COMPUTER GRAPHICS AND IMAGE PROCESSING 9, 213–233 (1979)

Toward a Low-Level Description of Dot Clusters: Labeling Edge, Interior, and Noise Points

STEVEN W. ZUCKER

Department of Electrical Engineering, McGill University, Montreal, Quebec, Canada

AND

ROBERT A. HUMMEL

Department of Mathematics, University of Minnesota, Minneapolis, Minnesota 55455

Received October 4, 1978

Cluster analysis is an approach to finding structure in raw data patterns. Most of the available algorithms for finding clusters are not generally applicable, however, because their performance varies with the shape of the cluster. This paper presents the first stage of an approach to computing the shapes of dot clusters. It is founded on the general principle that the concept of shape is derived from interactions with the visual world. In particular, a relaxation process for labeling the functional roles that dots appear to be playing in a given arrangement is presented together with many examples of its performance. These labels should provide the primitive descriptions out of which global shape descriptions can be computed. Also, since dot patterns are a simple form of binary image, the relevance of this approach to low-level vision is described.

1. INTRODUCTION

Cluster analysis is an attempt to find structure within a collection of data objects without a precise model for the process that produced those objects. More specifically, given a set of objects and a set of measurements (observations) on each object, the problem is to group the objects into one or more distinct classes. Each class indicates a grouping of objects that are similar in some sense, with similarity normally expressed in terms of a metric relation defined over those objects. One approach to cluster analysis is to define a purely mathematical criterion for clustering, such as within-cluster variance, and then to proceed by finding the grouping that minimizes this criterion. Another approach might begin with a probabilistic model for clusters and then attempt to find the best parametric description.

There is an intuitive side to cluster analysis, however, and the assumptions necessary for a purely mathematically motivated approach can often lead to results which are at odds with these intuitions. For instance, the visual appearance of clusters within dot patterns represents these intuitions nicely, and Zahn [25]

0146-664X/79/030213-21$02.00/0

used this visually imposed structure to reveal shortcomings in many of the classical approaches. He also argued effectively for incorporating psychological notions, in particular the Gestalt law of proximity [13], into clustering algorithms. Similar arguments have been advanced by other researchers in non-parametric cluster analysis (e.g., [18, 23]).

We agree with Zahn that, in the absence of detailed, a priori information about generating processes, there should be a correspondence between the results of a clustering algorithm and the visual appearance of clusters within the given data. We have also attempted to extend the perceptual content of clustering algorithms by including a Gestalt notion of good continuation. Our ideas were motivated by our conception of how the visual system processes information. We believe that such processing requires the construction of successively more abstract descriptions of the data, and we also believe that these descriptions are symbolic. From this point of view, our description of dot clusters is in terms of the functional roles that individual dots can play either as parts of clusters or as outlying (noise) points. An algorithm for labeling the dots with assertions about the roles they appear to be playing in specific arrangements will then be presented. We hope that the labels will eventually provide the constructive [9], low-level framework from which general shape descriptors for dot clusters can be computed. More recent work [26] has begun to support this conjecture.

Our data consists of two-dimensional dot patterns. These were chosen for two primary reasons. First, two-dimensional patterns are the easiest to visualize, so their use will facilitate the evaluation of our algorithm's performance. Secondly, two-dimensional dot patterns can be considered as simplified binary images in which clusters of (black) dots correspond to (dark) regions in a picture. This suggests an important analogy between cluster analysis and scene analysis. In particular, computing descriptions of the cluster structure in dot patterns is analogous to computing a low-level description of the structure in an image. Because this analogy offers some insight into the properties of clustering algorithms, it will be discussed further in the next section.

2. AN OVERVIEW OF CLUSTERING AND SEGMENTATION

There are strong parallels between the approaches that have been developed for segmenting images and those developed for analyzing clusters. Algorithms based either on similarities or differences have emerged in both fields; however, neither has attempted to combine both in a uniform fashion. Several of these algorithms are reviewed in this section to indicate the kinds of properties on which similarity and difference measures have been based. This, in turn, provides the specific background out of which our approach evolved. For a more formal review of clustering algorithms, see [4].

Cluster analysis begins with a given set of N objects and a vector of p measurements associated with each object.[1] (This paper assumes $p = 2$). Initially, each object is represented as a point in this vector space of measurements. By specifying a notion of object similarity based on a metric in this measurement–vector

[1] Scaling procedures will not be considered in this paper.

domain, clusters can be thought of as dense regions of objects. In other words, they can be thought of as groupings of objects which lie close to one another. This qualitative notion is very difficult to translate into an algorithm, however, because of the many possibilities for specifying precisely what "dense" or "close to" means. For example, given a starting point, one could successively link the next nearest (with respect to the distance metric) unattached point to the current grouping until all points are attached. Such techniques construct descriptions of the data that are hierarchical, dendritic trees, and that are commonly found in numerical taxonomy [24]. However, once the tree has been built, there exists the problem of determining (from the tree) how many "natural" clusters are in the data. One possibility is to examine the distances encompassed by the linkages at the various levels and to search for abrupt changes. When the clusters are globular, these changes will often correspond to the larger spaces separating them and can then be used to define cluster boundaries. However, this is not true for other, more elongated distributions in which chaining may completely destroy the natural groupings [1]. The use of the linking technique thus seems to presuppose knowledge of the cluster shapes.

Other techniques provide more direct approaches to finding the low-density valleys separating clusters. Zahn [25], as has already been mentioned (see also Hubert [11]), first builds a minimum spanning tree (MST) description of the data, and then searches either for maxima in the lengths of the edges in this MST or for various "inconsistent edges" or "neck configurations." Koontz and Fukunaga [14], effectively move a boundary marker so that it lies in the valley between clusters. However, these techniques also work best when well-defined valleys exist between clusters.

Analogous to such valley-seeking or divisive techniques in clustering algorithms are edge-based image segmentation techniques. Edge detectors are local operators designed to respond strongly when they are centered over intensity differences. While many such operators exist [20], it has been impossible to find one that works for all intensity configurations [16, 28].

There is a second major class of techniques for cluster analysis that complement the valley-seeking approach perfectly. These agglomerative techniques concentrate not on the differences between points in a local neighborhood, but rather on their similarities. One approach, the k-means algorithm [15], begins with a collection of k seed points and then links each point to its nearest seed point. This defines an initial clustering. After computing the center of gravity for each cluster, points lying closer to a different cluster center than the one to which they are currently assigned are reassigned. This algorithm is iterated, recomputing the cluster centers of gravity until no further changes occur. Other examples of algorithms with this general orientation are often described as "mode seeking" (e.g., [5]).

The analog to such procedures in image segmentation is region growing. This approach is complementary to edge-based techniques, and an introductory survey can be found in [20]. The parallel between agglomerative/divisive clustering techniques and split-and-merge segmentation techniques [10] is

especially worth nothing, because it provides an example of the healthy interplay that can be achieved between these two disciplines.

The problem with approaches such as the k-means technique described above, as with the valley-seeking techniques, is that they too are sensitive to the shapes of the clusters. While the k-means approach works well with globular clusters and can be generalized to handle elongated or ellipsoidally shaped clusters [3], it still may produce distorted results when spurious noise points or irregular, interlocking clusters are present.

