

Scene Labeling by Relaxation Operations

AZRIEL ROSENFELD, FELLOW, IEEE, ROBERT A. HUMMEL, AND STEVEN W. ZUCKER, MEMBER, IEEE

Abstract—Given a set of objects in a scene whose identifications are ambiguous, it is often possible to use relationships among the objects to reduce or eliminate the ambiguity. A striking example of this approach was given by Waltz [13]. This paper formulates the ambiguity-reduction process in terms of iterated parallel operations (i.e., relaxation operations) performed on an array of (object, identification) data. Several different models of the process are developed, convergence properties of these models are established, and simple examples are given.

I. INTRODUCTION

SUPPOSE that we are analyzing a picture or scene, with the aim of describing it, and that we have detected a set of objects a_1, \dots, a_n in the scene, but have not identified them unambiguously. The relationships that exist among the objects can often be used to reduce, or even eliminate, the ambiguity.

To illustrate this idea, let us suppose that the "objects" are the individual line segments in an ideal line drawing representing a set of polyhedra. Each line segment has several interpretations; it may represent a) a convex dihedral angle, with both faces visible, b) a concave dihedral angle, with both faces visible, or c) a dihedral angle (in the limiting case, a flat cutout) with only one face visible, so that the line is an occluding edge. In c) there are two subcases, since the visible face can be on either side of the line. When lines meet at a vertex, however, not all of the possible combinations of these interpretations will be consistent, so that the ambiguity of the drawing can be reduced. This problem domain has been investigated by Clowes [1], Huffman [5], Waltz [13], and others.

As a simple example, consider the triangle shown in Fig. 1(a). This can be interpreted as

- α) a triangular cutout floating above the background (Fig. 1(b)),
- β) a triangular hole in the background (Fig. 1(c)),

(In both of these cases, all the lines are occluding edges, but in α) the triangle face is visible, while in β) the background face is visible.)

- γ) a triangular flap of the background, folded toward the viewer along one of the edges (Figs. 1(d)–(f)); here one edge is concave, and the other two are occluding, with the triangle in front,
- δ) the same, but folded back (Figs. 1(g)–(i)); one edge is convex, and the other two are occluding, with the triangle in back.

Manuscript received July 16, 1975; revised January 16, 1976. This work was supported by the Division of Computing Research, National Science Foundation, under Grant GJ-32258X.

The authors are with the Computer Science Center, University of Maryland, College Park, MD 20742.

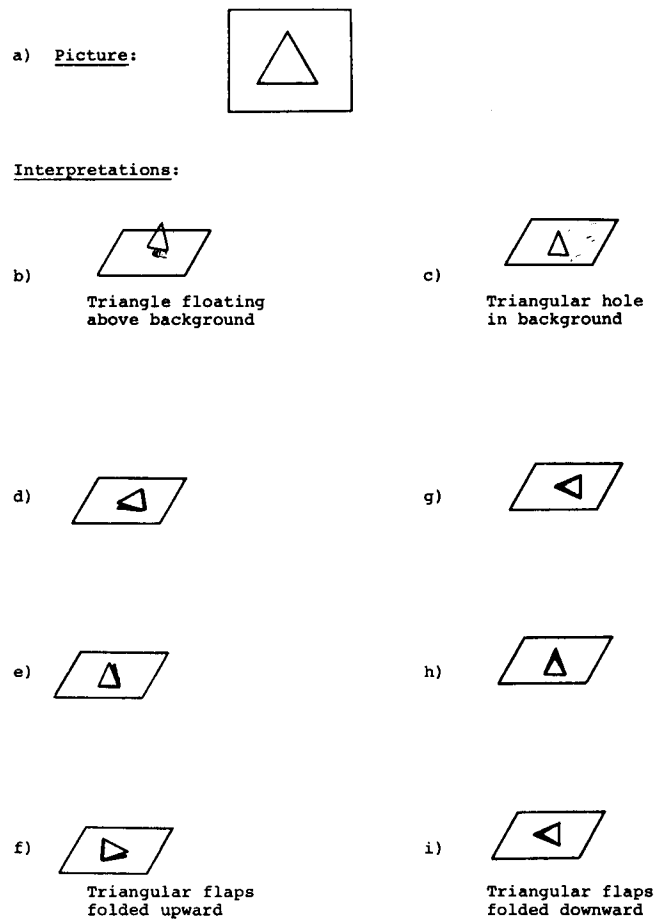


Fig. 1. Simple example of scene labeling.

We thus have a total of eight cases, but this is only a fraction of the $4^3 = 64$ possible combinations. In some circumstances, there could be a unique consistent interpretation, or even no consistent interpretation (i.e., we have an "impossible object"); these possibilities will be discussed further elsewhere in this paper.

Up to now, we have treated each interpretation as either possible or impossible; fuzzy or probabilistic interpretations were not allowed. This may be reasonable in the ideal line drawing case, where the individual lines are completely ambiguous, and we have no evidence favoring one interpretation over another, so that there is no basis for introducing weights or probabilities. (We exclude here the possibility that the observer has preferences—e.g., he might favor occluding edges over convex or concave edges; see Section VI.) Suppose, however, that we were given a gray-scale image, rather than a line drawing. It then may not be equally likely that the edges in this image represent convex, concave, or occluded dihedrals. For example, under some

conditions of illumination we might expect a convex dihedral to have a slight highlight along it and a concave dihedral to have a slight shadow. If the image is noisy, we may not be certain that we have detected such highlights or shadows, but we may still be able to attach weights or probabilities to the possible interpretations of each edge.

This paper describes several models for ambiguity-reduction processes. Section II presents a discrete model in which each interpretation of each object is either possible or impossible. The process is similar to the filtering scheme of Waltz [13, Section 2.4], except that it is implemented as an iterated parallel operation on the array of (object, interpretation) pairs, rather than sequentially, one object at a time. It is proved that if one initially assigns every possible interpretation to every object and then discards incompatible interpretations until no further discarding is possible, what remains is the greatest set of consistent interpretations (compare Turner [11, Section 10.2]). This set can be used as a starting point in constructing consistent ambiguous interpretations.

Section III introduces a fuzzy model which generalizes the discrete model by allowing each interpretation to have a weight between 0 and 1. Here again, we initially assign every interpretation to every object with weight 1 and then repeatedly weaken the weights, as dictated by quantitative compatibility relations, by performing an iterated parallel operation on the array of (object, interpretation weight) pairs. It is shown that this process converges to a strongest set of compatible weights.

In Section IV we develop a probabilistic model in which the sum of the weights of the interpretations of any given object is required to be 1, so that the weights can be interpreted as probabilities. We first study a linear process of weight modification (again, performed in parallel on the array of (object, weight) pairs), but this turns out, in many cases, to converge to a fixed limit no matter how the initial weights are chosen. Section V discusses nonlinear weight modification processes which do not have this disadvantage.

An early use of constraint satisfaction in problem solving is described by Fikes [3]. More recently, Gaschnig [4] has used discrete relaxation techniques for applying constraints in solving cryptarithmic problems, and Tenenbaum [10] proposes the use of a generalized relaxation scheme for region merging in a scene analysis system. The present authors and their colleagues are also applying relaxation methods to a variety of scene processing and labeling problems, including curve detection, cluster detection, noise cleaning, and template matching; the results will be described in a series of forthcoming reports.

II. THE DISCRETE MODEL

The model developed in this section generalizes the ideal line drawing example discussed in Section I. As already mentioned, it is basically a parallel version of the filtering process implemented by Waltz [13].

Let $A = \{a_1, \dots, a_n\}$ be the set of objects to be labeled, and $\Lambda = \{\lambda_1, \dots, \lambda_m\}$ the set of possible labels. For any given object a_i , not every label in Λ may be appropriate;

for example, if the objects are the junctions (rather than the lines) in an ideal line drawing, some labels will be appropriate for L -junctions, others for fork junctions, others for arrow junctions, etc. (see Waltz [13, Fig. 2.3]). Let $\Lambda_i \subseteq \Lambda$ be the set of labels that are compatible with (i.e., possible for) object a_i , $1 \leq i \leq n$.

For each pair of objects (a_i, a_j) , where $i \neq j$, some pairs of labels may be compatible and others may not. Let $\Lambda_{ij} \subseteq \Lambda_i \times \Lambda_j$ be the set of compatible pairs of labels; thus $(\lambda, \lambda') \in \Lambda_{ij}$ means that it is possible that a_i has label λ and a_j has label λ' . Here Λ_{ij} depends on the relationship between a_i and a_j in the scene. If a_i and a_j are irrelevant to one another, then there are no restrictions on the possible pairs of labels that they can have, so that $\Lambda_{ij} = \Lambda_i \times \Lambda_j$. In the line drawing models studied by Waltz *et al.*, two junctions constrain (i.e., are relevant to) one another only if there is a line connecting them. For uniformity of notation we define $\Lambda_{ii} = \{(\lambda, \lambda) \mid \lambda \in \Lambda_i\}$ for all i . We can, if desired, also assume that $\Lambda_{ij} = \Lambda_{ji}$ for all i and j .

By a *labeling* $\mathcal{L} = (L_1, \dots, L_n)$ of A we mean an assignment of a set of labels $L_i \subseteq \Lambda$ to each $a_i \in A$. We say that the labeling \mathcal{L} is contained in the labeling $\mathcal{L}' = (L'_1, \dots, L'_n)$ if $L_i \subseteq L'_i$, $1 \leq i \leq n$; in this case we write $\mathcal{L} \subseteq \mathcal{L}'$. By the union $\mathcal{L} \cup \mathcal{L}'$ of two labelings \mathcal{L} and \mathcal{L}' , we mean the labeling that assigns to each $a_i \in A$ the label set $L_i \cup L'_i$; the intersection can be defined similarly.

