felt that models of this type can be applied more efficiently to general service bureaus since university centers must operate under unique conditions. Be that as it may, it should be remembered that the simulation presented here is based on the assumption that decision process does not change during simulated intervals. This fact does not lessen the value of SD.

The detail program list of this simulation is available from the authors.

References

[1] H. Miyahara, Y. Teshigawara, and T. Hasegawa, "A model of users' behavior in computer network," Univ. of Hawaii, Honolulu, Aloha Syst. Tech. Rep. CN-74-10, Jan. 1974.

[2] J. W. Forrester, *Urban Dynamics*. Cambridge, MA: MIT Press, 1969.

Relaxation Processes for Scene Labeling: Convergence, Speed, and Stability

STEVEN W. ZUCKER, MEMBER, IEEE,
E. V. KRISHNAMURTHY, AND ROBERT L. HAAR

Abstract—**Relaxation labeling processes are techniques for reducing local ambiguities. They are currently being used in a number of applications, especially within computer-based vision systems. This paper studies theoretical aspects of the probabilistic rule used in these processes to iteratively evaluate local evidence. In particular, the convergence properties of the rule are studied and techniques for improving the speed of convergence are described. These results are supported by examples.**

I. Introduction

Relaxation labeling processes are a class of iterative, parallel techniques for using contextual information to reduce local ambiguities [1], [2]. Such techniques have been found to be useful in many applications, with emphasis to date on their applications in vision systems (e.g., [3]–[10]; see the overview of applications in [2]).

There are several different ways in which relaxation labeling processes can be approached. One possibility is to consider them as providing a framework for using heuristic knowledge, and then to evaluate both their performance in specific applications and their usefulness within larger "understanding" systems. In this correspondence, however, we consider a different aspect of these processes: namely, certain mathematical properties of the rule used for combining local evidence. In light of the number of current applications of relaxation labeling processes, such a theoretical study would certainly seem to be warranted.

In an earlier paper, a formal description of these processes was suggested [1]. This description included both discrete and probabilistic models for the way in which interactions could occur. While it was possible to study one of these models mathematically (the discrete model), only empirical results on the iterative behavior of the probabilistic updating algorithm have so far been obtained.[1] This correspondence is concerned with the abstract study of the convergence properties of the probabilistic algorithm. It attempts to characterize the conditions under which the algorithm will converge as well as the nature of the terminal states. Furthermore, it leads to a possible scheme for improving the speed of convergence and to considerations about the stability of these processes.

The model for relaxation labeling processes in [1] was motivated by a class of problems arising within the field of scene analysis or image understanding. Suppose, for example, that at one level of processing there are a set of objects a_i, $i = 1, 2, \cdots, n$, with a specified neighbor relation holding over this set. Attached to each of the objects is a set of labels λ_j, $j = 1, 2, \cdots, m$, where each label indicates a possible interpretative assertion about that object. For example, the objects might be vertices in a line drawing, and the labels indicators of the underlying physical edge configuration [9]. Or, at a lower level, the objects might be picture points and the labels assertions about whether the points are part of line or curve elements [3]. Then the relaxation algorithm attempts to use constraint or compatibility relationships defined over pairs of labels (possible interpretations) attached to neighboring objects in order to eliminate inconsistent or unlikely combinations of labels. This effectively reduces the number of labels attached to each object until only locally consistent label sets remain.

Labels can be attached to objects either in an all-or-none fashion, in which they are either possible or impossible (the discrete model), or they can have a measure of likelihood or confidence associated with them. This latter case is referred to as the probabilistic model because the likelihood can be interpreted as an estimate of the probability that a label λ is appropriate for an object a_i. This measure will be denoted by $p_i(\lambda)$.

In the discrete model, constraint relations indicate which neighboring pairs of labels are allowable. Operating on all the objects in parallel, the algorithm then discards labels (from the initial label set of each object) which are not consistent (i.e., do not form an allowable pair) with at least one of the labels residing on each of the neighboring objects. For a more detailed discussion of this algorithm, see [1].

In the probabilistic model, the probability estimates for an object's labels are updated on the basis of the probabilities distributed over the label sets on neighboring objects. These probabilities interact through a set of compatibility functions defined over pairs of labels on neighboring objects. More specifically, the compatibility of label λ on object a_i with label λ' on object a_j will be denoted by $r_{ij}(\lambda,\lambda')$. Compatibilities assume values in the range $[-1,1]$ according to the following criteria:

$$r_{ij}(\lambda,\lambda') = \begin{cases} 1, & \text{if } \lambda \text{ at } a_i \text{ is highly compatible with } \lambda' \text{ at } a_j; \\ 0, & \text{if } \lambda \text{ at } a_i \text{ is independent of } \lambda' \text{ at } a_j; \\ -1, & \text{if } \lambda \text{ at } a_i \text{ is incompatible with } \lambda' \text{ at } a_j. \end{cases}$$

Of course, intermediate values of the compatibility functions are possible. It is these compatibilities which provide the means for incorporating heuristic knowledge into the process (see the discussion in [2]).

Manuscript received October 13, 1976; revised July 21, 1977. This work was supported by the NSF under Grant MCS-72-03610.

S. W. Zucker was with the Computer Science Center, University of Maryland, College Park, MD 20742. He is now with the Department of Electrical Engineering, McGill University, Montreal, PQ, Canada.

E. V. Krishnamurthy was with the Computer Science Center, University of Maryland, College Park, MD 20742. He is permanently with the Indian Institute of Science, Bangalore 560012, India.

R. L. Haar is with the Computer Science Center, University of Maryland, College Park, MD 20742.

[1] For a related study of a restricted relaxation process embedded within a heuristic search, see [10, appendix A-1].