To produce techniques which will work on arbitrary clusters, or to determine when a technique designed to work on a given class of cluster shapes is applicable, Kendall [12] has pointed out the need for a theory of the shape of clusters. This is precisely the issue to which the approach developed in the next section is addressed. Rather than relying on a purely mathematical theory, it is motivated by the kinds of computational modeling currently used in machine perception. The proposed approach will make use both of edge (dissimilarity) and of region (similarity) data. However, rather than considering these information sources distinct as the previous approaches did, it will make use of both simultaneously. To avoid the inherent control problem of selecting one or the other kind of process for execution, they will be embedded in a cooperative computational (in particular relaxation labeling) environment. Furthermore, since global perspectives can be important in obtaining proper shape descriptions [17] but are expensive to implement, the algorithm is formulated locally but then allowed to iterate.

3. LABELING DOT CLUSTERS

A visual examination of dot patterns reveals that, in general, there are three primary functional roles which the dots can fulfill. They can be points interior to a cluster, points marking the border around a cluster, or isolated points exterior to a cluster. These isolated points, which can be thought of as either outlying noise points or clusters consisting of only single points, will here be referred to as noise points for simplicity. Since the three functional roles are sufficient for characterizing most dot patterns and since they are sufficient for representing the information sources described in the previous section, an attempt will be made to label each dot with assertions indicating which of these roles it appears to be playing in a specific data pattern. The labels will have measures of certainty attached to them indicating the relative strengths of alternative possibilities (cf. [21]).

3.1. The Relaxation Process for Labeling Dots

Relaxation labeling processes (RLPs) can be considered, for the purpose of this paper, as techniques for eliminating inconsistent (dot) interpretations on the basis of local context [27]. When the context allows multiple interpretations, the relaxation process should order the interpretations with respect to their con-

sistency with the interpretations of neighboring objects. Consistency is defined by a priori knowledge about how local interpretive labels can interact.

Abstractly, relaxation labeling processes are defined over a graph structure. The nodes of this graph represent the underlying objects and the edges represent pairs of objects considered as neighbors. For this application each dot will be represented as a node in the graph. The edge set is obtained by connecting each dot with its k nearest neighbors. Note that this edge set defines the local perspective that each point has on the other points in the pattern.[2]

For a clustering algorithm to be generally applicable, no restrictive assumptions relating to the number or shape of clusters should be made. Without any such information, it is initially possible that each point in the pattern might be acting either as an EDGE[3] point, as an INTERIOR point, or as a NOISE point. Thus all three of these possible assertions must be attached to each point. Furthermore, EDGEs have orientations associated with them, which, for this application, were quantized into eight possible directions. Each oriented EDGE label was indicated as a separate assertion.[4] In total, then, there are 10 possible assertions (labels) attached to each node: 8 indicating that the dot acts like an oriented edge segment (EDGE(OR1), ..., EDGE(OR8)), one indicating that it acts like an interior (INTERIOR), and one indicating that it acts like an isolated noise point (NOISE).

To disambiguate the label assignments, a probabilistic relaxation model was chosen. It permits evidence to be gathered incrementally. Also, in the event that more than one label is consistent for an object, it imposes an ordering on them derived from neighborhood consistency.

There are two essential aspects to the formulation of a probabilistic relaxation labeling process, one involving its symbolic structure and one involving the functional means for representing and accumulating evidence. The symbolic structure is specified by the label set and by the allowable interactions between labels. The local evidence is represented by a measure of likelihood or confidence (i.e., a probability) attached to each label. More specifically, if we let a_i, $i = 1, 2, \ldots, N$ be the set of objects (dots) and let λ_r, $r = 1, 2, \ldots, 10$ be the ten labels attached to each object, then $p_i(\lambda_r)$ is an estimate of the probability that label λ_r is correct for object a_i.

The relaxation algorithm iteratively uses the contextual information contained in the neighborhood around each node to update the probability estimate attached

[2] The choice of building the object graph with k equal to a constant number of neighbors for each point was an early design assumption made to simplify the programming. Some of the consequences of this assumption are examined further in Section 5.2. For the experiments described in this paper $k = 8$ was chosen. However, many of the experiments were recomputed with $k = 4$, with effectively no change in the final results.

[3] The specific assertions will be indicated by capital letters.

[4] It would also have been possible to include orientation as a parameter within the EDGE assertion [20]. Then only one EDGE label would be necessary, which would both reduce the computational cost of this system and make higher-dimensional extensions more feasible.

to each label. This is done in parallel on every object according to the following formula [19]:

$$p_i^{(k+1)}(\lambda) = \frac{p_i^{(k)}(\lambda)[1 + q_i^{(k)}]}{\sum_\lambda [p_i^{(k)}(\lambda)[1 + q_i^{(k)}]]},$$

where

$$q_i^{(k)}(\lambda) = \sum_j d_{ij}[\sum_{\lambda'} r_{ij}(\lambda, \lambda')p_j^{(k)}(\lambda')].$$

The $(k + 1)$st estimate for each probability is a normalized function of the product of the kth estimate for that probability and the neighborhood contribution $[1 + q_i^k(\lambda)]$.

The neighborhood contribution $(q_i^k(\lambda))$ is computed as a weighted sum over all current label probabilities at every neighboring node. The contribution of each label probability is controlled by the a priori compatibility $r_{ij}(\lambda, \lambda')$ that label λ on object a_i has with label λ' on object a_j. These compatibilities represent the primary interface between the symbolic and the evidential aspects of the process. They must take values in the range $[-1, 1]$ according to their semantic relationships and such that

(i) when λ and λ' are highly compatible, $r_{ij}(\lambda, \lambda') \to 1$;
(ii) when λ and λ' are independent, $r_{ij}(\lambda, \lambda') \to 0$; and
(iii) when λ and λ' are highly incompatible, $r_{ij}(\lambda, \lambda') \to -1$.

Situations intermediate to these cases will be assigned correspondingly intermediate real values. Thus if λ' is highly probable for a_j and is also strongly compatible with λ on a_i, then $p_i(\lambda)$ will receive a positive contribution from $p_j(\lambda')$. On the other hand, if λ' is inconsistent with λ, $p_j(\lambda')$ will detract support from λ at a_i. For a discussion of the convergence properties of this algorithm, see [29].

The computational complexity of an RLP is bounded by both the number of objects (N) and the number of labels per object (M). On a sequential machine the total amount of computation is $O(KMN)$, where K is the number of iterations required for convergence. However, the complexity on a machine capable of updating the N objects (dots) in parallel would only be $O(KM)$, while the K may be reduced by noniterative algorithms for accomplishing the same computation as relaxation [31].

3.1.1. The Compatibility Functions

The compatibility functions provide the means for embedding our intuitive model for dot clusters into the relaxation process. This model is general purpose in the sense that it requires no specific assumptions other than that clusters have some spatial extent. Analyzing this assumption more closely, we see that clusters consist of groupings of interior points surrounded by edge points. Isolated points (trivial clusters of one point) are considered as noise points. This general model is consistent with other intuitive descriptions of clusters, e.g., to quote Kendall

[12, p. 292],

> When we refer to a cluster in ordinary colloquial speech we have in mind a collection of members which are all close together; and further that the members in the middle are usually more densely distributed than those on the periphery.

Such models imply that interior points will often be surrounded by other, nearby interior points and that edge points should see interior points on their inside, occasional noise points on their outside, and other edge points oriented appropriately with respect to them. Also, interior points should never lie outside of edge points, nor should noise points fall inside edges. It is these basic local conditions which are translated into the compatibility relations that are treated individually below.

