SCENE ANALYSIS, ARRANGEMENTS, AND HOMOMORPHISMS

Robert M. Haralick

Electrical Engineering and Computer Science Departments
University of Kansas
Lawrence, Kansas

Hanson and Riseman (eds.)
Academic Press, Inc., New York

ISBN 0-12-323550-2

ABSTRACT

A number of scene analysis tasks can be understood and solved from a point of view which we call the theory of arrangements. An arrangement is a set of one or more labeled N-ary relations. The theory of arrangements suggests that the solution to some seemingly different high and low level scene analysis tasks can be found by the construction of a homomorphism from one arrangement to another, if such a homomorphism exists. In this paper we discuss the arrangement concept and its application to scene analysis. Then we illustrate how a general discrete constraint relaxation method can be used to construct homomorphisms from one arrangement to another.

I. Introduction

Scene analysis and image understanding tasks encompass everything from low level preprocessing and image enhancement operations through boundary delineation, feature extraction for color, texture and shape, to labeling objects in an image and interpreting their relationships. It is often the case that each kind of scene analysis task is explored independently from the tasks which preceed it and the tasks which follow it. In part, this is due to the enormous complexity of the problem and the lack of any unified conceptual view by which the whole problem can be understood.

Since the technical language of analysis is the language of mathematics, the conceptual problem is really a mathematical one. If there were a consistent way of expressing the high-level and low-level operations and their compositions, we could begin to ask what we are really doing in scene analysis, what we are trying to optimize, and why it is that a particular sequence of compositions of image operations produces a best result. Without a conceptual view and a corresponding notational system, scene analysis and image understanding will be forever a piecemeal problem.

It is the purpose of this paper to explore in a unified approach (1) the way in which the different levels of scene analysis depend on one another and (2) the way in which world model information can be integrated into the processing on any given level. We will show that a data structure, which we call an arrangement, is often a suitable data structure for representing information about a scene at any given level of processing, as well as representing world model information. Furthermore, we suggest that the way in which world model information can be integrated into any level of processing requires the establishment of a homomorphism between the world model information and the image information.

II. Scene Analysis

Scene analysis consists of a sequence of information extraction tasks. The initial tasks work on the raw image which can be considered as a low-level, noisy data source. The later tasks work on successively higher level data sources, finally producing a concise description of the scene. At each level of processing, information from a world model may be available to help guide the interpretations being produced.

The basic units being processed at each level are different. In the early stages the units might be groups of pixels with their associated gray tones or colors. Early processing can consist of textural feature extraction, edge detection, or small homogeneous region delineation. The basic characteristic of the early processing is its almost exclusive reliance on local properties to perform detection or produce descriptions. This is illustrated in Figure 1 which shows a local property extractor moving across the image examining a row of 4 pixels at a time.

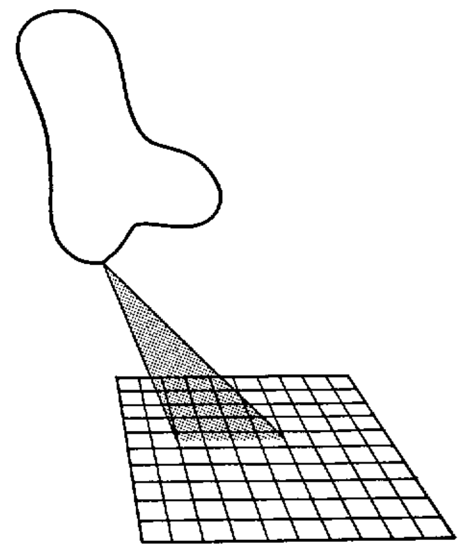


Figure 1 illustrates a local property extractor moving across the image and generating a property value for each row of 4 pixels.

The basic characteristics of the later stages of processing are their use of larger units; their emphasis on integrating, in some consistent fashion, the piecemeal information produced by the earlier processing; and their greater reliance on world model information to help guide the interpretation of the global scene. If, for example, an early stage of processing detects and labels edges according to type, a later stage of processing might use the labeled edges as the basic units and employ a world model to help make the edge labeling more complete and consistent.

Often, a stage of local processing will be followed by a stage of global processing guided by world model information. It is this pair of processing steps which we want to discuss in greater detail in order to characterize its general form.

To do this, we first illustrate a low-level processing example: boundary delineation. The first step in boundary delineation is micro-edge property extraction. Any pixel can be labeled as a micro-edge if it has on either side of it parallel, elongated, homogeneous areas of significantly different average gray tones. The label micro-edge consists of a quadruple whose components are the angular orientation, the elongation, and the average gray tones of each of the two homogeneous areas. Figure 2 illustrates a simple set of micro-edge property masks. Each micro-edge can be in one of eight orientations and have one of two possible elongations: straight and corner. Each homogeneous region consists of a connected set of pixels with a corresponding average gray tone. Its homogeneity can be measured by some criterion such as the variance of its gray tones. Since it is not guaranteed that each pixel will be associated with any micro-edges, a pixel may have associated with it none, one, or more than one micro-edge label. We consider that the local processing stage assigns each pixel the label "no edge" plus some possible micro-edge labels.

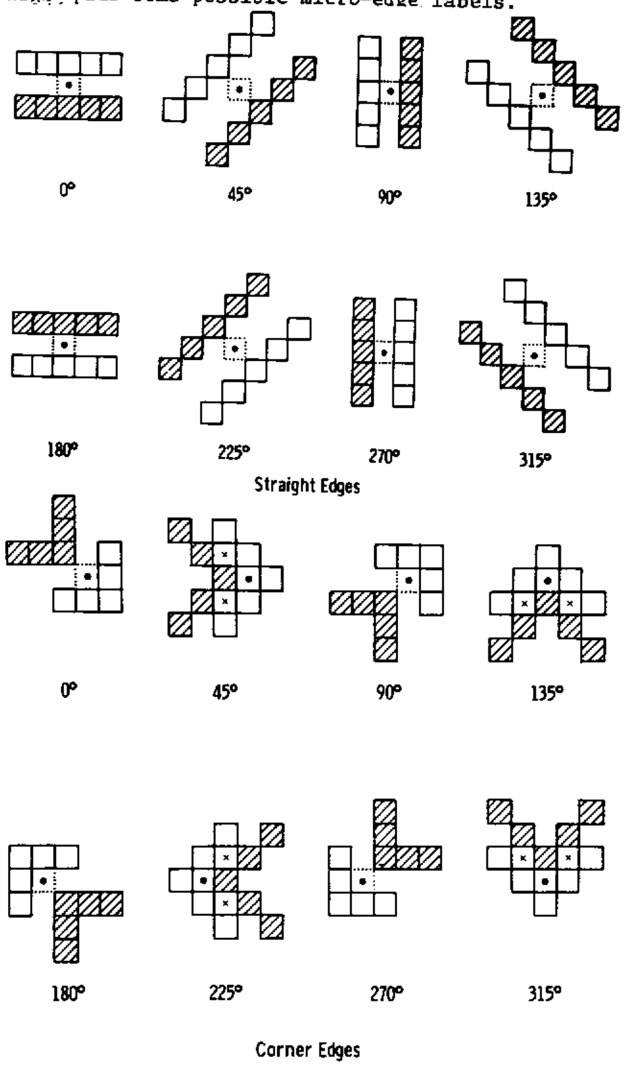


Figure 2 illustrates a simple set of masks which a local processing operation might use to detect straight or corner edges oriented at 8 possible angular directions. The striped resolution cells indicate that it is on the darker side of the edge.

The problem with the local labeling process is that the labels may not be unique and there may be incompatibilities of labels from one group of units to another. The world model information helps guide the next stage of processing by constraining labels of related groups of units to be compatible. By eliminating the incompatibilities, the ambiguity of the labeling can be reduced.