0018-9472/78/0100-0041$00.75 ©1978 IEEE

The initial probabilities are obtained either by performing a measurement of some sort or, in the absence of any *a posteriori* information, are assigned uniform values. Each probability is then updated by a rule of the form

$$p_i^{(k+1)}(\lambda) = F(p_i^{(k)}(\lambda);\, q_i^{(k)}(\lambda))$$

where k is the iteration number and $q_i^{(k)}(\lambda)$ denotes the neighborhood contribution. In this correspondence we shall consider the class of updating rules which can be modeled by the following updating formula (for further motivation behind this rule, see [1]):

$$p_i^{(k+1)}(\lambda) = \frac{p_i^{(k)}(\lambda)[1 + q_i^{(k)}(\lambda)]}{\sum_{\lambda} p_i^{(k)}(\lambda)[1 + q_i^{(k)}(\lambda)]} \qquad (1)^2$$

where

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

The coefficients (C_{ij}) represent a possible weighting over the neighboring objects a_j and insure that q_i is in the range $[-1,1]$. This rule is used to update the probability of each label on each object in parallel, and is then iterated until no further changes occur.

In order to study the conditions under which this rule iterates to a stable value, it is useful to first translate it into a matrix representation. Then the theory of iterative matrix processes can be used both to examine questions of boundedness and convergence, and also to suggest techniques for accelerating the rate of convergence.

II. Matrix Formulation of the Probabilistic Relaxation Process

A. Definitions and Notation

The local updating rule (1) can be written as a matrix process in the following way. Recall that there are m possible labels for each object, and let $P_i^{(k)}$ be an $(m \times m)$ diagonal matrix with diagonal elements $p_i^{(k)}(\lambda_j)$ for $j = 1, 2, \cdots, m$. Similarly, let $Q_i^{(k)}$ be an $(m \times m)$ diagonal matrix with elements $q_i^{(k)}(\lambda_j)$ for $j = 1, 2, \cdots, m$. Thus

$$P_i^{(k)} = \begin{pmatrix} p_i^{(k)}(\lambda_1) & 0 & 0 & \cdots & 0 \\ 0 & p_i^{(k)}(\lambda_2) & 0 & \cdots & 0 \\ & & & \ddots & \\ 0 & 0 & & \cdots & p_i^{(k)}(\lambda_m) \end{pmatrix}$$

$$Q_i^{(k)} = \begin{pmatrix} q_i^{(k)}(\lambda_1) & 0 & \cdots & 0 \\ 0 & q_i^{(k)}(\lambda_2) & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & q_i^{(k)}(\lambda_m) \end{pmatrix}.$$

Each of the P matrices is defined for each object a_i, $i = 1, 2, \cdots, n$, and the resulting n matrices are combined into a global $(mn \times mn)$ diagonal matrix with n blocks of $(m \times m)$ diagonal matrices $P_i^{(k)}$. Thus

$$P^{(k)} = \begin{pmatrix} P_1^{(k)} & 0 & \cdots & 0 \\ 0 & P_2^{(k)} & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & P_n^{(k)} \end{pmatrix}.$$

The global neighborhood contribution matrix $Q^{(k)}$ is, similarly, an $(mn \times mn)$ diagonal matrix with n blocks of $(m \times m)$ diagonal matrices $Q_i^{(k)}$:

$$Q^{(k)} = \begin{pmatrix} Q_1^{(k)} & 0 & \cdots & 0 \\ 0 & Q_2^{(k)} & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & Q_n^{(k)} \end{pmatrix}.$$

As a standard reference on matrix analysis, see [11].

B. Matrix Iterative Model

Using these definitions, the iterative sequence (1) can be written in matrix form for each object a_i:

$$P_i^{(k+1)} = P_i^{(k)}[I + Q_i^{(k)}](T_i^{(k)})^{-1} \qquad (2a)$$

where $T_i^{(k)}$ is an $(m \times m)$ diagonal matrix with elements $t_{jj}^{(k)} = \text{trace}\,\{P_i^{(k)}[I + Q_i^{(k)}]\}$ and I is the identity matrix of order $(m \times m)$. Here $P_i^{(0)}$ represents our initial estimate of the label probabilities for object i.

The matrix iterative sequence can also be written in the global form (where each P^k consists of blocks of P_i^k)

$$P^{(k+1)} = P^{(k)}[I + Q^{(k)}](T^{(k)})^{-1} \qquad (2b)$$

where $T^{(k)}$ is an $(mn \times mn)$ diagonal matrix with elements

$$t_{jj}^{(k)} = \text{trace}\,\{P_i^{(k)}[I + Q_i^{(k)}]\}$$

for $j = (i - 1)m + 1$ to im, and where I is the identity matrix of order $(mn \times mn)$.

We can now give a physical interpretation of the recursions (2). In our matrix notation, it is clear that the iteration process has no effect on a $P^{(k)}$ matrix in which each $P_i^{(k)}$ submatrix has only one nonzero diagonal element of value unity. Such a $P^{(k)}$ matrix will be called a *unique interpretation matrix* (UIM). It corresponds to each object having a unique label with probability l, and all other labels of that object having probability 0. From a Markov process point of view, a UIM corresponds to an absorbing state.

C. Convergence Criteria and Fixed Points

A matrix iterative sequence such as (2a) is said to be convergent (or to define a contractive mapping) if it has limiting points.

We will now discuss the convergence criteria for a specific i $(i = 1, 2, \cdots, n)$. Since each $P^{(k)}$ is composed of the $P_i^{(k)}$ blocks, and the $Q^{(k)}$ matrix is also appropriately blocked, the convergence of $P^{(k)}$ follows from the argument for convergence of each of the $P_i^{(k)}$ matrices.

In order to discuss the convergence criteria we need to associate with any vector or matrix A (say) a single nonnegative scalar that in some sense provides a measure of its magnitude. We will use the Euclidean norm defined by

$$\|A\| = (\sum \sum a_{ij}^2)^{1/2}$$

which is the sum of squares of the matrix elements a_{ij}. Clearly, $\|A\| = 0$ if and only if A is a zero matrix.

