TR-88-004

A HEURISTIC METHOD FOR SEPARATING CLUSTERS FROM NOISY BACKGROUND

TR-88-004

A Heuristic Method for Separating Clusters from Noisy Background

Jun S. Huang and Wen R. Shieh
Computer Vision Laboratory
Institute of Information Science
Academia Sinica
Taipei, Taiwan, Rep. of China
April 1988

Abstract:When points of clusters are confronded with points having a uniform density (noisy background) , the separation of the clusters from the background is computed by heuristic statistical method, where the distribution of each point in the clusters is assumed to be bivariate normal. We try to maximize the log likelihood function of the clustering configuration. A modified hill–climbing pass algorithm is studied and the simulation results indicate that the algorithm is reliable and efficient. Also real data from astronomical photograph are tested and the result is really good.

KEY WORDS

Heuristic statistical method

Cluster analysis

Hill climbing method

I. Introduction

Cluster analysis has wide applications in biology, sociology, and many other fields (see Anderberg, M. R. [1]). It's primary concern is on how to separate a set of data into a limited number of groups, which are called "clusters" (In the following, the terms group and cluster are used interchangably.). Based on this idea, various kinds of "cluster separation" have been proposed according to an extreme large number of applications. In the simplest case, clustering of data is to divide a set of data disjointly into groups. Each datum belongs to one and only one group. Cluster analysis of this kind can be modeled as searching a partition of data according to a specific rule (criterion). Other variations of cluster analysis had also been used which included fuzzy clustering [2], overlapped clustering (or called clumping) [3], and cluster separation from noisy background.

In this paper, we propose an clustering algorithm which deal with the case of cluster separation from noisy background. Our algorithm concerns on the problem of separating one or more clusters, each is assumed with normal distribution, from a uniformly distributed noisy background. Fig. 1a shows an example of this kind of problem, which is an astronomical photograph. In this picture, there are two stellar clusters and the background is full of small stars that are smoothly distributed. Our problem is to separate the two stellar clusters from the sky background. A possible result of this extraction is shown in Fig. 1b.

The technique of separating clusters from noisy background is suitable for any system composed of individual points [4]. Here we apply this technique in stellar cluster extraction.

Separating clusters from noisy background has been studied by Yahil and Brown [4]. In their paper, Yahil and Brown considered methods of maximum likelihood, minimum χ^2 of the fit of counts in concentric annuli about the mean, and several ad hoc procedures. These methods are compared by application to computer simulated clusters where the density of the points in the clusters is assumed to form a circularly symmetric bivariate distribution. In this paper, we drop the requirement that the cluster is circularly symmetric. We attack this problem directly by considering the joint density of points in the plane. The joint density is the product of some bivariate normal densities and some uniform densities. We try to maximize this joint density directly by moving points from the set of background points into the sets of cluster points, and vice versa. A modified hill–climbing pass algorithm is used (see Friedman and Rubin [5], Fukunaga [6] and also Scott and Symons [7]), and by simulation we find this algorithm is reliable and efficient. Also some real astronomical data are tested and the results are really good.

In section II, we show two heuristic algorithms for finding the approximated centers of clusters. In section III, we discuss the model of MLE clustering method. Next we show a heuristic modified hill–climbing pass algorithm for the MLE clustering method in section IV. In section V, experimental results of both simulation data and real data are given to show the performance of the clustering algorithm.

II Center finding

Because the clustering algorithm we proposed needs an initial center for each cluster. Therefore in this section, we will show two heuristic algorithms for solving this problem. It is worth to note that because we only consider the case of 2–dimensional data. A set of data can be thought as a set of points distributed on a plane. So, from now on, the terms point and datum have the same mean.

In the follows, we introduce two heuristic algorithms for finding approximated cluster centers in a set of points. Both of these two algorithms find the approximated cluster centers by first represent the plane using a quadtree and then locate the regions that are "dense" from the quadtree. The approximated cluster centers are obtained by computing the means of the points contained in these "densed" regions. Before the details of these two algorithms, we should give some brief descriptions of quadtree.

Given a picture (or an image), its quadtree representation is obtained by recursively decompose the picture into four equal–sized quadrands. A tree of degree 4 can be constructed to represent this decomposition process. The root node corresponds to the entire picture. Each son of a node represents a quadrant of the region represented by that node. The division process of a region is terminated based on some criterion (gray level, density, coverage, etc.). In our problem, we divide a reigon by considering the number of points contained in this region. If the number of points contained in this region exceed a predefined threshold then the division process is continued, otherwise the division process in this region is terminated. A leaf node in the quadtree is called a block. Fig. 2 shows a picture and its quadtree representation. Two further items, 4–connected region and 8–connected region, are explained as follows:

4 connected region: A region R is said to be 4 connected region if for any block p, q in R, there exist a sequence of blocks $p=p_0, p_1, \ldots, p_l=q$ in R such that p_{i+1} is 4 adjacent to p_i, $0 \leq i < l$. Two blocks are called 4 adjacent if they are adjacent to each other in the horizontal or vertical directions.

8 connected region: The 8 connected region is defined analogously except the diagonal adjacencies are included in adjacency consideration.

For a survey of quadtree, the reader is urged to consult [8].

In the following, two heuristic algorithms for cluster center finding are introduced. We will use the data plane in Fig. 3 as the illustration example for both algorithms.

Algorithm 1:

The following algorithm describes a method for finding approximated cluster centers in a set of points on a plane. This algorithm first convert a plane of points into its quadtree representation. Then uses the k–means clustering method to find the approximate cluster centers.

Input: Given a plane of points $X=\{\underline{x}_1, \underline{x}_2, ..., \underline{x}_n\}$, $\underline{x}_i=(x_{i1}, x_{i2})$, $1 \le i \le n$, and m, the number of clusters.

Output: Find m approximate centers $O=\{\underline{o}_1, \underline{o}_2, \ldots, \underline{o}_m\}$, $\underline{o}_k=(o_{k1}, o_{k2})$, $1 \le k \le m$.

1. Convert the plane representation of X into its quadtree representation (see Fig. 4a).
2. Assume the quadtree has s leaf nodes. Labeling these leaf nodes as $L_1, L_2, ..., L_s$.
3. Compute the block center for each leaf node L_i, $1 \le i \le s$, obtain $\underline{l}_1, \underline{l}_2, ..., \underline{l}_s$. $\underline{l}_j$ is computed as $l_{j1}=[L_{i\cdot\text{right margin}} - L_{i\cdot\text{left margin}}]/2$, $l_{j2}=[L_{i\cdot\text{upper margin}} - L_{i\cdot\text{lower margin}}]/2$.
4. Arbitrary choose m block centers $\underline{t}_1, \underline{t}_2, ..., \underline{t}_m$ as the initial means from $\underline{l}_1, \underline{l}_2, ..., \underline{l}_s$.
5 Using k–means clustering method with k equal m to find a partition of $\underline{l}_1, \underline{l}_2, ..., \underline{l}_s$, and therefore a partition of $L_1, L_2, ..., L_s$. We label this partition as $P_1, P_2, ..., P_m$ (see Fig. 4b).
6. Obtain the approximate cluster centers $\underline{o}_1, \underline{o}_2, ...,\underline{o}_m$ by computing the means of the points contained in the blocks in each set of $P_1, P_2, ..., P_m$.

