Understanding Objects with Curved Surfaces from a Single Perspective View of Boundaries

Shih Jong Lee and Robert M. Haralick
Machine Vision International, Ann Arbor, MI 48104, U.S.A.

Ming Chua Zhang
Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, U.S.A.

Recommended by H.H. Nagel

ABSTRACT

Given a single image of a scene containing a perspective view of three-dimensional objects, we would like a computer vision system to be able to perceive the objects in it. This paper describes an efficient way of labeling a line-drawing image and shows how to utilize the line-labeling and junction-labeling information to group together faces of the same object volume and predict the simplest arrangement and types of the hidden vertices caused by the overlapping of objects. This scheme is valid for both planar and curved-surface objects. In addition, it can handle some instances of multi-type vertices.

1. Introduction

In order to interpret three-dimensional scenes, a number of labeling schemes have been proposed during the past few years. An early program by Guzman [5], called SEE, began an area of research toward interpreting line drawings of polyhedral scenes without the need for specific object prototypes. Huffman [6] and Clowes [3] independently provided a firm theoretical foundation to account for the remarkable performance of Guzman's simple heuristics. They exhaustively cataloged vertices that could arise in line drawings of trihedral solids (solids whose corners are formed by exactly three meeting edges), and then used the catalog to interpret lines as corresponding to convex or concave solid edges. Waltz [9] expanded the classification of lines to eleven types, including cracks and shadow edges, as well as illumination information for the surfaces on either side. The resulting vertex catalog contained thousands of entries.

In addition to these, a number of people have attempted to extend the concept of a vertex catalog beyond trihedral solids. Turner [8] treated a domain involving objects with curved surfaces and developed a catalog with tens of thousands of entries. Kanade [7] allowed sheets as well as solids, and was able to interpret drawings of origami (paper-folding) constructions as well as simple indoor scenes with planar surfaces. Chakravarty [1, 2] proposed a simple junction catalog which can handle a wide variety of curved objects. This catalog is much smaller than what Turner [8] produced.

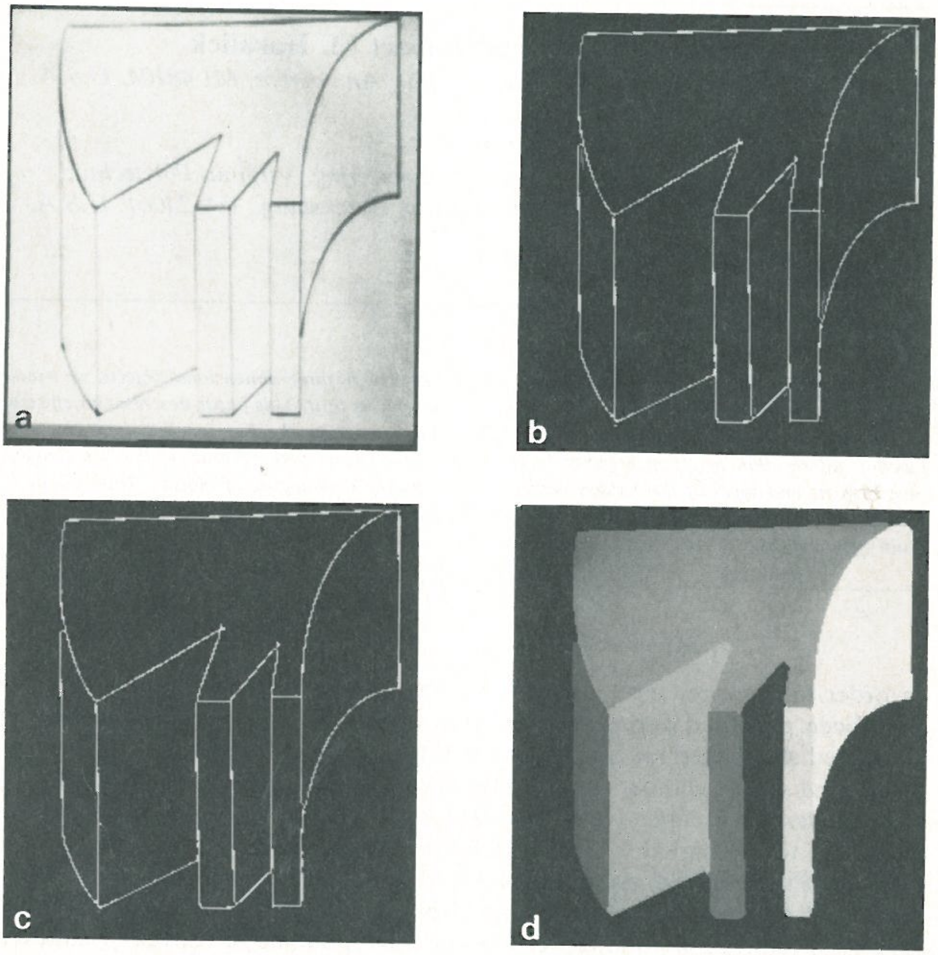


FIG. 1. Initial preprocessing. (a) The picture of the line drawing of an object. (b) The result after applying the ridge-valley operator to the image shown in (a). The white lines represent ridge edges; the grey portions represent valley areas. (c) The result after taking away the valley label of the image shown in (b). (d) The result after labeling the connected regions of the image shown in (c).

In this paper we start from a modification of Chakravarty's [1] labeling scheme by introducing not only junction labeling but also line labeling in the search process. This scheme is valid for both planar and curved-surface bodies. Based on this labeling method, we can obtain both junction labels and line labels. By means of these labels we develop a method of face grouping which can group faces belonging to the same body volume. Furthermore, using the labeling and grouping information, we describe a method which can predict the simplest arrangement and types of the hidden vertices caused by the overlapping of objects.

We will limit ourselves first to planar-faced or curved-surface solid bodies having trihedral vertices only. Then, we will extend our discussion to cover most of the multi-vertices coming from the combinations of more than one trihedral vertex and some objects possessing multi-vertices themselves.

The general labeling procedure can be summarized as follows:

(1) Use a video camera to take the pictures of solid block-like objects or line drawings of them and then digitize the picture.

(2) By means of edge-based segmentation obtain and label each connected region. Then, create a property file including line, region, vertex and chain code data. Fig. 1 shows some of the intermediate pictures during the processing.

(3) Search for the background region, and label each edge associated with this region with a '>' with proper orientation.

(4) Search around the boundary of each region, one region after another, label the boundary and vertices in a way which combines line labeling and junction labeling together to obtain a more efficient search.

The contribution of the labeling scheme developed in this paper is its efficiency and the way we can use it to group together faces of the same object volume.

2. The Labeling Scheme

According to Chakravarty [1], there are seven types of possible junctions which are shown in Fig. 2. The junction types are defined by the number of visible regions that correspond to the projection of faces and the type of faces it is associated with. By definition, junction types A and S are associated with the limb lines; type A is limited to one visible region and type S to two visible regions. Type M and r are limited to two and three visible regions, respectively. Types V and T are the only junctions where the number of visible regions may vary depending upon body structure. A type V junction may have either one or two visible regions associated with it; that is, either one or both of the two visible regions, inside and outside the V, may correspond to the projection of a face. At a type T junction, one or two faces may be occluded. For a concave-curved line it is possible to have more than one label for the same line.

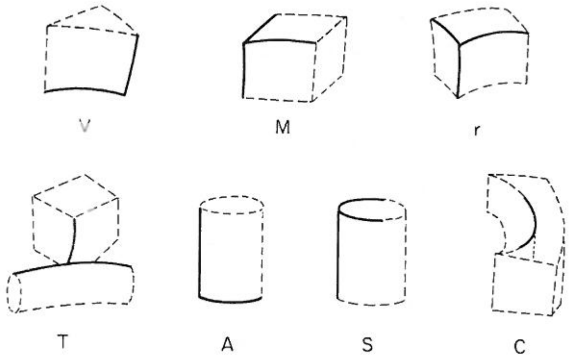


FIG. 2. The seven possible junction types. V: two lines (no limb line) meet at a junction. M: three lines (no limb line) meet at a junction and form two visible regions. r: three lines (no limb line) meet at a junction and form three visible regions. T: one line meets another line at somewhere between two ends of the second line segment. A: two lines (one limb line) meet at a junction and form one visible region. S: three lines (one limb line) meet at a junction and form two visible regions. C: the intersection of a concave-curved line with an invisible virtual edge.