Hence, the next level of processing uses world model information. It groups neighboring pixels if they are labeled with micro-edge shapes and orientations which make smooth transitions from one to the other. For example, the world model may permit a straight micro-edge oriented horizontally to be connected to other neighboring straight micro-edges when such edges are oriented no further than 45^{o} from the horizontal. Also it may be connected to a corner micro-edge, if the corner micro-edge is to its left and oriented at 180^{o}. A world model for this level can be a list showing the pairs of micro-edges which join to form a smooth transition. Such a list is given in Table 1. Robinson (1977) uses a similar constraint. More complex world models could have a greater variety of elongations, orientations, and include probabilities or conditional probabilities for each allowed pair of micro-edge connections.

Type of Spatial Relationships

unordered pairs	(s0,s0)	(s135,s135)	(s90,s90)	(s45,s45)
	(s0,s45)	(s135,s180)	(s90,s135)	(s45,s0)
	(s0,s315)	(s135,s90)	(s90,s45)	(s45,s90)
	(s180,s180)	(s315,s315)	(s270,s270)	(s225,s225)
	(s180,s225)	(s315,s0)	(s270,s315)	(s225,s180)
	(s180,s135)	(s315,s270)	(s270,s315)	(s225,s180)
ordered pairs	(s0,c90)	(s135,c315)	(s90,c270)	(s45,c225)
	(s180,c0)	(s315,c45)	(s270,c0)	(s225,c315)
	(c180,s0)	(c135,s315)	(c90,s270)	(c45,s225)
	(c270,s180)	(c225,s135)	(c180,s90)	(c135,s45)

Table 1 lists the allowable connections between micro-edges as specified in an example world model. There are four relationship types: vertical, horizontal, left diagonal, and right diagonal. Under each relationship type is a list of the micro-edge pairs which can be smoothly connected when they are in the given relationship. The pair (s0,c90) under [•|•] specifies that straight micro-edge oriented at 0^{o} to the left of corner micro-edge oriented at 90^{o} is allowable. Under any of the four types of spatial relationships, any micro-edge can be next to a pixel labeled no edge.

The grouping at this level of processing has the following effect: those micro-eges which cannot be grouped with some other micro-edges lose their status as micro-edges and become pixels labeled no edge. Each micro-edge which can be grouped with some other micro-edge is allowed to retain its label. In this manner micro-edges are allowed to reinforce one another. If the reinforcement is done iteratively until there are no more changes, and all effects have had a chance to propagate from each part of the image across the entire image, then the resulting micro-edge image is fully consistent with the world model information and the number of pixels falsely labeled as micro-edges is reduced.

Thus the first two processing levels in boundary delineation consist of a local property extraction stage followed by a processing stage that uses world model information. The local property extraction processing at each pixel of the image does not influence the property extraction at any other pixel of the image. The world model processing locates those pixels labeled as micro-edges that are not compatible with anything around them and changes their labels. This status change can affect the pixel's neighbors, its neighbors neighbors, and so on until the whole image has been affected. Because such processing propagates changes throughout the image, each pixel change can affect every other pixel, and we say the processing is global.

III. Scene Analysis and Homomorphisms

In the last section we described how the local property extraction step which groups units together, defining larger units, and giving each of the larger units one or more labels is followed by a global processing step which determines the relationships among the larger units. The global processing step specifies these relationships in a structure which we will define to be a arrangement (a labeled N-ary relation). Then the world model constraints are imposed to force the labels assigned to the larger units by the previous process to be mutually compatible. The world model constraint itself is an arrangement and in this section we will show that the imposing of its constraints amounts to determining a homomorphism from the first arrangement to the world model constraint arrangement which is consistent with the labels assigned by the lower processing level.

We illustrate this idea with the help of a simple abstract example. Let us suppose that our initial units are pixels and that some of the pixels in an image are named p_1 to p_{12} and q_1 to q_9. To do local processing, each group of related pixels must be examined. To keep our example simple, we will only concern ourselves with the related groups of pixels in the set p_1 to p_{12} and q_1 to q_9. As Figure 3 illustrates, there are four groups of related pixels:

$$\{p_n \mid n = 1,\dots,7\},\ \{p_n \mid n = 5,\dots,12\},$$

$$\{q_n \mid n = 1,\dots,5\},\ \text{and}\ \{q_n \mid n = 6,\dots,9\}$$

Each of these four groups are given names indicated by the shapes associated with these groups in Figure 3. The values the local property extractor associates with each unit group comes from the label set {a,b,c,d,e}. The labels for each unit group appear inside the shape which names the groups. Thus, the labels a and b are associated with the unit group $\{p_n \mid n = 1,\dots,7\}$.

Relative to our earlier boundary delineation example, a group of related units is any set of pixels in one of the spatial configurations of the micro-edges masks shown in Figure 2. The label set consists of the label no edge and the various different kinds of micro-edges distinguished by their type and angular orientation. Note that in the boundary delineation example, the name of each group of related units is the pixel in the output image into which the property extractor places the labels associated with the unit groups. Hence, different unit groups may have the same name.

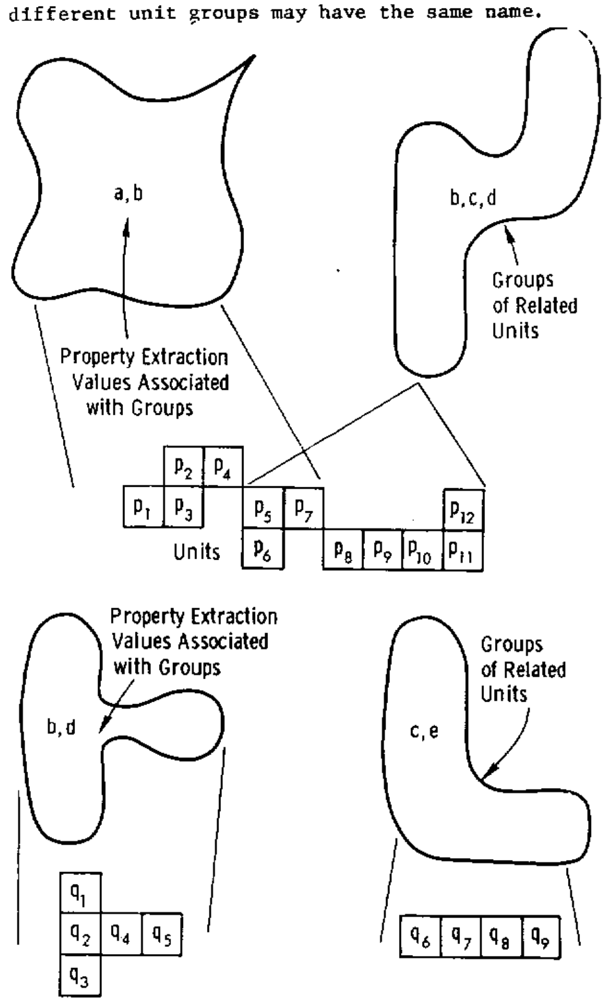


Figure 3 shows related units being grouped together and a local property extraction operation associating labels from the set {a,b,c,d,e} with each of the unit groups.

The world model in the global processing step names the relationships among sets of unit groups. For a two-dimensional world model, relationships between pairs of unit groups are named. For the boundary delineation example, there are four spatial relationships which are named in the top of Table 1. In general, the specifying of relationships between unit groups can be illustrated as in Figure 4. Then such a world model specifies the allowable or meaningful property extraction label pairs that can exist for each kind of unit group relationship pair. For the boundary delineation example, these are the constraints listed in Table 1. The specifying of the kind of constraint a two dimensional world model can impose on the labels associated with unit groups is illustrated in

Figure 5. The arrow from label a to label c for relationship x means that if a pair (G,H) of unit groups is related by relationship x, then it is allowable or meaningful for group G to have label a and group H to have label c. However, since there are no arrows from label a to labels b, d, or e, if the unit group G only has label a, the unit group H cannot have the labels b, d, or e.