(i) INTERIOR–INTERIOR. Since INTERIOR points will often be surrounded by other nearby INTERIOR points, the compatibility function defined between the INTERIOR label on a point and the INTERIOR label on a neighboring point should be a function of the distance d separating them. For a very small d the support should be large and positive. As d increases the support should decrease monotonically until it becomes slightly negative. This negative contribution reflects our assumption that neighboring INTERIOR points within the same cluster will never be far apart. A general functional form for this compatibility can be obtained by imagining that support propagates in straight lines through the dot cluster space. If this space were filled with a uniform, viscous medium, then the amount of support for a separation of d units would be $W_{\mathrm{II}}e^{(-\alpha_{\mathrm{II}}d)}$. α_{II} is a decay constant that controls the rate at which support falls off with distance and W_{II} is the maximal support for $d \approx 0$. To force this function to become negative for large d, it is necessary to add a small negative quantity. Therefore, the complete INTERIOR–INTERIOR compatibility function is

$$r_{ij}(\mathrm{INTERIOR, INTERIOR}) = W_{\mathrm{II}}(e^{-\alpha_{\mathrm{II}}d} - C_{\mathrm{II}}).$$

An initial value of $W_{\mathrm{II}} = 1.0$ was chosen because of the restriction that r_{ij} have an upper bound of 1. $\alpha_{\mathrm{II}} = 0.5$ was selected so that all of the points located within a radius of two normalized distance units would exert the strongest influence. (Note that this includes the eight nearest neighbors on an integer Cartesian grid.) Finally, $C_{\mathrm{II}} = 0.1$ was selected for the small negative shift. Empirical tests indicate that this process is sufficiently stable so that other functions which approximate the form of this compatibility (e.g., linear or piecewise-linear approximations) would yield equally good results; see the discussion of coefficient variations in the next section and also the related discussions in [28, 31]. Furthermore, it has recently been shown that as long as these heuristically chosen compatibilities are (or can be approximated by) a linear function of the true ones, they will behave as if they were the true ones [31].

(ii) INTERIOR–EDGE. The model for clusters presupposes that INTERIOR points are on the inside of EDGE points. To include this relationship explicitly in the compatibility function, an angular difference term between the orientation of the EDGE label and the vector connecting the two points is necessary. (See Fig. 1.) In particular, let ORVEC (i, j) be a vector directed from the current

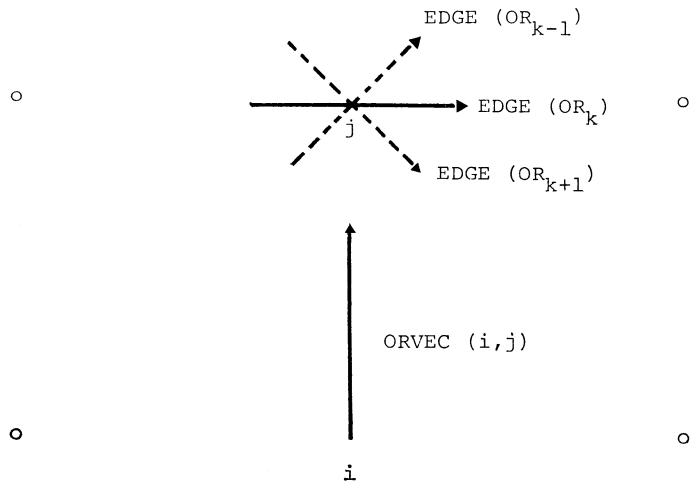

FIG. 1. Diagram indicating angular relationships for INTERIOR–EDGE compatibility function.

point i to the neighboring point j and let EDGE(OR_k) be the EDGE label at j which is clockwise perpendicular to ORVEC (i, j). Then the support of the INTERIOR label at i from EDGE labels at j will be proportional to cos (θ), where θ is the angle of rotation for the EDGE orientations relative to EDGE(OR_k). For example, each of the adjacent quantized edge elements, EDGE(OR_{k-1}) and EDGE(OR_{k-1}) contributes cos $(\pi/4)$ less support then EDGE(OR_k). As in the previous compatibility, the support that INTERIOR receives from EDGE also decreases exponentially with distance. In summary, then,

$$r_{ij}(\text{INTERIOR, EDGE}) = \cos\theta e^{-\alpha_{IE}d}.$$

Since edge points need not be as tightly packed as interiors, the power of the exponential was decreased slightly, i.e., we initially chose $\alpha_{IE} = 0.4$. Also, for implementation, a straight line approximation to the cosine was used.

(iii) INTERIOR-NOISE. Interior points are affected by noise points oppositely to the way they are affected by other interior points. That is, they should only receive support from noise points which are very far away; nearby noise points should detract support. Thus,

$$r_{ij}(\text{INTERIOR, NOISE}) = W_{IN}(e^{-\alpha_{IN}d} - C_{IN}),$$

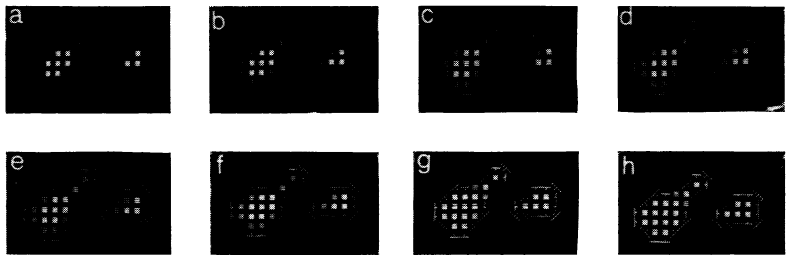

FIG. 2. A bottle-shaped cluster with a nearby blob-shaped cluster. (a) the initial probabilities; (b)–(f) 1 to 5 iterations; (g) 10 iterations; (h) 20 iterations.

with $W_{IN} = -1$, $\alpha_{IN} = 0.4$, and $C_{IN} = 0.3$. The somewhat stronger negative shift was selected because of the relative sparsity of NOISE points.

(iv) EDGE UPDATING. Since all interior/edge compatibility relationships are symmetric, the EDGE–INTERIOR compatibility is defined analogously to the INTERIOR–EDGE compatibility. The EDGE–EDGE compatibilities must depend on two angular relationships as well as on the distance separating the labeled points. The first angle describes the relative EDGE orientations, and the second describes the relative positions between the two points. Rather than evaluating the contribution from all of the angular components individually ([29]; see also [22]), an approximation to the total EDGE probability at point j was used. This was an average EDGE label computed as a weighted sum of the individual labels. Now only one cosine term involving the angle between the average EDGE orientation and the relative point directions is necessary and the compatibility becomes analogous to the INTERIOR–EDGE compatibility.[5]

(v) NOISE UPDATING. The updating of NOISE labels is analogous to the updating of INTERIOR labels, with the signs on the compatibilities reversed; that is, NOISE points must be on the outside of EDGEs and should be relatively far away from one another. This latter condition suggests that the NOISE compatibilities should change more slowly than the INTERIOR compatibilities, so that the value for the exponential decay constant was initially set to 0.4.

3.1.2. The Initial Probabilities

We have specified the underlying graph structure, the label set, and the compatibility relations for a relaxation process. Now we must specify a procedure for obtaining the initial label probabilities. Several different approaches are possible, all of which should eventually lead to the same results.