The labeling \mathcal{L} is called *consistent* if, for all i, j , we have

$$(\{\lambda\} \times L_j) \cap \Lambda_{ij} \neq \emptyset, \quad \text{for all } \lambda \in L_i.$$

For $i \neq j$, this means that for each pair of objects (a_i, a_j) and each label λ in L_i there exists a label λ' in L_j that is compatible with λ , i.e., $(\lambda, \lambda') \in \Lambda_{ij}$. Note that if $\Lambda_{ij} = \Lambda_i \times \Lambda_j$, this is no restriction, provided $L_j \neq \emptyset$. For $i = j$, the condition reduces to

$$\lambda \in L_i \text{ implies } (\lambda, \lambda) \in \Lambda_{ii},$$

in other words, to $L_i \subseteq \Lambda_i$, which means that every label in L_i is a possible label for a_i .

There always exist consistent labelings; in particular, the null labeling $\mathcal{L}_0 \equiv (\emptyset, \dots, \emptyset)$ is trivially consistent (the "for all $\lambda \in L_i$ " condition is vacuously satisfied). On the other hand, if $\mathcal{L} = (L_1, \dots, L_n)$ is a nonnull consistent labeling, then *every* L_i must be nonempty. Indeed, if we had $L_i \neq \emptyset$ and $L_j = \emptyset$ for some i, j , the definition of consistency would be immediately violated. We can also prove that there exists a greatest consistent labeling, i.e., a labeling $\mathcal{L}^{(\infty)}$ such that

- 1) $\mathcal{L}^{(\infty)}$ is consistent.
- 2) For any consistent labeling \mathcal{L} we have $\mathcal{L} \subseteq \mathcal{L}^{(\infty)}$.

We first prove the following proposition.

Proposition 1: If \mathcal{L} and \mathcal{L}' are consistent, so is $\mathcal{L} \cup \mathcal{L}'$.

Proof: For all $\lambda \in (L_i \cup L'_i)$ we have $\lambda \in L_i$ or $\lambda \in L'_i$. In the former case there exists $\lambda' \in L_j$, for any j , such that $(\lambda, \lambda') \in \Lambda_{ij}$, and in the latter case there exists such a $\lambda' \in L'_j$. Thus in either case there is such a $\lambda' \in (L_j \cup L'_j)$.

Corollary 2: There is a greatest consistent labeling $\mathcal{L}^{(\infty)}$.

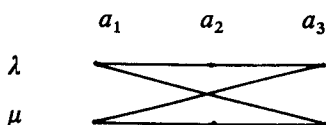
Proof: There are only finitely many possible labelings, since A and Λ are finite. Thus by induction on Proposition 1, any union of consistent labelings is consistent. (In fact, the proof of Proposition 1 could be easily extended to infinite unions, if necessary.) Thus the union $\mathcal{L}^{(\infty)}$ of all consistent labelings is consistent, and clearly any consistent labeling \subseteq this union.

It should be pointed out that $\mathcal{L}^{(\infty)}$ may be null; in other words, there may not exist a nonnull consistent labeling. Even if $\mathcal{L}^{(\infty)}$ is not null, it is possible that there is no way to assign *single* labels consistently to the objects. For example, suppose that $A = \{a_1, a_2, a_3\}$, and that

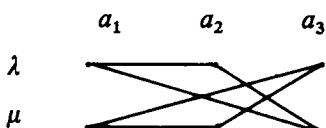
$$\begin{aligned} \Lambda &= \Lambda_1 = \Lambda_2 = \Lambda_3 = \{\lambda, \mu\} \\ \Lambda_{12} &= \Lambda_{23} = \{(\lambda, \lambda), (\mu, \mu)\} \\ \Lambda_{13} &= \{(\lambda, \mu), (\mu, \lambda)\}. \end{aligned}$$

Then readily, the labeling $\mathcal{L}^{(\infty)} = (\Lambda, \Lambda, \Lambda)$ is consistent, but it can be verified that no labeling using single labels is consistent. (If we discard, let us say, μ from a_1 , we must discard μ from a_2 and λ from a_3 ; this in turn implies that we must discard μ from a_3 , so that the whole labeling becomes null.) Thus this situation corresponds to an "impossible object." We call a labeling *unambiguous* if it is consistent and assigns only a single label to each object.

A useful way of representing labelings is in terms of what we shall call the *labeling network*. This is a graph G whose nodes are the pairs (i, λ) , for all $1 \leq i \leq n$ and all $\lambda \in \Lambda_i$. The nodes (i, λ) and (j, λ') are joined by an arc if and only if $(\lambda, \lambda') \in \Lambda_{ij}$. To any labeling $\mathcal{L} = (L_1, \dots, L_n)$ there corresponds a subgraph $G_{\mathcal{L}}$ of G whose nodes are the pairs (i, λ) for all $\lambda \in L_i$. For example, the graph corresponding to the labeling $\mathcal{L}^{(\infty)}$ in the preceding paragraph can be represented schematically by



\mathcal{L} is consistent if and only if, for each node (i, λ) of $G_{\mathcal{L}}$ and each j , there exists a node (j, λ') of $G_{\mathcal{L}}$ that is joined to (i, λ) by an arc. \mathcal{L} is unambiguous if and only if it is consistent and has only one node (i, λ) , for each i . Readily, if \mathcal{L} is unambiguous, the subgraph $G_{\mathcal{L}}$ is a clique, since every (i, λ) must be joined by an arc to every other. An ambiguous consistent labeling, on the other hand, need not even have a connected graph. For example, the graph defined by



has two disjoint connected components, but corresponds to a consistent labeling. When the network is disconnected, the scene labeling problem can be broken up into disjoint subproblems, each involving only the labels associated with a particular connected component.

To illustrate these ideas in greater detail, let us examine more closely the triangle example of Fig. 1. We shall use the $(+, -, \rightarrow)$ line labels used by Waltz [13] *et al.*; their meaning is as follows:

Label	Meaning
+	convex,
-	concave,
\rightarrow	occluding, with the right-hand region (when one faces along the line in the direction of the arrow) hiding the left-hand region.

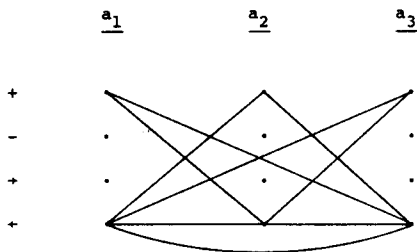
Note that when a line is labeled with an arrow, the arrow can point in either direction; thus the arrow should actually be regarded as two different labels. *A priori*, each line can have any one of these labels, but at an L junction, only six of the $4^2 = 16$ possible label pairs are compatible:

Label pair	Meaning
	junction floating above background
	junction on hole in background
	one line of junction is a concave edge, the other is above background
	one line of junction is a convex edge, the other is on a hole in background

In our triangle example, for each of the three lines a_1, a_2, a_3 we have $\Lambda_i = \{+, -, \leftarrow, \rightarrow\}$, and for each pair of lines (a_i, a_j) , Λ_{ij} is the set of six pairs shown above. The network for this concept of compatibility is shown in Fig. 2(a). It consists of two components, one involving only $+$'s and \leftarrow 's, the other only $-$'s and \rightarrow 's; only the first of these components is shown in the figure. Each of these components contains four cliques, as shown in Fig. 2(b); these cliques correspond to the unambiguous labelings of Fig. 1.

We now establish the validity of a parallel algorithm for constructing $\mathcal{L}^{(\infty)}$, the greatest consistent labeling. As indicated earlier, this algorithm is basically a parallel version of the filtering process used by Waltz [13]. In terms of the network, this algorithm discards any node (i, λ) if, for some j , there does not exist a node (j, λ') that is joined to (i, λ) by an arc. Such nodes cannot possibly belong to any unambiguous labeling; hence if our main interest is in unambiguous labelings, it is safe to discard such nodes.

Our algorithm operates as follows. We start with the initial labeling $\mathcal{L}^{(0)} = \{\Lambda_1, \dots, \Lambda_n\}$. Let $\mathcal{L}^{(k)}$ be the labeling at the k th application of the algorithm. To obtain the labeling at the $(k + 1)$ st step, we discard from each $L_i^{(k)}$ any label λ such that $(\{\lambda\} \times L_j^{(k)}) \cap \Lambda_{ij} = \emptyset$ for some j . In other words, we keep the label λ at a_i if, for every a_j , there is a label $\lambda' \in L_j^{(k)}$ at a_j which is compatible with λ ; otherwise,

a) Labelling network for this example

(only the component involving +'s and +'s is shown; the other component is an isomorphic graph involving -'s and +'s instead.)

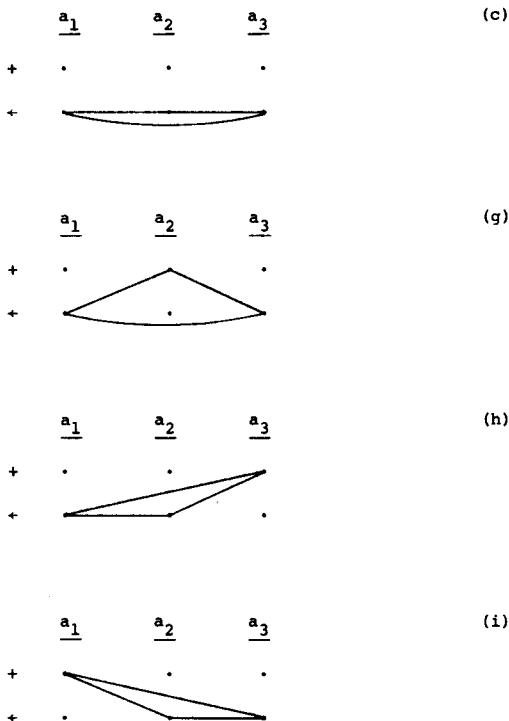
b) Cliques (for the component involving +'s and +'s)Corresponding interpretations of Figure 1

Fig. 2. Compatibilities for example in Fig. 1.

we discard λ . We shall refer to this algorithm in what follows as Δ .

