of course possible to use other type functions, such as triangular, to reflect the effect of the distance of the data point from the node point.

III. Estimation of the Cluster Centers from the Mountain Function

The basic idea of identification of cluster centers estimates is simple and straightforward. When using the mountain function we are systematically looking for global maxima and associating them with cluster centers. We start by looking for the grid node with the maximal mountain function value, the mountain range peak. If there are more than one maxima, we select randomly one of them. Let us denote the maximal value of the mountain function as M_1^* thus

$$M_1^* = \mathrm{Max}_i[M(N_i)].$$

We note it is the global maximum of the mountain function. Let the grid node N_1^* indicate the point in the hypercube where this maximal score of the mountain function is attained, it is a peak of the mountain range formed by the data set. We shall denote this peak node, N_1^*, and select it as the first cluster center.

It should be noted that this peak is *usually* surrounded by a number of grid points that also have high scores, this is due to the process of constructing the mountain function and the inherent continuity of the hypercube.

In order to find the next cluster center we must first eliminate the effects of the cluster center just identified. In order to accomplish this we subtract from our current mountain function a value at each node, this gives us a revised mountain function. The amount subtracted at each node point is proportional to the distance of the point from the maximal, the newly assigned cluster center, as well as being proportional to the current maximal value, M_1^*. In this manner we now form a new revised mountain function $\widehat{M}^2$ defined on the set of all nodes:

$$\widehat{M}^2(N_i) = \widehat{M}^1(N_i) - M_1^* \sum_{k=1}^{n} e^{-\beta\ d(N_1^*, N_i)} \qquad \text{(II)}$$

In the above $\widehat{M}^1(N_i)$ is actually the original mountain function, the value M_1^* is the current maximal and β is a constant.

To find the next cluster center we proceed in a similar manner. We find the node point, N_2^*, that maximizes the new reduced mountain function, $\widehat{M}^2$, this becomes our second cluster center we then remove the effects of this center in a manner similar to II giving us a new modified mountain function $\widehat{M}^3$ which we use in the next stage. More generally expression II can be written as

$$\widehat{M}^k(N_i) = \widehat{M}^{k-1}(N_i) - M_{k-1}^* \sum_{k=1}^{n} e^{-\beta\ d(N_{k-1}^*, N_i)} \qquad \text{(III)}$$

In the above $\widehat{M}^k$ is the new mountain function, $\widehat{M}^{k-1}$ the old mountain function, M_{k-1}^* the peak value of $\widehat{M}^{k-1}$, and the location of this peak value, the newly identified cluster center, is N_{k-1}^*.

We can look at this process as a process of destroying the mountain function. This process will guarantee that those nodes closer to the new identified cluster center will have their mountain more strongly reduced then the those further away. In some sense the idea of this approach is in the same spirit as Kohonen's [6] approach to self organization.

In the formulation we have suggested we have used an exponential function to introduce the effects of adding data and removing cluster centers one could use other functions, in particular a linear function may we used.

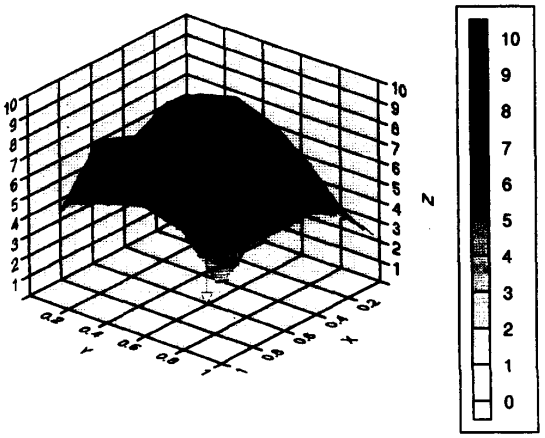

Fig. 5. Example 2: The mountain function.

In the examples that follow we have found it more effective to bound the new mountain functions below by zero, thus

$$\widehat{M}^k(N_i) = \max[\widehat{M}^{k-1}(N_i) - M_{k-1}^* \sum_{k=1}^{n} e^{-\beta\ d(N_{k-1}^*, N_i)}, 0].$$

We shall demonstrate the workings of the mountain method in the following example.