It is well known [11] that all norms are equivalent, in the sense that if the sequence

$$\|x^{(0)} - x\|_1,\ \|x^{(1)} - x\|_1,\ \cdots,\ \|x^{(n)} - x\|_1,\ \cdots$$

[2] While this updating scheme is rather simple in that it uses only a weighted sum of the neighboring label probabilities to update a given probability, such simple schemes appear to be rich enough to model a variety of real applications, ranging from line drawing interpretation to curve enhancement. Furthermore, most of the algebraic results derived in this paper only require the neighborhood contributions $q_i^{(k)}(\lambda)$, not the explicit scheme for calculating them.

vanishes in the limit, with respect to any one norm, then the same is true with respect to any other norm. Likewise, a sequence that is bounded in one norm is bounded in every norm.

We see that sequence (2a) starts with an initial approximation $P^{(0)}$ (where $\|P^{(0)}\| \neq 0$) and computes successive iterates

$$P_i^{(1)} = P_i^{(0)}[I + Q_i^{(0)}](T_i^{(0)})^{-1} \tag{3}$$

$$\begin{aligned} P_i^{(2)} &= P_i^{(1)}[I + Q_i^{(1)}](T_i^{(1)})^{-1} \\ &= P_i^{(0)}[I + Q_i^{(0)}](T_i^{(0)})^{-1}[I + Q_i^{(1)}](T_i^{(1)})^{-1} \end{aligned} \tag{4}$$

and so on with

$$P_i^{(k)} = P_i^{(0)} \prod_{r=0}^{k-1} [I + Q_i^{(r)}](T_i^{(r)})^{-1} \tag{5}$$

and

$$\begin{aligned} P_i^{(k+1)} &= P_i^{(0)} \prod_{r=0}^{k} [I + Q_i^{(r)}](T_i^{(r)})^{-1} \\ &= P_i^{(k)}[I + Q_i^{(k)}](T_i^{(k)})^{-1}. \end{aligned} \tag{6}$$

Therefore, using our concept of norm we can say that the sequence (6) converges if $\|P_i^{(k+1)} - P_i^{(k)}\|$ vanishes in the limit as $k \to \infty$, or

$$\lim_{k\to\infty} \|P_i^{(k+1)} - P_i^{(k)}\| = 0.$$

In terms of practical computation, this is tantamount to

$$\|P_i^{(k+1)} - P_i^{(k)}\| < \varepsilon$$

for sufficiently large values of k, where ε is a small positive number depending upon the precision used in our calculations.

Using (6), we see that for convergence we require

$$\begin{aligned} &\lim_{k\to\infty} \|P_i^{(k+1)} - P_i^{(k)}\| \\ &\quad = \lim_{k\to\infty} \|P_i^{(k)}[(I + Q_i^{(k)})(T_i^{(k)})^{-1} - I]\| = 0. \end{aligned} \tag{7}$$

Now, clearly, $0 < \|P_i^{(k)}\| \leq 1$ for each $k = 0, 1, 2, \cdots, n$. (Note that trace $P_i^{(k)} = 1$ for all k by the definition of the sequence (2a).) Hence the sequence (2a) is bounded or nonexpansive. We will now show that 1) in the following cases the sequence converges and 2) if the sequence converges, then one of the following cases result.

Case 1: $p_i^{(k)}(\lambda_j) \neq 0$ for all i and j and $Q_i^{(k)} = \beta I$, where β is a scalar (including zero).

Case 2: $p_i^{(k)}(\lambda_\alpha) = 0$ for some (but not all) α, $1 \leq \alpha \leq j$. The corresponding $q_i^{(k)}(\lambda_\alpha)$ are arbitrary, however the remaining $q_i^{(k)}(\lambda_j) = \beta$ (a scalar, including zero) for $j \neq \alpha$.

Case 3: $p_i^{(k)}(\lambda_\alpha) = 1$ for exactly one value of $j = \alpha$, $1 \leq \alpha \leq j$, and $p_i^{(k)}(\lambda_j) = 0$ for $j \neq \alpha$. This corresponds to the case in which $p_i^{(k)}$ is a UIM.

By substituting the values of $p_i^{(k)}(\lambda_j)$ and $q_i^{(k)}(\lambda_j)$ listed above, we can easily verify that these cases correspond to stationary points for the process (2a).

To prove that when the process converges, these or their combinations are the only stationary points, we consider the individual elements of

$$[P_i^{(k+1)} - P_i^{(k)}] = P_i^{(k)}[(I + Q_i^k)(T_i^k)^{-1} - I].$$

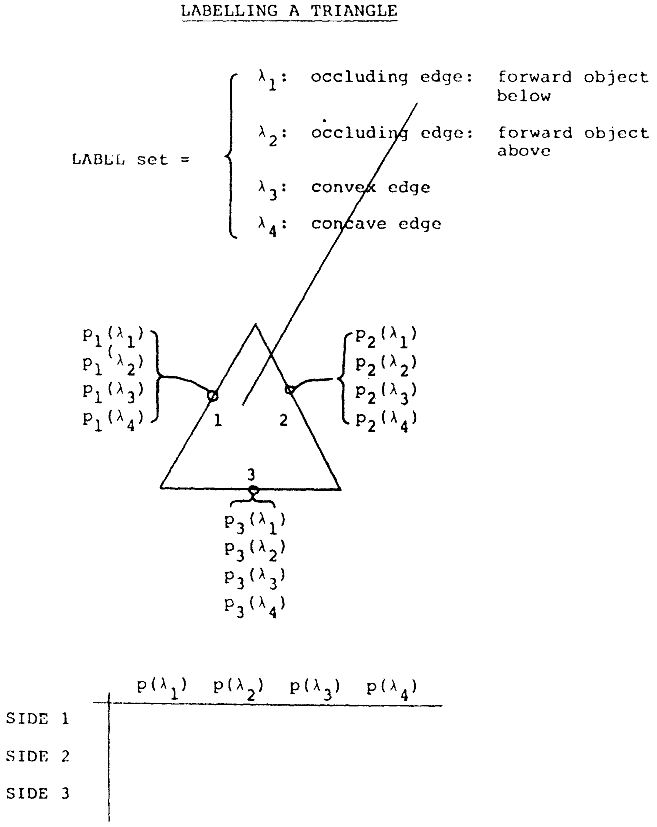

Fig. 1. Labeling a triangle.