To resolve this inconsistency, the type C junction is defined. This junction is formed by the intersection of a concave-curved line with an invisible virtual edge.

From the above definitions we know that the junction types are not specified in terms of rigid geometric characteristics only. For example, a type M junction can have a great deal of different forms with only the restriction that three lines meet at a vertex and form two visible regions. Similarly, a type r junction is restricted by the structure that three lines meet at a vertex and form three visible regions.

The labeling scheme includes junction labeling and line labeling. Junction labeling assigns to each junction in the image a label describing the junction type and the number of visible regions associated with it. The labels are of the form *N*-*a* where '*N*' could be 1, 2 or 3, which indicates the number of visible regions at the junction and *a* indicates the junction type, which could be any one in the set {A, S, M, r, V, T, C}. Fig. 3 shows an example of an object whose junctions are labeled.

In addition to the junction labeling, the lines must also be labeled. The line labeling is based on the Huffman and Clowes' rules [3, 6] extended to cover curved edges. It has:

(1) a label + for a convex edge;

(2) a label − for a concave edge;

(3) a label > for edges with one visible face, where the nearer occluding surface is on the right when looking along the arrow;

(4) a label < for the inverse situation of (3).

The correspondence between the junction types shown in Fig. 2 and the

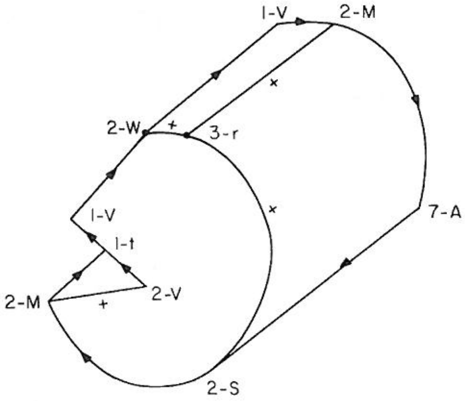


FIG. 3. An example of an object whose junctions are labeled.

junction types and line labels proposed by Huffman and Clowes are summarized in Fig. 4. Note that from now on the junction types of Huffman and Clowes are considered as the new types which are valid on both straight lines and curved lines.

In the labeling process, line-label information can provide some useful constraints. With the aid of these constraints, we show how the search process can be faster and more reliable compared with what Chakravarty did. Moreover, combining the junction-type and line-label information provides enough information for face grouping and the prediction of hidden vertices. This is discussed in Sections 5 and 6.

We close this section by describing how the new labeling scheme proceeds on the line drawing of Fig. 5.

(1) Collect all the junctions and lines of Fig. 5 into two sets *H* and *G* respectively and label all the boundary lines adjacent to the background region with a '>' or '<'. In this example, all the boundary lines 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 26 are labeled '>'.

(2) Label all the junctions associated with boundary lines and the interior lines which meet at these junctions. In this example, these are junctions 1(1-V), 2(2-M), 3(1-V), 4(1-V), 6(1-V), 8(2-M), 9(1-V), 11(2-M), 12(1-V), 13(2-S) and lines 14(+), 25(+), 21(+), 15(+), 14(+). The junctions and lines which can not be labeled so far are junctions 5, 7, 10, 14, 15, 16, 17, 18, 19, 20 and lines 4, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25.

(3) Search around the lines belonging to region 2 and label all the junctions and lines adjacent to them. In this example, the junctions 14(3-r), 15(2-V), 16(2-M), 17(3-r), 18(2-V), 19(2-M), 20(2-M) and the lines 16(+), 17(>), 18(>), 19(+), 20(+), 22(>), 23(+), 24(>) and 4(>) can be successfully labeled. When checking the line-label consistency between junctions, it is found that line 25 is labeled + at one end and labeled > at another end. In order to solve this

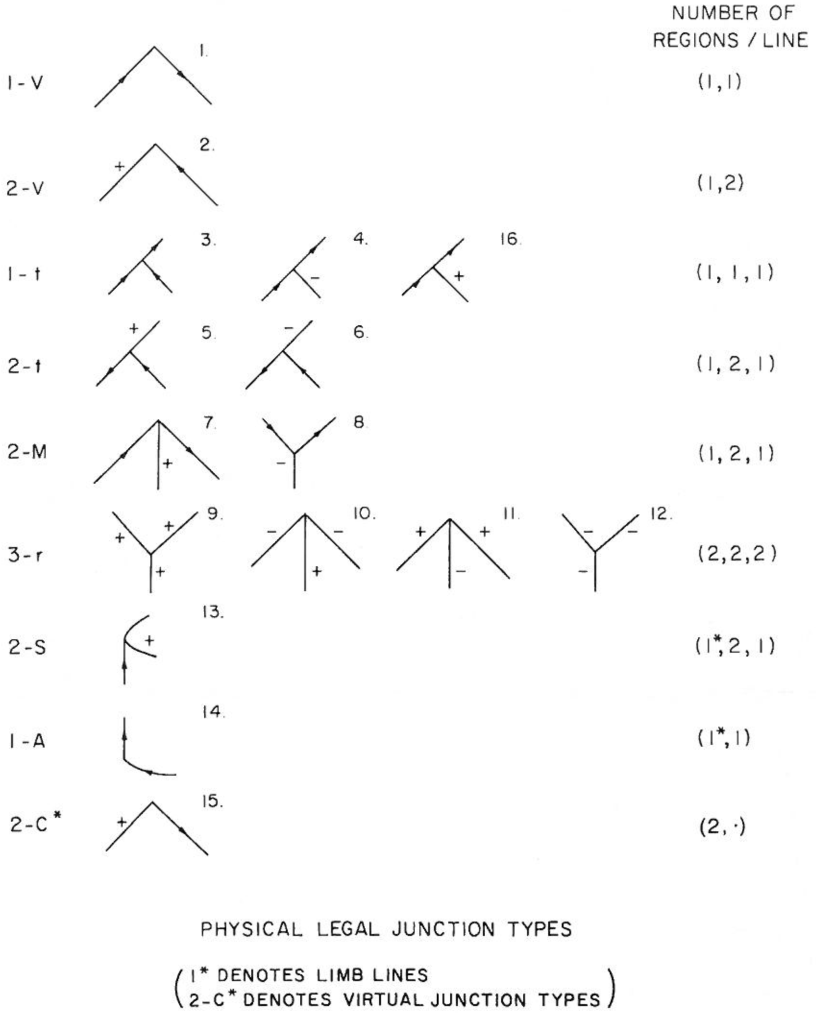


FIG. 4. The correspondence between the junction catalog and Huffman and Clowes and the one shown in Fig. 2.

inconsistency, one more virtual junction, junction 25, is introduced, which is labeled type 2-C. After this step, there are only three unlabeled junctions in set *H*. These are junctions 5, 7 and 10. Note that all the lines in set *G* have already been labeled.

(4) The same process as step (3) is applied to regions 4, 6 and 8 and junctions 10(1-t), 7(1-t) and 5(1-t).

From the above process, we can find that all of the junctions and lines of the example can be labeled with a few steps. There are only five search cycles.

