MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence
Memo No. 204 July 1970

EXTENDING GUZMAN'S SEE PROGRAM

Martin Henry Rattner

Adolfo Guzman's SEE program groups the regions of a two-dimensional scene into bodies, using local evidence in the scene to link regions together. This paper discusses an extended version of the SEE procedure that makes extensive use of evidence in the scene which indicates that two regions should be split into separate bodies.

The new procedure is better in several ways: 1) it correctly analyzes many scenes for which SEE makes mistakes; 2) it can interact with a higher-level object-recognizing program; 3) it can provide alternative solutions on demand.

This paper's original form was a B.S. thesis supervised by P.H. Winston.

* * *

Work reported herein was supported by the Warren McCulloch Laboratory, an M.I.T. research program sponsored by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract number NOOO14-70-A-0362-0002.

TABLE OF CONTENTS

1 Introduction 5

1.1 Computer vision 5

1.2 Possible improvements to SEE 10

1.3 SEEMORE: an extension of SEE 12

1.4 A note on nomenclature 14

2 The basic SEEMORE procedure 18

2.1 Phase 1: generate initial splits and links . . 18

2.2 Phase 2: generate other plausible splits . . . 30

2.3 Phase 3: link and conglomerate 43

3 Producing Alternatives 51

3.1 The nature of a complaint 51

3.2 The alternative-list 51

3.3 The processing of a complaint 56

4 Some examples 59

4.1 A detailed example 59

4.2 Other examples 64

5 Failures of SEEMORE and recommendations 68

1 Introduction

1.1 Computer vision

For the last several years, the Artificial Intelligence Group of M.I.T.'s Project MAC has been seeking to develop more complex and powerful ways for computers to interact with their environment. One of the products of this research is a primitive scene-perceiving system, which is represented schematically in the flow diagram of figure 1-1.

1.1.1 Pre-processing

The system begins with an actual scene, from which all information ultimately derives. The eye of the system is a television camera specifically designed to interact with a computer. A program developed by Griffith [1] processes the output of the camera to produce a line drawing of the scene. This line drawing, appropriately corrected and encoded, is used as the representation of the scene itself by all higher-level programs in the system.

1.1.2 Processing of the line drawing

Much of the following description of the MAC vision system is taken from Patrick H. Winston's Ph.D. thesis [2].

1.1.2.1 Classification of vertices

A program written by H. N. Mahabala [3] classifies and labels the vertices in the scene according to the number of converging lines at each vertex and the angles between them. Figure 1-2 displays the vertex types which are recognized. Notice that pairs of Ts with

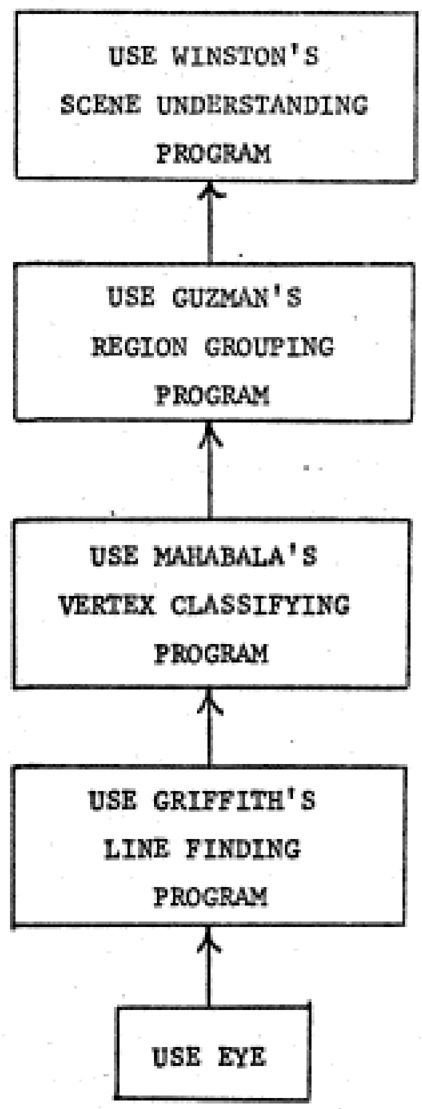

USE WINSTON'S
SCENE UNDERSTANDING
PROGRAM
USE GUZMAN'S
REGION GROUPING
PROGRAM
USE MAHABALA'S
VERTEX CLASSIFYING
PROGRAM
USE GRIFFITH'S
LINE FINDING
PROGRAM
USE EYE