As an immediate consequence of the definitions of Δ and of consistency, we have the following propositions.

Proposition 3: $\Delta\mathcal{L} = \mathcal{L}$ if and only if \mathcal{L} is consistent.

Proposition 4:

$$\mathcal{L}^{(\infty)} \subseteq \dots \subseteq \mathcal{L}^{(k+1)} \subseteq \mathcal{L}^{(k)} \subseteq \dots \subseteq \mathcal{L}^{(0)}$$

Proof: Since Δ discards labels, but never adds them, we clearly have $\mathcal{L}^{(k+1)} \subseteq \mathcal{L}^{(k)}$ for all $k = 0, 1, 2, \dots$. We now prove by induction that $\mathcal{L}^{(\infty)} \subseteq \mathcal{L}^{(k)}$ for all k . Since $\mathcal{L}^{(\infty)}$

is a consistent labeling, we have $L_i^{(\infty)} \subseteq \Lambda_i$ for all i , from the definition of consistency; thus $\mathcal{L}^{(\infty)} \subseteq \mathcal{L}^{(0)}$. Suppose that $\mathcal{L}^{(\infty)} \subseteq \mathcal{L}^{(k)}$ and let $\lambda \in L_i^{(\infty)} \subseteq L_i^{(k)}$. Then for every j , there exists $\lambda' \in L_j^{(\infty)} \subseteq L_j^{(k)}$ such that $(\lambda, \lambda') \in \Lambda_{ij}$, by definition of consistency. Hence λ is not discarded from object a_i at the k th step, so that $\lambda \in L_i^{(k+1)}$. Since i and λ were arbitrary, we have thus shown that $L_i^{(\infty)} \subseteq L_i^{(k+1)}$ for all i , so that $\mathcal{L}^{(\infty)} \subseteq \mathcal{L}^{(k+1)}$.

We can now prove the following theorem.

Theorem 5: There exists a k such that $\mathcal{L}^{(k)} = \mathcal{L}^{(\infty)}$.

Proof: A and Λ are finite, so that we cannot have $\mathcal{L}^{(k+1)} \subsetneq \mathcal{L}^{(k)}$ for infinitely many k , since $\mathcal{L}^{(k+1)} \subseteq \mathcal{L}^{(k)}$ means that at least one label was discarded from at least one object. Hence for some k we have $\mathcal{L}^{(k)} = \mathcal{L}^{(k+1)}$, and by Proposition 3, this means that $\mathcal{L}^{(k)}$ is consistent. However, $\mathcal{L}^{(\infty)} \subseteq \mathcal{L}^{(k)}$ (Proposition 4), and $\mathcal{L}^{(\infty)}$ is the greatest consistent labeling (Corollary 2); hence we must have $\mathcal{L}^{(\infty)} = \mathcal{L}^{(k)}$.

We have thus shown that Δ , after a finite number of repetitions, must stop at the unique greatest consistent labeling $\mathcal{L}^{(\infty)}$. The number of repetitions required is at most mn , since as just pointed out, at each step before the process stops, at least one label must be discarded from at least one object.

To actually find unambiguous labelings, we can use a tree search procedure similar to that used by Waltz [13]. The number of combinations to be tested is, in general, much smaller when we start with $\mathcal{L}^{(\infty)}$ than if we had to start with $\mathcal{L}^{(0)}$. The search procedure can be as follows: for any object a_i such that $L_i^{(\infty)}$ is not a singleton, pick a particular label $\lambda \in L_i^{(\infty)}$ and discard all other labels. If the labeling

$$\mathcal{L}' = \{L_1^{(\infty)}, \dots, L_{i-1}^{(\infty)}, \{\lambda\}, L_{i+1}^{(\infty)}, \dots, L_n^{(\infty)}\}$$

is consistent, we can repeat the process for another object a_j . If \mathcal{L}' is not consistent, we can repeatedly apply Δ to \mathcal{L}' . By the proofs of Propositions 3 and 4 and Theorem 5, this will stop, after finitely many steps, at the greatest consistent labeling \mathcal{L}'' that is contained in \mathcal{L}' . If \mathcal{L}'' is null, there is no unambiguous labeling in which a_i has label λ , and we should start again with a different λ or a different a_i . If \mathcal{L}'' is nonnull, we have a consistent labeling which is now unambiguous at i , and we can repeat the process for some other a_j for which L_j'' is not a singleton. In this way we can eventually find all the unambiguous labelings of the given scene, if any exist.

III. A FUZZY MODEL

In this section we present a generalization of the discrete model, in which the labels have weights between 0 and 1, rather than simply being either absent or present. The weights of the labels of a given object are not required to sum to 1; the model is fuzzy, rather than probabilistic. (On probabilistic models see Sections IV and V.) This fuzzy model is not as satisfactory as the probabilistic models, because the algorithm for applying compatibility constraints in the fuzzy model can only decrease the weights of labels, but never increase them. Nevertheless, we present the model here because it does constitute a generalization of the

discrete model, and it also can serve to introduce the idea of weighted scene labeling.

Let $A = \{a_1, \dots, a_n\}$ and $\Lambda = \{\lambda_1, \dots, \lambda_m\}$ be the object and label sets, as defined in Section II. For each i , we are given a fuzzy label set Λ_i associated with the object a_i . This Λ_i is a fuzzy subset of Λ , i.e., a mapping from Λ into the interval $[0,1]$. We can think of it as defining, for each $\lambda \in \Lambda$, the degree to which λ is compatible with the object a_i . (In the discrete case, Λ_i can take on only the values 0 and 1, so that we can regard it as a subset of Λ , consisting of those λ 's that are mapped into 1; these are the λ 's that are compatible with a_i .)

In addition, for each pair of objects (a_i, a_j) , where $i \neq j$, we are given a fuzzy set Λ_{ij} of pairs of labels; this is a mapping from $\Lambda \times \Lambda$ into $[0,1]$. Here $\Lambda_{ij}(\lambda, \lambda')$ can be thought of as the degree to which label λ of object a_i is compatible with label λ' of object a_j . We shall assume here that

$$\Lambda_{ij}(\lambda, \lambda') \leq \inf(\Lambda_i(\lambda), \Lambda_j(\lambda'))$$

for all i, j, λ, λ' ; this corresponds to the requirement, in Section II, that $\Lambda_{ij} \subseteq \Lambda_i \times \Lambda_j$. (In the discrete case, Λ_{ij} takes on only the values 0 and 1, so that it can be regarded as a subset of $\Lambda_i \times \Lambda_j$, consisting of those pairs (λ, λ') that are mapped into 1.) Thus a_i and a_j impose no constraints on one another provided that $\Lambda_{ij}(\lambda, \lambda') = \inf(\Lambda_i(\lambda), \Lambda_j(\lambda'))$ for all λ, λ' . As in Section II, for uniformity of notation we define $\Lambda_{ii}(\lambda, \lambda) = \Lambda_i(\lambda)$ for all λ , and $\Lambda_{ii}(\lambda, \lambda') = 0$ for all $\lambda \neq \lambda'$. We can, if desired, also assume that $\Lambda_{ij} = \Lambda_{ji}$ for all i and j .

By a fuzzy labeling $\mathcal{L} = (L_1, \dots, L_n)$ of A we mean an assignment of a fuzzy subset L_i of Λ to each a_i , $1 \leq i \leq n$; in other words, for each a_i , we are given a mapping L_i from Λ into $[0,1]$, which assigns a weight to each label. We say that $\mathcal{L} \leq \mathcal{L}' = (L_1', \dots, L_n')$ if $L_i \leq L_i'$ (i.e., $L_i(\lambda) \leq L_i'(\lambda)$ for all λ), $1 \leq i \leq n$. We also define $\mathcal{L} \vee \mathcal{L}' = (L_1 \vee L_1', \dots, L_n \vee L_n')$, where \vee means "sup." Here $L_i \vee L_i'$ is the mapping which gives any $\lambda \in \Lambda$ the value $\sup(L_i(\lambda), L_i'(\lambda))$. These definitions are the standard fuzzy-set analogs of inclusion and union, respectively.

The fuzzy labeling \mathcal{L} is called *consistent* if, for all i, j , and λ , we have

$$\sup_{\lambda'} [L_j(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')] \geq L_i(\lambda)$$

where \wedge means "inf." For $i = j$, this reduces to $\Lambda_i(\lambda) \geq L_i(\lambda)$ for all i , which means that no label has greater weight at any object than its compatibility with that object. For $i \neq j$, the definition means that for each pair of objects (a_i, a_j) , and each label λ , there exists a label λ' whose weight at a_j and whose compatibility Λ_{ij} with λ are both at least as great as the weight of λ at a_i . It is not hard to see that if the fuzzy sets involved are allowed to take on only the values 0 and 1, this definition of consistency reduces to that in Section II. Indeed, $L_i(\lambda) = 1$ requires that $\sup_{\lambda'} [L_j(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')] = 1$, so that for some λ' we have $L_j(\lambda') = \Lambda_{ij}(\lambda, \lambda') = 1$. In other words, if $\lambda \in L_i$, there exists λ' such that $\lambda' \in L_j$ and $(\lambda, \lambda') \in \Lambda_{ij}$, just as in Section II.