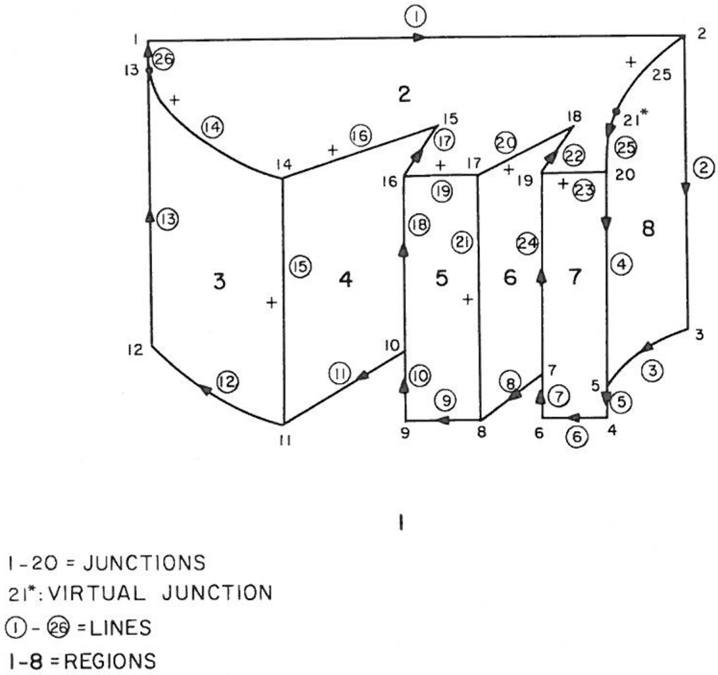


FIG. 5. A test graph of the search process.

For comparison, consider the Chakravarty scheme. It uses a table of permissible junction transitions when it transverses from one junction to another. In this example, if it starts from junction 1 which is labeled 1-V, and proceeds according to the allowable junction transitions, the adjacent junction, junction 26, will be labeled 2-S. Similarly, junctions 14 and 15 can also be uniquely labeled 3-r and 2-V, respectively. Junction 16 can be labeled 2-M or 3-r according to the allowable junction transition table. In either case junction 17 can be labeled 2-M or 3-r. Although Chakravarty proposed some additional criteria to check the whole search cycle, the use of these criteria will increase the computation complexity.

In this example which we believe to be representative, we find that the search process converges very fast. Most of the junction labels and line labels can be well determined by just searching around the region boundary which the vertices and edges belong to. In addition, using this method we can avoid most of the labels which correspond to the impossible objects of Chakravarty's method. The reason is that junction labels combined with line labels introduce more constraints than junction labels alone.

3. Grouping Body Faces into Volumes

A face is a portion of a surface bounded by edges or closed on itself. An image region is a connected part of the projection of a face. A fundamental problem in the analysis of pictures is the problem of partitioning a picture into regions which are understood as body faces and then associating together all the faces which enclose a volume. In this section, we discuss an approach to this problem. The approach is based on the junction labels and line labels obtained from the previous section.

One approach to the face grouping problem for planar objects was introduced by Duda and Hart [4] is based on the following rules:

(1) A Y vertex gives evidence that the three image regions which touch it correspond to three faces which enclose the same volume.

(2) A W vertex gives evidence that the two regions included between the narrow angles of it correspond to two faces which enclose the same volume.

(3) Two collinear T vertices give evidence for grouping together the faces corresponding to the regions on the same side of the stem of the T's; if a single region lies on the same side of both stems, the evidence is disregarded.

A graph illustration of the rules is shown in Fig. 6. While the rules can work well in many cases, it can also be easily tricked. Fig. 7 shows some troublesome scenes for these rules.

In our method, we take into consideration not only junction types but also the line labels of the vertices. As a result not all faces corresponding to regions meeting at Y or W vertices are grouped. Furthermore, we are also able to group faces corresponding to regions meeting at some S-type, T-type and V-type vertices. All the legal junction types and line labels for the purpose of grouping are shown in Fig. 8. Note that the only edges adjacent to the grouped faces are +-labeled edges. Therefore, the grouping rules can be simplified:

> The regions adjacent to +-labeled edges of W-type, V-type, S-type, V-type and stems of T-type vertices correspond to faces which enclose the same body volume.

By using this simple but powerful rule, we can correctly group all the troublesome scenes shown in Fig. 7 (see Fig. 9). Furthermore, this method can take care of objects with curved surfaces as well.

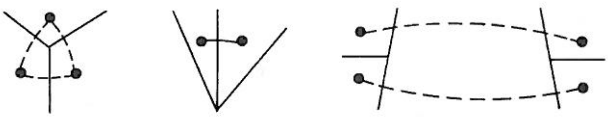


FIG. 6. A graph illustration of Duda and Hart's grouping rules. The three regions corresponding to a type Y junction are grouped. Similarly, the two regions of the type W junction and the two-region pairs of the type T junction are grouped.

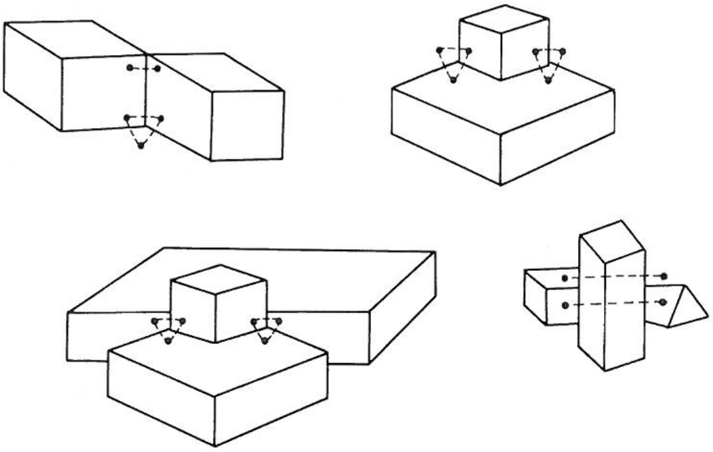


FIG. 7. Some troublesome scenes of Duda and Hart's grouping rules.

To illustrate the grouping strategy, we proceed by means of an example. Fig. 10(a) shows a simple scene consisting of one small object resting on another object. An inspection of the vertices of the labeled graph results in the link information given in Fig. 10(b), where we represent the regions of the picture by the nodes of a graph and the links between regions as arcs connecting the nodes. For clarity, each of the arcs (links) has been labeled with the vertex which it is derived from. For example, the link between nodes (regions) 1 and 3 results from the fact that vertex F has the Y-configuration and concommitant links of Fig. 10(a).

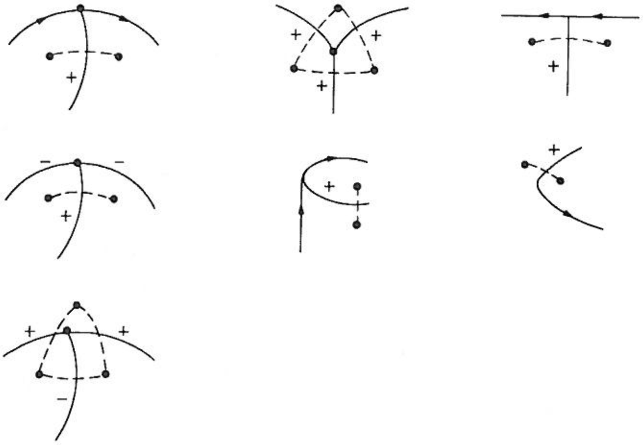


FIG. 8. All the legal types and line labels for the purpose of grouping.

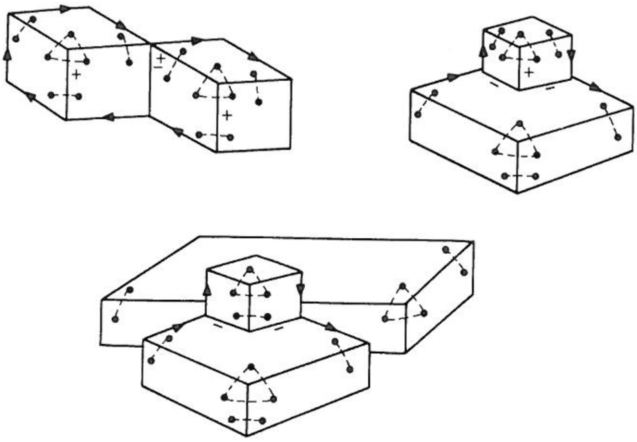


FIG. 9. The graphs shown give correct face grouping with the new grouping mechanism. They cannot be grouped well when Duda and Hart's rules are applied.