The simplest possible approach to initialization would be to sidestep the design of an initialization operator completely and to assign to each label the same initial probability. These totally uninformed estimates would be disambiguated by the relaxation process, which would first rely on the distance conditions embedded in the compatibilities. After the distance relationships begin to focus on the proper interpretations for each dot, the remaining symbolic structure of

[5] A second implementation that used all of the individual label contributions appeared to yield effectively the same results as the approximate version.

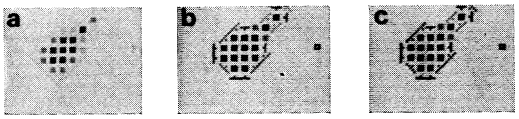

FIG. 3. A bottle-shaped cluster with a noise point. (a) initial probabilities; (b) 10 iterations; (c) 20 iterations.

the process would exert a more discriminating influence. This uniform initial probability approach was in fact taken for one of the examples in [25], which revealed that on the order of 2 to 3 iterations were required to obtain probability distributions comparable to those from the operator discussed next.

A second possibility would be to make use of the interpoint distance relationships to obtain initial probability estimates. Since most of the clustering algorithms in existence are based primarily on these distance properties, any whose results can be interpreted into label probabilities could in principle be adopted.

For the current implementation, a compromise between these two extremes was adopted: The initial probabilities were determined by operators that made use of a subset of the conditions described for the compatibility functions.[6] This choice allowed a close examination of the power of the relaxation labeling process (RLP) together with some of the computational savings obtained by using reasonable initial values. More specifically, the INTERIOR label probability is obtained by summing the distance contributions of a point's k nearest neighbors. This contribution is positive within a fixed radius (taken as 3 to 5 times the average dot spacing) and negative outside, and is determined by a linear function.[7] Thus, if most of the neighbors of a point were very close to it, a high initial INTERIOR probability is obtained. On the other hand, if most neighbors are distant, a small value is obtained.

The initial EDGE probabilities are obtained by summing the EDGE–INTERIOR compatibilities over the k-nearest neighbors. This process functions much like a center of gravity detector, in the sense that, if the center of gravity (or average position) of the k nearest neighbors is displaced in one direction, the EDGE label whose orientation is perpendicular to that direction is given the most weight. The initial NOISE probabilities are obtained from NOISE–INTERIOR compatibilities modified so that INTERIOR points outside the predefined radius of influence contribute most strongly.

Represented in this fashion, the local metric data are first scaled into positive numbers and then normalized into the range [0, 1]. Finally, since any label is possible initially, the resultant probabilities are smoothed slightly so that no label begins with a zero probability.

[6] Experimental results have indicated that the initial values obtained using this scheme are in fact very similar to those obtained from the uniform probabilities after one or two iterations. This suggests that the initialization process can be considered a modified RLP run for one iteration.

[7] The linear function was chosen as an approximation to the exponential function in the INTERIOR–INTERIOR compatibility because it allows more influence from the very distant points.

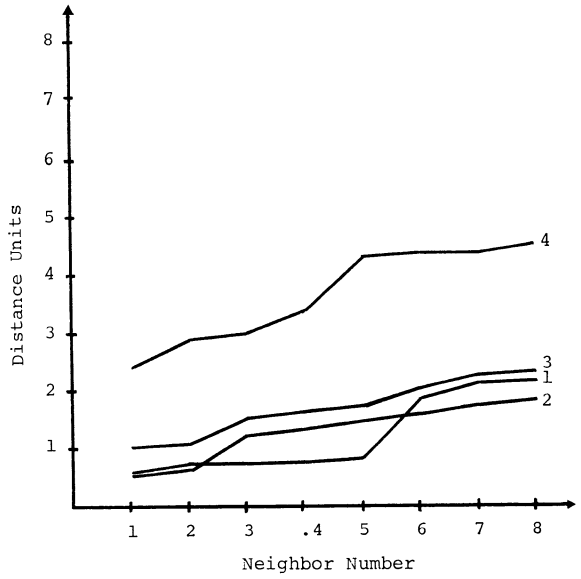

FIG. 4. Distances for Gaussian clusters.

4. EXPERIMENTS: CONSTANT-DENSITY CLUSTERS

Our first experiments for investigating the operation of the relaxation process were carried out with constant-density point distributions. Figures 2 and 3 show two of these experiments. The first contains two distinct clusters and the second contains an isolated noise point. To display the iterative results of the RLP, INTERIOR labels are shown as solid squares, EDGE labels as oriented line segments (with an extra dot in the middle to indicate the "dark" or interior side of the edge), and NOISE labels as hollow squares. For each point in the pattern, only the label with the highest probability is displayed with an intensity proportional to this probability.

In the figures, we can see that, although several labels were initially given incorrect high probabilities (Figs. 2a and 3a), by the end of the iteration process (Figs. 2h and 3c), the point distributions were correctly labeled. After 10 to 15 iterations, all the points had unique labels (i.e., $p_i(\lambda) \approx 1.0$), except for the two neck points in the bottle-shaped cluster. The ambiguity of these points derives from the coarse quantization of both the grid and the edge orientations. After 10 iterations, the probability was approximately 0.4 that the points were INTERIOR and 0.5 that they were EDGE. In the figure, these points are displayed as INTERIOR because the EDGE probability is distributed between the two orientations adjacent to the intermediate smooth curvature.

Figures 2 and 3 also provide evidence for the stability of the RLP with respect to compatibility coefficient variations. For Fig. 2, the coefficients are as described in the text; for Fig. 3, all of the coefficients have been decreased by various amounts up to a maximum of 50%, with the support given to INTERIOR points by EDGE points increased by 25%. There is virtually no change in the final

pattern. The process is, within reasonable bounds for this input pattern, invariant to changes in the coefficients. The important factor is to keep the coefficients in proper proportion to each other [28].

5. EXTENDING THE RLP FOR VARIABLE-DENSITY CLUSTERS

In the previous section, the compatibilities were specified for uniform clusters. Realistic clusters, however, normally contain point distributions of varying density. These density differences are reflected in the distribution of the distances separating neighboring points. An examination of this distribution suggests a scheme for converting d into a relative distance parameter. In Fig. 4, there is a plot of the Euclidean distances that separate a point from each of its k nearest neighbors. The points were all chosen from a single cluster (Fig. 8), and were selected from increasingly less dense areas. The important information for the labeling process is contained in the shape of these curves and not in their absolute value. In particular, a typical point in the interior of a dense cluster will have a number of immediate neighbors at approximately the same distance, and then there will be a jump in distance to the next nearest group of neighbors. This was the case for the points in the previous example and it is also the case for curve 1 in Fig. 4. The other curves, shifted upward toward larger values, derive from neighborhoods which are more and more sparse. However, they do not show a substantial difference in shape; all of these curves are for INTERIOR points.

If the distance scales for the neighborhood around each point were normalized to a standard amount (for a given k), then the compatibilities would be more a function of the relative shapes of these curves than their absolute height. One normalization technique is to scale the distance of the farthest neighbor to a standard amount, thus superimposing the distance curves. For our examples, the maximum neighborhood distance was chosen to be 2.4 relative distance units (for $k = 8$), with the remaining $(k - 1)$ distances scaled by the same factor for each point. The specific value of 2.4 was selected so that, with the scaled values resembling distributions on a unit grid, most INTERIOR–INTERIOR support initially comes from within 2.4 distance units. Without this scaling, the distance components of the compatibility functions would have an overpowering effect since the angular terms are also based on a unit grid.