Algorithm 2:

This algorithm will yield a better accuracy and a smaller computation time.

Input: Same as algorithm 1.

Ouptut: Same as algorithm 1.

1. Same as algorithm 1's step 1, convert the plane representation of **X** into a quadtree representation.
2. Let i=1.
3. Delete all leaf nodes that are in the ith level in the quadtree.
4. Check if there result m distinquish 8–connected regions. If not, set i=i+1, go to step 3, else go to step 5 (see fig 5.a, b. c).
5. From the above, obtain the approximate cluster centers $\underline{o}_1, \underline{o}_2, ...,\underline{o}_m$ by computing the means of the points contained in each one of the m distinquish 8–connected region.

III. The clustering model

We consider the astronomical photograph as a plane which contains a set of points $X = \{\underline{x}_1, \underline{x}_2, ..., \underline{x}_n\}$, where $\underline{x}_i = (x_{i1}, x_{i2})$, $1 \leq i \leq n$. Each point corresponds to a bright pixel on the photograph. Some points in X belong to stellar clusters, some are not. Our goal is to extract the points belong to stellar clusters from X, and partition these extracted points into m groups, where m is the number of stellar clusters.

We assume the points in cluster i come from bivariate normal distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, $1 \leq k \leq m$, and each point in the sky background comes from uniform distribution in a circular disc with center $\underline{\mathbf{a}}$ and radius r. Let the set of $\underline{x}_i$'s assigned to kth cluster denoted as C_k, $1 \leq k \leq m$, and the set of $\underline{x}_i$'s in sky background as B. Futhermore, we assume that all points are independent so that the joint density function of X is

$$\mathscr{L} = \prod_{C_1} f(\underline{x}_i, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \prod_{C_2} f(\underline{x}_i, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \cdots \prod_{C_m} f(\underline{x}_i, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

$$\prod_{B} \frac{1}{\pi r^2} I_{\{|\underline{x}_i - \underline{a}| \leq r\}}(\underline{x}_i)$$

, where $\boldsymbol{\mu}_k = \begin{bmatrix} \mu_{k1} \\ \mu_{k2} \end{bmatrix}$, $\boldsymbol{\Sigma}_k = \begin{bmatrix} \sigma_{k1}^2 & \rho_k \sigma_{k1} \sigma_{k2} \\ \rho_k \sigma_{k1} \sigma_{k2} & \sigma_{k2}^2 \end{bmatrix}$, $1 \leq k \leq m$, and

$$I_A(\underline{x}) = \begin{cases} 1, & \text{if } \underline{x} \in A, \\ 0, & \text{if } \underline{x} \notin A. \end{cases}$$

$$f(\underline{x}_i, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi \sigma_{k1} \sigma_{k2} \sqrt{1-\rho_k^2}} \exp \left\{ \frac{1}{2(1-\rho_k^2)} \left[\left(\frac{x_{i1} - \mu_{k1}}{\sigma_{k1}} \right)^2 - 2\rho_k \left(\frac{x_{i1} - \mu_{k1}}{\sigma_{k1}} \right) \left(\frac{x_{i2} - \mu_{k2}}{\sigma_{k2}} \right) + \left(\frac{x_{i2} - \mu_{k2}}{\sigma_{k2}} \right)^2 \right] \right\}.$$

To simplify the computation, we take log likelihood function, and find it is

$$\ell=\log \mathscr{L} = -n_1\log(2\pi\sigma_{11}\sigma_{12}\sqrt{1-\rho_1^2}) - \frac{1}{2(1-\rho_1^2)} \sum_{C_1} [(\frac{x_{i1}-\mu_{11}}{\sigma_{11}})^2 - 2\rho_1(\frac{x_{i1}-\mu_{11}}{\sigma_{11}})(\frac{x_{i2}-\mu_{12}}{\sigma_{12}}) + (\frac{x_{i2}-\mu_{12}}{\sigma_{12}})^2] - n_2\log(2\pi\sigma_{21}\sigma_{22}\sqrt{1-\rho_2^2}) - \frac{1}{2(1-\rho_2^2)} \sum_{C_2} [\ldots] - n_3\log(\ldots) - \ldots - n_b\log\pi r^2 + \log\prod_{B} I_{\{|\underline{x}_i-\underline{a}|\le r\}}(\underline{x}_i).$$

Where n_k is the number of points in cluster C_k, $1 \le k \le m$, and n_b is the number of points in sky background. Thus the maximum likelihood estimates of $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ are

$$\hat{\mu}_{kj} = \overline{x_{kj}} = \sum_{C_k} \frac{x_{ij}}{n_k} \qquad 1 \le k \le m, 1 \le j \le 2$$

$$\hat{\sigma}_{kj}^2 = \frac{1}{n_k} \sum_{C_k} (x_{ij} - \hat{\mu}_{kj})^2 \qquad 1 \le k \le m, 1 \le j \le 2$$

and $$\hat{\rho}_k = \frac{\sum_{C_k} (x_{i1} - \hat{\mu}_{k1})(x_{i2} - \hat{\mu}_{k2})}{n_k\hat{\sigma}_{k1}\hat{\sigma}_{k2}}.$$

To find the maximum likelihood estimates of $\underline{a}$ and r, note that when r is too small, then $\log\prod_{B} I_{\{|\underline{x}_i-\underline{a}|\le r\}}(\underline{x}_i) = -\infty$ can happen with high probability, and when r is too large, $-\log(\pi r^2)$ tends to $-\infty$. Thus the maximum likelihood estimates exists and they are

$$\hat{\underline{a}} = \frac{1}{n_b} \sum_{B} \underline{x}_i,$$

and $\hat{r} = \max_{B} |\underline{x}_i - \hat{\underline{a}}|.$

By substituting these estimates back into log $\mathscr{L}$ we get log likelihood function

$$\ell = \log \mathscr{L}(\hat{\underline{\mu}}_1, \hat{\underline{\mu}}_2, \ldots, \hat{\underline{\mu}}_m, \hat{\underline{\sigma}}_1, \ldots, \hat{\underline{\sigma}}_m, \hat{\rho}_1, \ldots \hat{\rho}_m, \hat{\underline{a}}, \hat{r})$$

$$= -n_1 \log\{\frac{2\pi}{n_1} \sqrt{\sum_{C_1}\sum_{C_1} (x_{i1} - \overline{x_{11}})^2 (x_{i'2} - \overline{x_{12}})^2 - [\sum_{C_1} (x_{i1} - \overline{x_{11}})(x_{i2} - \overline{x_{12}})]^2}$$