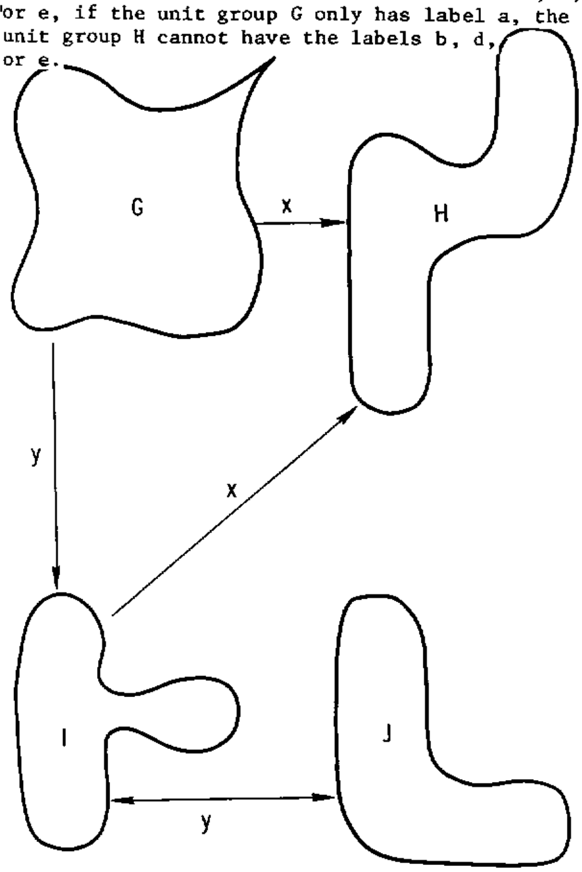


Figure 4 shows the names x or y a two-dimensional world model may give to pairs of unit groups. We call this relation the unit constraint relation.

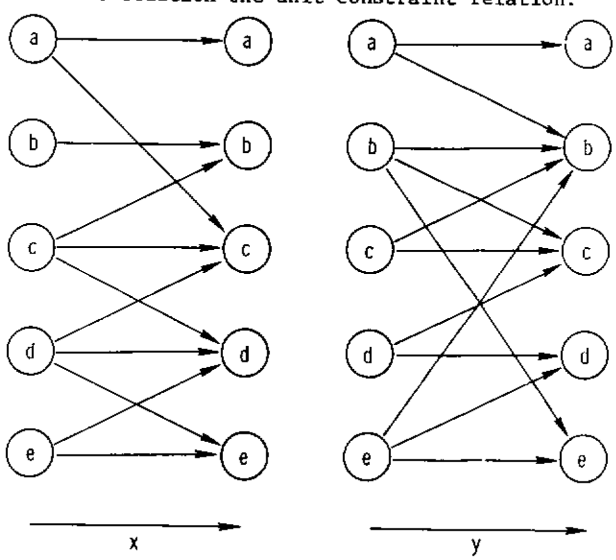


Figure 5 shows for each different unit pair relationship the constraints a world model can impose on the labels of unit groups. An arrow from label a to label c under relationship type x means that if a pair (G,H) of unit groups is related by relationship x, then it is allowable or meaningful for group G to have label a and group H to have label c. We will call this relation the label constraint relation.

To make our data structures take a uniform appearance, we show in Figure 6 the binary relation produced by the property extraction step illustrated in Figure 3. The constraint imposed by the world model can readily be understood by first examining Figures 4 and 6. Figure 4 indicates that a pair of unit groups, say (G,H), has the relationship x. Figure 6 indicates that unit group G can have label a and unit group H can have label d. Therefore, there is an induced relationship x on the label pair (a,d). The induced relationship is obtained as the composition of the relations in Figures 4 and 6.

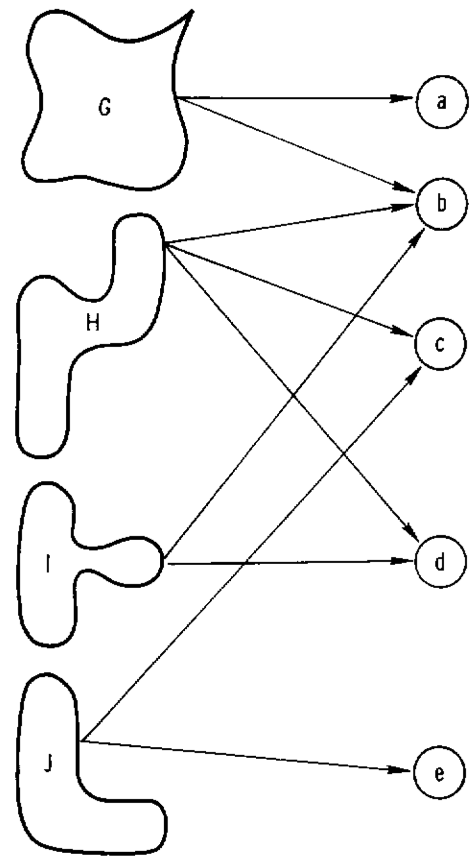


Figure 6 shows the binary relation produced by the property extraction step which assigns labels to units. This binary relation carries the same information as that shown in Figure 3.

The composition is done in the following way. If one group has labels a and b and is related by relationship type x to another unit having labels b and d, then links (a,b), (a,d), (b,b), and (b,d) are added to a graph for relationship x. The results of such a composition are shown in Figure 7.

Notice that there are links which are not in the world model constraint relation of Figure 5. Such links, e.g., (a,b), for relationship type x, indicate that there is some group having label a which is related by relationship x to a group having label b and that this pair of labels is not compatible with the world model. Upon eliminating all incompatible pairs of labels, the labeling of Figure 8a results.

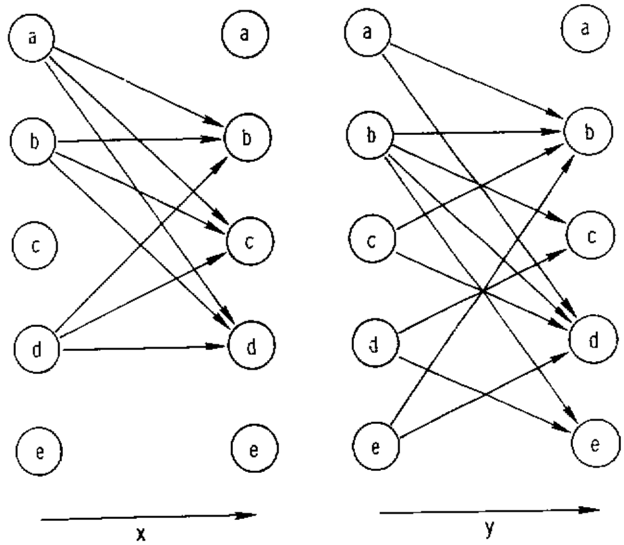


Figure 7 shows the composition of the relation of Figure 4 with the labeling relation of Figure 6. Notice that there are links shown here which are not shown in the relation of Figure 5, the world model constraints. Such links, like (a,b) for relationships type x, indicate that there is some group having label a which is related by relationship x to a group having label b and that this pair of labels is not compatible with the world model.

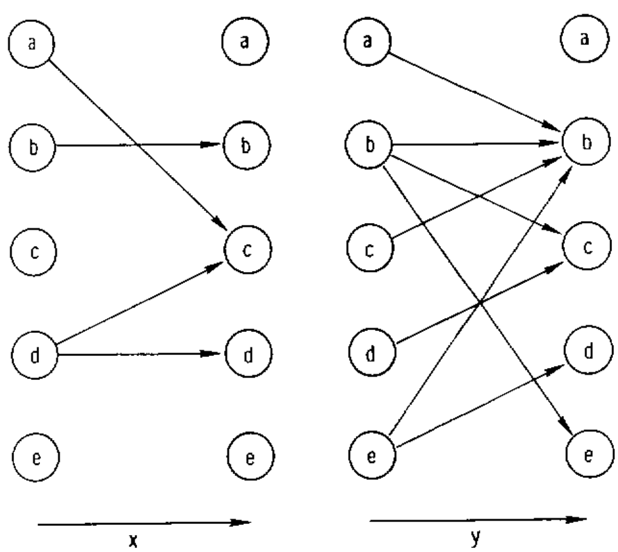


Figure 8 shows the compatible pairs of lables of Figure 7 after removing those not in Figure 5.