5.1. Experiments: Variable-Density Clusters

The first example of a variable density dot pattern (Fig. 5) was obtained by copying the "touching clusters" figure from Zahn [25, Fig. 3e]. The RLP labeled this figure correctly, and in the process exhibited two interesting features of its behavior. The first can be seen in the lower right corner of Fig. 5. The initialization operator incorrectly placed most of the weight upon the NOISE label for the two most distant points. As the iterations proceeded, the INTERNAL labels became more certain of their interpretation and began to group into regions. These regions then established a more consistent overall interpretation which, in turn, propagated its influence outward to rectify the two incorrect labelings. Thus,

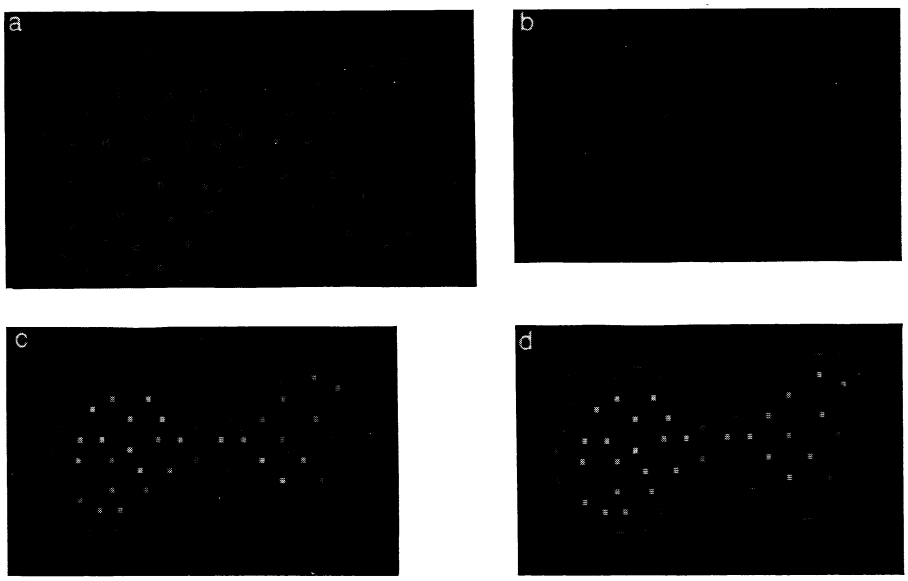

FIG. 5. Figure-eight shaped cluster; (a) Original point distribution; (b) initial probabilities; (c) 5 iterations; (d) 10 iterations

although the RLP is operating in parallel over the entire pattern, the primary changes appear to take place in a best-first fashion [26].

The second feature involves a fundamental ambiguity in the interpretation of the neck points of the pattern. This ambiguity derives from the fact that the pattern can be seen either as two clusters which are just touching or as one cluster with a thin neck. Zahn attempted to resolve this ambiguity by employing an algorithm for finding cut sets in the minimal spanning tree. This was an extremely difficult problem, and Zahn's method is not generally applicable. On the other hand, the relaxation process does not attempt to resolve the ambiguity in terms of local features. Rather, it terminates with a labeling that makes the ambiguity explicit. This labeling can then be passed to a more informed, global process for a resolution. In particular, after 10 iterations most of the points in Fig. 5 were labeled uniquely. However, for the neck points the RLP placed a weight of approximately 0.5 on the INTERIOR label and left the rest of the weight on the appropriate EDGE labels. To show that this kind of ambiguity was inherent in the structure of the pattern and is not an artifact of the current compatibilities, the contribution that EDGE points get from the INTERIOR points was increased 25%; the result was unchanged (Fig. 6).

The next example (Fig. 7) contains two intertwined C-shaped clusters which are linearly inseparable. The relaxation process correctly labeled these clusters with distinct EDGES surrounding extremely thin INTERIORS. This result remained effectively invariant over changes in the exponential coefficients (increasing all the α coefficients 25% so that they become equal to α_{II}) and over 25% increases in the EDGE–EDGE and EDGE–INTERIOR weights. The only ambiguities resulted from quantization effects. This example does begin to indicate

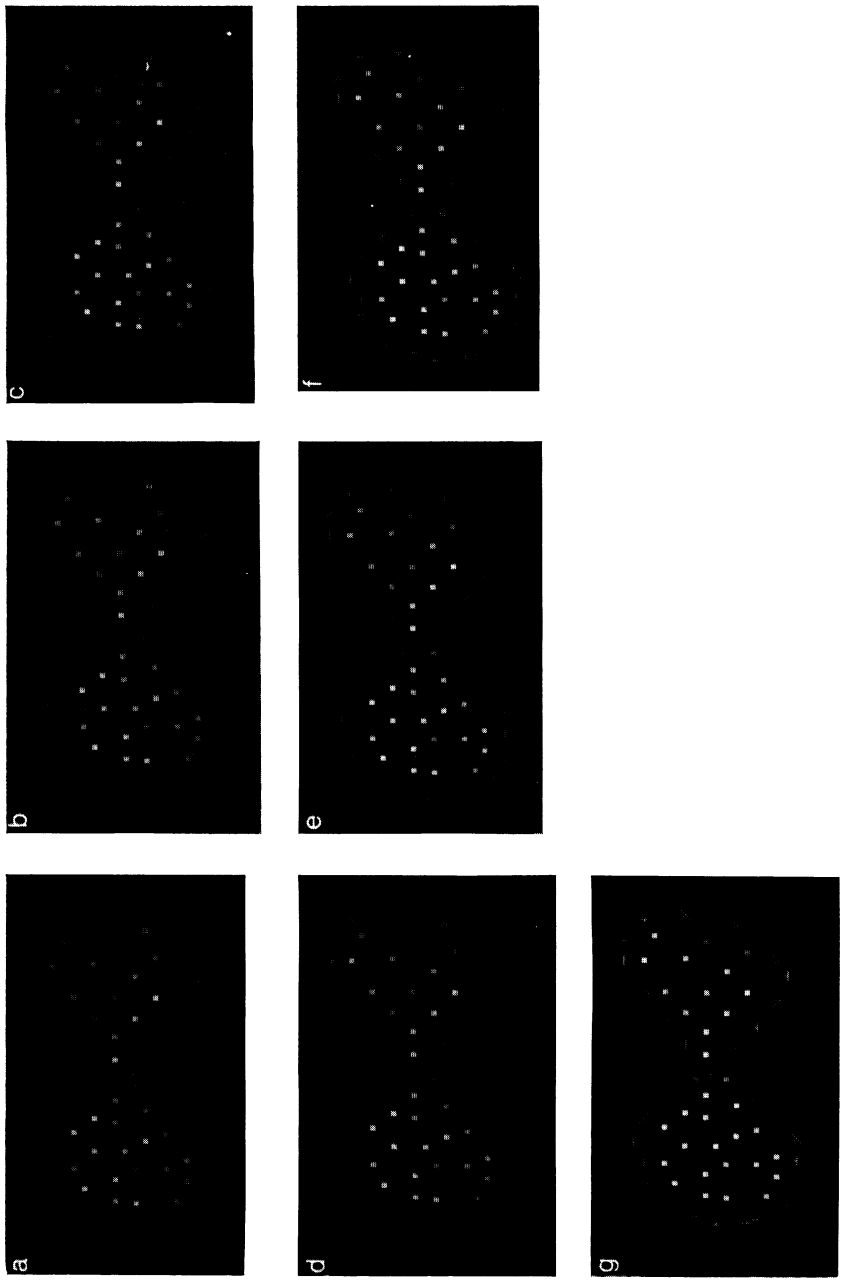

FIG. 6. Figure-eight shaped cluster—enhanced EDGE–INTERIOR support. (a) Initial probabilities; (b)–(f) 1 to 5 iterations; (g) 10 iterations.

the limitations of purely binary compatibilities, however. The EDGE around the thin, concave portions of these clusters is weak in places where neighboring, quantized EDGEs seem to encompass them. To detect these extreme situations, ternary compatibilities would seem to be necessary.