$$-n_2 \log\{\frac{2\pi}{n_2} \sqrt{\sum_{C_2}\sum_{C_2} (x_{i1} - \overline{x_{21}})^2 (x_{i'2} - \overline{x_{22}})^2 - [\sum_{C_2} (x_{i1} - \overline{x_{21}})(x_{i2} - \overline{x_{22}})]^2}$$

$$- \ldots\ldots$$

$$-n_m \log\{\frac{2\pi}{n_m} \sqrt{\sum_{C_m}\sum_{C_m} (x_{i1} - \overline{x_{m1}})^2 (x_{i'2} - \overline{x_{m2}})^2 - [\sum_{C_m} (x_{i1} - \overline{x_{m1}})(x_{i2} - \overline{x_{m2}})]^2}$$

$$-n_b \log(\pi \hat{r}^2).$$

IV. The heuristic MLE clustering algorithm

Let $C_1, C_2, \cdots, C_m$ be the sets of points considered to be bivariate normal and $\mathbf{B}$ be the set of points considered to be background points. Initially, each of C_k, $1 \le k \le m$, should contain at least 3 points. The heuristic MLE clustering modified hill–climbing pass algorithm is described as follows.

1. Find the cluster centers $\underline{o}_k$, $1 \le k \le m$, by appling one of the cluster center finding algorithms, which are discussed in section II, on the plane. Set $\ell = -10^8$.
2. For each cluster center $\underline{o}_k$ find three points which are nearest to it and include these three points into C_k, $1 \le k \le m$. Thus we obtain m sets, each contains three points. After the completion of this algorithm, C_k will contain the set of points belong to cluster k.

3. Let the sets $E_1, E_2, \cdots, E_m$ to be empty.
4. Let k=1. Repeat step 5 to step 9 m times. Each time consider the set C_k.
5. Compute the center of C_k.
6. Find a point, called nearest point, in **B** which is besides those points in E_k, nearest to the center of C_k. Then compute new log likelihood function ℓ^* by considering C_j, $j \neq k$, and C_k with the nearest data point as m bivariate normal distributions.
7. If $\ell^* > \ell$, go to step 8, otherwise go to step 9.
8. Let $\ell = \ell^*$. Put the nearest point into C_k and return points of E_k to **B**. Let E_k to be empty. Set k=k+1. If k>m, then go to step 10., otherwise go to step 5.
9. Put the nearest point into E_k. Let k=k+1. If k>m, then go to step 10., otherwise go to step 5.

10. If the numbers of points of E_k, $1 \leq k \leq m$, are all greater than a properly choosen number, say 10, then stop. Otherwise go to step 4.

V. Experimental results

Both simulation data and real data have been used to test the performance of the MLE clustering algorithm. The results show a good performance of the algorithm

Simulation Data

Simulation data are generated in a form that m bivariate normal clusters each with n_k points, $1 \leq k \leq m$, are superimposed in a bivariate uniform noisy background with n_b points. The generation of bivariate normal and uniform points can be seen in [9].

Two kinds of simulation data have been generated, one is with 2 clusters, and the other kind of simulation data is with 4 clusters. We show two examples of these simulation data and the results after applying cluster center finding algorithm and then the MLE clustering algoritrhm.

Fig. 6a shows a 2–cluster simulation data. One cluster in this 2–cluster simulation data is with $\mu_1=\begin{bmatrix} 2 \\ 3 \end{bmatrix}$, $\sigma_{11}^2=1$, $\sigma_{12}^2=1.5$ and $\rho_1=0.8$. The other one is with $\mu_2=\begin{bmatrix} -2 \\ -3 \end{bmatrix}$, $\sigma_{21}^2=1$, $\sigma_{22}^2=1.5$ and $\rho_2=0.8$. Each cluster has 50 points. Also there are 100 uniform points on a disc with radius 5 and center at (0,0). Fig. 6b is the result after applying the MLE clustering algorithm to Fig. 6a. 73 points are classified as cluster 1 including the original 43 cluster 1 points, 29 noisy background points and 1 cluster 2 points. On the other hand, 76 points are classified as cluster 2 with 42 points from cluster 2, 34 points from noisy background and no point from cluster 1.

Fig. 7a shows another simulation data with 4 clusters. The parameters of the 4 clusters are

$$\mu_1=\begin{bmatrix} 2 \\ 3 \end{bmatrix} \qquad \sigma_{11}^2=0.5 \qquad \sigma_{12}^2=0.5 \qquad \rho_1=0.5$$

$$\mu_2=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \sigma_{21}^2=0.5 \qquad \sigma_{22}^2=0.5 \qquad \rho_2=0$$

$$\mu_3=\begin{bmatrix} -1 \\ -3 \end{bmatrix} \qquad \sigma_{31}^2=0.5 \qquad \sigma_{32}^2=0.5 \qquad \rho_3=-0.5$$

$$\mu_4=\begin{bmatrix} 4 \\ -1 \end{bmatrix} \qquad \sigma_{41}^2=0.7 \qquad \sigma_{42}^2=0.4 \qquad \rho_4=0.7$$

, each cluster has 50 points. The noisy background is same as Fig. 6a, that is 100 uniform points on a disc with radius 5 and center at (0,0) are generated. The result of applying the MLE clustering algorithm to this simulation data is showed in Fig. 7b. The relationships of points before and after applying the MLE clustering algorithm are shown in table 1.

original clusters / result clusters	noisy background	cluster 1	cluster 2	cluster 3	cluster 4	total
noisy background	51	2	0	2	1	56
cluster 1	9	47	1	0	0	57
cluster 2	12	1	47	0	0	60
cluster 3	14	0	0	48	0	62
cluster 4	14	0	2	0	49	65
total	100	50	50	50	50	300

Table 1

Other simulation work has been done with simulation data with only 1 cluster. The mean of the only 1 cluster is kept constant since its change has little effect on the separation. the range of σ_2^2 is 1, 2, 3, 4, 5, whereas σ_1^2 is kept to be 1 (By symmetry, σ_1^2 need not be changed). The range of ρ is 0, 0.2, 0.4, 0.6, 0.8. Thus there are 25 cases to be run. For each case 100 uniform data, with radius 10 and center at (0,0), and 50 bivariate normal data are generated. It is found that the rate of misclassification increases with increase σ_2^2. The change of ρ has no significant influence on the separation. However when σ_2^2 has values 1, 2, and 3, the rates of misclassification are all less than 5 percent. This indicates that the proposed algorithm are reliable when σ_2^2 is not too large.

Real Data

Beside the simulation data, we have applied the MLE clustering algorithm to several astronomical photographs. The photographs are first scanned by a Microtek M–300A scanner with 75 dpi to obtain the binary image data. And then the clustering algorithm are applied to extract the stellar clusters. Fig. 8 shows three pictures of the binary image data of photographs with stellar M46 and M47, M33, and M31 Fig. 9 shows the results of cluster center finding. Fig. 10 shows the extracted stellar after applying the MLE clustering algorithm to Fig. 8 with the cluster centers found in Fig. 9.