There always exist consistent fuzzy labelings; in particular, the null labeling $\mathcal{L}_0 = (0, \dots, 0)$ (each L_i is the constant function 0, corresponding to the empty set) is trivially consistent. If \mathcal{L} is a nonnull consistent fuzzy labeling, then no L_j can be 0; if $L_i \neq 0$ and $L_j = 0$, i and j cannot satisfy the consistency condition. We now prove, as in Section II, that there exists a greatest consistent fuzzy labeling $\mathcal{L}^{(\infty)}$, such that for any consistent fuzzy labeling \mathcal{L} we have $\mathcal{L} \leq \mathcal{L}^{(\infty)}$.

Proposition 6: If \mathcal{L} and \mathcal{L}' are consistent, so is $\mathcal{L} \vee \mathcal{L}'$. Indeed, the sup of any set of consistent fuzzy labelings is consistent.

Proof: For all \mathcal{L} we have $\sup_{\lambda'} [L_j(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')] \geq L_i(\lambda)$ and $\sup_{\lambda'} [L_j'(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')] \geq L_i'(\lambda)$; hence, the sup of the left members \geq the sup of the right members, i.e.,

$$\sup_{\lambda'} [(L_j \vee L_j')(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')] \geq (L_i \vee L_i')(\lambda).$$

The same argument holds for an arbitrary set of consistent \mathcal{L} 's, taking the sup of all the left members and the sup of all right members.

Corollary 7: There is a greatest consistent fuzzy labeling $\mathcal{L}^{(\infty)}$.

Proof: The sup of all the consistent fuzzy labelings is consistent, by Proposition 6.

Note that here again, there is no guarantee that $\mathcal{L}^{(\infty)}$ is not null. Even if it is not, there may not exist *unambiguous* consistent fuzzy labelings, which give nonzero weight to just one label for each object.

As in Section II, we can define a *fuzzy labeling network* in which the node (i, λ) has weight $\Lambda_i(\lambda)$, and the arc joining (i, λ) to (j, λ') has weight $\Lambda_{ij}(\lambda, \lambda')$. It can be verified that this network is a fuzzy graph in the sense of [7]. If we threshold these weights (e.g., we keep only those nodes and arcs whose weights > 0), we obtain an ordinary graph.

To illustrate these ideas, consider the triangle example of Figs. 1 and 2. Suppose that the *a priori* weights of the labels \rightarrow for any line are 1; and of $+$ and $-$, 0.7. Suppose further that the compatibilities of the pairs of labels, when lines meet at an L -junction, are 1 for \wedge and \vee ; 0.7, for the other four pairs shown in Section II; and 0, for all other pairs. Thus the fuzzy labeling network for this example is as shown in Fig. 2(a), except that the nodes and arcs should have weights attached to them as just specified. It is not hard to see that for this choice of *a priori* weights, the greatest compatible fuzzy labeling of the triangle assigns weight 1 to the \rightarrow labels, and weight 0.7 to $+$ and $-$, at each line. The unambiguous fuzzy labelings are as in Fig. 2, except that the first one has all weights 1, while the other three involve weights of 0.7.

We can define a label weakening algorithm Ω that is guaranteed to converge to $\mathcal{L}^{(\infty)}$; it is a generalization of the algorithm Δ in Section 2. We start with the initial fuzzy labeling $\mathcal{L}^{(0)} = (\Lambda_1, \dots, \Lambda_n)$. To obtain $\mathcal{L}^{(k+1)}$ given $\mathcal{L}^{(k)}$, we let

$$L_i^{(k+1)}(\lambda) = \inf_j \left[\sup_{\lambda'} [L_j^{(k)}(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')] \right]$$

for all i, j , and λ . If only the values 0 and 1 are allowed, this definition reduces to

$$L_i^{(k+1)}(\lambda) = 1 \Leftrightarrow (\forall j)(\exists \lambda')[L_j^{(k)}(\lambda') = 1 \text{ and } \Lambda_{ij}(\lambda, \lambda') = 1]$$

which is the same as the criterion for keeping label λ at object a_i in Section II. It is easy to establish the following propositions.

Proposition 8: $\Omega\mathcal{L} = \mathcal{L}$ if and only if \mathcal{L} is consistent.

Proposition 9: $\mathcal{L}^{(\infty)} \leq \dots \leq \mathcal{L}^{(k+1)} \leq \dots \leq \mathcal{L}^{(0)}$.

Proof: The $j = i$ term of $L_i^{(k+1)}(\lambda)$ is $\sup_{\lambda'} [L_i^{(k)}(\lambda') \wedge \Lambda_{ii}(\lambda, \lambda')] = L_i^{(k)}(\lambda) \wedge \Lambda_{ii}(\lambda, \lambda) \leq L_i^{(k)}(\lambda)$; hence, the inf (on j) of all the terms is also $\leq L_i^{(k)}(\lambda)$, for all i and λ , so that $\mathcal{L}^{(k+1)} \leq \mathcal{L}^{(k)}$ for all k . It remains to show that $\mathcal{L}^{(\infty)} \leq \mathcal{L}^{(k)}$ for all k . Since $\mathcal{L}^{(\infty)}$ is consistent, we have $\mathcal{L}^{(\infty)} \leq \mathcal{L}^{(0)}$ by the case $i = j$ of the definition of consistency; we proceed by induction. If $\mathcal{L}^{(\infty)} \leq \mathcal{L}^{(k)}$, we have

$$\begin{aligned} L_i^{(k+1)}(\lambda) &= \inf_j [\sup_{\lambda'} [L_j^{(k)}(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')]] \\ &\geq \inf_j [\sup_{\lambda'} [L_j^{(\infty)}(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')]]. \end{aligned}$$

Since $\mathcal{L}^{(\infty)}$ is consistent, the expression in the outer brackets on the right side is $\geq L_i^{(\infty)}(\lambda)$ for all j ; hence the inf of these is also $\geq L_i^{(\infty)}(\lambda)$, so that we have proved $L_i^{(k+1)}(\lambda) \geq L_i^{(\infty)}(\lambda)$, i.e., $\mathcal{L}^{(\infty)} \geq \mathcal{L}^{(k+1)}$.

By definition of Ω , each $L_i^{(k+1)}(\lambda)$ is either one of the $L_j^{(k)}(\lambda')$'s or one of the $\Lambda_{ij}(\lambda, \lambda')$'s; hence, the $L_i^{(k)}(\lambda)$'s can take on only finitely many possible values ($L_i^{(0)}(\lambda)$'s = $\Lambda_{ii}(\lambda)$'s, or $\Lambda_{ij}(\lambda, \lambda')$'s). Thus the sequence $\mathcal{L}^{(0)}, \mathcal{L}^{(1)}, \dots$ must stop after finitely many steps, just as in Section II. When it stops, let us say at $\mathcal{L}^{(m)}$, we have $\Omega\mathcal{L}^{(m)} = \mathcal{L}^{(m)}$, so that $\mathcal{L}^{(m)}$ is consistent (Proposition 8). Since $\mathcal{L}^{(m)} \geq \mathcal{L}^{(\infty)}$, which is the greatest consistent fuzzy labeling, this implies $\mathcal{L}^{(m)} = \mathcal{L}^{(\infty)}$.

Even if the $L_i^{(k)}(\lambda)$'s can take on infinitely many values (as would be the case, e.g., if we used \cdot in place of \wedge in the definition of Ω), we can still show that we converge to $\mathcal{L}^{(\infty)}$ when Ω is iterated, although we do not necessarily reach $\mathcal{L}^{(\infty)}$ in a finite number of steps. Indeed, by Proposition 9 the sequence $\mathcal{L}^{(0)}, \mathcal{L}^{(1)}, \mathcal{L}^{(2)}, \dots$ produced by applying Ω repeatedly is monotonically nonincreasing, and is bounded below by $\mathcal{L}^{(\infty)}$. Hence this sequence must converge to a limit $\mathcal{L} \geq \mathcal{L}^{(\infty)}$. If we can show that \mathcal{L} is consistent, then it must actually be $\mathcal{L}^{(\infty)}$, since $\mathcal{L}^{(\infty)}$ is the greatest consistent fuzzy labeling. Now a fuzzy labeling that does not change under Ω is consistent (Proposition 8). In fact, \mathcal{L} cannot change under Ω , since Ω is a continuous operator, so that we have

$$\begin{aligned} \Omega(\mathcal{L}) &= \Omega(\lim_k \mathcal{L}^{(k)}) = \lim_k (\Omega\mathcal{L}^{(k)}) = \lim_k (\mathcal{L}^{(k+1)}) \\ &= \lim_k (\mathcal{L}^{(k)}) = \mathcal{L}. \end{aligned}$$

Thus \mathcal{L} is consistent, and so must be $\mathcal{L}^{(\infty)}$, which proves the following theorem.

Theorem 10: $\lim_k \mathcal{L}^{(k)} = \mathcal{L}^{(\infty)}$.

Given $\mathcal{L}^{(\infty)}$, we can look for unambiguous fuzzy labelings by a tree search procedure analogous to that discussed at the end of Section II. If we want to find such labelings in which the unique nonzero label weight for each object is as high as possible, we can design the search procedure to try the labels having the highest weights first. For each choice, Ω is applied to obtain a consistent labeling, and the process is repeated. If desired, this procedure can be cut short if all weights for some object drop below a prespecified threshold.