5.2. Touching Gaussian Clusters

Defining the neighborhood of a point as its k nearest neighbors can lead, for certain variable-density patterns, to an improperly skewed local perspective. This phenomenon is illustrated in Fig. 8, the "touching Gaussian clusters" example in Zahn [25]. It receives a highly ambiguous initial labeling, mainly because the initialization operator is based on Euclidean distance measurements. Since the density decreases toward the EDGE, the k-neighbor set associated with each point is concentrated on the side nearest to the cluster center. Thus, as the iterations proceed, each point acts as if it were the most outlying point in a region of decreasing density, and becomes labeled as an EDGE point. After 15 iterations there are three concentric EDGE rings around the tight cluster centers.

O'Callaghan [18] has criticized the standard k nearest neighbor definition for the "neighborhood of a point" and has proposed a visually more satisfying one. His new definition tends to decrease the asymmetries induced by the k nearest neighbor rule by eliminating points obscured from the view of a central point. Thus the neighborhood set is defined dynamically according to the relative positions of nearby points as well as the local density.

This more symmetrical neighborhood definition can be incorporated into the relaxation process when the underlying graph structure is built. However, it is possible to simulate its effect more simply by increasing the influence exerted by sparse points located behind EDGE points. This compensates for the deficiency of points toward the edge of the cluster and, after the normalizing component in the updating rule is taken into account, has the effect of making the neighborhood contribution more symmetrical. The results of this change are shown in Fig. 9. Note that now the clusters are labeled much as those in Fig. 5, and that, again, ambiguities are retained in the neck region.

6. CONCLUSIONS

Cluster analysis techniques are often employed to discover structure within raw experimental data. At such times, clustering algorithms must not presuppose specific structural classes for the dots or models for generating them. Rather, the algorithms must rely on more general principles. The basic general principle for the work presented in our paper is that the concept of shape should, at least in part, be a visual one. This position has two immediate consequences. The first is that the kinds of information processing which take place in the visual system can be applied to the development of cluster analysis algorithms. Since we believe that visual information processing involves the construction of successively more abstract descriptions of the data, we developed a relaxation algorithm to provide a low-level description of the roles which dots can play in cluster patterns. In particular, dots which acted as portions of the edge around a cluster or as points

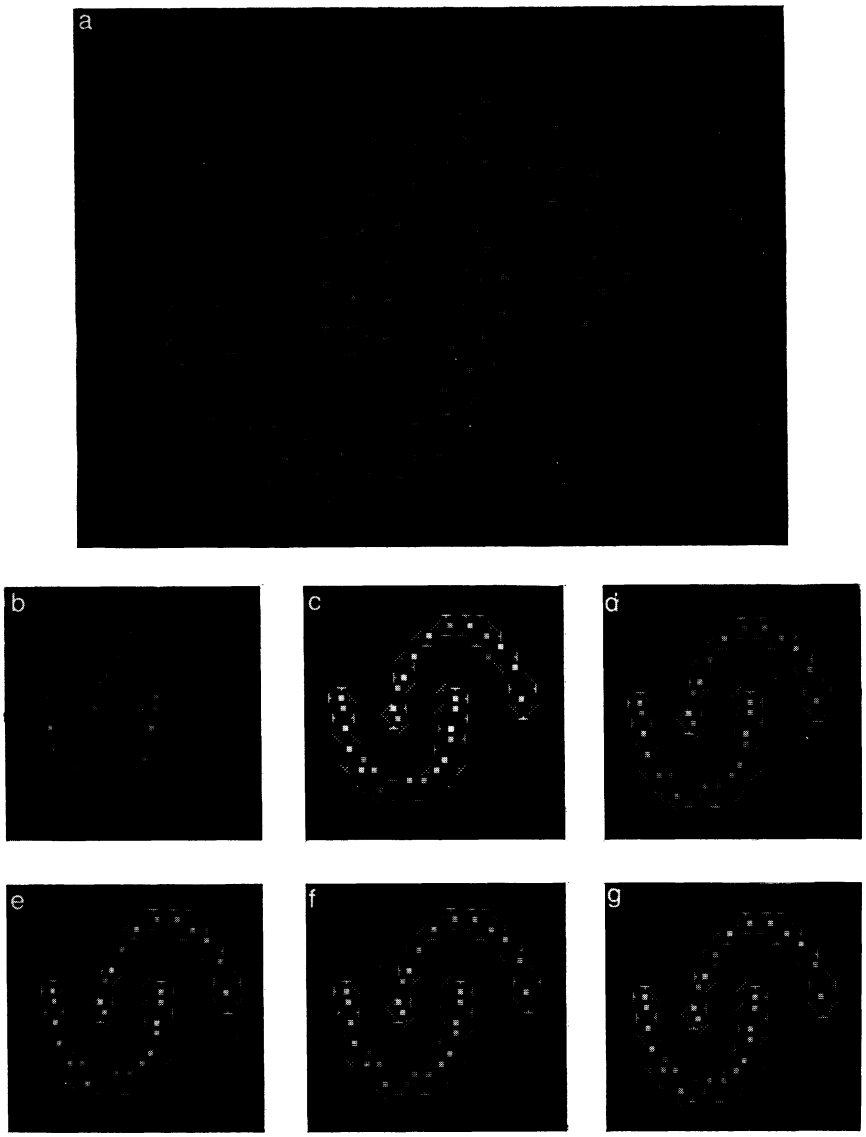

FIG. 7. Intersecting C-shaped clusters: (a) Original point distribution; (b)–(g) 0, 5, 10, 15, 20, and 25 iterations.

interior to a cluster were labeled appropriately. Also, isolated points outside of clusters were identified. The final assignments of these interpretive labels were, furthermore, in exact agreement with the visual appearance of the patterns. This brings us to the second consequence of our position—the general evaluation criterion for the algorithm. More specifically, when detailed, a priori information is lacking, the cluster structure imposed by an algorithm on a collection of data should agree with the visual appearance of clusters in that data.