By an examination of Fig. 10(b) we can find that a single link should not be considered a strong enough evidence for irrevocably grouping together the two linked regions as part of the same object; region 1 and 4, which are 'clearly' part of separate objects, are linked. To avoid this difficulty, we will adopt the following heuristic rule for grouping together nodes (regions)

> Two nodes are to be grouped if there are at least two links between them. Any links from these two nodes to other nodes remain connected to the newly formed 'group node'.

Fig. 10(c) illustrates the application of this rule to the graph of Fig. 10(b). There are several pairs of double-linked nodes in Fig. 10(b); we have arbitrarily applied the rule to node 2 and 3 and to node 5 and 6. Notice how, in accordance with the second part of the rule, the links from the remaining nodes are attached to the new group nodes. Since the graph of Fig. 10(c) contains nodes linked by at least two arcs, the same heuristic rule can be applied again. Fig. 10(d) gives the result of this application; there are now no nodes connected by at least two arcs so the process terminates. The final grouping of regions is defined by the nodes of the final graph. In this case regions 1, 2, and 3 are grouped together as one object; region 4, 5, and 6 form another object; and region 7, the background, forms a third object. In this example, then, the process of first finding links and then merging nodes yields a satisfactory result.

While the method presented above works satisfactorily for most of the scenes, it is not perfect. It can not group the faces which belong to the same body when their corresponding regions are separated by other objects and are not adjacent in the two-dimensional perspective image, such as the one shown in Fig. 11.

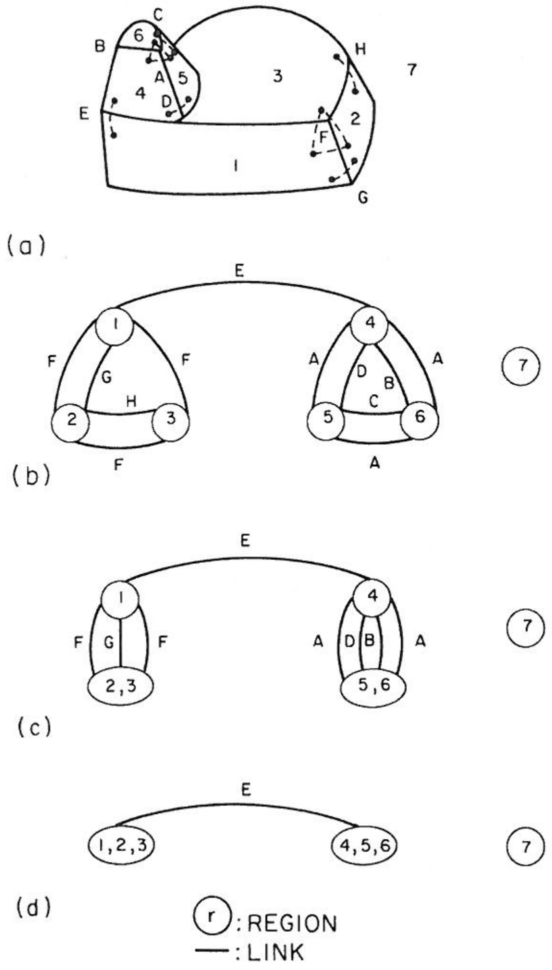


FIG. 10. An example of face grouping. The nodes represent regions; the arcs represent links between regions. On the basis of (d), we can find that there are two objects and one background.

As we can easily find out from the figure, the missing evidence is caused by object overlapping. That is, some useful vertices are hidden and some evidence is lost. Thus, if we could predict the hidden vertices of overlapped objects, then, this kind of trouble could be easily solved. The technique of predicting hidden vertices is discussed in Section 5. Before proceeding to this discussions, we generalize the labeling technique to higher-order vertices than trihedral.

Two images which can be well grouped by means of our method are shown in Fig. 12.

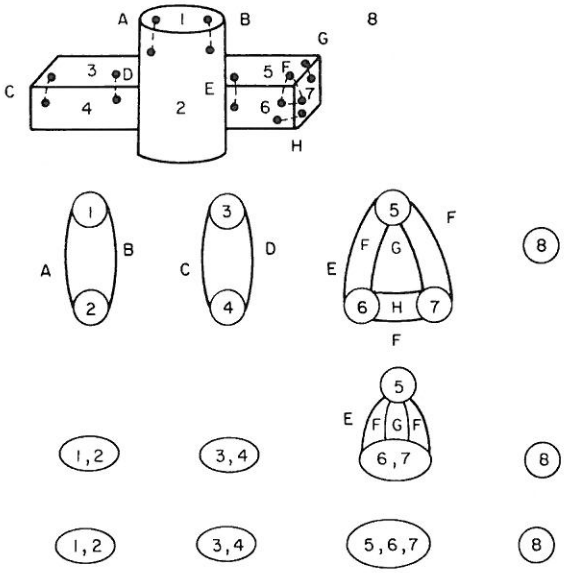


FIG. 11. An example of incompleteness of the grouping strategies. It is easy to find that the group composed of regions 3 and 4 and the group consists of regions 5, 6 and 7 should be grouped together.

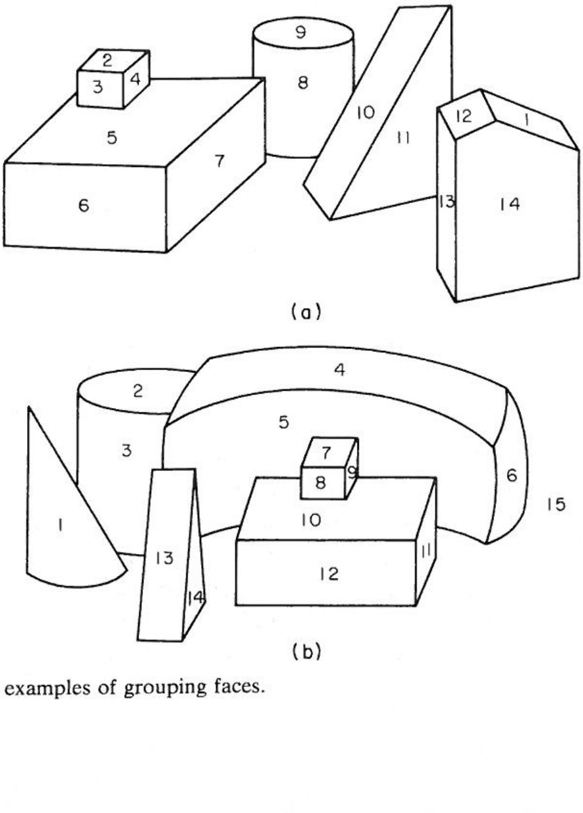


FIG. 12. Two test examples of grouping faces.

4. Label Multi-Type Junctions by Means of Trihedral Vertices

There are two major advantages which can be gained from region groupings. One is that we can obtain context information from each region group of an image. The other advantage of region grouping is that by combining the region-grouping process with the labeling process, we can label most of the multi-type junctions by means of trihedral vertex concepts only. This creates a simple and fast multi-type junction-labeling scheme.

We will discuss here seven commonly used multi-type junctions. These junctions include peak, K, X, multi, XX, KA and KX as shown in Fig. 13. The labeling process is sketched as follows:

(1) Label all the edges adjacent to trihedral vertices. The procedure is the same as the one described in Section 2 of this paper.

(2) If all the edges have been successfully labeled, exit with success. If not, go to the next step.

(3) Group regions according to the method introduced in Section 3. In this case, the two faces adjacent with a +-labeled multi-type edge are considered to come from faces which enclose the same volume.