A difficulty with algorithm Ω is that low weights are contagious. For example, suppose that for some object a_j , we have $L_j^{(k)}(\lambda') \leq \frac{1}{2}$ for all λ' . Then for all i , when we apply Ω , we obtain

$$L_i^{(k+1)}(\lambda) \leq [\frac{1}{2} \wedge \sup_{\lambda'} \Lambda_{ij}(\lambda, \lambda')] \leq \frac{1}{2}.$$

In other words, if all weights are low for *some* object, applying Ω causes all weights to become low at *all* objects. This is analogous, in the discrete case, to the fact that if some object has an empty label set, and we apply algorithm Δ , the label sets of all objects become empty. (Note that as long as some weight is high for object a_j , let us say $L_j^{(k)}(\lambda'') = 1$, we can keep object j from having any effect on object i by setting $\Lambda_{ij}(\lambda, \lambda'') = 1$, since this implies

$$\begin{aligned} L_i^{(k+1)}(\lambda) &\leq \sup_{\lambda'} [L_j^{(k)}(\lambda') \wedge \Lambda_{ij}(\lambda, \lambda')] \\ &= L_j^{(k)}(\lambda'') \wedge \Lambda_{ij}(\lambda, \lambda'') = 1 \end{aligned}$$

which is no restriction.) This behavior of algorithm Ω results from the fact that weights can decrease, but never increase, under it. In Sections IV and V we discuss algorithms which allow weights to either increase or decrease; this will permit us to give the weights a probabilistic interpretation.

IV. A LINEAR PROBABILISTIC MODEL

We now consider a model in which we require that the sum of the label weights for each object is 1. We shall call such a weighted labeling a *stochastic labeling*. We can think of the weight assigned to label λ of object a_i as the probability that λ is the correct label of a_i . Thus in a stochastic labeling, to each object a_i there corresponds a probability vector p_i , where for each $\lambda \in \Lambda$, $p_i(\lambda)$ is the weight of label λ at object a_i , where $0 \leq p_i(\lambda) \leq 1$, and $\sum_{\lambda=1}^m p_i(\lambda_j) = 1$. The vector p_i is essentially the same thing as the fuzzy subset L_i of Λ in Section III, but it will be more convenient for us to use vector terminology here.

The *entropy* of the object a_i under a given stochastic labeling is defined to be

$$H_i \equiv - \sum_{j=1}^m p_i(\lambda_j) \log p_i(\lambda_j).$$

It is well known that H_i is zero if and only if p_i is a unit vector of the form $(0, \dots, 0, 1, 0, \dots, 0)$ and that H_i takes on its maximum value when the $p_i(\lambda_j)$ are all equal (to $1/m$). Thus H_i is a measure of our uncertainty as to the nature of object a_i . A desirable feature of a weight modification algorithm is that it should tend to decrease the entropies of the objects, i.e., to make us increasingly certain about

blockwise; the k th power of the block corresponding to eigenvalue ρ_i is

$$\begin{pmatrix} \rho_i^k & \binom{k}{k-1} \rho_i^{k-1} & \binom{k}{k-2} \rho_i^{k-2} & \cdots \\ 0 & \rho_i^k & \binom{k}{k-1} \rho_i^{k-1} & \cdots \\ 0 & 0 & \rho_i^k & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Since $|\rho_i| < 1$, the terms of this matrix all go to zero as $k \rightarrow \infty$. Thus J^k converges to a matrix J^∞ consisting entirely of 0's, except for a single 1 in its upper left corner. This J^∞ simply selects the eigenvector that has eigenvalue 1 out of the basis. Hence applying F to $J^\infty(\mathcal{L}^{(0)})$ leaves it unchanged, i.e., $J^\infty(\mathcal{L}^{(0)})$ is the fixed point $\mathcal{L}^{(\infty)}$ of F .

Next, suppose that the eigenvalue 1 is not simple. The argument just given still shows that J^k converges, but the limit matrix J^∞ now has more than a single 1 on the main diagonal. In this case we still obtain a fixed point of F (a linear combination of eigenvectors that each have eigenvalue 1), but it is no longer the unique fixed point. Conditions under which 1 is a simple eigenvalue of F will be discussed in a moment.

If there are other complex eigenvalues that have absolute value 1, we can eliminate them by the trick of working with the operator $\frac{1}{2}(F + I)$, where I is the identity, rather than with F itself. Readily, $\frac{1}{2}(F + I)$ has the same eigenvectors with eigenvalue 1 that F has; but all other eigenvalues are shifted to the midpoints between 1 and their original positions in the complex plane. Thus $\frac{1}{2}(F + I)$ cannot have complex eigenvalues with absolute value 1, except for 1 itself.

In order to formulate conditions for the simplicity of the eigenvalue 1, let us now suppose that the p_{ij} 's and c_{ij} 's are all nonnegative (this assumption has not been necessary up to this point). Then by the Perron-Frobenius theorem [12] on nonnegative matrices, 1 is a simple eigenvalue provided that the matrix of F is irreducible. A matrix M is called reducible if there exists a permutation matrix P such that PMP^T has the block form

$$\begin{pmatrix} A & B \\ O & C \end{pmatrix}.$$

Let us further assume that $c_{ij}p_{ij}(\lambda | \lambda') \neq 0$ if and only if $c_{ji}p_{ji}(\lambda' | \lambda) \neq 0$; this is far weaker than the symmetry condition mentioned in Sections II and III. (Note that, in fact, $p_{ij}(\lambda | \lambda')p_j(\lambda') = p_{ji}(\lambda' | \lambda)p_i(\lambda)$ is the joint probability that objects a_i and a_j have labels λ and λ' , respectively.) Thus if F has a reducible matrix, it must be of the form

$$\begin{pmatrix} A & O \\ O & C \end{pmatrix}.$$

If the blocks A and C are themselves reducible, we can repeat the process, to eventually obtain a matrix R for F that is composed of irreducible blocks along the main

diagonal, e.g.,

$$R = \begin{pmatrix} M_1 & & O \\ & \ddots & \\ O & & M_q \end{pmatrix}.$$

Thus we see that if F is reducible, we can partition the mn -dimensional space of (object, label) pairs into subsets on each of which F operates independently of the others.

We can interpret the reducibility of F in terms of the labeling network discussed in Sections II and III. Here the nodes, as usual, are the (object, label) pairs, and we join the two nodes (i, λ) , (j, λ') by an arc if and only if $c_{ij}p_{ij}(\lambda | \lambda') \neq 0$ (which, we recall, is equivalent to $c_{ji}p_{ji}(\lambda' | \lambda) \neq 0$). Thus two nodes are joined if and only if the corresponding (object, label) pairs influence one another under the operation F . It is evident that if F has the block diagonal form shown above, the subgraphs corresponding to the blocks are connected components of the network.

In summary, if F is irreducible, iteration of F always yields a unique limiting stochastic labeling derived from the eigenvector of F that has eigenvalue 1. If F is reducible, we can decompose the labeling problem into subproblems, corresponding to the connected components of the labeling network, and we obtain convergence to a unique limit on each of these subspaces. Any linear combination of these limits is a fixed point of F .

To illustrate these remarks, let us assign conditional probabilities to pairs of labels in the triangle example of Fig. 1. We will restrict our attention to the connected component of the network shown in Fig. 2(a), which involves the labels \leftarrow and \rightarrow only, in order to insure that the matrix of F is irreducible. A similar analysis can be applied to the other connected component.

The conditional probability of event A given event B is obtained from the formula

$$p(A | B) = \frac{p(A \& B)}{p(B)},$$

i.e., it is the joint probability divided by the probability of B . One possible source of conditionals for labelings of line drawings is statistics based on a collection of possible interpretations of many actual line drawings. In Fig. 3, we have collected these statistics for labels on lines meeting at L junctions by analyzing the eight possible interpretations of the triangle. For ordinary applications, this sample set is probably too small to give realistic conditionals. However, since we will only apply the model to the triangle example, within our limited labeling "world" the resulting conditionals are realistic.

If we set the weighting constants to $c_{ij} = \frac{1}{3}$ for all i, j , then the resulting matrix corresponding to the operator F for the \leftarrow, \rightarrow component of the network is

$$\begin{pmatrix} 1 & 0 & \frac{2}{3} & 1 & \frac{2}{3} & 1 \\ 0 & 1 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ \frac{2}{3} & 1 & 1 & 0 & \frac{2}{3} & 1 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 1 & \frac{1}{3} & 0 \\ \frac{2}{3} & 1 & \frac{2}{3} & 1 & 1 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 1 \end{pmatrix}.$$

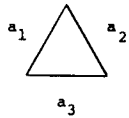


Figure 1 case	Labels		
	a ₁	a ₂	a ₃
b	+	+	+
c	+	+	+
d	+	-	+
e	+	+	-
f	-	+	+
g	+	+	+
h	+	+	+
i	+	+	+

Based on this set of cases, the probabilities of the labels + and - will be taken to be 3/8 each, and those of the labels + and - to be 1/8 each.

Events		Probabilities	
A	B	p(A&B)*	p(A B)
+	+	1/4	2/3
+	+	0	0
+	-	1/8	1
+	+	0	0
-	+	0	0
+	+	1/4	2/3
+	-	0	0
+	+	1/8	1
-	+	1/8	1/3
-	+	0	0
-	-	0	0
-	+	0	0
+	+	0	0
+	+	1/8	1/3
+	-	0	0
+	+	0	0

Fig. 3. Conditional probabilities for example in Fig. 1. *Probability that A and B occur consecutively, in clockwise order.

It is easy to verify that this matrix has a single stochastic eigenvector corresponding to the eigenvalue 1, given by

$$e = (\frac{3}{8}, \frac{1}{8}, \frac{3}{8}, \frac{1}{8}, \frac{3}{8}, \frac{1}{8})^T$$

Thus the linear model yields a probability assignment of ($\frac{3}{8}, \frac{1}{8}$) for the labels \leftarrow and \rightarrow , respectively, for each of the three lines. These numbers reflect a bias toward interpretation (c) of Fig. 1, the triangular hole, as opposed to interpretations (g)-(i), the flaps.

Iteration of the relaxation process will converge to this eigenvector regardless of the initial distribution. Even if we begin with an initial stochastic labeling corresponding to an unambiguous consistent labeling (for example, ((1,0), (0,1), (1,0))), the process will diffuse the probabilities through the connections in the network at rates corresponding to the conditionals. The resulting eigenvector is an unbiased distribution, dependent only on the compatibilities of the label interpretations. A label will have a high probability if one or more fairly probable interpretations include that label at the given object.