Figure 1-1

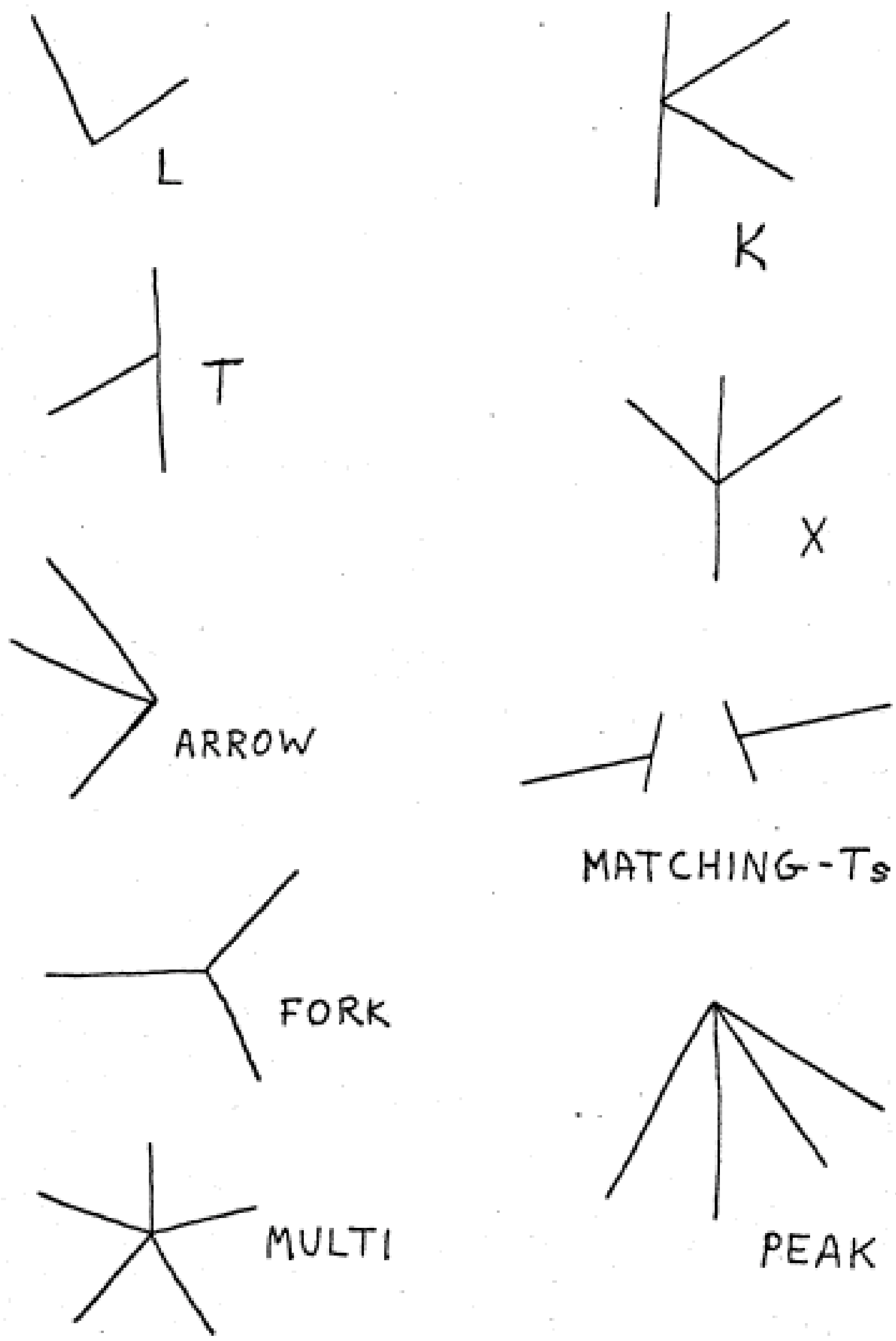


Figure 1-2

crossbars lying between collinear uprights are found by Mahabala's program. These are called matching-Ts.

The program then proceeds to create names for all of the regions in the scene. Rigorously, "region" as used here simply refers to any maximal area in which one can move from any point to any other point without crossing a line. For example, figure 1-3 has eight regions, not counting the background. Various properties are calculated and stored for these regions. Among these are a list of the vertices and bordering regions which surround each region.

1.1.2.2 The SEE procedure

These results are then supplied to the program named SEE developed by A. Guzman [4]. This program conjectures about which regions belong to the same objects. For figure 1-3, the end result of the program is the commentary:

Body 1 consists of A B C

Body 2 consists of D E F G H

Surprisingly the program contains no explicit models for the objects it expects to see. It simply examines the vertices and uses the vertex classifications to determine which of the neighboring regions are likely to be part of the same object. ARROWs, for example, strongly suggest that the two narrow-angle regions belong to the same body (figure 1-4). This evidence is represented by SEE as a strong link placed between the regions which are suspected of belonging to the same body. Figure 1-5 depicts a link which has been placed between regions 1 and 2 as a result of the evidence provided by the