VI. Summary

In this paper, we consider the problem of cluster separation from noisy background. The problem is modeled as a number of clusters superimposed on a set of uniformly distributed points. All clusters are assumed with normal distributions. The clustering method is try to separate clusters from background by maximizing the joint density of the points of clusters and the points of background. A heuristic modified hill–climbing pass algorithm is used for the clustering method. Both exprimental and real data from astronomical photograph have been used to study the performance of the clustering algorithm and the results are really good.

Refernence

[1]. M. R. Anderberg, "Cluster analysis for application," Academic Press, New York, 1973.

[2]. J. C. Bezdek, "Some current trends in fuzzy clustering." in Proc. Meet. North Amer. Soc. Gen. Syst. Res. 1977, J. White, Coord. Washington, DC: SGSR press, PP. 347–353.

[3]. R. M. Needham, "Automatic classification in linquistics." The Statistician, 17, pp. 45–54, 1967.

[4]. Amos Yahil and Morton B. Brown, "On separating clusters from background," Technometrics, vol. 18, no. 1, pp. 55–58, Feb. 1976.

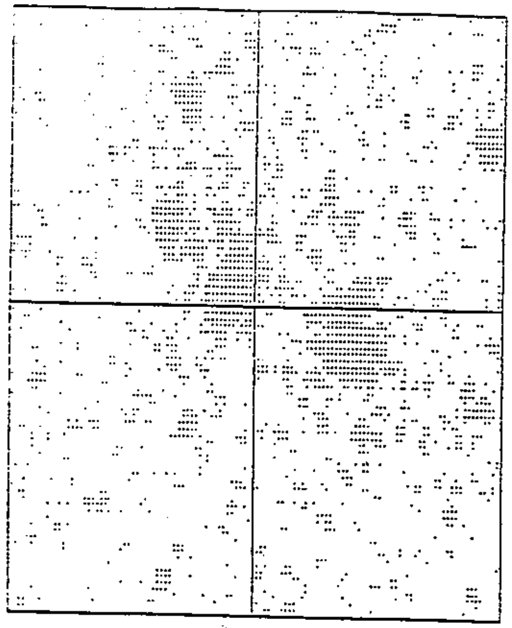
[5]. H. P. Friedman and J. Rubin, "On some invariant criteria for grouping data," J. Amer. Statist. Assoc. 62, pp. 1159–1178, 1967.

[6]. K. Fukunaga, "A criterion and an algorithm for grouping data," IEEE Trans. on Computers, C–19, pp. 917–23, 1970.

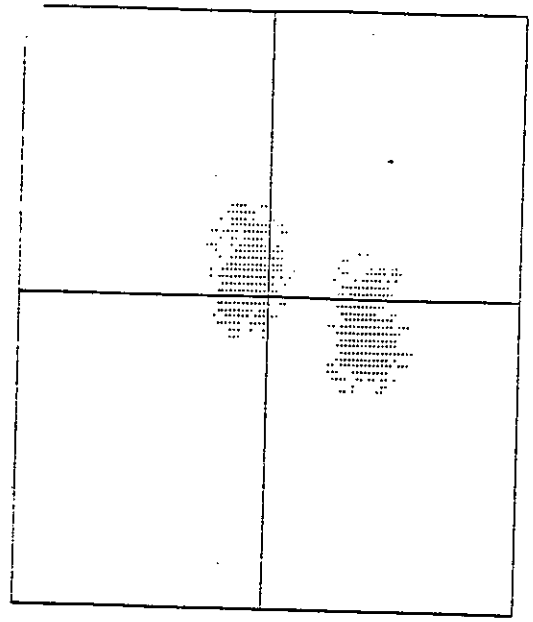
[7]. A. J. Scott and M. J. Symons, "Clustering methods based on likelihood ratio criteria," Biometrics, 27, pp. 387–97, 1971.

[8]. Hansan Samet, "The quadtree and related hierachical data structure," Computing Survey, vol. 16, no. 2, June 1984.

[9]. S. J. Yakowitz, "Computational Probability and Simulation," Addison–Wesley Co. Reading, Mass. 1977.



a



b

Fig. 1a. An example of astronomical photograph with two stellar clusters.

1b. A possible result of cluster extraction from Fig. 1a.

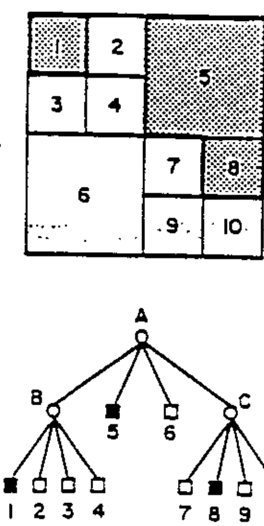
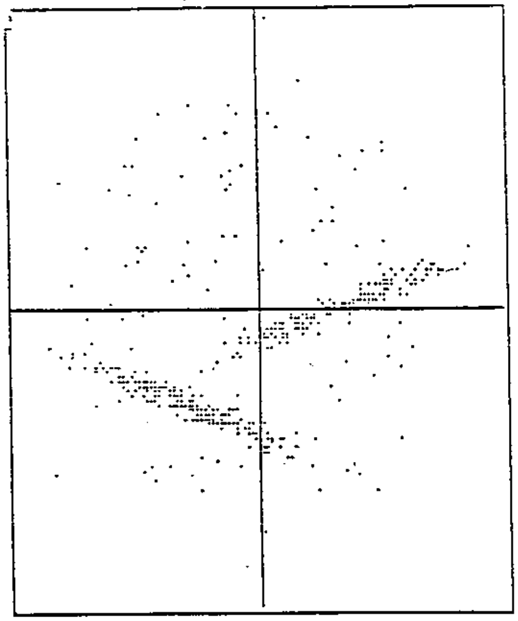


Fig 2. A picture and its quadtree representation.



```
cluster 1
mean of x = -2.0000000000E+00
mean of y =  2.0000000000E+00
variance of x =  2.0000000000E+00
variance of y =  2.5000000000E-01
correlation =  9.5000000000E-01

cluster 2
mean of x =  2.0000000000E+00
mean of y =  0.0000000000E+00
variance of x =  2.0000000000E+00
variance of y =  2.5000000000E-01
correlation = -9.5000000000E-01
```

Fig. 3. A plane of data.