(4) Label edges of group boundaries. The strategy is to consider only the edges belonging to the same group at a time. Therefore, we can separate a multi-type vertex, divide it into a set of trihedral vertices which belong to different volumes. The labeling of the edges of the multi-type vertex can be done first by labeling independently each volume consisting of trihedral vertex partitions and then combining all the labels together. During the labeling of each trihedral vertex, it can happen in some cases that there is more than one possible boundary label. The way of solving this ambiguity is to consider the label of the edges which belong to the multi-type vertex as >- or <-labels first; if this is not acceptable, then consider them as +-labels and then as −-labels. It is also possible that in some cases there is no consistent labeling for some of the trihedral vertex partitions. The reason is that some of the edges in the trihedral

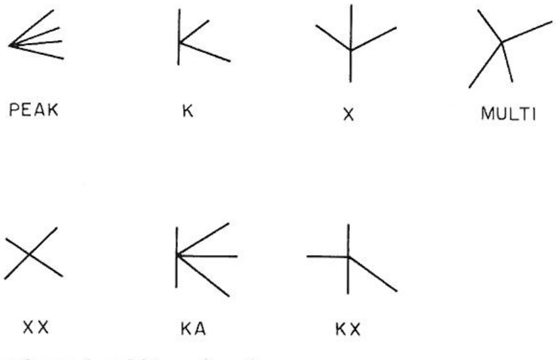


FIG. 13. The commonly used multi-type junctions.

vertex partition are formed by the overlapping of the other volumes and do not exist in reality. Thus, the separations of these trihedral vertices are inproper. The way of solving this situation is skipping this redundant trihedral vertex labeling from the labeling of the edges of the multi-type vertex.

(5) If all the edges have been successfully labeled, exit with success. If not, exit with fail.

Some examples of this universal labeling scheme are shown in Fig. 14. Fig. 14(a) shows an example of labeling a 'peak'-type vertex. In this example, it happens that all the edges are labeled by labeling trihedral vertices only. The reason is that all regions of this image belong to the same object. Fig. 14(b) is

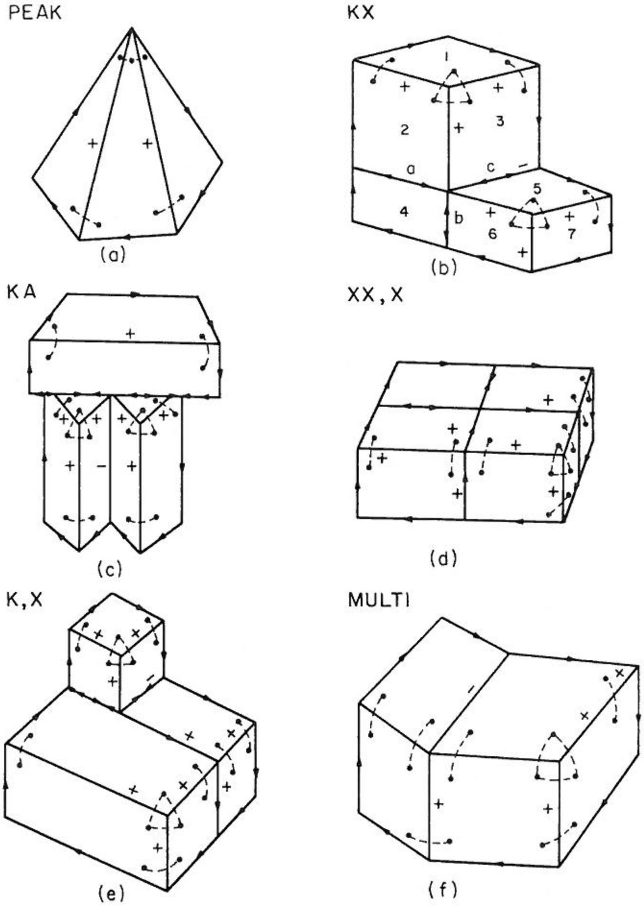


FIG. 14. Examples of a universal labeling method applied to peak-, KX-, KA-, XX-, X-, K- and multi-type vertices.

an example of labeling a 'KX'-type vertex. In this example, after labeling the trihedral vertices, there are two edges, *a* and *b*, which can not be labeled. According to the labeled edges, the regions are divided into three groups. One is composed of region 1, 2, and 3; the second is region 4; and the third group is composed of region 5, 6 and 7. In the process of labeling group boundaries, edges *b* and *c* will be each labeled '>' by a 'W'-type vertex of group 5, 6, 7; edges *a* and *c* will be labeled '<' by a 'W'-type vertex of group 1, 2, 3; edges *a* and *b* will be labeled '>' by a 'V'-type vertex of group 4. In this example all the edges *a*, *b* and *c* are double labeled. This is allowable and it means that more than one object meets at these edges. Examples of labeling other multi-type vertices are also shown in Fig. 14.

5. Prediction of Hidden Vertices

In general, not all the objects of an image are fully visible. That is, the occlusion of one object by another is a natural extension of the single-object situation. In this section, we introduce a method of predicting hidden vertices for such overlapped objects.

By observing the pictures of overlapped objects, such as the pictures of Fig. 12, we find that at the overlapped portions, where the edges of an object are broken by other objects, some T-type vertices are formed. Actually, all the predictions of the hidden trihedral vertices can be made on the basis of edges which are stems of T-type vertices.

Before going into the prediction step, all the T-type stem pairs which are separated by other objects but lying on the same line must be connected. When connecting this kind of T-type stem pairs, the edges of each pair will be considered as a single edge. Therefore, the connected T-type pairs can be disregarded, and the regions on the same side of the stem pair can be grouped together. An example of this process is shown in Fig. 15. It is easy to find from the figure that junctions *a* and *b* which are stems of T-type vertices lie on the same line. According to above rule, we will connect *a*, *b*, give a link of grouping between group 1 and group 2, and another link between group 4 and group 5. Similarly, the process is applied to junction pairs *cd*, *ef*, *gh*, *ij* and *kl*. Finally the twelve junctions are disregarded and the groups 1, 2 and 3 are merged as one region group only. It is easy to find that this process can solve most of the incompleteness of groupings as shown in Fig. 11. After this preliminary work, we can start the prediction process.

The rules for predicting the hidden vertices corresponding to the edges which are stems of T-type vertices are based on the assumption of symmetry and can be summarized as follows:

Rule 1. For a '+'-labeled T-type vertex stem, if the other end of the edge, named *d*, is a W-type vertex, there should be a corresponding hidden Y-type vertex; if the other end of the edge is a Y-type vertex; there should be a

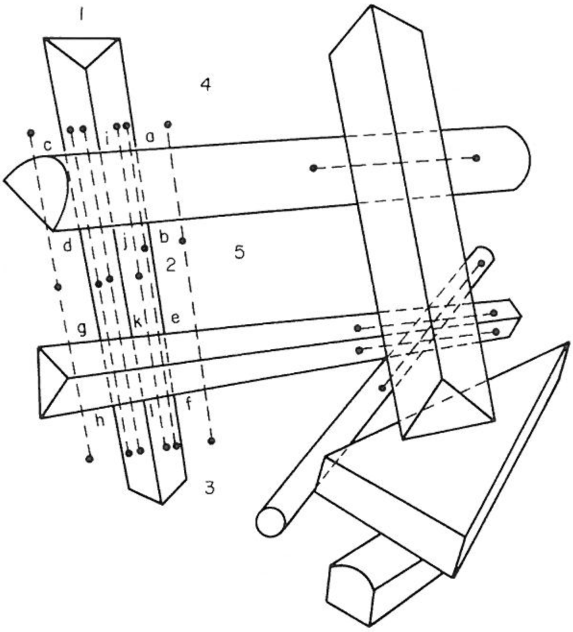


FIG. 15. An example of the preliminary work.

corresponding hidden W-type vertex; if the other end of the edge is an S-type vertex; there should be a corresponding hidden S-type vertex.

A graph illustration and some graph examples of this rule are shown in Fig. 16.