V. NONLINEAR PROBABILISTIC MODELS

The linear model of Section IV always converges to the same limit, irrespective of any initial bias on the label probabilities, if the matrix of conditionals is irreducible. In this section we briefly consider a nonlinear model which exhibits a more useful type of behavior, i.e., it converges to a limit that does depend on the initial probabilities.

Before defining such a nonlinear model, let us consider the desirable characteristics of a stochastic relaxation

operator F . The probability $p_i(\lambda)$ of a given label for a given object a_i should be increased by F if other objects' labels that have high probabilities are highly compatible with λ at a_i . Conversely, $p_i(\lambda)$ should be decreased if other high-probability labels are incompatible with λ at a_i . On the other hand, labels having low probabilities should have little influence on $p_i(\lambda)$, whether or not they are compatible with it. These characteristics can be summarized in tabular form as follows:

		Compatibility of λ' with λ	
		High	Low
Probability of λ'	High	+	-
	Low	0	0

where + means that $p_i(\lambda)$ should increase, - means that it should decrease, and 0 that it should remain relatively unchanged.

The linear model presented in Section IV, however, behaves essentially as follows.

		Compatibility of λ' with λ	
		High	Low
Probability of λ'	High	+	0
	Low	0	-

Here $p_i(\lambda)$ decreases the most when the $p_j(\lambda')$'s have both low probability and low compatibility with $p_i(\lambda)$.

We could come closer to the desired behavior by redefining the linear model to make use of compatibility coefficients that can take on both positive and negative values, let us say in the range $[-1, 1]$. Such coefficients could be regarded as representing correlations (or covariances), rather than conditional probabilities. Let us denote the compatibility of label λ on a_i with label λ' on a_j by $r_{ij}(\lambda, \lambda')$, rather than by $p_{ij}(\lambda | \lambda')$, to avoid confusion with the compatibilities of Section III. We would like the r 's to behave as follows. If λ' on a_j frequently co-occurs with λ on a_i , then $r_{ij}(\lambda, \lambda')$ is positive, if they rarely co-occur, $r_{ij}(\lambda, \lambda')$ is negative, and if their occurrences are independent, $r_{ij}(\lambda, \lambda') \doteq 0$.

The covariance of two events A and B is defined as

$$p(A \& B) - p(A)p(B) \equiv \text{cov}(A, B)$$

and the correlation of A and B is defined as

$$\begin{aligned} \text{cor}(A, B) &\equiv \frac{\text{cov}(A, B)}{[(p(A) - p^2(A))(p(B) - p^2(B))]^{1/2}} \\ &\equiv \frac{\text{cov}(A, B)}{\sigma(A)\sigma(B)} \end{aligned}$$

For example, with reference to Fig. 3, the standard deviations of the events \rightarrow and \leftarrow are $(3/8 - 9/64)^{1/2} = \sqrt{15/64}$ each, while for the events $+$ and $-$ they are $(1/8 -$

Events		cov	cor
A	B		
+	+	7/64	7/15
+	+	-9/64	-3/5
+	-	5/64	5/√105
+	+	-3/64	-3/√105
+	+	-9/64	-3/5
+	+	7/64	7/15
+	-	-3/64	-3/√105
+	+	5/64	5/√105
-	+	5/64	5/√105
-	+	-3/64	-3/√105
-	-	-1/64	-1/7
-	+	-1/64	-1/7
+	+	-3/64	-3/√105
+	+	5/64	5/√105
+	-	-1/64	-1/7
+	+	-1/64	-1/7

Fig. 4. Covariances and correlations for example in Fig. 1.

$1/64)^{1/2} = \sqrt{7/64}$ each. The covariances and correlations of pairs of these events are tabulated in Fig. 4.

If we let $r_{ij}(\lambda, \lambda')$ be the correlation between the events that a_i has label λ and a_j has label λ' , then readily we always have $-1 \leq r_{ij}(\lambda, \lambda') \leq 1$. If these events (call them A and B) always co-occur, we have $p(A \& B) = p(A) = p(B)$, so that $\text{cov}(A, B) = p(A) - p^2(A)$, which implies $r_{ij} = 1$. If they never co-occur, we have $p(A \& B) = 0$, so that $r_{ij} = -p(A) \cdot p(B) / \sigma(A)\sigma(B) < 0$. (It is not hard to show that this expression is a minimum when $p(A) = p(B) = \frac{1}{2}$, which yields $r_{ij} = -1$.) If A and B are independent, $p(A \& B) = p(A)p(B)$, so that $\text{cov}(A, B) = 0$, implying $r_{ij} = 0$. Thus taking r_{ij} to be the correlation gives it the desirable properties described above.

In order to define our new F operator, let us first set

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

where the d 's are coefficients, analogous to the c 's in Section IV. By the remarks in the preceding paragraph, this $q_i^{(k)}(\lambda)$ behaves just the way we would want the *change* in $p_i^{(k)}(\lambda)$ to behave, as indicated at the beginning of this section. Indeed, if $p_j^{(k)}(\lambda')$ is high, and $r_{ij}(\lambda, \lambda')$ is very positive or very negative, then the label λ' at a_j makes a substantial positive or negative contribution to $q_i^{(k)}(\lambda)$; while if $p_j^{(k)}(\lambda')$ is low, λ' at a_j makes relatively little contribution to $q_i^{(k)}(\lambda)$, irrespective of the value of $r_{ij}(\lambda, \lambda')$.¹

¹ Note that for $i = j$ we have $r_{ii}(\lambda, \lambda) = 1$ for $\lambda = \lambda'$, and $= -p_i(\lambda)p_i(\lambda') / \sigma_i(\lambda)\sigma_i(\lambda')$ for $\lambda \neq \lambda'$, so that r_{ii} takes on only extreme values. Thus if $p_i(\lambda)$ is large, r_{ii} makes a large positive contribution to $q_i(\lambda)$ and a negative contribution to the q_i 's of other labels. In other words, as might be expected, r_{ii} is biased in favor of the currently preferred label of a_i . If desired, this bias can be counteracted by making the weights d_{ii} small.

Figure 1 case	Matrix	Figure 1 case	Matrix
b	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$	c	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
d	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$	g	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
e	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	h	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
f	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$	i	$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

Key to Matrices

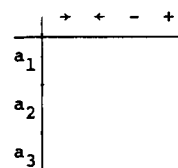


Fig. 5. Weight matrices for eight unambiguous interpretations in Fig. 1.

These observations suggest that we might define a new linear F operator by letting

$$p_i^{(k+1)}(\lambda) = p_i^{(k)}(\lambda) + q_i^{(k)}(\lambda).$$

However, this definition would not guarantee that the p 's remain nonnegative. Instead, we shall define a nonlinear operator by setting

$$p_i^{(k+1)}(\lambda) = p_i^{(k)}(\lambda) [1 + q_i^{(k)}(\lambda)] / \sum_{\lambda'} p_i^{(k)}(\lambda) [1 + q_i^{(k)}(\lambda)].$$

Here the denominator serves to guarantee that the p 's continue to sum to 1. Moreover, they remain nonnegative, since readily q_i is in the range $[-1, 1]$ (provided $\sum_j d_{ij} = 1$), so that $1 + q_i$ is nonnegative. The discussion in the preceding paragraph still applies; a very positive or very negative contribution to q_i contributes an increase or decrease to p_i (since $p_i^{(k+1)}$ is obtained by multiplying $p_i^{(k)}$ by $(1 + q_i^{(k)})$), whereas a small contribution to q_i contributes little change to p_i . Clearly, many other F operators with these properties could be defined, but the one given here is especially simple.

We have not yet been able to establish the convergence properties of this nonlinear operator, but we have found that in specific examples, it does exhibit the desired type of behavior. To illustrate this, we once again consider the triangle example. The weights assigned to the four labels $\rightarrow, \leftarrow, -, +$ on the three sides a_1, a_2, a_3 (see Fig. 3) constitute a 3-by-4 matrix; these matrices, for the eight unambiguous interpretations of Fig. 1(b)–(i), are shown in Fig. 5. We shall assume that the node weights d_{ij} are $d_{ii} = 0.2$, $d_{ij} = 0.4$ for $i \neq j$, and that the r_{ij} 's are the correlation values given in Fig. 4. The behavior of the

Case	Initial weights	After 2 to 3 iterations	After 20 to 30 iterations	Limit
A	.25 .25 .25 .25	.3 .3 .2 .2	.33 .33 .17 .17	.37 .37 .13 .13
	.25 .25 .25 .25	.3 .3 .2 .2	.33 .33 .17 .17	.37 .37 .13 .13
	.25 .25 .25 .25	.3 .3 .2 .2	.33 .33 .17 .17	.37 .37 .13 .13
B	.5 0 .5 0	.8 0 .2 0	.98 0 .02 0	1 0 0 0
	.5 0 .5 0	.8 0 .2 0	.98 0 .02 0	1 0 0 0
	.5 0 .5 0	.8 0 .2 0	.98 0 .02 0	1 0 0 0
C	.5 0 .5 0	.62 0 .37 0	1 0 0 0	1 0 0 0
	.4 0 .6 0	.49 0 .51 0	.97 0 .03 0	1 0 0 0
	.5 0 .5 0	.62 0 .37 0	1 0 0 0	1 0 0 0
D	.5 0 .5 0	.64 0 .36 0	1 0 0 0	1 0 0 0
	.3 0 .7 0	.36 0 .64 0	.07 0 .93 0	0 0 1 0
	.5 0 .5 0	.64 0 .36 0	1 0 0 0	1 0 0 0
E	.3 0 .7 0	.5 0 .5 0	.95 0 .05 0	1 0 0 0
	.3 0 .7 0	.5 0 .5 0	.95 0 .05 0	1 0 0 0
	.5 0 .5 0	.84 0 .16 0	1 0 0 0	1 0 0 0
F	.2 0 .8 0	.3 0 .7 0	.06 0 .94 0	0 0 1 0
	.3 0 .7 0	.51 0 .49 0	1 0 0 0	1 0 0 0
	.5 0 .5 0	.83 0 .17 0	1 0 0 0	1 0 0 0
G	.3 .2 .3 .2	.41 .13 .32 .14	.98 0 .02 0	1 0 0 0
	.3 .2 .3 .2	.41 .13 .32 .14	.98 0 .02 0	1 0 0 0
	.3 .2 .3 .2	.41 .13 .32 .14	.98 0 .02 0	1 0 0 0
H	.3 .2 .3 .2	.38 .17 .29 .16	1 0 0 0	1 0 0 0
	.25 .25 .25 .25	.35 .20 .25 .20	1 0 0 0	1 0 0 0
	.2 .2 .4 .2	.23 .16 .45 .16	.2 0 .8 0	0 0 1 0

Fig. 6. Examples of nonlinear relaxation process in triangle case.

label weights as the nonlinear relaxation process is iterated is illustrated in Fig. 6 for various initial weight assignments. The following comments can be made on the cases shown in this figure.