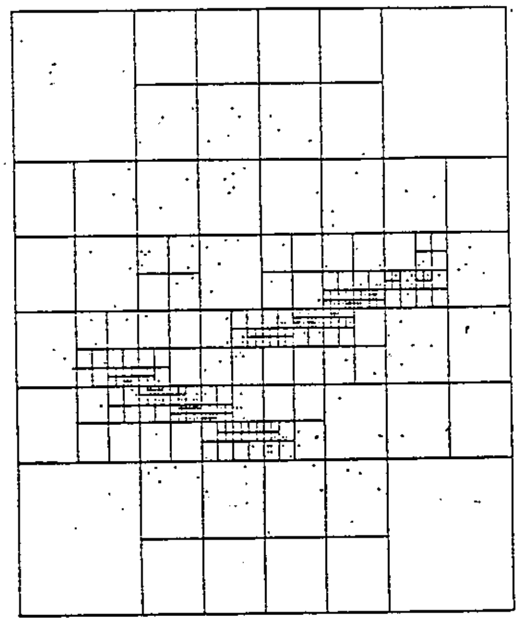
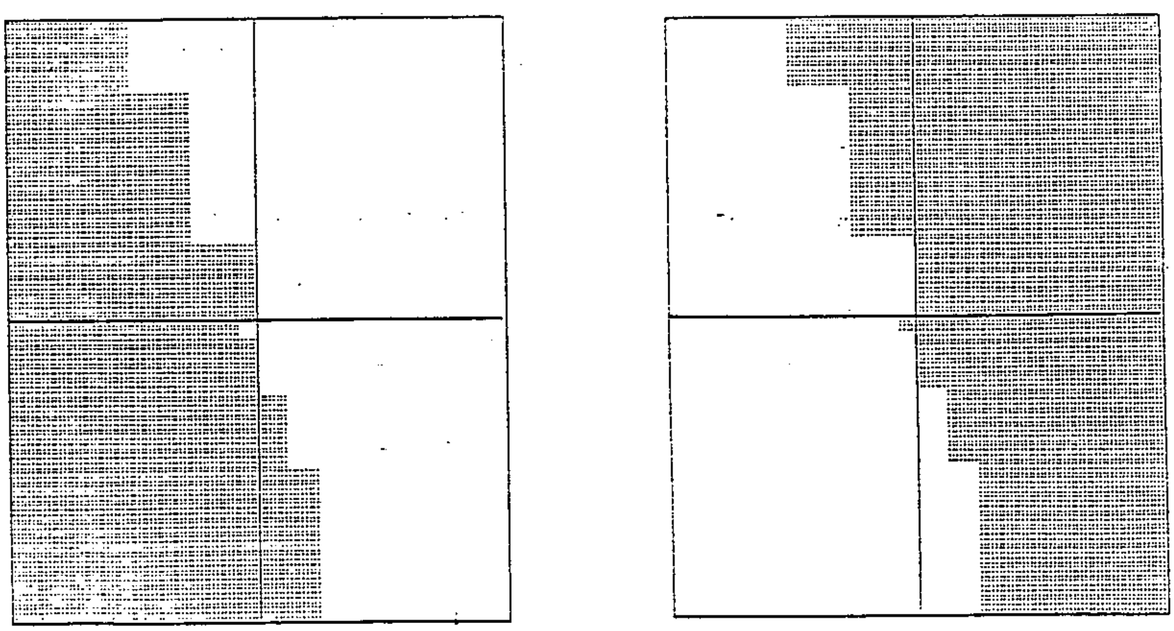


Fig. 4 a. Quadtree representation of Fig. 3.



```
center 1 X= -2.0139622642E+00 Y=  1.5577358491E+00
center 2 X=  2.0042553191E+00 Y= -2.9787234043E-01
```