Rule 2. For a '−'-labeled T-type stem, if the other end of the edge is a W-type vertex, there is a hidden Y vertex; if the other end of the edge is a Y-type vertex, there is a hidden W-type vertex. A graph illustration and some examples of this rule are shown in Fig. 17.

Rule 3. For the '>' or '<'-labeled T-type vertex stems, we will consider one pair of these edges at a time, each pair of these edges includes one edge labeled '>' and another edge labeled '<'. The rule can be further subdivided as follows:

(i) There is no '+' or '−'-labeled T-type stem located between a pair of '>'-labeled stems.

(a) The minimum number of edges across the two other ends of the '>' and '<'-labeled edges is one. In this case, if the vertex corresponding to one end of a stem, named *d*, is an A-type vertex, there is a corresponding hidden S-type vertex; if *d* is an S-type vertex, there is a corresponding hidden A-type vertex.

(b) The minimum number of edges across the other two ends of the '>'- and '<'-labeled edges is greater than one. In this case, if either of the other ends of a stem is an S-type vertex, there should be a hidden A-type vertex. Otherwise,

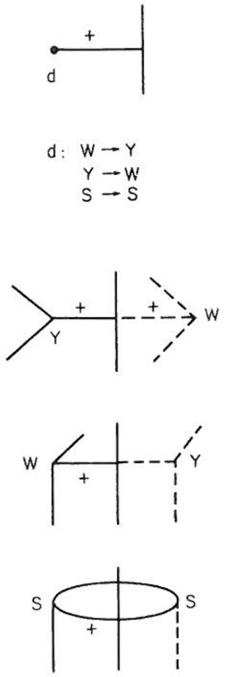


FIG. 16. A graph illustration of Rule 1. The mappings between visible and hidden vertices in this case are W to Y, Y to W and S to S.

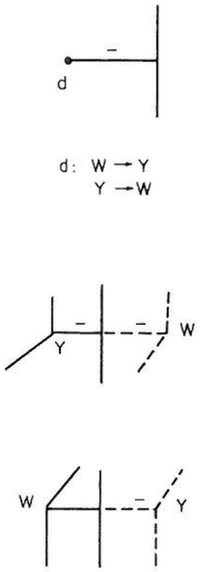


FIG. 17. A graph illustration of Rule 2. The mappings between visible junctions and hidden junctions are W to Y and Y to W.

there should be a hidden V-type vertex. A graph illustration and some examples of these rules are shown in Fig. 18.

(ii) There is at least one '+'- or '−'-labeled T-type stem located between a pair of '>'- and '<'-labeled stems.

(a1) If a '>'- or '<'-labeled edge and its adjacent '+'- or '−'-labeled T-type stems share the same *d* vertex (the other end of the stem of a T-type vertex), there should be one V-type hidden vertex and one Q vertex (the Q vertex could be either a W- or a V-type vertex and can be determined by Rules (ii)–(b1) and (ii)–(b2). A graph illustration of this situation is shown in Fig. 19.

(a2) If the vertices *d* of a '>'- or '<'-labeled edge and its adjacent '+'- or '−'-labeled T-type stems are adjacent, there should be one Q vertex. A graph illustration of this situation is shown in Fig. 20.

(a3) Otherwise, there should be no Q vertices. A graph illustration of this situation is shown in Fig. 21.

(b1) If the adjacent T stem comes from a W-type vertex, then, the Q vertex should be a W-type vertex and there should be one more V vertex. A graph which illustrates this situation is shown in Fig. 22.

(b2) If the adjacent T stem comes from a Y-type vertex, then, the Q vertex

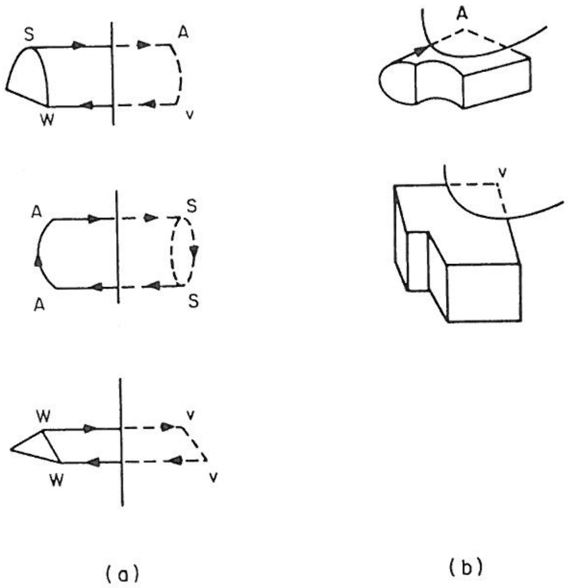


FIG. 18. A graph illustration of Rule 3(i). In (a) there are two hidden vertices. While, in (b) there is only one hidden vertex.

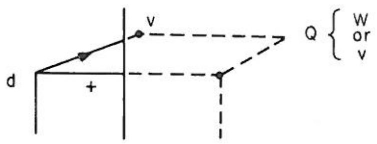


FIG. 19. A graph illustration of Rule 3(ii)–(a1). In this case one V-type and one Q-type hidden vertex correspond to vertex *d*.

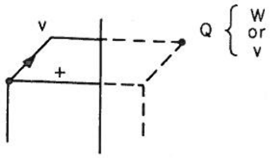


FIG. 20. A graph illustration of Rule 3(ii)–(a2). In this case one Q vertex corresponds to vertex *d*.

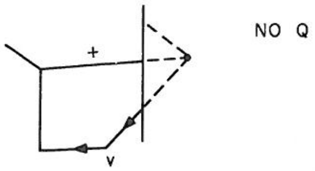


FIG. 21. A graph illustration of Rule 3(ii)–(a3).

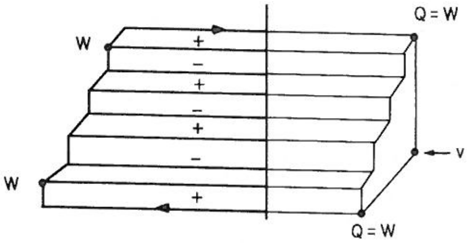


FIG. 22. A graph illustration of Rule 3(ii)–(b1). The Q vertex in this case is W-type.

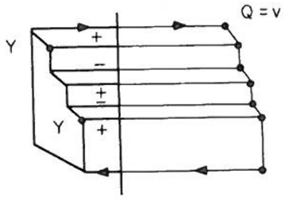


FIG. 23. A graph illustration of Rule 3(ii)–(b2). The Q vertex in this case is V-type.

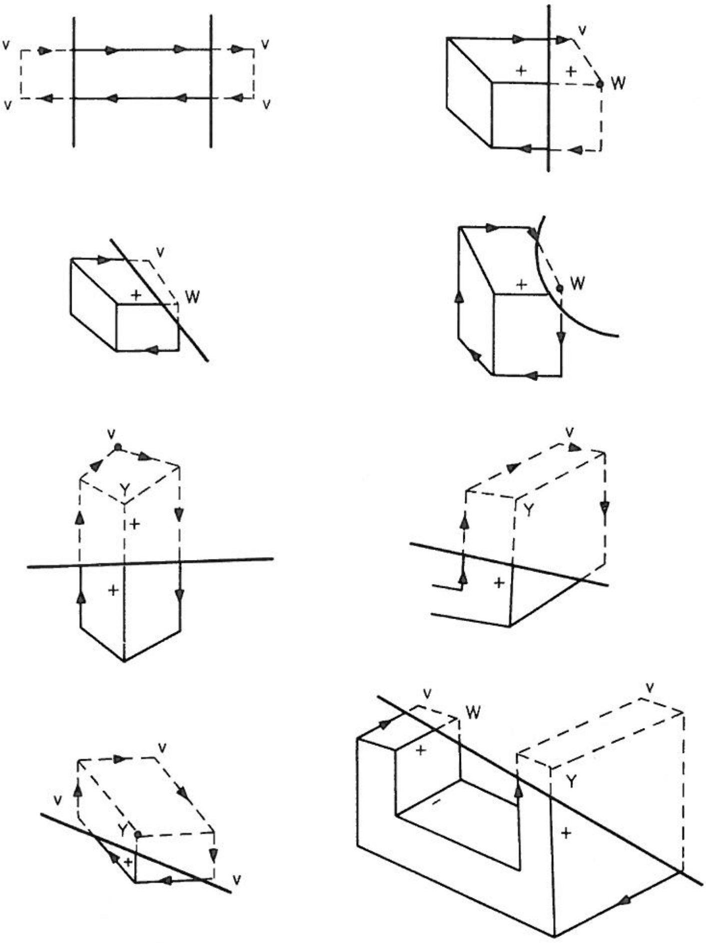


FIG. 24. Some simple examples of hidden vertices which can be successfully solved by the predicting rules.