Given that the process converges, we have

$$[P_i^{(k+1)} - P_i^{(k)}] = 0 \qquad (\text{i.e., } < \varepsilon).$$

In terms of matrix elements, this implies

$$p_i^{(k)}(\lambda_j)\left[\frac{(1 + q_i^{(k)}(\lambda_j))}{\sum_j p_i^{(k)}(\lambda_j)[1 + q_i^{(k)}(\lambda_j)]} - 1\right] = 0. \tag{8}$$

This means

$$1 + q_i^{(k)}(\lambda_j) = \sum_j p_i^{(k)}(\lambda_j)[1 + q_i^{(k)}(\lambda_j)]$$

for $p_i^{(k)}(\lambda_j) \neq 0$. Since $\sum_j p_i^{(k)}(\lambda_j) = 1$, we obtain

$$q_i^{(k)}(\lambda_j) = \sum_j p_i^{(k)}(\lambda_j) q_i^{(k)}(\lambda_j).$$

Since this expression is a constant it corresponds to the Case 1, viz., $Q_i^{(k)} = \beta I$.

If $p_i^{(k)}(\lambda_\alpha) = 0$ for some α, $(1 \leq \alpha \leq j)$, then the requirement that the bracketed expression in (8) must be equal to zero implies that

$$\begin{aligned} q_i^{(k)}(\lambda_j) &= \sum_j p_i^{(k)}(\lambda_j) q_i^{(k)}(\lambda_j) \\ &= \beta \text{ (a scalar)}, \qquad j \neq \alpha. \end{aligned}$$

This further implies that the $q_i^{(k)}(\lambda_\alpha)$ can be arbitrary, which corresponds to Case 2.

A particular example of this occurs when $p_i^{(k)}(\lambda_j) = 0$ for all j except one value $j = \alpha$, for which $p_i^{(k)}(\lambda_\alpha) = 1$. This is a UIM which corresponds to Case 3.

D. Examples of Convergence

In order to give examples of these classes of fixed points, for continuity with earlier work we shall use the example described in

TABLE I
CORRELATION BETWEEN LABELS FOR THE TRIANGLE LABELING EXAMPLE

$\mathrm{Cor}(\lambda_1,\lambda_1)$ =	.467
$\mathrm{Cor}(\lambda_1,\lambda_2)$ =	−.600
$\mathrm{Cor}(\lambda_1,\lambda_3)$ =	.488
$\mathrm{Cor}(\lambda_1,\lambda_4)$ =	−.293
$\mathrm{Cor}(\lambda_2,\lambda_1)$ =	−.600
$\mathrm{Cor}(\lambda_2,\lambda_2)$ =	.467
$\mathrm{Cor}(\lambda_2,\lambda_3)$ =	−.293
$\mathrm{Cor}(\lambda_2,\lambda_4)$ =	.488
$\mathrm{Cor}(\lambda_3,\lambda_1)$ =	.488
$\mathrm{Cor}(\lambda_3,\lambda_2)$ =	−.293
$\mathrm{Cor}(\lambda_3,\lambda_3)$ =	−.143
$\mathrm{Cor}(\lambda_3,\lambda_4)$ =	−.143
$\mathrm{Cor}(\lambda_4,\lambda_1)$ =	−.293
$\mathrm{Cor}(\lambda_4,\lambda_2)$ =	.488
$\mathrm{Cor}(\lambda_4,\lambda_3)$ =	−.143
$\mathrm{Cor}(\lambda_4,\lambda_4)$ =	−.143

[1], which involves labeling a line drawing of a triangle. That is, given a line drawing of a triangle, we wish to determine the physical edge configuration underlying this drawing. For simplicity, we shall allow only four different edge configurations (see Fig. 1(a)). The compatibilities between pairs of these labels are taken as the correlations derived in [1]; these are listed in Table I. Since each of these labels is initially a possible interpretation for each line, the entire label set is attached to each line. Then, beginning with a set of initial probability estimates, the process iterates according to (1) or (2).

To demonstrate Case 1 solutions, uniform initial probability assignments were selected. These are represented in a matrix of the form shown in Fig. 1(c). The results of successive iterations toward a fixed point and the corresponding $q_i(\lambda)$ matrices are shown in Table II. Note that the process, beginning with no bias for any specific solution (labeling), terminates at effectively the *a priori* likelihood for each label, and that the q matrix approaches zero. This example corresponds to [1, fig. 6, case A].

The second example, shown in Table III, demonstrates a Case 3 solution. (This example corresponds to [1, fig. 6, Case G].) Beginning with a slight bias toward one solution, even with potentially contradictory evidence, it terminates with a UIM.

There are many other examples of relaxation labeling processes applied to realistic problems that supply further empirical evidence on the behavior of these processes (see, e.g., [2]–[6]).

Once convergence properties for a process have been determined, the question of the rate of convergence becomes important. Indeed, for some applications, the number of iterations required may turn out to be prohibitive. In the next section we consider techniques for accelerating the rate of convergence.