Certainly those pairs of labels in Figure 7 which are not in Figure 5 are not compatible and we may eliminate them. The resulting relation of compatible labels is shown in Figure 8. This relation may be used to help disambiguate labels assigned by the local processor to unit groups G, H, I, and J.

Suppose group G can keep label a. Since H is related to G by relationship x, then H can only keep label c because the only label that can relate to label a by relationship x is c (See Figure 8). Now if H keeps the label c, since I relates to H by relationship x, then I can only have labels a or d. But the local processor assigned labels b or d to I. Hence I must take the label d.

Now consider the fact that G relates to I by relationship y. G has the label a and I has the label d. Figure 8 indicates that label a does not relate to label d by relationship y. Hence, unit group G cannot take the label a.

Suppose that unit group G keeps the label b. Then since H is related to G by relationship x and the only label that can relate to label b by relationship x is label b (see Figure 8), H must take label b. Also, unit group I must take label b. Now I is related to G by relationship y. Since the label b can relate to itself by relationship y, everything is still all right. Unit groups I and J are related by relationship y. Since label b can relate to labels c or e by relationship y, unit group J can keep both its labels c and e.

By this process of tracing compatible labels around, it is possible to reduce the ambiguity of the initial labeling done by the local processing stage. For our example, the resulting labeling is shown in Figure 9a. In general, the result is not single-valued as can be seen from this example.

Figure 9b shows the relation which is the composition of the unit constraint relation (Figure 4) with the labeling of Figure 9a. Notice that all links in Figure 9b are links in the label constraint relation of Figure 5. Whenever the composition of one relation with a second results in a relation which is contained in a third relation, we call the second relation a homomorphism from the first relation to the third relation. Thus the labeling of Figure 9a is a homomorphism from the unit constraint relation to the label constraint relation.

In the next sections we will develop the precise mathematical idea of arrangement homomorphism, describe how a variety of scene analysis tasks can be posed as problems in finding arrangement homomorphisms, and describe a general method for tracing compatible labels around in order to eliminate the incompatible labels.

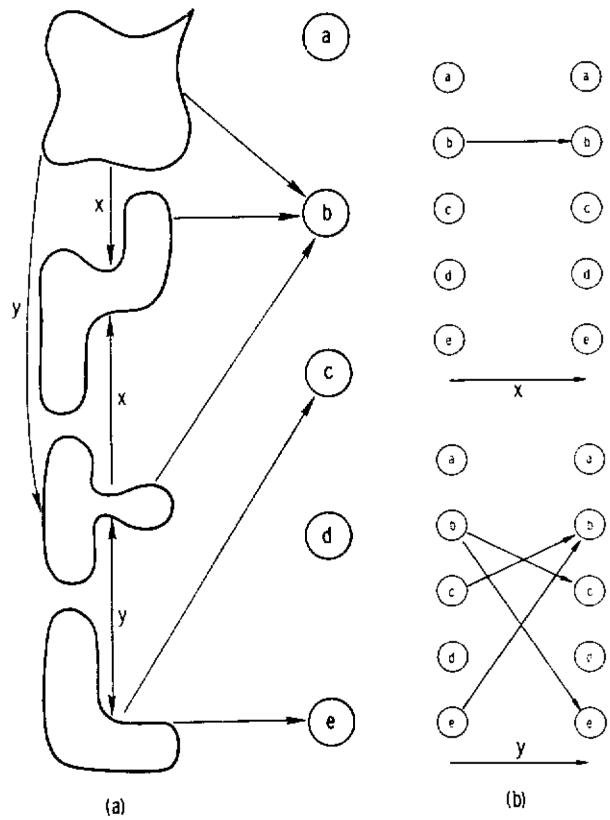


Figure 9a shows the unit labeling compatible with the world model constraints and unit group relationships. This labeling contains two homomorphisms from the unit constraint relation of Figure 4 to the label constraint relation of Figure 5.

Figure 9b shows the composition of the relation of Figure 4 with the label assignment of Figure 9a. Notice that all links in Figure 9b are also in Figure 5. This indicates that the labeling of Figure 9a contains homomorphisms.

IV. The Arrangement

In this section we give the definition for arrangements and arrangement homomorphisms. The definitions here are a generalization of that given by Haralick and Kartus (1976).

Let A be the set of elements whose arrangement is being described. Each group of related elements from A is given a label from the label set L. Let R be the labeled N-ary relation which consists of labeled N-tuples of elements from A. Then a simple order-N arrangement is a triple (R,A,L) where $R \subseteq A^N \times L$. When the sets A and L are understood, the relation R is called an arrangement for short.

A general arrangement is a set of simple arrangements, each simple arrangement being of different order, being defined on the same set, and having the same label set. If there are K simple arrangements in the arrangement $\mathcal{A}$, then we write

$$\mathcal{A} = \{R_1, R_2, \ldots, R_K;\ A, L\} \text{ where}$$

$$R_k \subseteq A^{N_k} \times L,\ k = 1, \ldots, K.$$

Let $\mathcal{A} = \{R_1, \ldots, R_K;\ A, L\}$ be a general arrangement and $H \subseteq A \times B$.

The composition of arrangement $\mathcal{A}$ with H results in an arrangement $\mathcal{B}$ which we define as

$$\mathcal{A} \circ H = \mathcal{B} = \{S_1, S_2, \ldots, S_K;\ B, L\}, \text{ where}$$

$$S_k = \{(b_1, b_2, \ldots, b_{N_k}, \ell) \in B^{N_k} \times L \mid \text{for some } (a_1, a_2, \ldots, a_{N_k}, \ell) \in R_k,\ (a_n, b_n) \in H,\ n = 1, \ldots, N_k\} = R_k \circ H,\ k = 1, \ldots, K.$$

An arrangement $\mathcal{A} = \{R_1, \ldots, R_K;\ A, L\}$ is contained in an arrangement $\mathcal{D} = \{T_1, \ldots, T_K;\ A, L\}$ if and only if

$$R_k \subseteq T_k,\ k = 1, \ldots, K.$$

In this case we write $\mathcal{A} \subseteq \mathcal{D}$.

Two arrangements $\mathcal{A} = \{R_1, \ldots, R_K;\ A, L\}$ and $\mathcal{B} = \{S_1, \ldots, S_N;\ B, M\}$ are comparable if the number of relations in each arrangement is the same (K=N), the label sets are the same (M=L), and the relation R_k has the same order as the relation S_k ($R_k \subseteq A^{N_k} \times L$ and $S_k \subseteq B^{N_k} \times L$).

Let $\mathcal{A} = \{R_1,\ldots,R_K; A,L\}$ and $\mathcal{B} = \{S_1,\ldots,S_K; B,L\}$ be two comparable arrangements. Let $H: A \rightarrow B$. The function H is a homomorphism from arrangement $\mathcal{A}$ to arrangement $\mathcal{B}$ if and only if $A \circ H \subseteq B$.

V. Examples

In this section we show how some boundary delineation tasks, scene labeling tasks, and image understanding tasks can be described within the mathematical framework of simple arrangements.

V.1 Boundary Delineation

The local processing operation for boundary delineation is usually some kind of gradient operator. We will assume that the gradient operator associates with each neighborhood of resolution cells a set of possible edge labels. Each such a set of labels includes the label no edge. The first part of the boundary delineation problem is to use some higher level world model information to retain in each neighborhood as many of the labels assigned by the local processing operation as possible and at the same time make sure that incompatible labeling situations are removed. The second part of the boundary delineation problem is to use the micro-edges retained by first part and fill in all likely gaps in the borders. Both parts have a similar mathematical description.

Let R be the set of resolution cells of the image and let $S \subseteq R^N$ be the group of spatially related resolution cells. We call S the set of neighborhoods of the image. In a simple case, N could equal 9 and S could be the set of all 3 x 3 neighborhoods. In our example in Section II, N is 49 and S is the set of all 7 x 7 neighborhoods. Let L be the set of the names for some of the possible relationships which groups of neighborhoods can be in. In our example in Section II, K is 2 and L has the names horizontal, vertical, left diagonal, and right diagonal. Let E be the set of possible micro-edge labels. In our example in Section II, E contains the labels no edge, straight edge at angular orientation θ, and corner edge at orientation θ where θ can be a multiple of 45^o.