Example 1 We consider a set of 10 data points in R^2 space shown in Fig. 1 and listed in Table I. The intervals I_1, I_2 are $[0, 1]$ and $[0, 1]$. The grid for discretization, with $r_1 = r_2 = 6$, is shown in the same figure. The numerical values of the mountain function calculated according to (I) with $\alpha = 5.4$ are listed in Table II. In Fig. 2 we depict the mountain function in the 3D space. It is seen that the maximal value of the mountain function appears at node (0.8, 0.4) of the grid. Therefore we associate this node with the first cluster center estimate. To find the next (less important) cluster center we remove the effect of the first cluster by destroying the mountain function according to expression (III) with $\beta = 5.4$. The modified mountain function, after removing the first custer center, is depicted in Fig. 3. Its numerical values are listed in Table III. It can be seen by comparison of the entries of Table II and Table III the extensive destroying of the mountain function around the cluster center estimate (0.8, 0.4). By inspecting the modified mountain function we find the global maximum at grid point (0.4, 0.8). It becomes the estimated second cluster center.

We continue the production of new cluster centers and reduction of the mountain function until the level of the current maximum M_{k-1}^*, compared with the original maximum M_1^* becomes too low. This means that there are only very few points around this cluster center and it can be omitted. We shall stop the process of destroying the mountain function when the ratio:

$$\frac{M_1^*}{M_{k-1}^*} < \delta \qquad \text{(V)}$$

where δ is given parameter; we shall denote by p^* the step that satisfies the stop criterion (V). Obviously it defines $(p^* - 1)$ cluster centers.

We shall summarize the above discussion in the following algorithm:

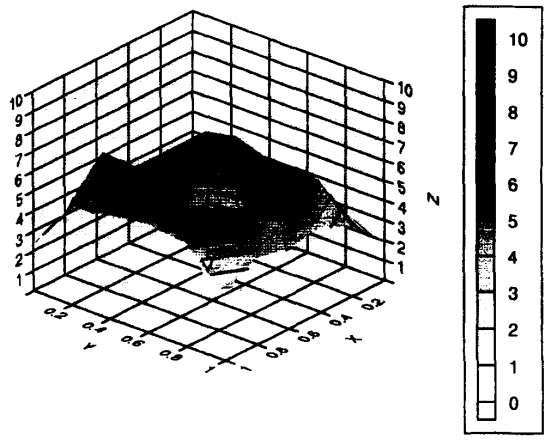

Fig. 6. Example 2: The mountain function after removing the first cluster center.

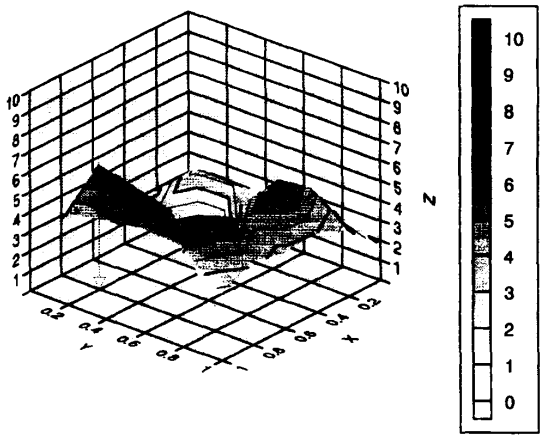

Fig. 7. Example 2: The mountain function after removing the second cluster center.

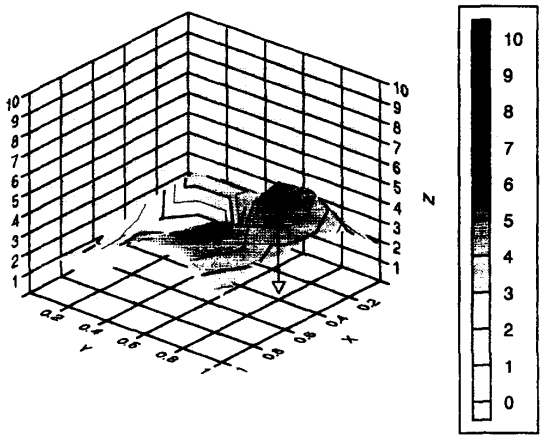

Fig. 8. Example 2: The mountain function after removing the third cluster center.