A) Initially all weights are equal; converges essentially to the *a priori* label probabilities ($\frac{3}{8}, \frac{3}{8}, \frac{1}{8}, \frac{1}{8}$).

B) Initially equal weights are in one component ($\rightarrow, -$), zero weights in the other; converges to the most probable interpretation in that component (case (b) of Fig. 1).

C) Initial weights are in the same component, slightly biased toward one of the less probable interpretations (case (c) of Fig. 1); still converges to the most probable interpretation.

D) Same as C), but now sufficiently biased; converges to the desired interpretation. The transition point defining "sufficient bias" depends on the d_{ij} 's.

E) Still in same component, but two contradictory biases, toward two of the improbable interpretations, cases (d) and (f) of Fig. 1; converges to the probable interpretation.

F) Similar to E), but two contradictory biases of unequal weight; converges as dictated by the stronger one.

G) Biased toward one of the two components; converges to the most probable interpretation in that component.

H) Similar to G), but biased toward a less probable interpretation in one component; converges to that interpretation.

VI. CONCLUDING REMARKS

As the examples given at the end of Section V indicate, the nonlinear probabilistic model exhibits very reasonable behavior in terms of converging to one of the possible interpretations of the scene, in accordance with the initial

biases of the probabilities. In practice, one would not be concerned with actual convergence, but only with the behavior of the weights after a few (or a few dozen) iterations. In our examples, the trend of convergence is usually apparent fairly early. One can think of the relaxation process as "enhancing" the initial probabilities, as influenced by the correlations r_{ij} .

It would seem from these examples that the model of Section V could serve as a useful preprocessor for imposing constraints on label probabilities. After allowing such a process to operate for a while, one could input the resulting enhanced probabilities to a semantics-based analyzer which would check the results and disambiguate them further if necessary. This analyzer would, of course, be more powerful than the relaxation model, since it would not be restricted to applying constraints that can be formulated in a homogeneous fashion, and it would probably operate sequentially. The relaxation process serves to (hopefully) reduce the number of possibilities that this sequential process has to consider. Of course, it is possible that the relaxation scheme could lead to unsatisfactory or unacceptable results. In such a case, the sequential process could go back to the original input data, or it could reinitiate the relaxation process with a different set of initial probabilities.

An approach of this type has many potential applications in scene analysis, as well as other areas. The following are just a few examples.

1) As an extension of our toy triangle example, one could attempt to implement a more complete probabilistic version of Waltz's labeling model. In this connection, the Appendix to this paper presents discrete, fuzzy, and probabilistic models for line drawings in which the objects are *L*-junc-

tions (rather than lines), and these can have six possible labelings, as shown in Section II.

2) More realistic situations might involve models for specific classes of real scenes. For example, suppose that we are analyzing pictures of human faces and have detected objects a_1, \dots, a_n , each of which can have labels $\lambda_1, \dots, \lambda_m$ representing "eye," "nose," "scar," "wrinkle," etc. Our facial-feature recognizer can assign a probability $p_i(\lambda_j)$ to each label λ_j at each object a_i , $1 \leq i \leq n$, $1 \leq j \leq m$. Evidently, the relative positions of the a_i 's impose constraints on these probabilities; for example, if a_r is a mouth, and a_s is below a_r , then a_s cannot be a nose. It should be possible to greatly reduce the initial ambiguity of the $p_i(\lambda_j)$'s by applying these constraints in the form of a relaxation process. Similar, though probably weaker, constraints could in principle be formulated for various classes of indoor and outdoor scenes; see [10].

3) The relaxation approach can also be used as an aid to low-level picture or scene segmentation. Suppose that we have detected and ambiguously identified a set of local features in a picture. We can then allow them to interact in such a way that "clusters" of features having a common label reinforce one another, while other labels become weaker. Similar processes have been studied in conjunction with neural network modeling (e.g., [2]). To give a specific example, suppose that at points a_1, \dots, a_n , a line or curve detection operator has an above-threshold value, corresponding to one of the possible slopes s_1, \dots, s_m . Let s_0 represent the possibility that there is no line or curve through a given point. Based on the values of the various directional line detection operators at the a_i 's, we can assign probabilities $p_i(s_j)$ to the labels s_j at the points a_i , $1 \leq i \leq n$, $0 \leq j \leq m$. We can then allow the relative positions of the a_i 's to impose constraints on these probabilities; e.g., if a_u and a_v are close together in relative direction s_k , we might want to strengthen $p_u(s_k)$ and $p_v(s_k)$. A relaxation process based on such constraints should tend to reinforce a_i 's that lie on smooth curves, while weakening "noisy" a_i 's.

4) Examples can also be found in areas other than picture and scene analysis. Suppose that a character-recognition device has scanned a string of characters a_1, \dots, a_n and identified them probabilistically as being various letters of the alphabet $\lambda_1, \dots, \lambda_m$; let $p_i(\lambda_j)$ be the probability that character a_i is letter λ_j . We could then impose constraints on these probabilities based on the observed frequencies with which characters co-occur (e.g., labels T and H on a pair of consecutive characters a_i, a_{i+1} might reinforce one another, while the labels T and G would have a weakening effect on each other). Of course, character recognition post-processors more powerful than this have been developed, but it would be of interest to see how much ambiguity reduction can be obtained from a simple relaxation process, based on co-occurrence constraints, as compared with more sophisticated methods.

5) As a final example, we mention a possible application in the area of medical diagnosis (or, more generally, any type of diagnostic testing). Let a_1, \dots, a_n be measurable phenomena (temperature, pulse rate, blood count, etc.) and

let $\lambda_1, \dots, \lambda_m$ be ranges of values for these measurements. *A priori*, we can specify a discrete probability density for the value of each measurement (i.e., $p_i(\lambda_j)$ is the probability that phenomenon a_i has value in range λ_j). At the same time, we can, in principle, specify correlations (or joint probabilities) for pairs of the measurements. If we actually carry out one of the measurements, its value becomes known, and we can use the correlations, implemented in the form of a relaxation process, to update the probability densities of the remaining measurements. This process can be repeated. The choice of which measurements to perform at a given stage can be guided by their degrees of ambiguity (or, more realistically, by the increases in some utility function that will result if these measurements are performed). An approach to medical diagnosis based on probabilistic networks is currently being investigated by Lemmer [6].

One can also speculate on the possible relevance of models such as those discussed in this paper to the perception of scenes by humans. There are many types of ambiguous scenes which most observers see as unambiguous, unless the ambiguity is pointed out to them. This may be due to an innate bias against certain interpretations, e.g., a preference for occluding edges over concave dihedrals or for solid objects over holes, in our triangle example. As we have seen, relaxation starting from a biased initial probability vector often results in convergence to an unambiguous interpretation. Of course, a sequential model could also account for this phenomenon, since a bias against certain interpretations would make it unlikely that these would be selected for testing as possible members of unambiguous labelings.

Similar remarks apply to situations in which two or more labelings are equally perceivable. Here the observer "sees" only one of these at a time, but he can usually switch from one to another at will (often by forcing himself to re-interpret one part of the scene), sometimes switching spontaneously. Voluntary switching could correspond to re-initiation of the relaxation process with a new bias on part of the scene (or to sequential testing of a new combination). Spontaneous switching could be a reinitiation as a result of "fatiguing" of the current bias.

At the other extreme, "impossible" scenes have no consistent labeling but are often locally well formed. Presumably, a relaxation process applied to such a scene would converge to an interpretation that always remained partially ambiguous.

Whatever the value of such speculations may be, the relaxation processes described in this paper do seem to have useful ambiguity-reduction properties. The authors plan to explore nontrivial applications of these processes, and they hope that other investigators will do likewise.