III. IMPROVING THE SPEED OF CONVERGENCE

The order or rate of convergence of norm-reducing iterative sequences such as (2a) is related to the deviation of the successive iterates from the true solution. If this deviation decreases as a linear function of the number of iterations, we say that the process exhibits linear convergence. While the theoretical rate of convergence of (2) depends upon the initial choice of $P_i^{(0)}$ as well as on the R and C matrices, experimentally it appears to exhibit a linear convergence. Therefore, it is necessary to improve the speed of such a scheme by a modification of the iterative sequence. This modification should be such that 1) probability vectors are mapped into probability vectors, 2) the stationary points are the UIM's, and 3) the speed of convergence is geometric. This can be accomplished by a sequence of the form

$$P_i^{(k+1)} = (P_i^{(k)})^\alpha (I + Q_i^{(k)})^\alpha (T_\alpha^{(k)})^{-1} \tag{9}$$

where α is a positive integer and $T_\alpha^{(k)}$ is the diagonal matrix whose elements are all equal to trace $(P_i^{(k)})^\alpha (I + Q_i^{(k)})^\alpha$. To show that sequence (9) satisfies the above properties, we note first that $\|P_i^{k+1}\| \le 1$, since trace $P_i^{k+1} = 1$ for all k. Therefore, the sequence (9) is bounded. As before, a sufficient condition for convergence is that

$$\lim_{k\to\infty} \|P_i^{(k+1)} - P_i^{(k)}\| = \lim_{k\to\infty} \|(P_i^{(k)})^\alpha (I + Q_i^{(k)})^\alpha (T_\alpha^{(k)})^{-1} - P_i^{(k)}\| = 0$$

or in terms of matrix elements

$$\frac{(p_i^{(k)}(\lambda_j))^\alpha (1 + q_i^{(k)}(\lambda_j))^\alpha}{\sum_j (p_i^{(k)}(\lambda_j))^\alpha (1 + q_i^{(k)}(\lambda_j))^\alpha} - p_i^{(k)}(\lambda_j) = 0.$$

Using arguments similar to those in Section II, we can show that for (9):

1) UIM's are stationary points;

2) cases for which $p_i(\lambda_j) = (1 + q_i(\lambda_j)) = 1/m$ are stationary points (note that this is a special case of the βI case in Section III);

3) $Q_i^{(k)} = 0$ is not a stationary point unless $P_i^{(k)}$ is a UIM.

We next show that sequence (9) has a geometrical rate of convergence. In other words, if sequence (6) takes k steps to converge, then sequence (9) would only take $[\log_\alpha k]$ steps. Since

$$P_i^{(k+1)} = P_i^{(k)}(I + Q_i^{(k)})(T_i^{(k)})^{-1}$$

we can approximate the $(k + s)$th step by

$$P_i^{(k+s)} = [P_i^{(k)}(I + Q_i^{(k)})(T_i^{(k)})^{-1}]. \tag{10}$$

Sequence (10) offers a sort of extrapolation to speed up (6) by carrying it over to the $(k + s)$th step. However, it lacks the property that trace $P_i^{(k+s)} = 1$. In order to take care of this we approximate

$$\left(\sum_j p_i^{(k)}(\lambda_j)(1 + q_i^{(k)}(\lambda_j))\right)^s$$

by

$$\left(\sum_j (p_i^{(k)}(\lambda_j))^s (1 + q_i^{(k)}(\lambda_j))\right)^s$$

thus obtaining the sequence

$$P_i^{(k+s)} = (P_i^{(k)})^s [I + Q_i^{(k)}]^s (T_s^{(k)})^{-1} \tag{11}$$

where $T_s^{(k)}$ is a matrix whose diagonal elements are trace $(P_i^{(k)})^s \times [I + Q_i^{(k)}]^s$. It is easily seen that (11) is of the form (9); indeed

$$P_i^{(k+1)} = (P_i^{(k)})^\alpha [I + Q_i^{(k)}]^\alpha [T^{(k)}]^{-1}.$$

TABLE II
TRIANGLE LABELING USING EQUAL INITIAL PROBABILITIES

Iteration number	Probabilities				q matrix (neighborhood contributions)			
0	.25	.25	.25	.25				
	.25	.25	.25	.25				
	.25	.25	.25	.25				
10	.295	.295	.205	.205	.007	.007	-.021	-.021
	.295	.295	.205	.205	.007	.007	-.021	-.021
	.295	.295	.205	.205	.007	.007	-.021	-.021
20	.321	.321	.179	.179	.004	.004	-.012	-.012
	.321	.321	.179	.179	.004	.004	-.012	-.012
	.321	.321	.179	.179	.004	.004	-.012	-.012
30	.335	.335	.165	.165	.002	.002	-.007	-.007
	.335	.335	.165	.165	.002	.002	-.007	-.007
	.335	.335	.165	.165	.002	.002	-.007	-.007
40	.343	.343	.157	.157	.001	.001	-.004	-.004
	.343	.343	.157	.157	.001	.001	-.004	-.004
	.343	.343	.157	.157	.001	.001	-.004	-.004
50	.347	.347	.153	.153	.000	.000	-.003	-.003
	.347	.347	.153	.153	.000	.000	-.003	-.003
	.347	.347	.153	.153	.000	.000	-.003	-.003

TABLE III
TRIANGLE LABELING USING UNEQUAL INITIAL PROBABILITIES

Iteration number	Probabilities				q matrix			
0	.3	.2	.3	.2				
	.3	.2	.3	.2				
	.3	.2	.3	.2				
3	.402	.143	.307	.148	.143	-.126	.033	-.101
	.402	.143	.307	.148	.143	-.126	.033	-.101
	.402	.143	.307	.148	.143	-.126	.033	-.101
6	.583	.054	.291	.072	.283	-.262	.088	-.165
	.583	.054	.291	.072	.283	-.262	.088	-.165
	.583	.054	.291	.072	.283	-.262	.088	-.165
9	.771	.007	.203	.018	.437	-.386	.140	-.209
	.771	.007	.203	.018	.437	-.386	.140	-.209
	.771	.007	.203	.018	.437	-.386	.140	-.209
12	.880	.000	.116	.003	.521	-.440	.216	-.220
	.880	.000	.116	.003	.521	-.440	.216	-.220
	.880	.000	.116	.003	.521	-.440	.216	-.220
15	.933	.000	.067	.000	.551	-.460	.281	-.225
	.933	.000	.067	.000	.551	-.460	.281	-.225
	.933	.000	.067	.000	.551	-.460	.281	-.225

Therefore, a single step of (9) is equivalent to α steps of (6); or, if we require k steps in (6) it is sufficient to perform $[\log_\alpha k]$ steps in (9). For instance, if $\alpha = 2$, what takes 100 iterations in (6) would take only about 7 steps in (9).