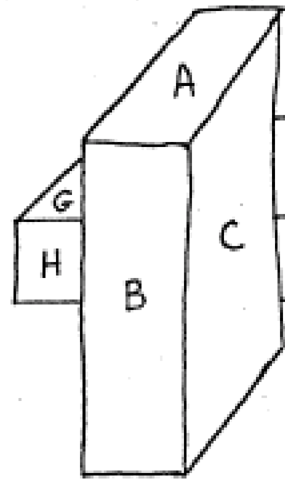


Figure 1-3

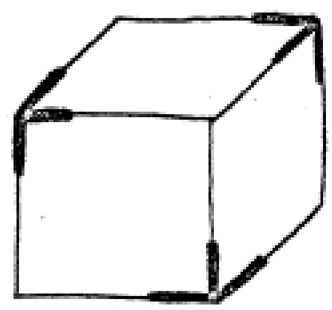


Figure 1-4

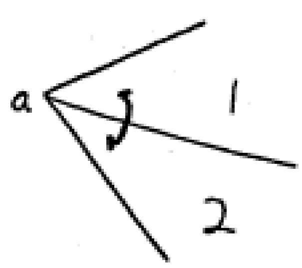


Figure 1-5

ARROW a.

Since links are often generated between regions belonging to different bodies, SEE is not satisfied with the evidence represented by a single link; two or more links are required to bind two regions together into a single nucleus. A nucleus represents a set of regions which will eventually be reported as belonging to a single object in the scene. Two or more links between a pair of nuclei will similarly result in their conglomeration into a still larger nucleus. The nuclei grow in this fashion until no two of them can be combined by a double link; they are then called maximal nuclei. SEE then uses weaker evidence in the form of weak links to further merge some nuclei. Finally, special heuristics are employed to attempt to eliminate nuclei consisting of a single region by combining them with other nuclei.

This fairly sophisticated procedure can sort out the regions in scenes as complicated as that in figure 1-6, borrowed from Guzman's thesis. Twelve objects are reported and the regions of each are remembered.

1.1.2.3 Higher-level analysis

Winston has developed and implemented procedures which use the output of SEE to create network-structured descriptions of scenes. These descriptions are used by Winston's programs for comparisons, concept learning, and identification.