Fig. 4 b Two partitions of blocks of the quadtree of Fig. 2

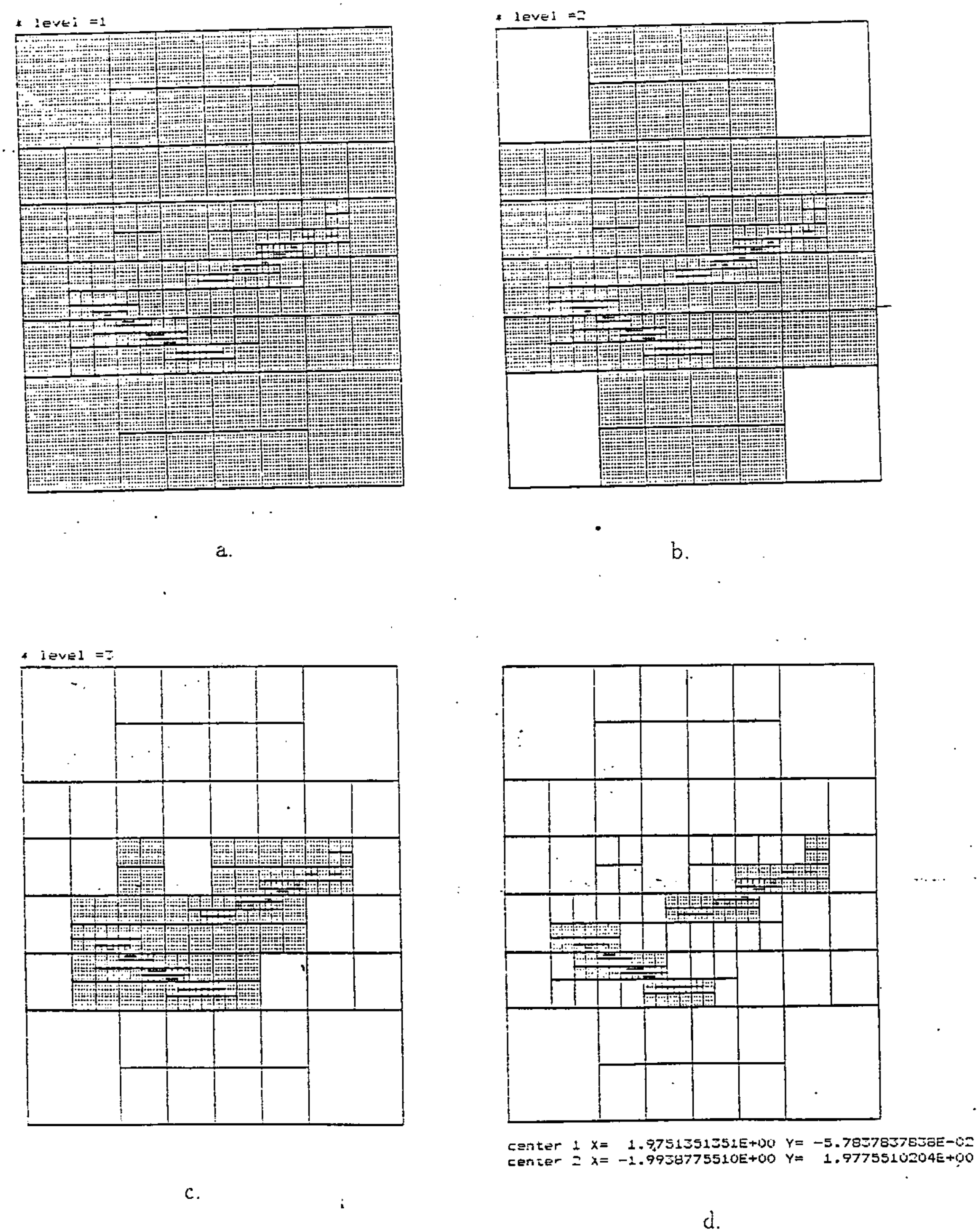
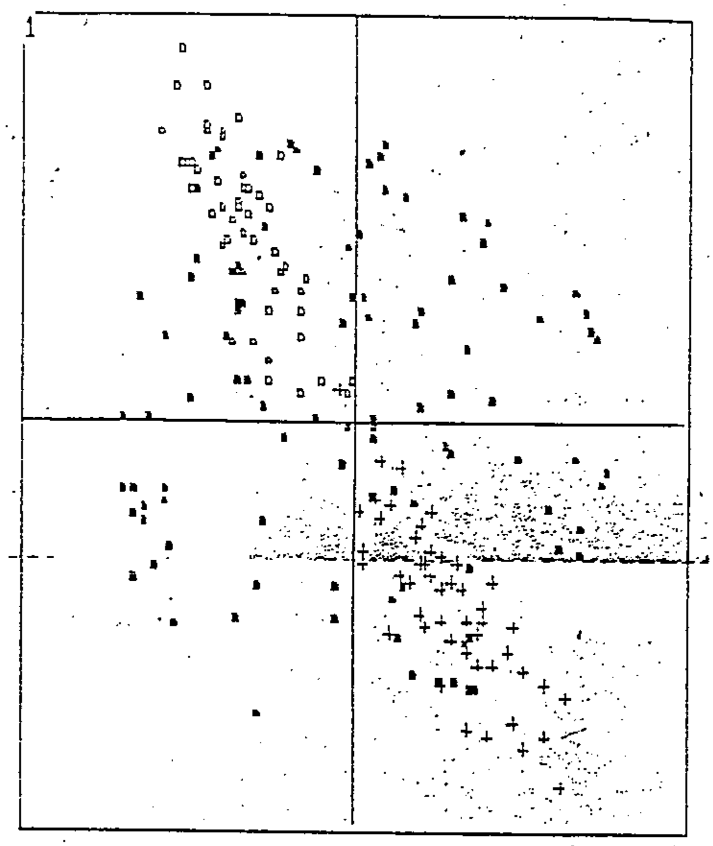
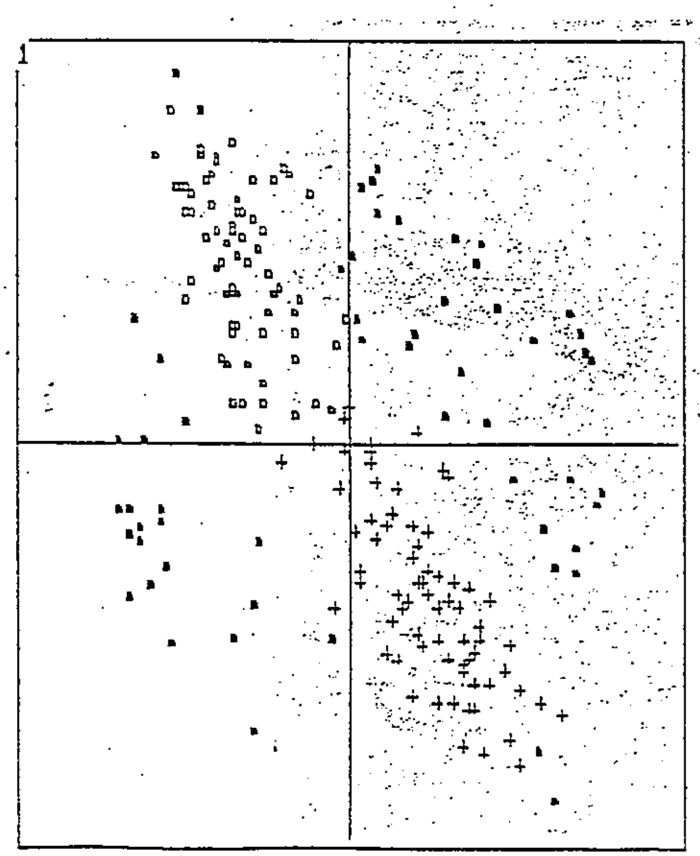


Fig. 5a, 5b, 5c and 5d. Four sequential pictures result from iteratively executed the step 3 and 4 of algorithm 2.



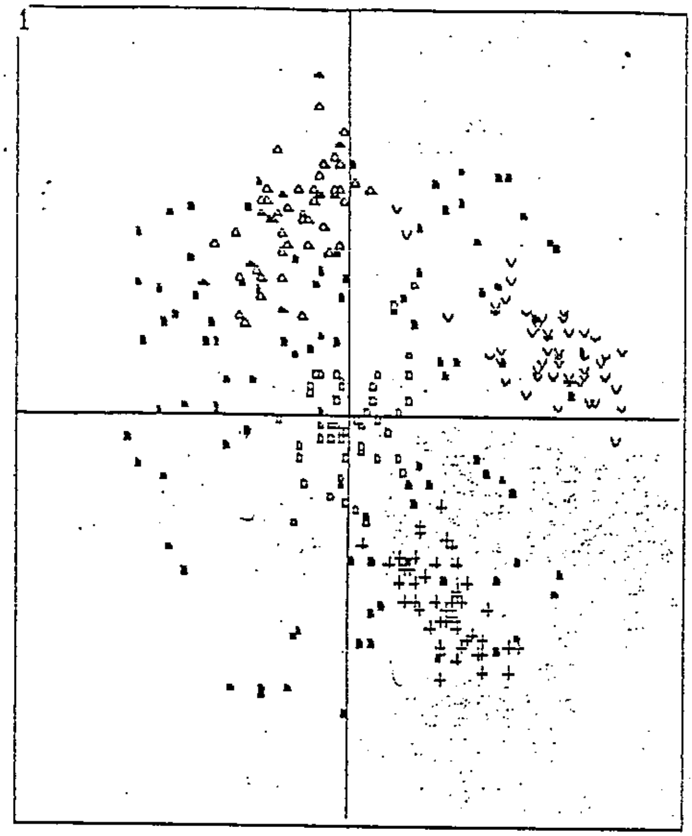
a.