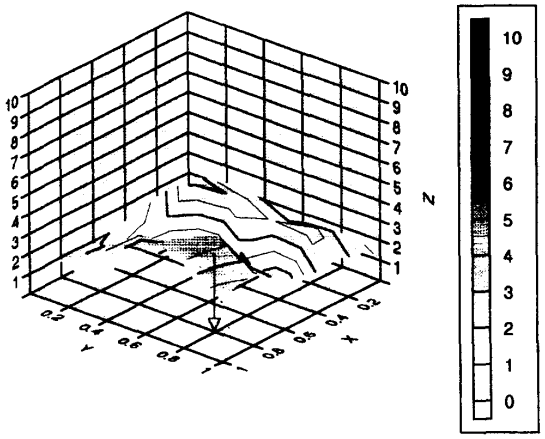

Fig. 9. Example 2: The mountain function after removing the fourth cluster center.

Algorithm

1) Calculate the intervals $I_j, j = (1, s)$.
2) Quantize the intervals and form the grid.
3) Calculate the mountain values according to (I).
4) Find the cluster centers estimates and modify the mountain function according to (III) until stopping rule (IV) is satisfied.

Example 2 We consider the two dimensional data set, that is presented on Fig. 4. Its mountain function is depicted in Fig. 5.

The first estimated cluster center estimate is found at the node (0.6, 0.6). The absolute maximum of the mountain function at this node is 9.94. The process of estimating the cluster centers and destroying the mountain function is illustrated in Figs. 6–9. The second center estimate is at node (0.4, 0.4); it is associate with an absolute maximum 6.94 of the modified mountain function that is presented on Fig. 6.

Similarly we obtain the next cluster center estimate at node (0.8, 0.2), where the peak value of the modified mountain function is 6.27, see Fig. 7.

In the same manner we proceed finding the next cluster center estimates by destroying the mountain function, see Fig. 8 and Fig. 9. This yields cluster centers (0.4, 0.8) and (0.8, 0.8).

The first five identified cluster centers are shown in Fig. 10. The result is consistent with the Fuzzy C-Means algorithm that finds cluster centers (0.83, 0.26), (0.25, 0.31), (0.62, 0.64), (0.90, 0.80), (0.32, 0.81).

Figs. 11 and 12 show the performance of the mountain method on two examples drawn from Dunn [7].

IV. CONCLUSION

A simple, easy to implement algorithm, for approximate clustering was presented in this paper. It is based upon the idea of griding the space and the concept of constructing and destroying the a mountain function. This method allows us to identify estimates of the cluster centers and can provide a quick and approximate technique for locating cluster centers. Its eventual combination with computer graphics will allow us to develop a natural and effective approach to clustering.

A number of issues regarding the workings of this approach require further investigation. The sensitivity of the performance of

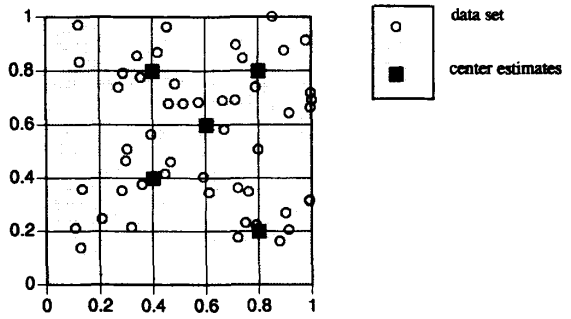

Fig. 10. Example 2: Original data set and estimated centers.

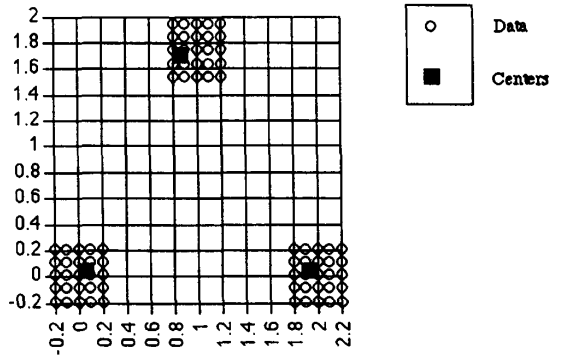

Fig. 11. Three clusters focused on vertices of an Isosceles triangle.