1.2 Possible improvements to SEE

Refer again to the flow diagram of figure 1-1. Winston has

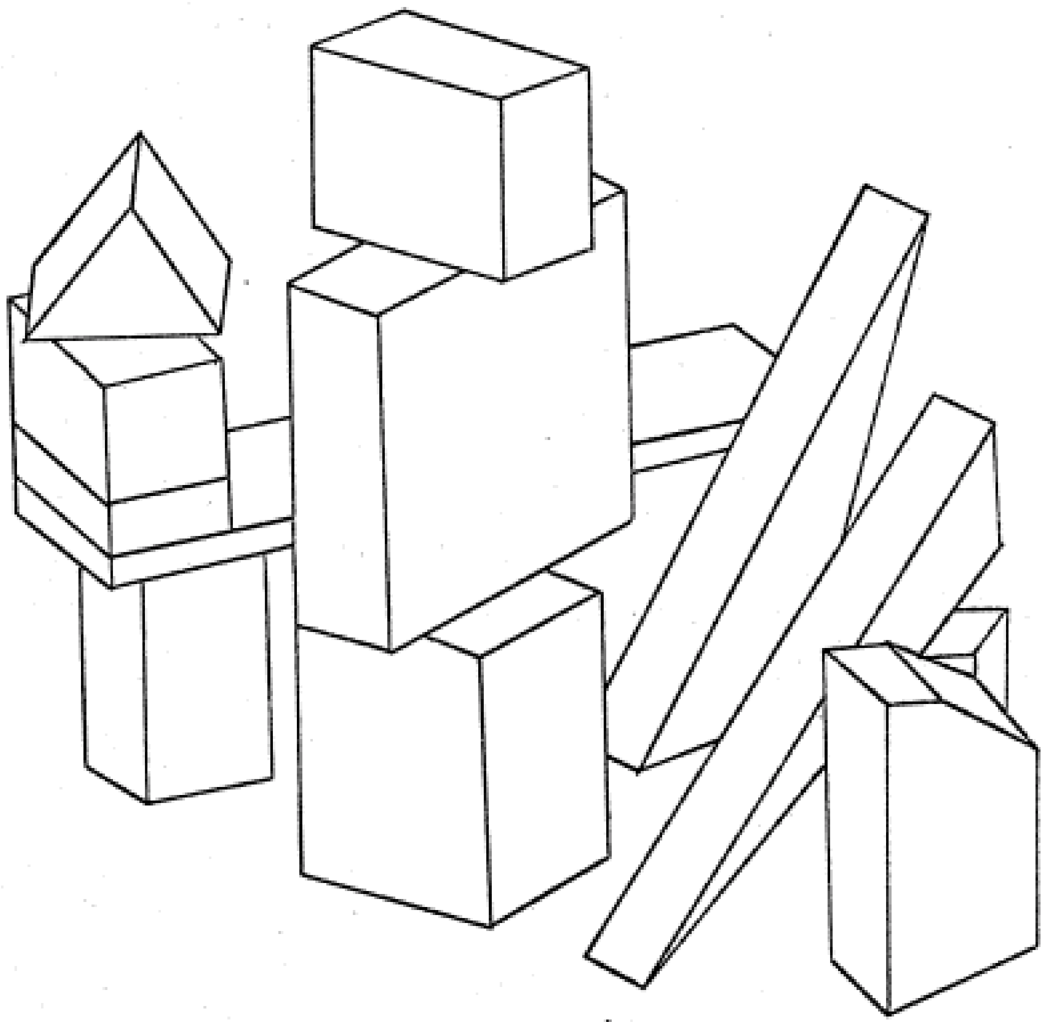


Figure 1-6

noted that all information in the system moves in one direction only, as indicated by all of the arrows in the diagram pointing from bottom to top. He remarks: "There is as yet no way a process can discourse with and modify the behavior of any process acting below it." [2] One of the aims of this research is to enable SEE to receive such discourse from above.