The local processing step associates edge labels with neighborhoods. For the first part of boundary delineation, the association is determined by some sort of a compass gradient operator on the original image. When a gradient at a particular orientation is high enough, the label micro-edge at the particular angular orientation is instantiated. For the second part of boundary delineation, the association is determined by some kind of neighborhood operator in the class of region growing operators operating on the edge image created in the first part of the boundary delineation operation. In either case, the local processing operation determines a binary relation $G \subseteq S \times E$ which pairs neighborhoods to edges.

The world model consists of a pair of arrangements (T,C) where $T \subseteq S^K \times L$ is the unit constraint arrangement and $C \subseteq E^K \times L$ is the label constraint arrangement. These relations are the general form for the unit constraint and label constraint relation used in our abstract example in Section III (Figure 4 and Figure 5). T indicates for each relationship type in L the ordered group of neighborhoods which have the relation. C indicates for each relationship type in L the ordered groups of micro-edge labels which are compatible when situated in neighborhoods of the given relationship type.

Each step in boundary delineation then corresponds to finding a mapping $H: S \rightarrow E$, $H \subseteq G$, satisfying the homomorphism condition $T \circ H \subseteq C$. Hence the problems of boundary delineation can be posed as a problem of finding a homomorphism H, contained in G, from the arrangement (T,S,L) to the arrangement (C,E,L).

V.2 Scene Labeling

Suppose a scene has been divided into segments $S = \{s_1,\ldots,s_K\}$. A low level feature extractor with decision rule using gray tone, color, shape, and texture of each segment assigns some possible description from a set D of descriptions to each segment. This operation defines a relation $G \subseteq S \times D$. The problem with this low-level assignment is that each segment may be associated with multiple descriptions. The desired labeling of the scene would have each segment described unambiguously.

A similar situation arises in the line labeling problem of Waltz (1972). Here, S is the set of line segments found in a scene and D is a set containing labels that can be associated with any line. The labels in D could be, for example, convex, concave, occluding left, occluding right. The relation F, determined from low level processes, associates with each line in S one or more labels from D. The desired line labeling would be some subset of F that associates each line with only one label.

One way of reducing the possibly ambiguous description a line or segment initially has is to use constraints from a higher level world model. Such a model specifies the relevant relationships between groups of related segments of lines and specifies the associated labeling constraints. To employ such a model, related (ordered) sets of N segments or lines must be determined. Segments can be related on the basis of their relative spatial positions. Lines can be related on the basis of the junctions they form. Then for each kind of relationship the model can specify a constraint which the labels of each kind of related segments or lines must satisfy.

For instance, pairs of segments in S could be related if they mutually touch each other. There could be different kinds of touching such as to the left, to the right, above, below, in front of, in back of, supported by, and contained in. Suppose L is the set of such relationship labels. Then the set of spatially related segments or lines could be specified by the relation $A \subseteq S \times S \times L$, where $(s,t,i) \in A$ if and only if label i describes the way segment s relates to segment t. In the general case, the relationships in L can describe the way N segments or lines are related so that the relation A is a labeled N-ary relation: $A \subseteq S^N \times L$ and is, therefore, a simple arrangement.

The world model also contains labeling constraints. For example, pairs of segments whose relationship label is i can be constrained by the world model to have associated with them only certain allowable description pairs. In this case the world model label constraint is an arrangement $C \subseteq D \times D \times L$, where $(d_1,d_2,i) \in C$ if and only if it is legal for a pair of segments s_1 and s_2 having relation i to have respective descriptions d_1 and d_2. In general, the relation C is a labeled N-ary relation, $C \subseteq D^N \times L$ which includes in it all labeled N-tuples of compatible descriptions for an ordered set of N related segments.

To summarize the information we have available:

(1) $G \subseteq S \times D$, the assignments of descriptions given by a low level operation;

(2) The world model (A,C) where $A \subseteq S^N \times L$ is the labeled sets of related N-tuples of segments and $C \subseteq D^N \times L$ is the N-ary relational labeling constraints.

The scene labeling problem is to use F,A, and C to determine a new labeling relation H which contains fewer ambiguous descriptions than G and which is consistent with the constraints specified by the world model. In essence we want

(1) $H: S \to D,\ H \subseteq G$

(2) $A \circ H \subseteq C$

Notice that (A,S,L) is a simple arrangement, (C,D,L) is a simple arrangement, and H is a binary relation which successfully translates the structure of arrangement (A,S,L) into the structure of arrangement (C,D,L). The binary relation H is the homomorphism from arrangement (A,S,L) into arrangement (C,D,L) which is contained in G.

Our discussion of scene labeling is more general than that of Rosenfeld, Hummel, and Zucker (1976) who consider only binary relational labeling constraints. We consider N-ary relational labeling constraints; any ordered set of N segments can have a N-ary relation labeling constraint. For the particular binary case (N=2), if we define a unique label for each pair of segments, then the treatment given here exactly corresponds to that in Rosenfeld, Hummel, and Zucker.

V.3 Image Understanding

We will illustrate the arrangement concept in image understanding by considering a few highly stylized problems. This example is taken from Haralick and Kartus (1976). Suppose we have a segment in terms of certain basic attributes, for example, shape discriminators. Using these attributes, we could assign a shape label to each of the segments. To define an arrangement from these labels, we can group related segments together, N at a time, and form the corresponding set of N-tuples of their labels.

Depending on the particular segmentation task, the order of the segments in the groups may or may not be important. For example, it may be reasonable in some kind of image understanding problems to order the segments in a left-right top-bottom manner. On the other hand, if the order of the segments in the group is not important, an arbitrary fixed order based on the segments shape can be used.

The label given to each N-tuple can be the name we might give to a group of related segments whose shapes are the components of the given N-tuple. Another possibility is to use the interpretation label as a counter. We can assign the integer label "1" to all N-tuples arising from a group of segments the first time the N-tuple is encountered. The label "k" can be assigned the kth time the same kind of N-tuple is encountered.

One criterion by which segments can be considered related is spatial connectivity or nearness. Two segments are eligible to be included in the same related group when their interaction lengths overlap. To make things simple in our examples we will use interaction lengths of zero. Thus, two segments are related only when they are touching. In the stylized examples we give, segments are represented by circles, squares, triangles, etc.

Arrangements can be used to establish the likeness of two images when one image is essentially the same as the other, but the order or placement of the image parts is different. In this case template matching the images will not work. Often geometric transformations of rotation, magnification, translation, skew will also not work. The example shown in Figure 10 illustrates one way of handling this problem using the notion of connectedness and simple order-3 arrangements. Suppose the image has five basic kinds of figures: squares, triangles, circles, arrows, and hexagons. A quadruple whose first three components are these shapes taken in the order square, triangle, circles, hexagons, and arrows will be considered to belong to the arrangement of the image if all three shapes touch each other in a pairwise manner. In general, we may use the criterion consider any N-tuple if enough of its components interact in a pairwise or K-wise manner. A label of 1 or 2 will also be associated with each triple of shapes to make the quadruple; in this example such a label will just count the number of times that the triplet it is associated with occurs. In Figure 10, there are four drawings. Each drawing has two triangles, one circle, one square, and one arrow. Using the order-3 arrangement concept, there are two pairs of drawings whose arrangements are isomorphic by the identity function. The drawings themselves, however, have their parts placed differently in absolute position and orientation. This isomorphism becomes clear upon examination of Figure 11 which shows the arrangements for the drawings. The drawings on the left are isomorphic to the arrangement labeled A. The drawings on the right are isomorphic to the arrangement labeled B.