One could also use a more general form

$$P^{(k+1)} = (P^{(k)})^{\alpha_1}(I + Q^{(k)})^{\alpha_2}(T^{(k)})^{-1}, \qquad \alpha_1 \geq 2, \quad \alpha_2 \geq 2$$

where $T^{(k)}$ is a matrix whose diagonal elements are

$$\{\text{trace } (P_i^{(k)})^{\alpha_1}(I + Q_i^{(k)})^{\alpha_2}\}.$$

This sequence also remains bounded, since trace $P_i^{(k+1)} = 1$ and $\|P_i^{(k+1)}\| \leq 1$.

In order to test these suggestions for improving the rate of convergence, the simple triangle example already described was used again. While this example already converged fairly rapidly, it can be readily seen that $\alpha = 2$ and $\alpha = 3$ (in recursion (11)) yielded even faster convergence (see Table IV).

For a more realistic test a program that computes spatial layouts was used (for a complete description, see [12]). This program translates symbolic spatial relations (such as NEAR and LEFT

TABLE IV
SPEEDUP OF CONVERGENCE BY USING A RECURSION BASED ON αTH POWERS

Iteration number	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$
0	.50 0 .50 0 .30 0 .70 0 .50 0 .50 0		
1		.62 0 .38 0 .17 0 .83 0 .62 0 .38 0	.68 0 .32 0 .08 0 .92 0 .68 0 .32 0
2		.85 0 .15 0 .04 0 .97 0 .85 0 .15 0	.97 0 .03 0 0 0 1.0 0 .97 0 .03 0
3	.70 0 .30 0 .32 0 .68 0 .70 0 .30 0	.99 0 .01 0 0 0 1.0 0 .99 0 .01 0	1.0 0 0 0 0 0 1.0 0 1.0 0 0 0
6	.87 0 .13 0 .28 0 .72 0 .87 0 .13 0	1.0 0 0 0 0 0 1.0 0 1.0 0 0 0	
12	.98 0 .02 0 .13 0 .87 0 .98 0 .02 0		

OF) between idealized objects into a probability distribution over the possible coordinate locations for the objects. The models for the symbolic relations are embedded in the compatibility functions and the label sets correspond to possible coordinate locations for the objects. The relaxation algorithm (2) is then used to disambiguate the positional locations for each object, given a set of spatial relations between pairs of objects. Beginning with a uniform distribution of probabilities over the label set (i.e., no *a priori* bias on the positions), the algorithms required a very large number (75 to 100) of iterations before disambiguation. The dramatic acceleration results obtained using (11) are shown in Fig. 2 for $\alpha = 2$, 3, and 4. For conciseness, rather than showing the actual probabilities, an information measure (one minus the entropy) is used. The information measure for object i is computed by

$$I(i) = 1 + \frac{\sum_{x,y} p_i(x,y) \log p_i(x,y)}{\log (1/N)}$$

where N is the number of position labels (coordinate pairs).

Note that whereas before well over a hundred iterations were required ($\alpha = 1$), now only 5 or 10 are sufficient. In order to examine the stability of this procedure, a number of different layout configurations, some involving implicit constraints, were computed. In each case, the resulting spatial layouts were the same for $\alpha = 1$, 2, 3, and 4.

Before using this process with a large α, however, a potential pitfall must be mentioned. This pitfall involves the digital implementation of these algorithms and the possibility of cumulative numerical roundoff errors driving the process to an instability. While an increase in the order of the process accelerates convergence, it also introduces instability in numerical problems. This is because products of numbers are involved which may be of substantially different magnitudes and, furthermore, powers of these numbers are involved in the products. Thus there is a tradeoff between the number of iterations required for termination and the numerical stability of the process. This tradeoff must be considered individually for each application. In the next section we discuss in more detail some guidelines for detecting numerical instabilities.

IV. NUMERICAL INSTABILITY

We will now indicate some of the numerical difficulties that might arise in using recursions (6) or (9). The types of instabilities may fall into the following (not necessarily exhaustive) categories.

1) $Q^{(k)} \approx 0$ in (6). Since the process is stationary for any choice of $P_i^{(k)}$ with trace $P_i^{(k)} = 1$, roundoff errors would cause instability.

2) $Q^{(k)} \approx -I$ in (6). Here factors of the form 0/0 may arise. These indeterminate situations can lead to large numerical instabilities.

3) When using sequence (9), since very small numbers are raised to powers, underflow or overflow may result if α is large.

Numerical instability is demonstrated in the triangle labeling example when a uniform initial labeling is used (see Table V). Up to about 60 iterations the process is slowly converging with the q-terms becoming smaller and smaller. Suddenly, just before iteration 70, the process becomes numerically unstable and begins to converge very rapidly toward a different fixed point. Fortunately, however, it is possible to use simple extrapolation techniques for detecting such instabilities.

One technique for determining whether a given sequence is converging to a stable stationary point is to extrapolate, by any of several well-known procedures [11], and then to compare the results. While the extrapolated sequence may not itself be stable, the difference between the extrapolation and the original sequence may be sufficient for detecting instabilities.

As an example of extrapolation, one can take the equally spaced iterates $P_i^{(m)}$, $P_i^{(2m)}$, and $P_i^{(3m)}$ and construct the new estimate using Aitken's δ^2 process [13]:

$$\hat{P}_i^{(3m)} = P_i^{(3m)} - \frac{(P_i^{(3m)} - P_i^{(2m)})^2}{(P_i^{(3m)} - 2P_i^{(2m)} + P_i^{(m)})}. \tag{12}$$

This extrapolated term should be a strong indicator of the direction of the process, and the difference $\|P_i^{(3m)} - \hat{P}_i^{(3m)}\|$ should indicate whether an instability is encountered.