Another possible improvement to SEE is indicated by the scene of figure 1-7. Vertices a, b, and c each generates one strong and one weak link, causing SEE to report a single body composed of regions 1, 2, 3, and 4. A more satisfactory analysis would find two abutting wedges.

In figure 1-8, which is ambiguous, SEE again finds one body. Either of the two alternative analyses depicted would be better.

1.3 SEEMORE: an extension of SEE

1.3.1 Basic properties of SEEMORE

In view of the problems of SEE noted above, it would be useful for a modified SEE procedure to fulfill the following requirements:

1) It should correctly handle all cases which SEE handles correctly.

2) It should correctly handle many cases, such as figure 1-7, which SEE handles incorrectly.

3) It should provide for interaction with a higher-level object-recognizing program such as Winston's.

4) In case its first analysis is unsatisfactory, or in case of ambiguous scenes such as figure 1-8, it should be capable of generating plausible alternative answers.

I have devised a procedure called SEEMORE which seems to satisfy all of the above criteria for many scenes. SEEMORE has not been

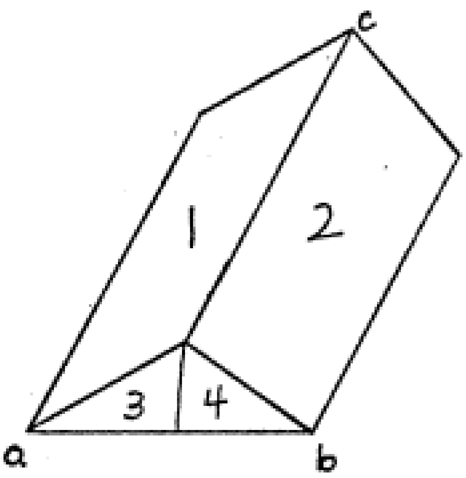


Figure 1-7

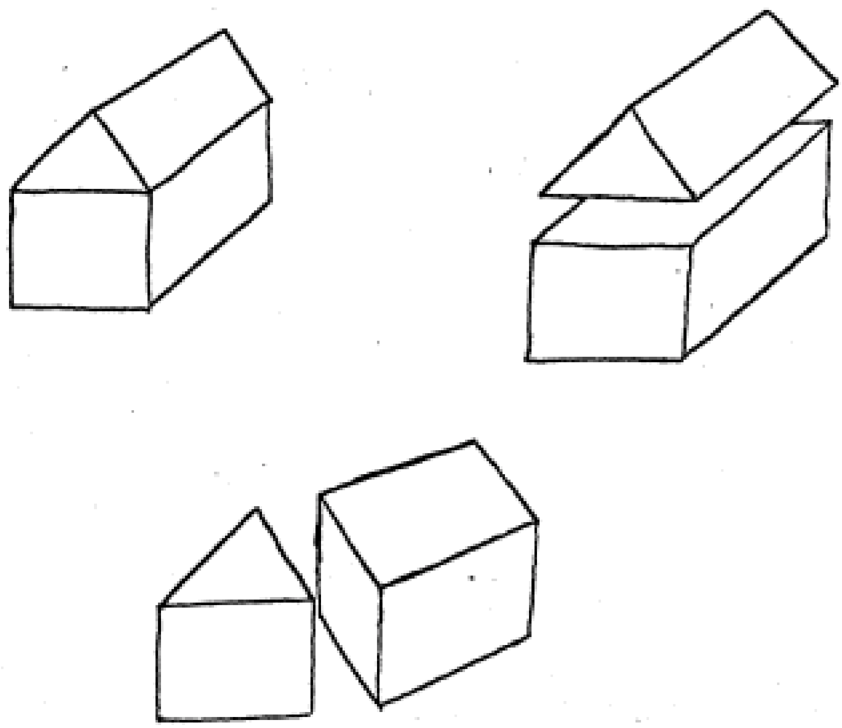


Figure 1-8

programmed, but has been developed extensively through hand-simulations on a large number of scenes. For the scene of figure 1-6, for example, SEEMORE produces the same answer as SEE, finding twelve bodies. In figure 1-7, SEEMORE finds the wedges (1 3) and (2 4). For the ambiguous scene of figure 1-8, SEEMORE produces both of the alternatives depicted.