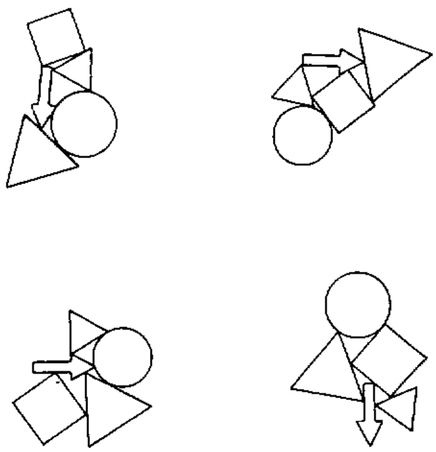


Figure 10 illustrates four drawings, each of which has two triangles, one square, one circle, and one arrow. Using the order-3 arrangements concept, there are 2 pairs of drawings whose arrangements are isomorphic.

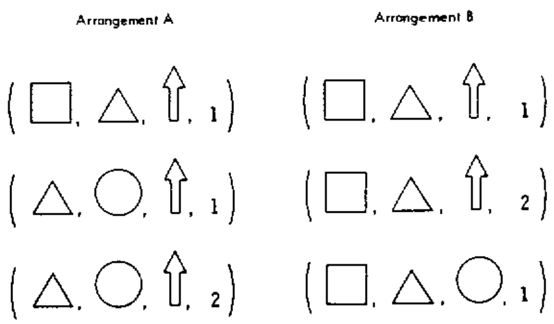


Figure 11 illustrates the quadruples in the order-3 arrangements for the drawings of Figure 10. The two drawings on the left in Figure 10 are isomorphic to Arrangement A and the two drawings on the right in Figure 7 are isomorphic to Arrangement B. The quadruple (□,Δ,↑,2) means that the drawing has a piece that consists of a square, triangle, and arrow pairwise touching each other and the label two designates that this is the second such piece in the drawing.

The situation becomes slightly more complicated when the function that establishes the isomorphism is not the identity function. This is illustrated in Figure 12 which also has four drawings. Each drawing has two squares, one circle, one hexagon, and one triangle. Taking the order as square, hexagon, triangle, and circle, and using the order-3 arrangement concept, there are two pairs of drawings in Figure 12 whose arrangements are isomorphic. Also the arrangement for each drawing in Figure 12 is isomorphic to the arrangement for one of the drawings in Figure 10. The isomorphism, however, is not the identity function: a square stays square, a hexagon becomes a triangle, a triangle becomes an arrow, and a circle remains a circle.

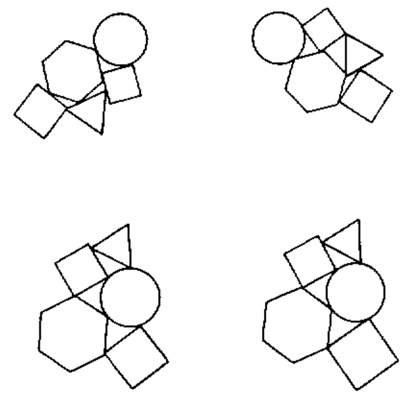


Figure 12 illustrates four drawings each of which has two squares, one circle, one hexagon, and one triangle. Using the order-3 arrangement concept, there are 2 pairs of drawings whose arrangements are isomorphic. The arrangement for each drawing is isomorphic to the arrangement for one of the drawings in Figure 10.

More complicated still is the case where the correspondence between one drawing and another is by an arrangement homomorphism which does not establish a one-one correspondence. Such a case is illustrated in Figure 10 which depicts two drawings. Taking the order as hexagon, circle, triangle, arrow, and square and using the name or label 1 for all triplets except the triplet (arrow, triangle, square) which gets the name 2, we may use the arrangement concept to establish the correspondence between one of the drawings (the one on the right) in Figure 13 and two of the drawings in Figure 10 (the ones on the left). The correspondence is a homomorphism and finding it, although easy, should begin to give the reader some idea of the combinatorial problems involved. The drawing on the left of Figure 13 is homomorphic to neither of the drawings in Figure 10.

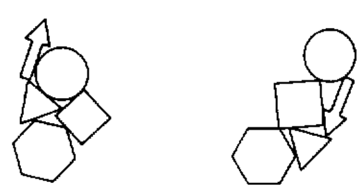
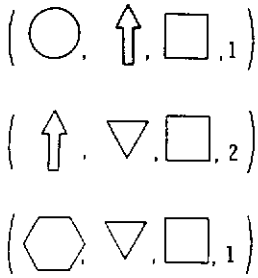
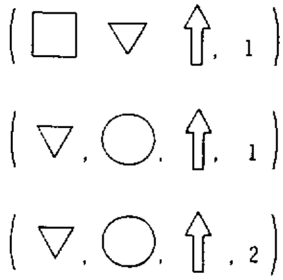


Figure 13 illustrates two drawings. Using the arrangement concept, labels of 1 or 2 can be assigned to each triplet to make one of the drawings in Figure 10 a homomorphic image of one of these drawings.

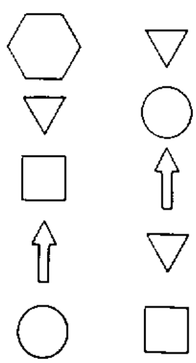
The problem of finding homomorphisms is truly one of establishing the correspondence using relationships. Figure 14 shows the quadruples in the arrangement for the right-hand drawing of Figure 13 and the arrangement for the left-hand drawing of Figure 10. The homomorphism which establishes the relationship between the arrangements appears in the central bottom part of Figure 14.



Arrangement for Right-hand Drawing of Figure 13.



Arrangement for Left-hand Drawing for Figure 1?.



Homomorphism

Figure 14 illustrates the arrangement for one of the drawings in Figure 13 and the arrangement for one of the drawings in Figure 10. Below the arrangement is the homomorphism.

VI. Discrete Relaxation

In the last few sections we have illustrated how a variety of scene analysis problems can be posed as problems of finding homomorphisms from one arrangement to another. In this section we describe a discrete relaxation algorithm which can be used to help determine all the homomorphisms from one arrangement to another. The form of the discrete relaxation algorithm given here is a generalization to N dimensions of Waltz filtering (Waltz, 1972) and the discrete scene labeling relaxation (Rosenfeld, Hummel, and Zucker, 1976) and network consistency relation (Mackworth, 1977).

Let (R,A,L) and (S,B,L) be two order-N simple arrangements. Let $T_0 \subseteq A \times B$ be a given constraint which must contain any homomorphisms we are interested in finding. By definition a function $H: A \rightarrow B$ is a homomorphism from arrangement (R,A,L) to arrangement (S,A,L) if and only if $R \circ H \subseteq S$. We would like to find all such functions H satisfying the constraint $H \subseteq T_0$.

To motivate the algorithm, consider that each labeled N-tuple of R must be translated through H to some labeled N-tuple in N. Since H is contained in T_0 we might begin by pairing each labeled N-tuple in R to those labeled N-tuples in S it can be translated to through any pairs in T_0.

So, for example, if R has the labeled triple (a,b,c,ℓ) and $T_0 = \{(a,\alpha), (a,\beta), (b,\beta), (c,\beta), (c,\alpha)\}$ then the list of triples $(\alpha,\beta,\beta,\ell)$, $(\alpha,\beta,\alpha,\ell)$, (β,β,β,ℓ), and $(\beta,\beta,\alpha,\ell)$ are all the possible labeled triplets to which T_0 can translate the labeled triplet (a,b,c,ℓ). Suppose of these four labeled triplets only $(\alpha,\beta,\beta,\ell)$ and $(\beta,\beta,\alpha,\ell)$ are in S. Then the pairing establishes the correspondence of (a,b,c,ℓ) to $(\alpha,\beta,\beta,\ell)$ and $(\beta,\beta,\alpha,\ell)$.

This correspondence of labeled N-tuples in R to labeled N-tuples in S carries information which can eliminate pairs in T_0 which cannot possibly contribute to any homomorphism. It can do so in the following way. If the pair (a,b) is to be in some homomorphism, then each labeled N-tuple in R which has a component with the value a must be able to be associated with some labeled N-tuple of S having the corresponding component with value b. The association of labeled N-tuples of S with labeled N-tuples of R is carried by the composition through T_0 as described in the previous paragraph.