This extrapolation has been calculated for the triangle labeling example, and the results are shown in Table V. There are other techniques for reducing numerical problems in iterative processes, such as using scaled variables, higher precision, and so on. While these techniques are available if needed, it should be stressed that in all of the realistic applications of relaxation labeling processes to date (e.g., [2]–[6]), numerical instability has not yet been encountered.

V. SUMMARY AND CONCLUSIONS

This correspondence has considered certain mathematical aspects of probabilistic relaxation processes. These aspects were studied by first representing the process in an iterative matrix form. The bounded norms of the successive P matrices immediately guarantee that the process is nondivergent. Algebraic considerations then provide a characterization of the different ways in which the process can converge.

The two interesting types of convergence are 1) convergence to an unambiguous interpretation matrix (UIM) in which a node has only one possible label, and 2) convergence to an ambiguous labeling, when the neighborhood contribution approaches a constant. This latter case corresponds, in an intuitive sense, to the situation in which the network contains no additional local information suitable for further resolving ambiguities. In such cases control should be passed to another process such as a search procedure.

The remaining possibility for convergence (Class 2 in Section II-C) suggests a consideration that may be important in applying

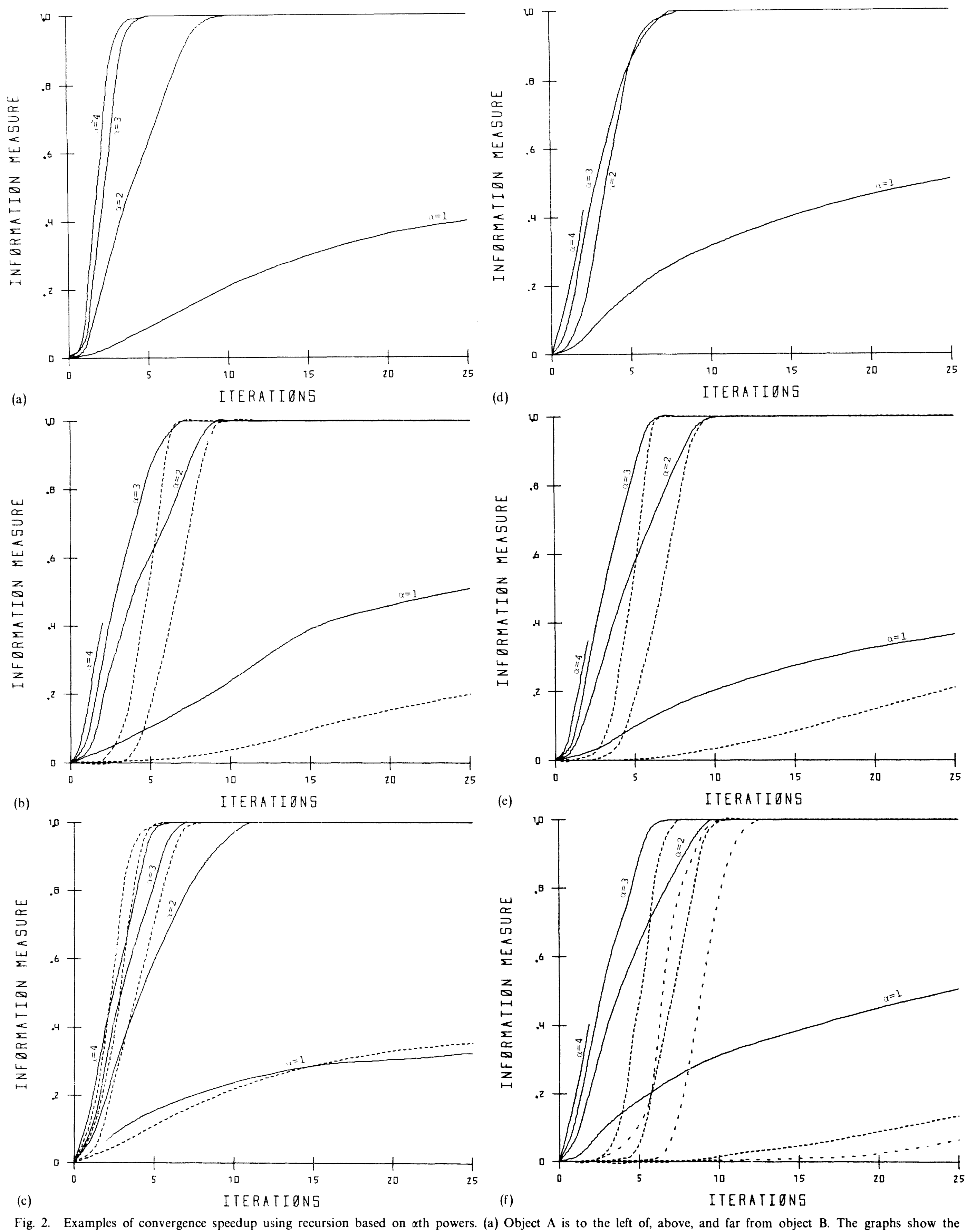

Fig. 2. Examples of convergence speedup using recursion based on αth powers. (a) Object A is to the left of, above, and far from object B. The graphs show the information measures for object A as a function of iteration number. (b) Object A is to the left of object B, which is to the left of object C. Solid curves: information measures for objects A and C; dashed curves: measures for object B. (c) Object A is to the left of objects B and C; object B is above object C. Solid curves: information measures for object A; dashed curves: measures for objects B and C. (d) Objects A, C, D, E are to the left of, to the right of, above, and below object B, respectively. The graphs show the information measures for objects A, C, D, and E. (e) Same as (d), and in addition, object D is above and to the right of A, and above and to the left of C, while object E is below and to the right of A, and below and to the left of C. Solid curves: measures for objects A, C, D, E; dashed curves: measures for object B. (f) A contradictory case: object A is to the left of B, B is to the left of C, C is to the left of A: D and E are above and below B, respectively. Solid curves: measures for D, E; dashed curves: for B; dotted curves: for A, C.