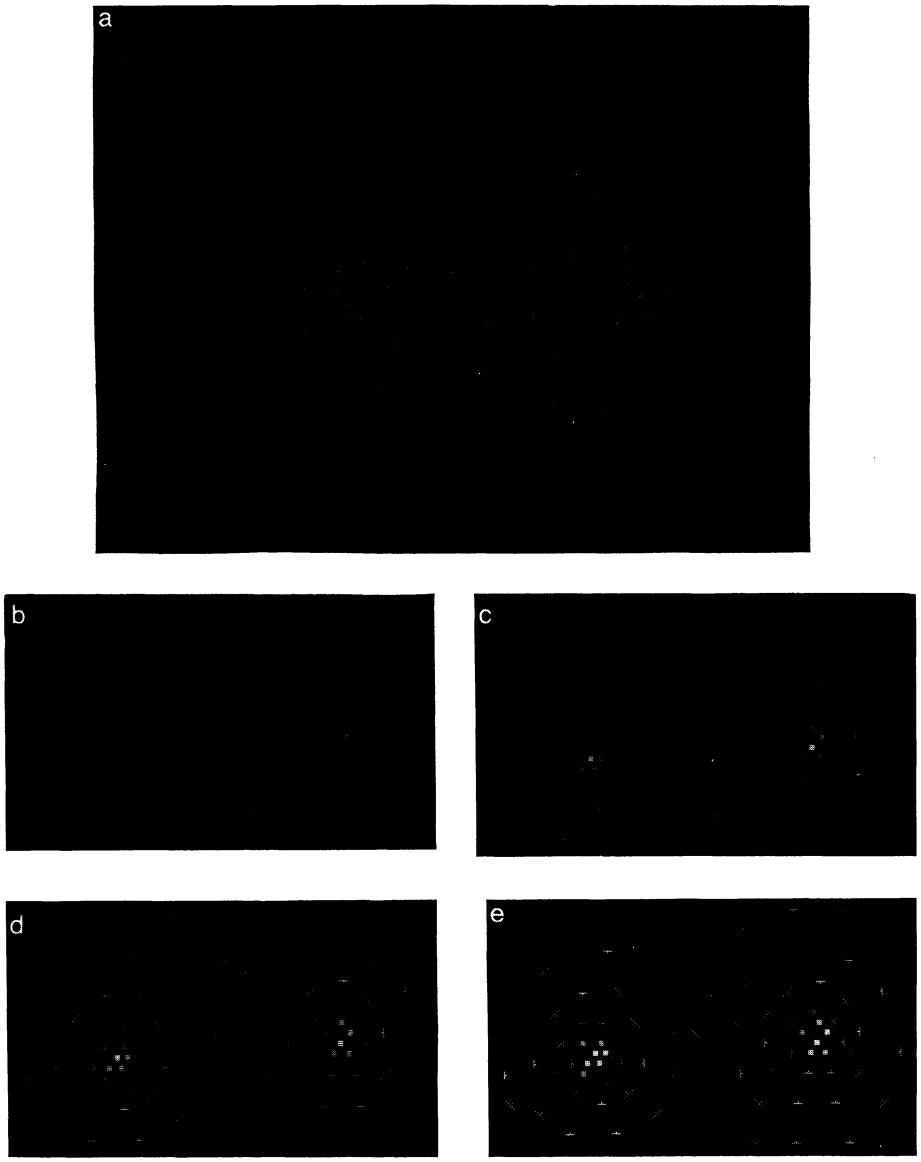

FIG. 8. Touching Gaussian clusters: (a) original point distribution; (b)–(d) 0, 5, 10, and 15 iterations.

Since dot patterns approximate a class of binary images, the study of relaxation processes for labeling dots is also relevant to the design of vision systems. Low-level vision, like dot labeling, is plagued with many sources of ambiguity and noise. Examples of noise in this paper were seen to arise from both poor initialization operator responses and from quantization errors. To counteract these noise effects, a closed-loop feedback configuration is required. RLPs provide one

structure in which this feedback can be accomplished to reduce noise while, at the same time, making certain implicit pattern ambiguities explicit (e.g., the neck points in Fig. 6). If we consider the RLP as a system of concurrent cooperating and competing subprocesses [26], the partial results from each subprocess are defined in part by the compatibilities, and, for this application, they include

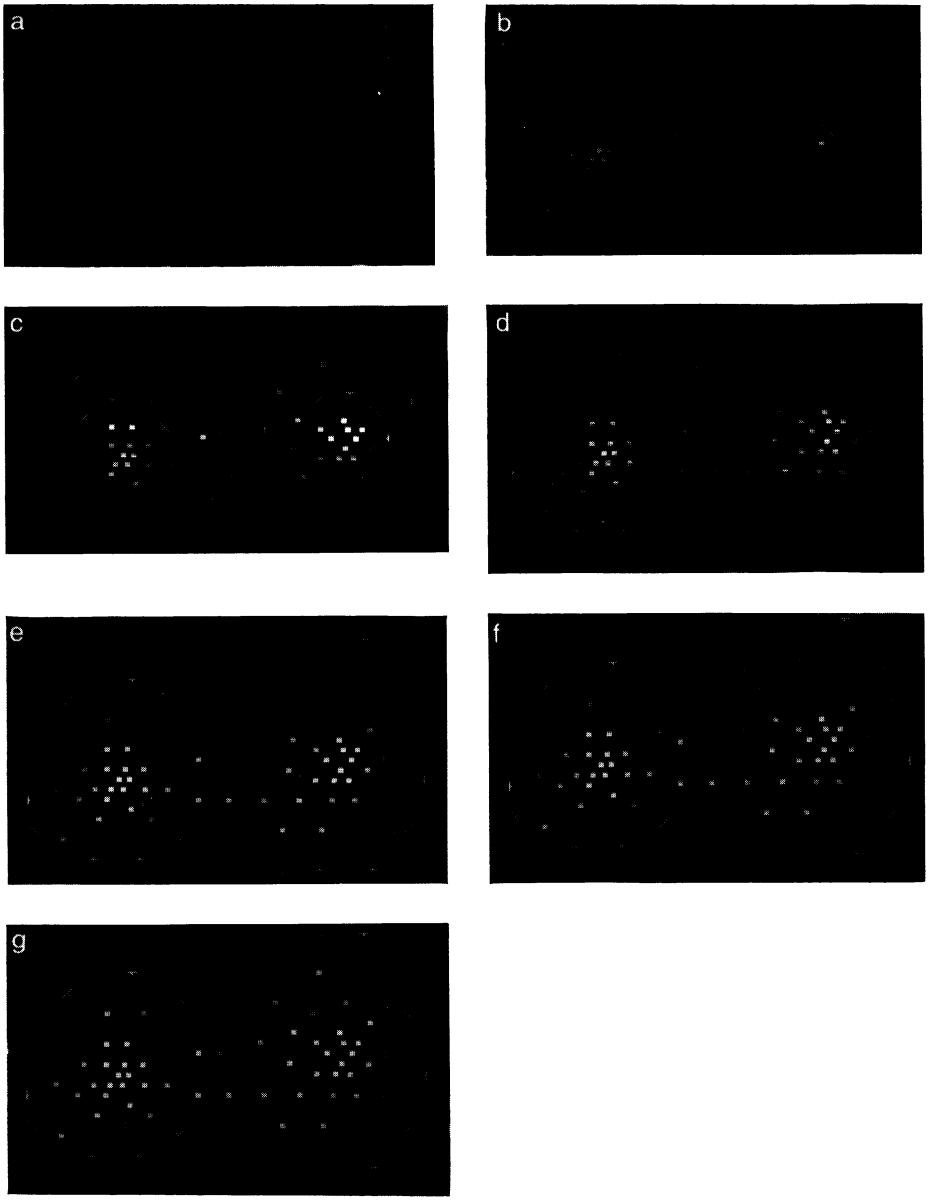

FIG. 9. Touching Gaussian clusters with neighborhood correction. (a)–(g) 0, 5, 10, 15 20. 25. and 30 iterations.

region-based interior and noise processes and an edge-based border process. It should be noted that only binary compatibilities were used in this paper, which indicates how locally determined the description for dot clusters really is. The modular nature of these compatibilities readily permits the inclusion of additional sources of structural information, such as the linear arrangements which appear in many dot patterns [17], as well as more subtle, higher-order compatibilities.