The discrete relaxation process iterates first using a T to establish a labeled N-tuple to labeled N-tuple correspondence and then using this correspondence to determine a smaller T which can be used to establish the labeled N-tuple correspondence for the next iteration. At each iteration T, which is assumed finite, becomes smaller. The iterations finally reach a fixed point since T is bounded below by the empty set. If a homomorphism H satisfying the constraint $H \subseteq T_0$ exists and is unique, then H will be the fixed point of the iterations. If more than one homomorphism exists, then the fixed point will not be single-valued and will contain all homomorphisms satisfying the constraint $H \subseteq T_0$. In the next section we will discuss a combination of a tree search and discrete relaxation process to divide a multi-valued relation into its component homomorphisms. In the remainder of this section we give a precise mathematical description of the discrete relaxation and illustrate its use on the abstract simple example we worked in an intrutive manner in Section III.

To begin we will need some notational conventions. Let $R \subseteq A^N \times L$, $S \subseteq B^N \times L$, and $T_0 \subseteq A \times B$. The iterations define $T_1, T_2, \ldots, T_K$ each relation being a restriction of the previous one. We will suppose that T_k is defined. The first part of the iteration goes through each labeled N-tuple of R and associates it with each labeled N-tuple of S it can by composition through T_k. We will want to have a way of describing such an association. So let $(a_1,\ldots,a_N,\ell)$ be a labeled N-tuple of R and let $G(a_1,\ldots,a_N,\ell;\, S,T_k)$ be the set of all labeled N-tuples in S which can be reached by the composition of $(a_1,\ldots,a_N,\ell)$ through the relation T_k.

$$G(a_1,\ldots,a_N,\ell;\, S,T_k) = \{(b_1,\ldots,b_N,\ell) \in S \mid (a_n,b_n) \in T_k,\ n = 1,\ldots,N\}.$$

We know that a pair $(a,b) \in T_k$ is not a contributor in any homomorphism if for some labeled N-tuple $(a_1,\ldots,a_N,\ell)$ in R having a component with value a and label ℓ, we cannot find in $G(a_1,\ldots,a_N,\ell;\ S,T_k)$ a labeled N-tuple having a corresponding component with value b. To write this easily we need a notation which lets us select from any set of labeled N-tuples all those N-tuples with a given label and which for a specified component have a particular value. So if we desire to select from R those N-tuples having label ℓ and value a for component n we need only write $R_n(a,\ell)$ which we define by

$$R_n(a,\ell) = \{(a_1,a_2,\ldots,a_N,\ell) \in R \mid a = a_n\}$$

The next step in the iteration process is to select a value a for component n and select a label ℓ. Then go through each labeled N-tuple in $R_n(a,\ell)$. Suppose $(a_1,\ldots,a_N,\ell) \in R_n(a,\ell)$. The labeled N-tuples of S which correspond to $(a_1,\ldots,a_N,\ell)$ can be found in $G(a_1,\ldots,a_N,\ell;\ S,T_k)$. The pair $(a,b) \in T_k$ can be a pair which participates in some homomorphism only if there is some labeled N-tuple in $G(a_1,\ldots,a_N,\ell;\ S,T_k)$ having its n^{th} component with value b. There is some labeled N-tuple in $G(a_1,\ldots,a_N,\ell;\ S,T_k)$ havings its n^{th} component with value b if and only if the projection of $G(a_1,\ldots,a_N,\ell;\ S,T_k)$ onto its n^{th} coordinate contains the element b.

We clearly need a notation for projection. For any $Q \subseteq C^N \times L$ we define the projection to the n^{th} coordinate by

$$\pi_n Q = \{c \in C \mid \text{for some } (c_1,\ldots,c_N,\ell) \in Q,\ c_n = c\}$$

Hence the set of values which exist in the n^{th} component for the labeled N-tuples in $G(a_1,\ldots,a_N,\ell;\ S,T_k)$ can be written as $\pi_n G(a_1,\ldots,a_N,\ell;\ S,T_k)$.

If the pair (a,b) has a possibility in participating in some homomorphism, then for each label ℓ, for each component n, and for each labeled N-tuple $(a_1,\ldots,a_N,\ell)$ in $R_n(a,\ell)$, there must exist some labeled N-tuple in $G(a_1,\ldots,a_N,\ell;\ S,T_k)$ whose n^{th} component has value b. Hence,

$$b \in \bigcap_{\ell\in L} \bigcap_{n=1}^{N} \bigcap_{(a_1,\ldots,a_N,\ell)\in R_n(a,\ell)} \pi_n G(a_1,\ldots,a_N,\ell;\ S,T_k)$$

The discrete relaxation defines the restriction T_{k+1} by

$$T_{k+1} = \{(a,b)\ T_k \mid b \in \bigcap_{\ell\in L} \bigcap_{n=1}^{N} \bigcap_{(a_1,\ldots,a_N,\ell)\in R_n(a,\ell)} \pi_n G(a_1,\ldots,a_N,\ell;\ S,T_k)\}$$

The following theorem proves that if $H \subseteq T_k$ and H is a homomorphism from R to S, then $H \subseteq T_{k+1}$. Hence, those pairs which were in T_k but not in T_{k+1} do not participate in any homomorphism.

<u>Theorem</u>: Let $R \subseteq A^N \times L$, $S \subseteq B^N \times L$, and $H \subseteq T \subseteq A \times B$. If $H: A \to B$ satisfies $R \circ H \subseteq S$, then $H \subseteq \phi(T)$ where

$$\phi(T) = \{(a,b) \in T \mid b \in \bigcap_{\ell\in L} \bigcap_{n=1}^{N} \bigcap_{(a_1,\ldots,a_N,\ell)\in R_n(a,\ell)} \pi_n G(a_1,\ldots,a_N,\ell;\ S,T_k)\}$$

<u>Proof</u>: Suppose $(a,b) \in H$. Since $H \subseteq T$ is given we need only show that

$$b \in \bigcap_{\ell\in L} \bigcap_{n=1}^{N} \bigcap_{(a_1,\ldots,a_N,\ell)\in R_n(a,\ell)} \pi_n G(a_1,\ldots,a_N,\ell;\ S,T).$$

So let $\ell\in L$ and $n \in \{1,\ldots,N\}$ be given. There are two cases: either $R_n(a,\ell) = \phi$ or $R_n(a,\ell) \neq \phi$. If $R_n(a,\ell) = \phi$, then the result is immediate since intersections over empty collections are always full. If $R_n(a,\ell) \neq \phi$, let $(a_1,\ldots,a_N,\ell) \in R_n(a,\ell)$. By definition of $R_n(a,\ell)$, we must have $a_n = a$.

Since H is a function, it is defined everywhere and there exists $b_1,\ldots,b_N \in B$ such that $(a_n,b_n) \in H$, $n = 1,\ldots,N$. Since $a_n = a$ and $(a,b) \in H$, we may take $b_n = b$.

By definition of $G(a_1,\ldots,a_N,\ell;\ S,T)$, $G(a_1,\ldots,a_N,\ell;\ S,T) = \{(b_1^*,\ldots,b_N^*,\ell) \in S \mid (a_n,b_n^*) \in H,\ n = 1,\ldots,N\}$. Now $(a_1,\ldots,a_N,\ell) \in R$ and $(a_n,b_n) \in H$, $n = 1,\ldots,N$ imply $(b_1,\ldots,b_N,\ell) \in R\circ H$. By assumption, $R\circ H \subseteq S$. Hence, $(b_1,\ldots,b_N,\ell) \in S$ and we must have $(b_1,\ldots,b_N,\ell) \in G(a_1,\ldots,a_N,\ell;\ S,T)$. Then $b = b_n \in \pi_n G(a_1,\ldots,a_N,\ell;\ S,T)$. Therefore,

$$b \in \bigcap_{\ell\in L} \bigcap_{n=1}^{N} \bigcap_{(a_1,\ldots,a_N,\ell)\in R_n(a,\ell)} \pi_n G(a_1,\ldots,a_N,\ell;\ S,T)$$

The next proposition proves that if H is a fixed point of the iteration process, then H single-valued implies that $R\circ H \subseteq S$. This means that if a function H is a fixed point of the iteration process, then H is a homomorphism.