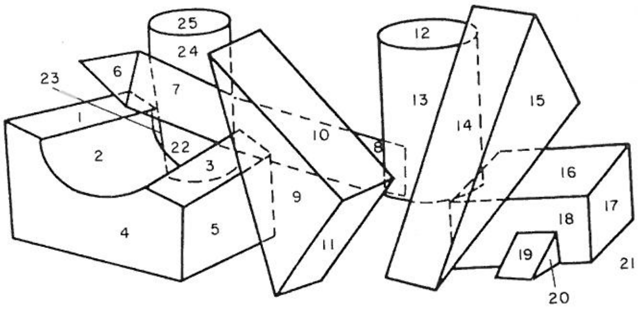


FIG. 25. A test example of the predicting rules. In this example all the hidden vertices (dashed portions) can be correctly predicted by the predicting rules.

should be a V-type vertex. A graph illustration of this situation is shown in Fig. 23.

Using the above rules, we can successfully predict most of the hidden vertices. Furthermore, as we have mentioned before, the hidden-vertex information obtained in this step can be used as an aid to group regions which correspond to body faces which enclose the same volume.

Fig. 24 illustrates some overlapped objects which can be predicted very well according to these rules. A test example and its result (dashed line portions) are shown in Fig. 25.

6. Context-dependent Hidden Vertex Prediction Method

The rules introduced in Section 5 are based on the assumption that objects are symmetric. However, the situation in the real world is more complex than what can be handled by symmetric assumptions only. Figs. 26–28 show many ambiguous cases for the above rules. Figs. 26(a) and 28(a) show that different view directions may cause different hidden vertices even though both cases follow the symmetry assumption. Fig. 28(c) shows that the higher the number of hidden vertices, the higher the degree of ambiguity.

In order to alleviate this ambiguity, we introduce a context-dependent predicting method. This method uses local vertex information and also global information extracted from all the objects of the same image or a set of similar images. The context information includes:

(1) The number of vertices of each region in the same group.

(2) The ordered junction-type sequence of each region.

(3) The edge type between each vertex pair, each edge being either a straight (s) or curved (c) line.

(4) The normalized length of each edge, where the normalization is applied to all edges belonging to the same object volume and makes the longest edge have unit length.

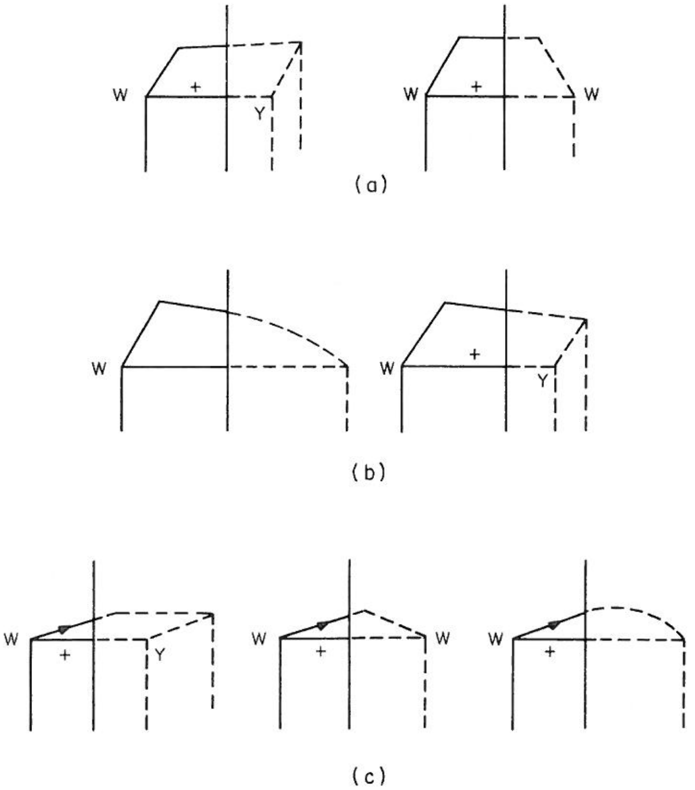


FIG. 26. Ambiguous case 1. For a '+'-labeled T-type stem of which one end meets a W-type vertex, there are many kinds of possible combinations of hidden vertices.

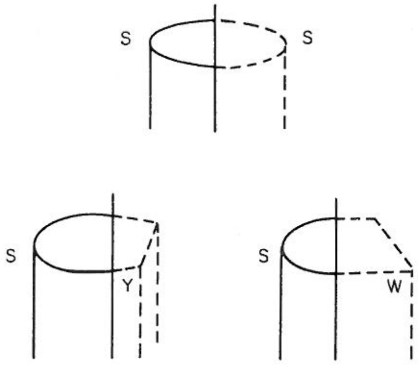


FIG. 27. Ambiguous case 2. For a '+'-labeled T-type stem of which one end meets an S-type vertex, there are three possible combinations of hidden-vertices combinations. (a) This case follows the symmetric assumption; (b) and (c) are basically the same kind of body with a different view direction.

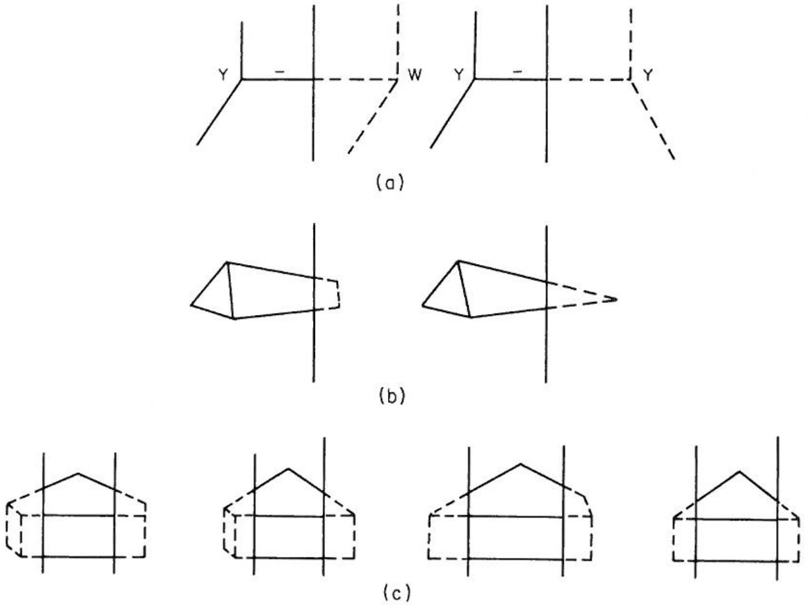


FIG. 28. Ambiguous case 3. Although both objects in (a) follow the symmetric assumption, different view directions will cause difference in hidden vertices. Only one object of (b) follows the symmetric assumption but both are possible objects. From (c) we can find that the higher the number of hidden vertices, the higher the degree of ambiguity.