TABLE V

Iteration number	Probabilities				q matrices				Extrapolated probabilities (K=10)			
0	.25	.25	.25	.25								
	.25	.25	.25	.25								
	.25	.25	.25	.25								
15	.31	.31	.19	.19	.006	.006	-.016	-.016	.35	.35	.15	.15
	.31	.31	.19	.19	.006	.006	-.016	-.016	.35	.35	.15	.15
	.31	.31	.19	.19	.006	.006	-.016	-.016	.35	.35	.15	.15
30	.33	.33	.17	.17	.002	.002	-.007	-.007	.35	.35	.15	.15
	.33	.33	.17	.17	.002	.002	-.007	-.007	.35	.35	.15	.15
	.33	.33	.17	.17	.002	.002	-.007	-.007	.35	.35	.15	.15
60	.35	.35	.15	.15	-.002	.002	-.003	-.001	.35	.35	.15	.15
	.35	.35	.15	.15	-.002	.002	-.003	-.001	.35	.35	.15	.15
	.35	.35	.15	.15	-.002	.002	-.003	-.001	.35	.35	.15	.15
70	.29	.42	.13	.16	-.059	.059	-.031	.027	.35	.35	.15	.15
	.29	.42	.13	.16	-.059	.059	-.031	.027	.35	.35	.15	.15
	.29	.42	.13	.16	-.059	.059	-.031	.027	.35	.35	.15	.15
73	.18	.55	.10	.18	-.161	.168	-.086	.075	.35	.35	.15	.15
	.18	.55	.10	.18	-.161	.168	-.086	.075	.35	.35	.15	.15
	.18	.55	.10	.18	-.161	.168	-.086	.075	.35	.35	.15	.15

relaxation labeling processes. This involves creating a balance between the c_{ij} and r_{ij} and the current probability assignments. While such situations are highly unlikely in realistic problems, they could conceivably arise if automatic techniques for inferring compatibilities (such as optimization techniques over a given sample space) were used.

Since the matrix $Q_i^{(k)}$ depends upon the correlation matrix r_{ij} and the weights c_{ij}, it is very difficult to determine the rate or speed of convergence theoretically. However, it has been demonstrated that sequence (9) can greatly improve the speed of convergence. To see why this works we recall that in effect the size of the neighborhood over which information can travel to influence each point increases with the iteration number [2]. When the information from the next larger neighborhood is only slightly different from the smaller neighborhood's information, then the terms in the updating rule are almost identical. Raising the updating rule to a power then provides an acceleration of this slow change by lumping the effects of more than one of the original iterations into each new iteration. The overall result of this is to speed up the convergence.

There is a danger involved with this speedup technique, however. Some of the ambiguous fixed points (Case 1 or Case 2 configurations) may be driven to a UIM by incorrectly biased initial information. To circumvent this difficulty, the original process (2) could be used for the early iterations, followed by the higher order process (9) for the later iterations. The proper point for conversion would then need to be determined either by a trend analysis or on the basis of local probability distributions.

This correspondence has dealt almost exclusively with mathematical properties of the mechanism employed for combining evidence in a relaxation labeling process. While the updating rule exhibits appropriate behavior in the abstract, the entire process must be considered before more concrete observations are possible. In particular, one must consider the symbolic structure of the process, i.e., the labels, their interpretations, and the graph structure specifying their interdependence. It is generally very difficult to study arbitrary symbol-manipulating processes independently of specific applications. However, the mechanism underlying relaxation labeling processes is characterizable in the abstract. Studying these processes from this point of view offers an understanding of their behaviors complementary to that obtainable through applications.

Acknowledgment

The continued encouragement and critical remarks of A. Rosenfeld are deeply appreciated, as is the help of S. Rowe and Y. Leclerc in preparing the text and drawings for this paper.

References

[1] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 420-433, 1976.

[2] S. W. Zucker, "Relaxation labeling and the reduction of local ambiguities," in *Proc. Third Int. Joint Conf. Pattern Recognition*, San Diego, Nov. 1976; see also C. H. Chen, Ed., *Pattern Recognition and Artificial Intelligence*. New York: Academic, 1977.

[3] S. W. Zucker, R. A. Hummel, and A. Rosenfeld, "An application of relaxation labeling to line and curve enhancement," *IEEE Trans. Comput.*, vol. C-26, pp. 394-403 and 922-929, 1977.

[4] H. Barrow and J. M. Tenenbaum, "Experiments in interpretation-guided segmentation," Artificial Intelligence Center, Stanford Res. Inst., Tech. Note 123, Mar. 1976.

[5] E. M. Riseman and M. A. Arbib, "Computational techniques in visual systems: Part II. Segmenting static scenes," Computer and Information Science, Univ. of Massachusetts, Amherst, MA, COINS Tech. Rep. 76-11, July 1976.

[6] L. S. Davis and A. Rosenfeld, "An application of relaxation labeling to spring-loaded template matching," *Proc. Third Int. Joint Conf. Pattern Recognition*, San Diego, Nov. 1976.

[7] J. R. Ullmann, "Associating parts of patterns," *Info. and Control*, vol. 9, pp. 583-601, 1966.

[8] ——, "An algorithm for subgraph isomorphism," *J. ACM*, vol. 23, pp. 31-42, 1976.

[9] D. Waltz, "Understanding line drawings of scenes with shadows," in *The Psychology of Computer Vision*, P. H. Winston, Ed. New York: McGraw-Hill, pp. 19-91, 1976.

[10] H. Barrow and J. M. Tenenbaum, "MSYS: A system for reasoning about scenes," Artificial Intelligence Center, Stanford Research Institute, Tech. Note 121, Apr. 1976.

[11] A. S. Householder, *Theory of Matrices in Numerical Analysis*. New York: Dover, 1975.

[12] R. Haar, "Computational models of spatial relations," Computer Science Center, Univ. of Maryland, Tech. Rep. 478, Aug. 1976. Also, *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 557-559, 1977.

[13] B. Jones, "Extrapolation for higher orders of convergence," *J. Inst. Math. and Its Appl.*, vol. 17, pp. 27-36, 1976.