<u>Proposition</u>: Let $R \subseteq A^N \times L$, $S \subseteq B^N \times L$, and $H \subseteq A \times B$. If

$$H = \{(a,b) \in H \mid b \in \bigcap_{\ell \in L} \bigcap_{n=1}^{N} \bigcap_{(a_1,\ldots,a_N,\ell)\in R_n(a,\ell)} \pi_n G(a_1,\ldots,a_N,\ell;\ S,T)\},$$

then H single-valued implies $R \circ H \subseteq S$.

Proof: Let $(b_1,\ldots,b_N,\ ) \in R \circ H$. Then for some $(a_1,\ldots,a_N,\ell) \in R$, $(a_n,b_n) \in H$, $n = 1,\ldots,N$. By definition of H,

$$b_k \in \bigcap_{\ell \in L} \bigcap_{n=1}^{N} \bigcap_{(a_1',\ldots,a_N',\ell)\in R_n(a_k,\ell)} \pi_n G(a_1',\ldots,a_N',\ell;\ S,H),\ k=1,\ldots,N.$$

Now for each $\ell \in L$ and $n \in N$ $b_k \in \pi_n G(a_1,\ldots,a_N,\ell;\ S,H)$ $k = 1,\ldots,N$. In particular, $b_k \in \pi_k G(a_1,\ldots,a_N,\ell;\ S,H)$, $k = 1,\ldots,N$. But H single-valued implies that for each $(a_1',\ldots,a_N',\ell)$ there exists a unique $(b_1',\ldots,b_N',\ell) \in G(a_1',\ldots,a_N',\ell;\ S,H)$. Hence, $(b_1,\ldots,b_N,\ell) \in G(a_1,\ldots,a_N,\ell;\ S,H)$ and by definition of $G(a_1,\ldots,a_N,\ell;\ S,H)$ this implies that $(b_1,\ldots,b_N,\ell) \in S$.

To provide an example of the relaxation process, Figure 15 puts in list form the unit constraint arrangement and label constraint arrangement of the world model for the abstract example in Section II. The initial T_0 is shown in Figure 16 where the reader can follow the steps of the iterations. The fixed point is reached in 3 iterations.

R	S
GHx	aax aay
GIy	acx aby
IHx	bbx bby
IJy	cbx bcy
	ccx bey
	cdx cby
	dcx ccy
	ddx dcy
	dex ddy
	edx eby
	eex edy
	eey
UNIT CONSTRAINT ARRANGEMENT	LABELING CONSTRAINT ARRANGEMENT

Figure 15 lists the triples in the unit constraint relation and the labeling constraint relation for the abstract example of Section II.

	T_0
G	ab
H	bcd
I	bd
J	ce

N-Tuple	Association
GHx	acx bbx
GIy	aby bby
IHx	bbx
IJy	bcy bey
JIy	cby eby

	T_1
G	ab
H	b
I	b
J	ce

N-Tuple	Association
GHx	bbx
GIy	aby bby
IHx	bbx
IJy	bcy bey
JIy	cby eby

	T_2
G	b
H	b
I	b
J	ce

N-Tuple	Association
GHx	bbx
GIy	bby
IHx	bbx
IJy	bcy bey
JIy	cby eby

	T_3
G	b
H	b
I	b
J	ce

	H_1	H_2
G	b	b
H	b	b
I	b	b
J	c	e

Figure 16 shows how in three iterations the initial labeling T_0 can be reduced. The two homomorphisms H_1 and H_2, which T_3 contains, are shown to the right of T_3.

The application of the discrete relaxation to general arrangements is simple. Let $\{R_1,R_2,\ldots,R_K;\ A,L\}$ and $\{S_1,S_2,\ldots,S_K;\ B,L\}$ be two comparable arrangements. Let $T_0 \subseteq A \times B$ be given. Let the discrete relaxation operate beginning with T_0 and using relations R_1 and S_1. Call the resulting relation T_1 and let the discrete relaxation operate with T_1 using relations R_2 and S_2. After the K^{th} relation has finished, use the resulting relation T in a discrete relaxation for relation R_1 and S_1 and continue cycling through in this manner until relation T does not change for a whole cycle. The limiting relation T will then contain all homomorphisms from the first arrangement to the second that T_0 does.

VII. Tree Search

Should the discrete relaxation not reduce the initial relation T far enough and it is desired to obtain a unique assignment of labels to units, a tree search must be done to determine all the homomorphisms T has. The tree search can proceed as follows. Find the first unit which has more than one label. Successively instantiate each of these possible labels to the unit, thereby branching the tree out. Each instantiation produces a T relation which is a restricted version of the previous level's relation.

Each restricted relation can be put through the discrete relaxation procedure yielding two possible outcomes. Either the fixed point relation is not defined everywhere, in which case that branch of the tree search terminates, or the fixed point relation is defined everywhere, in which case the branch may continue. If the fixed point relation is defined everywhere, then either the relation is single-valued or multi-valued. If the relation is single-valued, it is a homomorphism. If it is multi-valued, then we can again find a unit which has more than one label and successively instantiate these possible labels to the unit and continue to branch the tree out.

VIII. Generalizations: Probabilistic Models

The world model discussed in this paper has been the discrete model. Those labeled N-tuples in the unit constraint relation or label constraint relation had no weights or probabilities associated with them. One natural generalization of this model is to have a weight function defined on each of the constraint relations. The discrete relaxation, then becomes a probabilistic relaxation, which could be similar to that defined by Rosenfeld et al (1976), Davis and Rosenfeld (1976), or Hanson and Riseman (1977). Each of these researchers have reported some success with such procedures.

The problem with the probabilistic relaxation is that it is not yet known if, in fact, the normalized weights used have probability interpretations. Unlike the discrete relaxation which has been shown to preserve homomorphisms, it is not known what the various forms of probabilistic relaxation preserve or optimize. It appears to be a difficult theoretical problem on which more work needs to be done.

IX. Conclusion

In this paper we have introduced the concept of an arrangement as a set of labeled N-ary relations of different orders. We discussed some general scene analysis processes and have illustrated how some of these processes can be viewed as determining or identifying homomorphisms which are constrained by local processing results on the scene data. The homomorphisms are between the unit constraint arrangement and the label constraint arrangement defined by world model. Finally, we have discussed how discrete relaxation, followed by a tree search, can determine homomorphisms from one arrangement to another.

It is our hope that by illustrating the underlying mathematical unity of a diverse set of scene analysis processes, some generality and power can be gained in formulating the total scene analysis problem.

References

1. Haralick, Robert M. and Jess Kartus, "Theory of Arrangements," University of Kansas, Department of Electrical Engineering and Computer Science, June 1976 (to be published in IEEE SMC).

2. Rosenfeld, Azriel, Robert A. Hummel, and Steven W. Zucker, "Scene Labeling by Relaxation Operations," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-6, No. 6, June 1976, pp. 420-433.

3. Waltz, David L., "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, New Jersey, Prentice-Hall, 1975.

4. Hanson, Allen R. and Edward M. Riseman, "Pattern-Directed Boundary Formation Via Relaxation," Machine Vision Workshop, University of Massachusetts, Amherst, Massachusetts, June 1-3, 1977.

5. Mackworth, Alan K., "Consistency in Networks of Relations," Artificial Intelligence, Vol. 8, 1977, pp. 99-118.

6. Robinson, Güner, "Edge Detection by Compose Gradient Masks," Computer Graphics and Image Processing, Vol. 6, No. 5, 1977, pp. 492-501.

7. Davis, L.S. and Azriel Rosenfeld, "An Application of Relaxation Labeling to Spring-Loaded Template Matching," Proceedings of the Third International Joint Conference on Pattern Recognition, San Diego, California, November 1976, pp. 591-597.