An example of the context data format is shown in Fig. 29. The context data of the shown object is:

⟨3, 4, 4, 7 |
W,(s,0.3),T,(s,0.4),V,(s,0.4);
W,(s,0.5),V,(s,0.5),W,(s,0.5),Y,(s,0.5);
W,(c,0.7),A,(s,0.7),S,(c,0.8),Y,(s,0.5);
W,(s,0.4),V,(s,0.5),W,(s,0.5),Y,(c,0.8),S,(c,0.3),V,(s,1.0),V,(s,0.3)⟩.

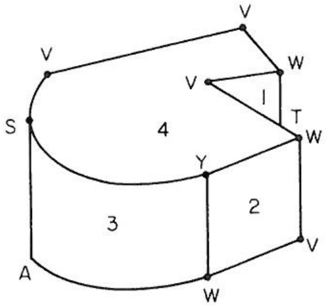


FIG. 29. An example of the context data format of an image.

Here the sequence of numbers 3, 4, 4, 7 means that there are four regions in the same group and the number of vertices in each region are 3, 4, 4 and 7, respectively. The first string W,(s,0.3),T,(s,0.4),V,(s,0.4) means that in region 1 the clockwise vertex sequence is W,T,V or clockwise rotations of W,T,V; W,(s,0.3),T means that the edge between vertices W and T is a straight line with normalized edge length 0.3; T,(s,0.4),V means that the edge between vertices T and V is a straight line with normalized edge length 0.4 and V,(s,0.4) means that the edge between vertices V and W is a straight line with normalized edge length 0.4. The rest of this context data includes the strings which give the information of the other three regions with the same format as that of the first region.

The way of applying the context-dependent data in the predicting process can be summarized as follows:

(1) Apply the vertex-prediction method described in Section 5 to all the unambiguous regions and skip all the region groups which have more than one labeling possibilities.

(2) Extract the context data from the region groups on which the prediction has already been successfully applied.

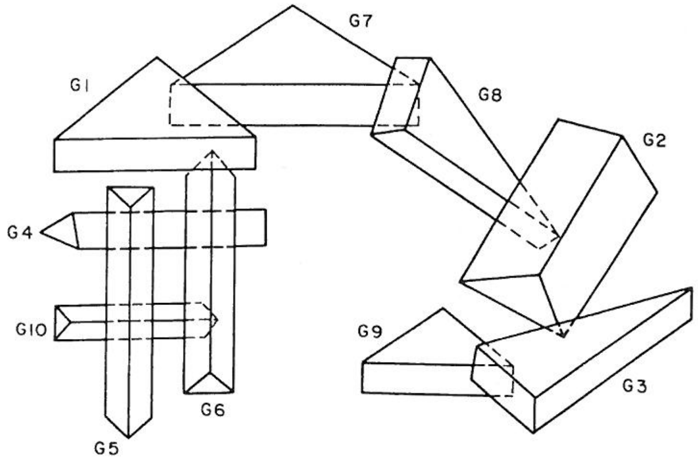
(3) Obtain the test data (which has the same format as the context data) of all the possible vertex combinations of each ambiguous group.

(4) Consider the consistency between the vertex number and junction sequence of the test data corresponding to each possible vertex combination of the ambiguous region groups and the context data obtained from the unambiguous region groups. If none of the test data of the same group matches any of the context data, choose the one which has less deviations; if more than one of the test data matches some of the context data, choose the one which has higher frequency of matching; if there are more than one test data which have same frequency of matching, choose the one which follows the symmetric assumptions.

(5) Match the edge type and normalized edge length information included in the context data with the selected best test data to estimate the edge types and the normalized edge lengths of the edges adjacent to the predicting vertices.

Fig. 30 shows an example of applying the context-dependent predicting method. In this example, there are six available unambiguous region groups, labeled G1 to G6, and four ambiguous region groups, labeled G7 to G10. The context-dependent data and the selected best test data are shown in the figure. The predicted results for the ambiguous region groups are shown as the dashed line portions in this figure.

One valuable feature of this method is that by considering the ratios of normalized edge lengths, we can predict not only hidden vertices but also the length of the edges adjacent to the hidden vertices. As shown in Fig. 30, in G8 the normalized length of the predicted edges *a*, *b* and *c*, according to the correspondence between the normalized edge lengths of G8 and G3, are



```
G1 < 3,4 : W,(S,0.75),V,(S,0.75),W,(S,1.0);W,(S,1.0),W,(S,0.2),V,(S,1.0),V,(S,0.2)>
G2 < 3,4,4: W,(S,0.3),Y,(S,0.3),W,(S,0.4);W,(S,1.0),V,(S,0.3),W,(S,0.8),Y,(S,0.3);
            W,(S,0.3),Y,(S,0.8),W,(S,0.3),V,(S,1.0)>
G3 < 3,4,4: W,(S,1.0),W,(S,0.9),Y,(S,0.4);W,(S,0.4),Y,(S,0.15),W,(S,0.4),V,(S,0.15);
            W,(S,0.15),Y,(S,0.9),W,(S,0.15),V,(S,0.9)>
G4 < 3,4 : W,(S,0.15),W,(S,0.2),V,(S,0.15);W,(S,1.0),V,(S,0.2),V,(S,1.0),W,(S,0.15)>
G5 < 3,4,4 : W,(S,0.15),W,(S,0.12),Y,(S,0.12);W,(S,0.12),Y,(S,1.0),W,(S,0.12),V,(S,1.0);
            W,(S,1.0),V,(S,0.12),W,(S,1.0),Y,(S,0.12)>

G6 = G5
G7 = G1
G8 = G3
G9 = G1
G10 = G6
```

FIG. 30. An example of applying the context-dependent prediction method.

predicted as 1.0, 0.9 and 0.9 respectively. Similarly, the normalized length of the predicted edges of G7, G9 and G10 can also be easily predicted according to the correspondence between G6 and G10 as well as the correspondence between G1 and both G7 and G9. Given an image the prediction of the normalized lengths make it possible to reconstruct both fully visible and part invisible objects. Another valuable feature is that the data format provides a possible way of identifying objects by means of data matching.

The context-dependent vertex prediction is not guaranteed to work in all cases. However, we believe that in most of the cases the degree of ambiguity will be greatly reduced.

7. Conclusions

The perceptional performance by a computer is still a major research problem, even though it can be trivially performed by human beings. The aim of this paper has been to develop some general rules for computer vision systems which can perceive objects in line-drawing images. Specifically, we have described an algorithm for labeling edges of curved-surface objects. Using the label information, we can group together regions which belong to the same objects and predict the types of the hidden vertices. Although the method starts with trihedral vertex objects, it is good for most of the objects which possess multi-type vertices.

REFERENCES

1. Chakravarty, I., A generalized line and junction labeling scheme with applications to scene analysis, *IEEE Trans. Pattern Anal. Machine Intelligence* **1** (2) (1979) 202–205.
2. Freeman, H. and Chakravarty, I., The use of characteristic views in the recognition of three-dimensional objects, in: E.S. Gelsema and L.N. Kanal (Eds.), *Pattern Recognition in Practice* (North-Holland, Amsterdam, 1980) 277–288.
3. Clowes, M.B., On seeing things, *Artificial Intelligence* **2** (1) (1971) 79–112.
4. Duda, R.O. and Hart, P., *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
5. Guzman, A., Computer recognition of three dimensional objects in a visual scene, MAC-TR-59 (Thesis), Project MAC, MIT, Cambridge, MA, 1968.
6. Huffman, D.A., Impossible objects as nonsense sentences, in: B. Meltzer and D. Michie (Eds.), *Machine Intelligence* (Edinburgh University Press, Edinburgh, 1971) 295–323.
7. Kanade, T., A theory of origami world, Tech. Rept. CMU-CS-144, Carnegie-Mellon University, Pittsburgh, PA, 1978.
8. Turner, K.J., Computer perception of curved objects using a television camera, Ph.D. Dissertation, Edinburgh University, Edinburgh, Scotland, 1974.
9. Waltz, D.L., Understanding line drawings of scenes with shadows, in: P. Winston (Ed.), *The Psychology of Computer Vision* (McGraw-Hill, New York, 1975).

Received October 1983