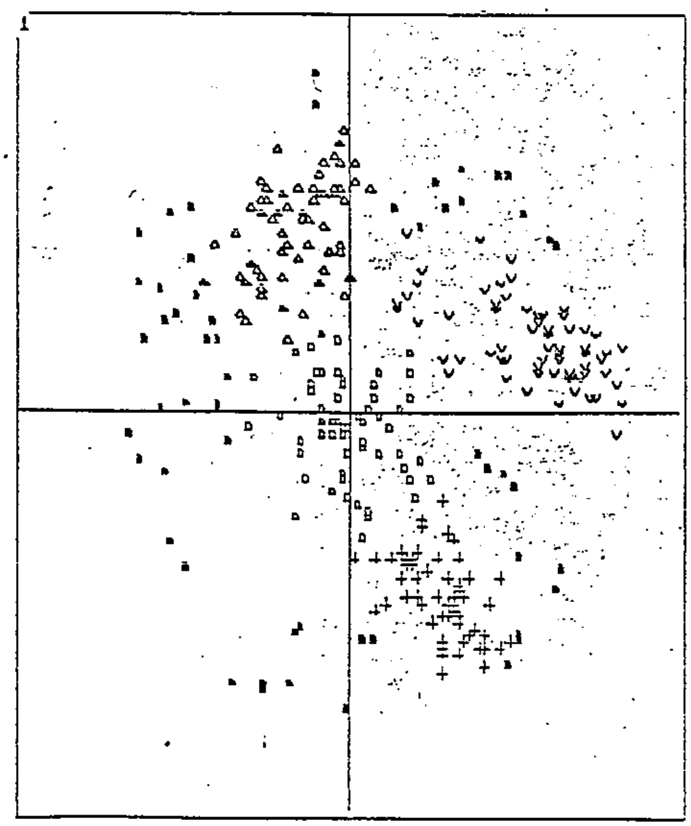
b.

Fig. 6a. 2–cluster simulation data.

6b. Result after applying the MLE clustering method to Fig. 6a.



a.



b.

Fig. 7a. Another simulation data with 4 clusters.

7b. Result after applying the MLE clustering method to Fig. 7a.

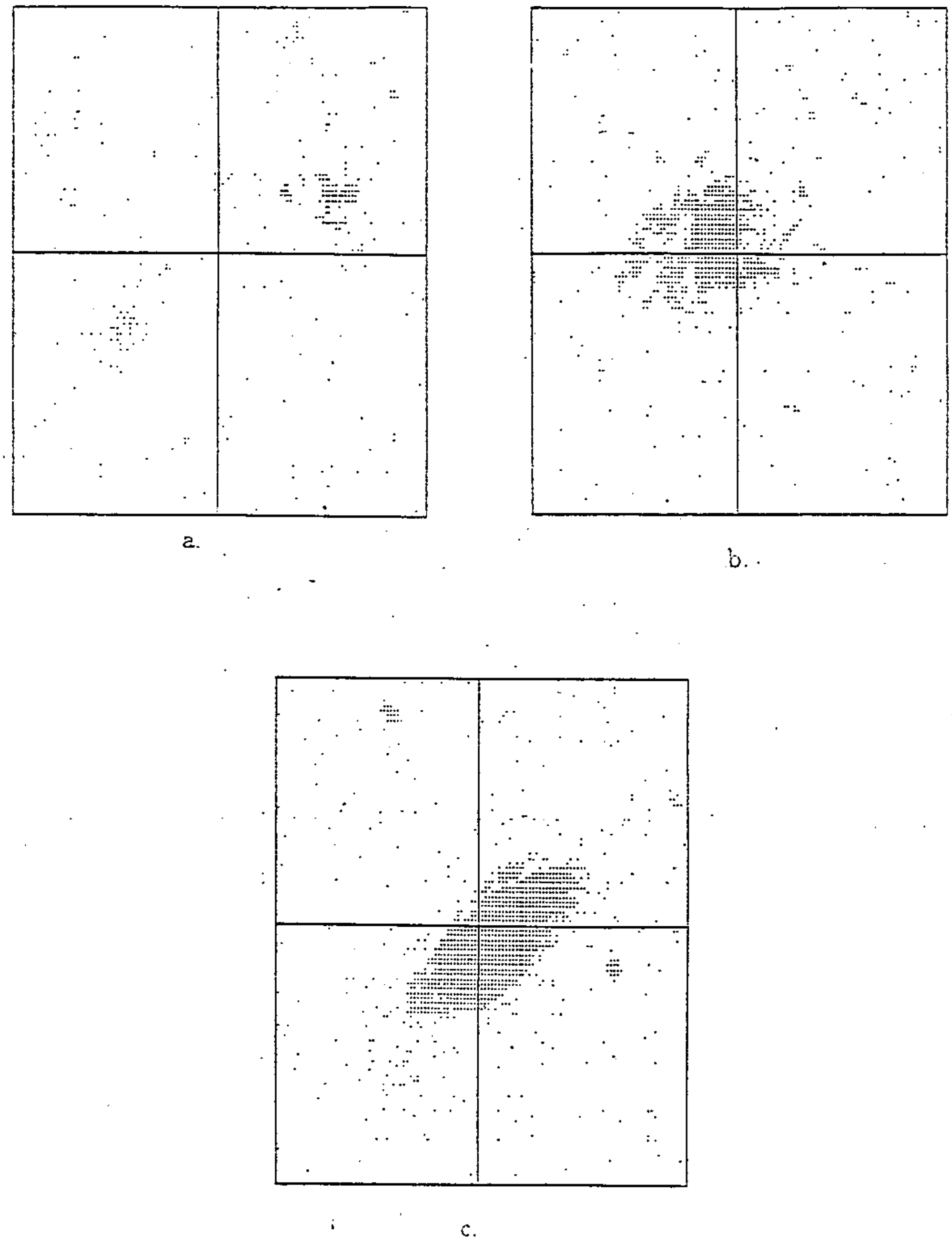
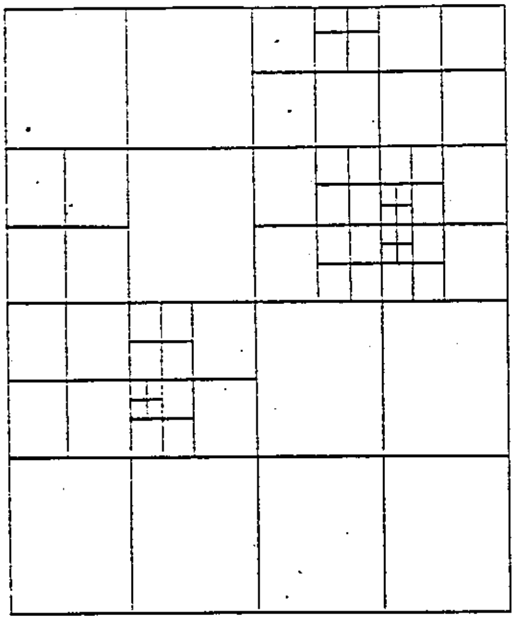
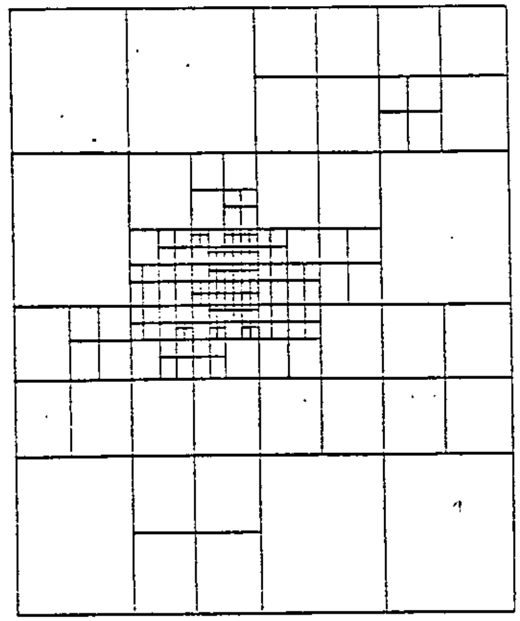


Fig. 8a, 8b and 8c. Three pictures of the binary image data of photographs with stellar clusters M46 and M47, M33, and M31.