The relevance of this design for vision systems is that edge-based and region-based processes can be designed to operate together and that the control of these processes can be data-directed. Predetermined strategies such as the ones in [6, 16] are not absolutely necessary. In the portions of the dot patterns where the proper labeling was obvious, the RLP converged rapidly. In other, more ambiguous positions, convergence was slower and apparently was dependent on improved contextual support. On the basis of the output displays, such behavior could be interpreted as resulting from a "best-first" labeling strategy [2, 8]. However, this strategy was only implicitly defined; hence processing constraints arising, for example, from sequential implementations need not restrict the power of these techniques.

The cost of implicit control is, of course, very expensive computationally. However, the main points of this paper with regard to cluster analysis were the isolation of the important information sources within dot patterns and their use in computing low-level cluster descriptions. RLPs provided a convenient framework for implementation. In the future as the precise computation that is performed by relaxation becomes better understood, more efficient algorithms for accomplishing it can be devised [29]. This should bring the computational complexity more in line with other approaches to clusting.

The labeling of the functional roles which dots can play in clusters is only half of the cluster analysis problem. The actual linking of these dots into identified clusters still remains. While the design of an algorithm for tracking edge points seems to be straightforward for the examples presented in this paper, the possibility of ambiguous situations still exists. Thus a more elegant solution might have another RLP designed to label the links between points. This second RLP could use the label information provided by the process described in this paper (for a preliminary design see [26]). Such a solution is also relevant to computer vision systems, in which many kinds of grouping algorithms must cooperate with one another. The approach to clustering in this paper has already led to a new approach to labeling edge, interior, and noise points is gray-level images [30], while the dot and link process in [26] has led to a new result about scheduling algorithms in complex systems. We hope that the further study of multiple hierarchical RLPs and their interactions in cluster analysis will provide the experience necessary to confront the remaining and more difficult vision problems.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under Grant MCS-72-03610. The continued interest of A. Rosenfeld in this work is gratefully acknowledged, as is the help of B. Rose in preparing this paper.

REFERENCES

1. M. R. Anderberg, "Cluster Analysis for Applications," Academic Press, New York, 1973.
2. H. G. Barrow and J. M. Tenenbaum, "MSYS: A System for Reasoning about Scenes," Technical Note 121, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, Calif., 1976.
3. H. Chernoff, Metric considerations in cluster analysis, in *Proc. Sixth Berkeley Symp. Math. Stat. Prob.* (L. M. Le Cam, J. Neyman, and E. L. Scott, Eds.), Vol. 1, Univ. of California Press, Berkeley, 1972.
4. E. Diday and J. C. Simon, Clustering analysis, in *Digital Pattern Recognition* (K. S. Fu, Ed.), Springer-Verlag, New York, 1976.
5. I. Gitman and M. D. Levine, An algorithm for detecting unimodal fuzzy sets and its application as a clustering technique, *IEEE Trans. Comput.* **C-19**, 1970, 583–593.
6. C. A. Harlow, Image analysis and graphs, *Comput. Graphics Image Processing* **2**, 1973, 62–80.
7. J. A. Hartigan, "Clustering Algorithms," Academic Press, New York, 1973.
8. F. Hayes-Roth and V. R. Lesser, Focus of attention in a distributed logic speech understanding system, in "Proc. 1976 IEEE Conference on Acoustics, Speech, and Signal Processing," Philadelphia.
9. J. Hochberg, In the mind's eye, in *Contemporary Theory and Research in Visual Perception* (R. N. Haber, Ed.), Holt, Rinehart & Winston, New York, 1968.
10. S. L. Horowitz and T. Pavlidis, Picture segmentation by a directed split-and-merge procedure, in *Proc. Second Intl. Joint Conf. Pattern Recognition*, 1974, pp. 424–433.
11. L. J. Hubert, Some applications of graph theory to clustering, *Psychometrika*, **39**, 1974, 283–309.
12. M. G. Kendall, Cluster analysis, in *Frontiers of Pattern Recognition*, (S. Watanabe, Ed.), Academic Press, New York, 1972.
13. K. Koffka, "Principles of Gestalt Psychology," Harcourt, Brace, & World, New York, 1935.
14. W. L. G. Koontz and K. Fukunaga, A nonparametric valley-seeking technique for cluster analysis, *IEEE Trans. Comput.* **C-21**, 1972, 171–178.
15. J. MacQueen, Some methods for classification and analysis of multavariate observations, in *Proc. Fifth Berkeley Symp. Math. Stat. Prob.* (L. M. Le Cam and J. Neyman, Eds.), Vol. 1, Univ. of California Press, Berkeley, 1967.
16. D. Marr, "Early Processing of Visual Information," A.I. Memo 340, Artificial Intelligence Lab., M.I.T., Dec. 1975.
17. J. F. O'Callaghan, Human perception of homogeneous dot patterns, *Perception*, **3**, 1974, 33–45.
18. J. F. O'Callaghan, An alternative definition for "neighborhood of a point," *IEEE Trans. Comput.* **C-24**, 1975, 1121–1125.
19. A. Rosenfeld, R. Hummel, and S. W. Zucker, Scene labeling by relaxation operations, *IEEE Trans. Systems, Man Cybernet.* **SMC-6**, 1976, 420–433.
20. A. Rosenfeld and A. Kak, "Digital Picture Processing," Academic Press, New York, 1976.
21. E. H. Ruspini, A new approach to clustering, *Inform. Contr.* **15**, 1969.
22. B. J. Schachter, A. Lev, S. W. Zucker, and A. Rosenfeld, "An Application of Relaxation Methods to Edge Reinforcement," Technical Report TR-476, Computer Science Center, University of Maryland, 1976.
23. P. H. A. Sneath, A method for curve seeking from scattered points, *Comput. J.* **9**, 1966, 383–391.
24. P. H. A. Sneath and R. R. Sokal, "Numerical Taxonomy," Freeman, San Francisco, 1973.
25. C. T. Zahn, Graph-theoretical methods for detecting and describing Gestalt clusters, *IEEE Trans. Comput.* **C-20**, 1971, 68–86.
26. S. W. Zucker, Vertical and horizontal processes in low-level vision, in *Computer Vision Systems* (E. Riseman and A. Hanson, Eds.), Academic Press, New York, 1978.
27. S. W. Zucker, Relaxation labeling and the reduction of local ambiguities, in "Proc. Third Intl. Joint Conf. Pattern Recognition," San Diego, Nov. 1976; Also in *Pattern Recognition and Artificial Intelligence* (C. H. Chen, Ed.), Academic Press, New York, 1977.

28. S. W. Zucker, R. Hummel, and A. Rosenfeld, An application of relaxation labeling to line and curve enhancement, *IEEE Trans. Comput.* **C-26**, 1977, 394–403, 922–929.
29. S. W. Zucker, E. V. Krishnamurthy, and R. Haar, Relaxation processes for scene labeling: Convergence, speed, and stability, *IEEE Trans. Systems Man Cybernet.* **8**, 1978, 41–48.
30. S. W. Zucker and Y. Leclerc, Intensity clustering by relaxation, in "Workshop on Pattern Recognition and Artificial Intelligence," Princeton University, April 1978.
31. S. W. Zucker and J. H. Mohammed, "Analysis of Probabilistic Relaxation Labeling Processes," Report No. 78-3R, Department of Electrical Engineering, McGill University, Montreal.