APPENDIX MODELS FOR L-JUNCTIONS

In the line drawing description schemes used by Waltz [13] and others, the junctions, rather than the individual lines, are the "objects." Thus the number of possible labels

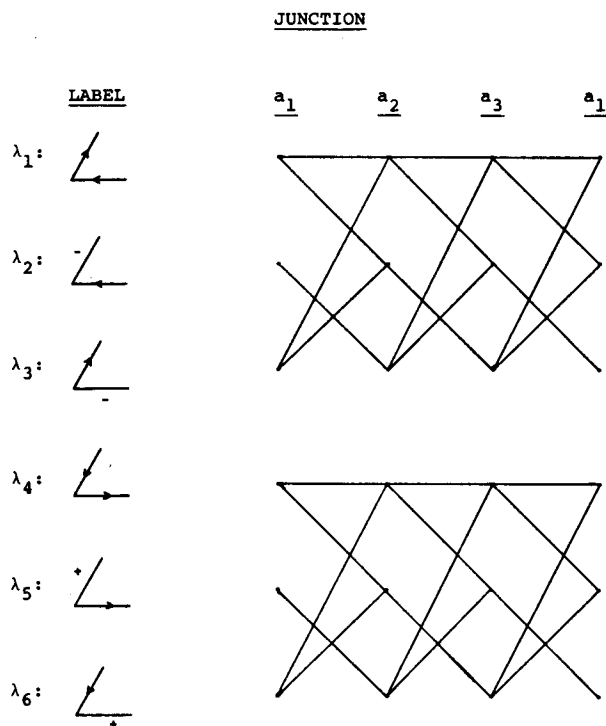


Fig. 7. Labeling network for triangle using junctions as "objects."

can be quite large, particularly if complex junctions can occur. The compatibility relations, on the other hand, tend to become simpler; for example, we can simply call two junction labels compatible if they give the same interpretation to the line (if any) that joins them.²

In this Appendix, we rework the triangle examples using the three *L*-junctions, rather than the three lines, as objects. There are six allowable *L*-junction types, as shown in Section II. These six types are the same as those listed by Waltz (if cracks and shadows are not allowed). If we define compatibility as in the preceding paragraph, the labeling network for the triangle is as shown in Fig. 7. (Object a_1 has been repeated to make the network easier to draw.) As before, there are eight unambiguous consistent labelings of the triangle; they correspond to the eight cases shown in Fig. 1.

We can assign weights to the junction compatibilities and formulate a fuzzy model. A plausible set of weights for the compatibilities in the first connected component of the network in Fig. 7 might be

Pair	Compatibility	Illustration
λ_1, λ_1	0.9	
$\left. \begin{matrix} \lambda_1, \lambda_2 \\ \lambda_2, \lambda_3 \\ \lambda_3, \lambda_1 \end{matrix} \right\}$	0.3	
λ_3, λ_2	0.1	

² Waltz's scheme is much more sophisticated than this; he takes into account the regions, as well as the lines, that are shared by the junction.

<u>A</u>	<u>B</u>	<u>p(A&B)</u>	<u>P_{ij} = p(A&B)/p(B)</u>	<u>r_{ij} = cor(A,B)</u>
λ_1	λ_1	1/8	1/2	1/3
λ_1	λ_2	1/8	1	$3/\sqrt{21}$
λ_1	λ_3	0	0	$-1/\sqrt{21}$
λ_2	λ_1	0	0	$-1/\sqrt{21}$
λ_2	λ_2	0	0	-1/7
λ_2	λ_3	1/8	1	1
λ_3	λ_1	1/8	1/2	$3/\sqrt{21}$
λ_3	λ_2	0	0	-1/7
λ_3	λ_3	0	0	-1/7

We assume $p(\lambda_1) = 1/4$; $p(\lambda_2) = p(\lambda_3) = 1/8$

Fig. 8. Statistics for triangle case.

with all other pairs having compatibility zero. It is not hard to show that the greatest consistent fuzzy labeling for this assignment is given by

Label	Junction		
	a_1	a_2	a_3
λ_1	0.9	0.9	0.9
λ_2	0.3	0.3	0.3
λ_3	0.3	0.3	0.3

and similarly for the other three labels. Eight unambiguous consistent fuzzy labelings exist, corresponding to the eight cases shown in Fig. 1, with all label weights equal to 0.9 (in cases (b) and (c)) or 0.3 (in the other six cases).

In order to apply the probabilistic models, we need conditional probabilities (p_{ij} 's) and correlations (r_{ij} 's). We can derive these from the eight interpretations of the triangle; they are tabulated in Fig. 8, for the labels in the first connected component of the network.

For the linear model, using $c_{ij} = \frac{1}{3}$ for all i and j , the 9×9 operator matrix (for the labels $\lambda_1, \lambda_2, \lambda_3$) is

$$\frac{1}{3} \begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} & 1 & 0 & \frac{1}{2} & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{2} & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{2} & 0 & 1 & 1 & 0 & 0 & \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & 0 & \frac{1}{2} & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{2} & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This is an irreducible matrix, and its unique stochastic eigenvector with eigenvalue 1 is

$$\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)^T$$

Case	Initial weights	After 2 to 3 iterations	After 20 to 30 iterations	Limit
A	.33 .33 .33	.4 .3 .3	1 0 0	same
	.33 .33 .33	.4 .3 .3	1 0 0	
	.33 .33 .33	.4 .3 .3	1 0 0	
B	.4 .3 .3	.5 .25 .25	1 0 0	same
	.33 .33 .33	.4 .33 .27	0 1 0	
	.33 .33 .33	.4 .27 .33	0 0 1	
C	.5 .25 .25	.6 .2 .2	1 0 0	1 0 0
	.4 .3 .3	.46 .3 .24	.01 .99 0	0 1 0
	.4 .3 .3	.46 .3 .24	.01 0 .99	0 0 1
D	.6 .2 .2	.7 .15 .15	1 0 0	1 0 0
	.5 .25 .25	.6 .22 .18	.99 .01 0	1 0 0
	.5 .25 .25	.6 .18 .22	.99 0 .01	1 0 0
E	.33 .33 .33	.37 .36 .27	0 1 0	same
	.3 .3 .4	.35 .25 .4	0 0 1	
	.33 .33 .33	.46 .27 .27	1 0 0	
F	.3 .3 .4	.4 .22 .38	.73 0 .27	.73 0 .27
	.3 .4 .3	.4 .38 .22	.73 .27 0	.73 .27 0
	.33 .33 .33	.34 .33 .33	.12 .44 .44	0 .5 .5
G	.3 .3 .4	.4 .24 .36	1 0 0	same
	.25 .5 .25	.28 .52 .20	0 1 0	
	.33 .33 .33	.31 .31 .38	0 0 1	

Fig. 9. Examples of behavior of nonlinear model.

corresponding to an assignment of probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$ to labels $\lambda_1, \lambda_2, \lambda_3$, respectively, at each junction. The linear model will converge to this assignment, whatever the initial probabilities.

The nonlinear model is once again much better behaved. Using $d_{ii} = 0.2$ for all i , and $d_{ij} = 0.4$ for all $i \neq j$, its behavior for the labels $\lambda_1, \lambda_2, \lambda_3$ is illustrated in Fig. 9. Comments on the seven examples shown here are as follows.

A) No initial bias; converges to the most probable interpretation (case (b) of Fig. 1), as in example (B) of Section V.

B) One junction favored as "floating," no bias on the other two; converges to an appropriate case of Fig. 1, where just the favored junction floats.

C) All junctions favored as floating, with one preferred; same result as in example B): only the preferred junction floats.

D) Same as example C), but stronger biases toward floating; result has all junctions floating (even though one was preferred).

E) One junction favored as involving a concave edge, no bias on the other two; converges to an appropriate case of Fig. 1, with the favored edge concave.

F) Two junctions equally favored as involving concave edges (which is impossible); converges to an ambiguous result.

G) Same as example F), but with unequal favoring; converges to the case in which only the more favored edge is concave.

ACKNOWLEDGMENT

The help of Mrs. Shelly Rowe in preparing this paper is gratefully acknowledged. The discrete model of Section II

is an expansion of Section 3 of [8]; it is an outgrowth of discussions with H. R. Barrow in the summer of 1974.

REFERENCES

- [1] M. B. Clowes, "On seeing things," *Artificial Intelligence*, vol. 2, pp. 79-116, 1971.
- [2] P. Dev, "Segmentation processes in visual perception: A co-operative neural model," COINS Technical Report 74C-5, University of Massachusetts at Amherst, June 1974.
- [3] R. E. Fikes, "REF-ARF: A system for solving problems stated as procedures," *Artificial Intelligence*, vol. 1, pp. 27-120, 1970.
- [4] J. Gaschnig, "A constraint satisfaction method for inference making," *Proc. 12th Allerton Conf. on Circuit and System Theory*, Urbana, Ill., Oct. 1974.
- [5] D. A. Huffman, "Impossible objects as nonsense sentences," in B. Meltzer and D. Michie, eds., *Machine Intelligence*, vol. 6, Edinburgh U. Press, 1971, pp. 295-323.
- [6] J. F. Lemmer, unpublished Ph.D. dissertation proposal, Dept. of Computer Science, University of Maryland, 1974 (advisor: Prof. L. N. Kanal).
- [7] A. Rosenfeld, "Fuzzy graphs," in L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura, eds., *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, Academic Press, New York, 1975, pp. 77-95.
- [8] A. Rosenfeld, "Networks of automata: Some applications," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-5, pp. 380-383, 1975.
- [9] D. R. Smart, *Fixed Point Theorems*. New York: Cambridge University Press, 1974.
- [10] J. M. Tenenbaum, T. D. Garvey, S. Weyl, and H. C. Wolf, "An interactive facility for scene analysis research," Technical Note 87, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, CA, Jan. 1974.
- [11] K. J. Turner, "Computer perception of curved objects using a television camera," Ph.D. dissertation, Dept. of Machine Intelligence, University of Edinburgh, 1974.
- [12] Richard S. Varga, *Matrix Iterative Analysis*. Englewood Cliffs, N.J.: Prentice-Hall, 1962.
- [13] D. L. Waltz, "Generating semantic descriptions from drawings of scenes with shadows," MIT Technical Report AI271, November 1972. For briefer versions see D. Waltz, "Understanding line drawings of scenes with shadows," in P. H. Winston, ed., *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975, pp. 19-91; and D. Waltz, "Automata theoretic approach to visual processing," in R. T. Yeh, ed., *Applied Computation Theory*, Englewood Cliffs, N.J., Prentice-Hall, 1975.