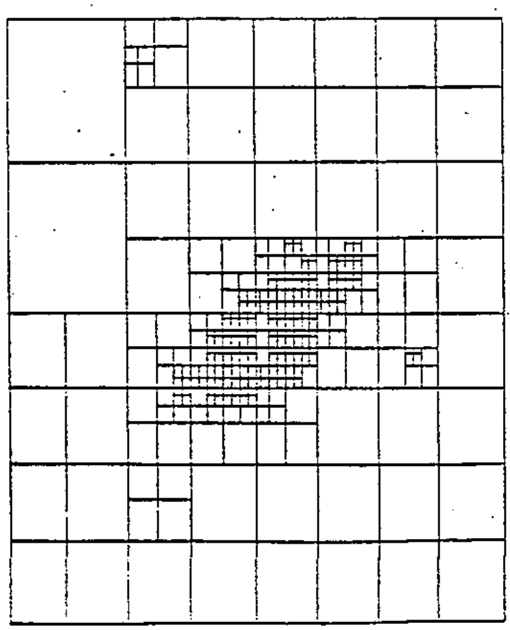
```
result
center 1 X=  3.2601941748E+01  Y=  5.0640776699E+01
center 2 X=  9.8500000000E+01  Y=  3.8263440860E+01
```

a.



```
result
center 1 X=  5.5867484663E+01  Y=  4.9065030675E+01
```

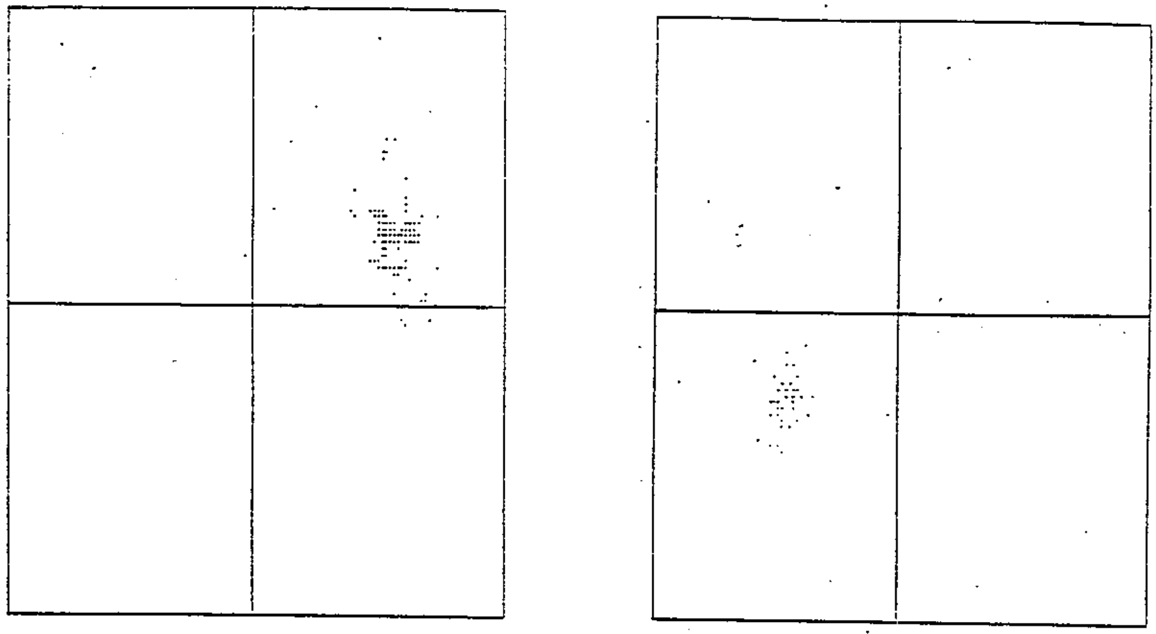
b.



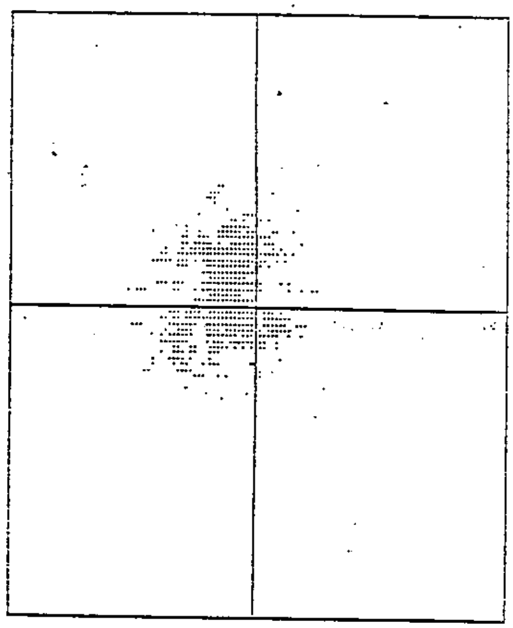
```
result
center 1 X=  6.5109137056E+01  Y=  5.4351945854E+01
```

c.

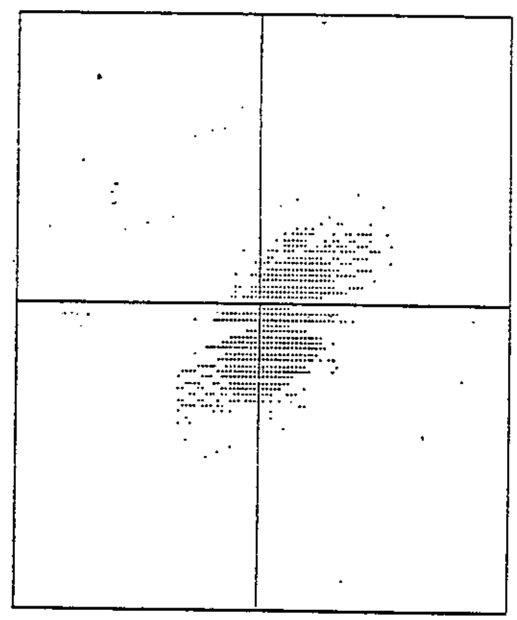
Fig. 9a, 9b and 9c. Results of approximated cluster center finding of Fig. 8a, 8b, and 8c.



a.



b.



c.

Fig. 10a, 10b, and 10c The extracted stellars after applying the MLE clustering algorithm to Fig. 8a, 8b, and 8c.