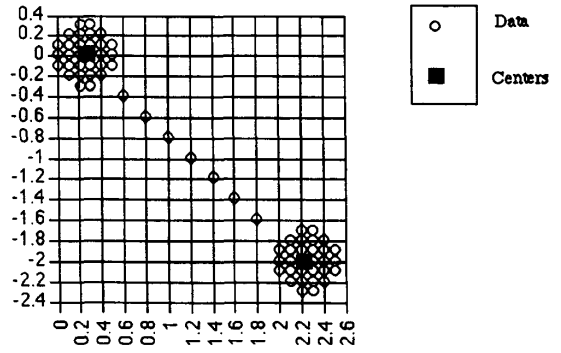

Fig. 12. Two clusters connected by a bridge.

the algorithm to the selection of the parameters α and β, the effect of the granularity of the griding and the selection of δ are some of the most important. The answer to these questions can best be obtained by experience using this approach.

In summary our initial investigations with this mountain method of clustering seem to indicate that it is a promising approach to fuzzy clustering and worth further investigation by those interested in clustering techniques.

References

[1] J. C. Bezdek and S. K. Pal, "Fuzzy models for pattern recognition," *IEEE Publications:* New York, 1992.
[2] J. C. Bezdek, "Numerical taxonomy with fuzzy sets," *Journal of Mathematical Biology* 1, 57–71, 1974.
[3] J. Bezdek, "Pattern recognition with fuzzy objective function algorithms," Plenum: NY, 1981.
[4] A. Kandel, *Fuzzy Techniques in Pattern Recognition,* Wiley-Interscience: New York, 1982.
[5] J. Bezdek, R. Hathaway, M. Sabin and W. Tucker, "Covergence theory for Fuzzy *C*-Means: counterexamples and repairs," *IEEE Trans. on Systems, Man & Cybernetics,* vol. SMC-17, 873–877, 1987.
[6] T. Kohonen, *Self-Organization and Associative Memory,* Springer-Verlag: Berlin, 1984.
[7] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters,"*Journal of Cybernetics 3,* 32–57, 1973.

A New Identification Jacobian for Robotic Hand/Eye Calibration

Hanqi Zhuang and Zhihua Qu

Abstract—**Hand/eye calibration is the process of identifying the unknown position and orientation of the camera frame with respect to the robot hand frame, when the camera is rigidly mounted on the robot hand. While computationally slightly more involved, one-stage iterative algorithms have two distinguished advantages over traditional two-stage linear approaches: (a) They are less sensitive to noise, and (b) they can handle cases in which the camera orientation information is not available.**

A more compact and lower dimensional Identification Jacobian is derived in this correspondence. The Jacobian, which relates measurement residuals to pose error parameters of the unknown hand/eye transformation, is a crucial component of one-stage iterative algorithms. The derivation procedure for the new Jacobian is straightforward and simple, owing to an alternative mathematical formulation of the hand/eye calibration problem. Observability conditions of the pose error parameters in the unknown hand/eye transformation are also provided based on this Identification Jacobian.

I. Introduction

A hand/eye configuration refers to the setup in which a camera is rigidly mounted on the hand of a robot manipulator. Hand/eye calibration is the process of computing the position and orientation of the camera frame with respect to the robot hand frame. The camera frame refers to the 3D coordinate frame used by the camera. The robot hand frame, also called the end-effector frame, refers to the 3D coordinate frame used by the robot controller software.

Shiu and Ahmad [1] mathematically formulated the hand/eye calibration problem as solving a system of homogeneous transformation equations of the form,

$$\mathbf{AX} = \mathbf{XB} \tag{1}$$

where $\mathbf{X}$ is the 4×4 homogeneous transformation from the robot tool coordinate frame to the sensor coordinate frame, $\mathbf{A}$ is the measurable

Manuscript received March 24, 1993; revised September 10, 1993.
H. Zhuang is with the Robotics Center and Department of Electrical Engineering, Florida Atlantic University, Boca Raton, FL 33431.
Z. Qu is with the Department of Electrical Engineering, University of Central Florida, Orlando, FL 32816 USA.
IEEE Log Number 9402288.