SEEMORE can also handle the difficult scene depicted in figure 1-9. In this case, four bodies are reported. These are composed of regions (2 3 4), (5 6), (8 9 10), and (1 7 11).

1.3.2 Splitting

Besides forming links in the manner of Guzman's program, SEEMORE also forms <u>splits</u>. While a link represents evidence that two regions are part of the <u>same</u> body, a split represents evidence that two regions belong to <u>distinct</u> bodies in the scene. To implement the splitting process I have developed a variety of heuristics, similar to those of SEE, which utilize simple local evidence to begin splits at vertices and extend them along edges of objects. These new heuristics, along with a modified version of SEE, are incorporated into a procedure which attempts to generate alternative answers, starting with the most plausible.

1.4 A note on nomenclature

Figure 1-10 illustrates some of the abbreviations and symbols which will be used in the text and diagrams of this paper. Numbers, such as 3 or 5, represent regions in the scene. The background, which is region 5 in this scene, will often not be labelled in the

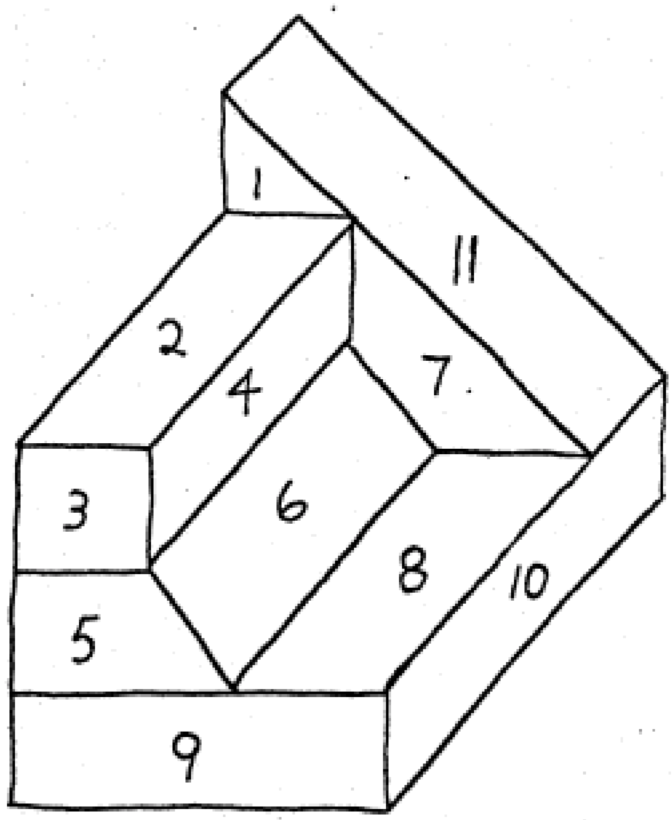


Figure 1-9

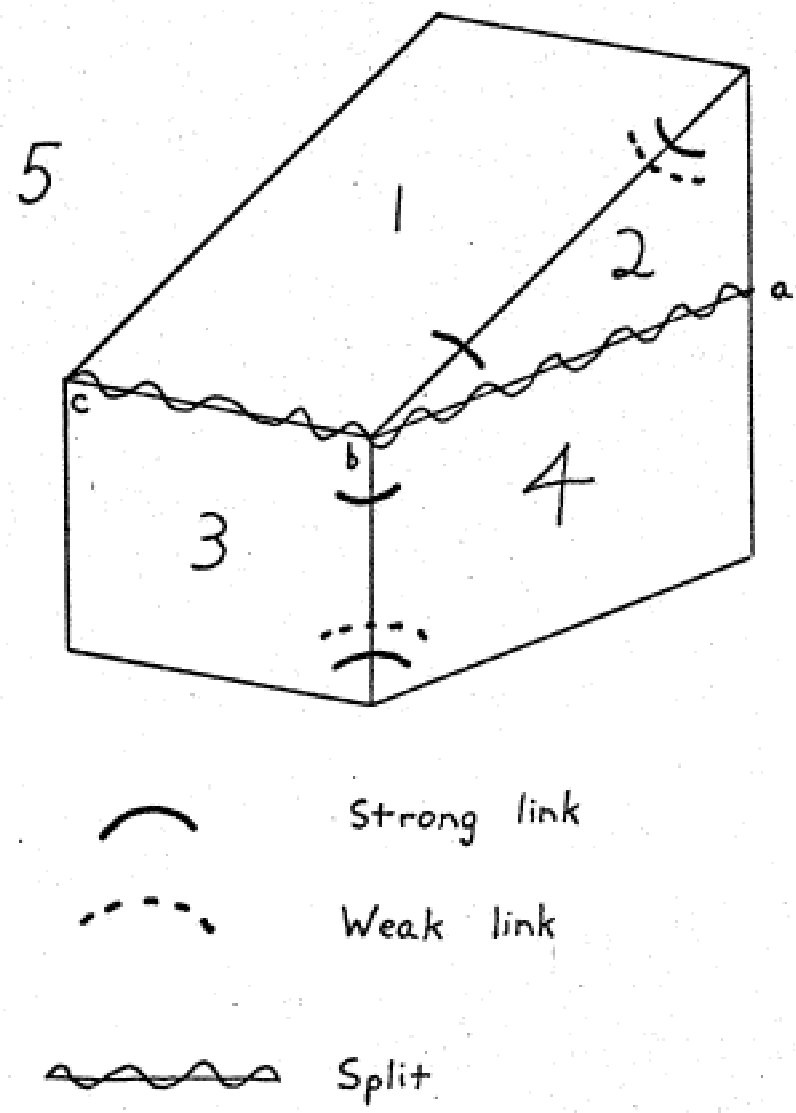


Figure 1-10

remaining diagrams. Lower-case letters, such as a, b, and c in the figure, will represent vertices in the scenes. The wavy lines superimposed on edges cb and ba represent splits. These will be designated in the text individually as c-b and b-a; or, equivalently, as one continuous split c-b-a. Strong and weak links are designated by short solid and dotted lines as shown, following Guzman's usage. Conglomerated regions will be indicated by enclosing lists of such regions in parentheses. In this example, the bodies found are (1 2) and (3 4).

2 The basic SEEMORE procedure

The flowchart of figure 2-1 illustrates the general structure of SEEMORE. The procedure begins by applying very cautiously those heuristics which are believed to be most reliable; this step is referred to as initial splitting and linking, or phase 1. The next step in SEEMORE's analysis of a scene, called phase 2, is a more tentative attempt to generate splits by applying heuristics thought to be less reliable than those of phase 1. If there is no splitting evidence in the scene, phase 3 is entered directly, and SEEMORE reduces to a slightly modified version of SEE.

Finally, phase 3 applies the rest of SEE's heuristics to the tentatively split scene and produces SEEMORE's first attempted solution. This is submitted to the higher-level program for approval. If the higher-level program is dissatisfied with the first analysis, it will return to SEEMORE with an appropriate complaint. The processing of this complaint is discussed in the next chapter.

At various times in phases 1 and 2, before the analysis of the scene is complete, SEEMORE may consult a higher-level program to find out if certain sub-parts of the scene can be recognized as unoccluded objects. This information is used by SEEMORE to generate additional splits and links.

2.1 Phase 1: generate initial splits and links

2.1.1 Splitting-vertices

SEEMORE begins by identifying those vertices in the scene which may be used to initiate or extend splits. I refer to these, naturally

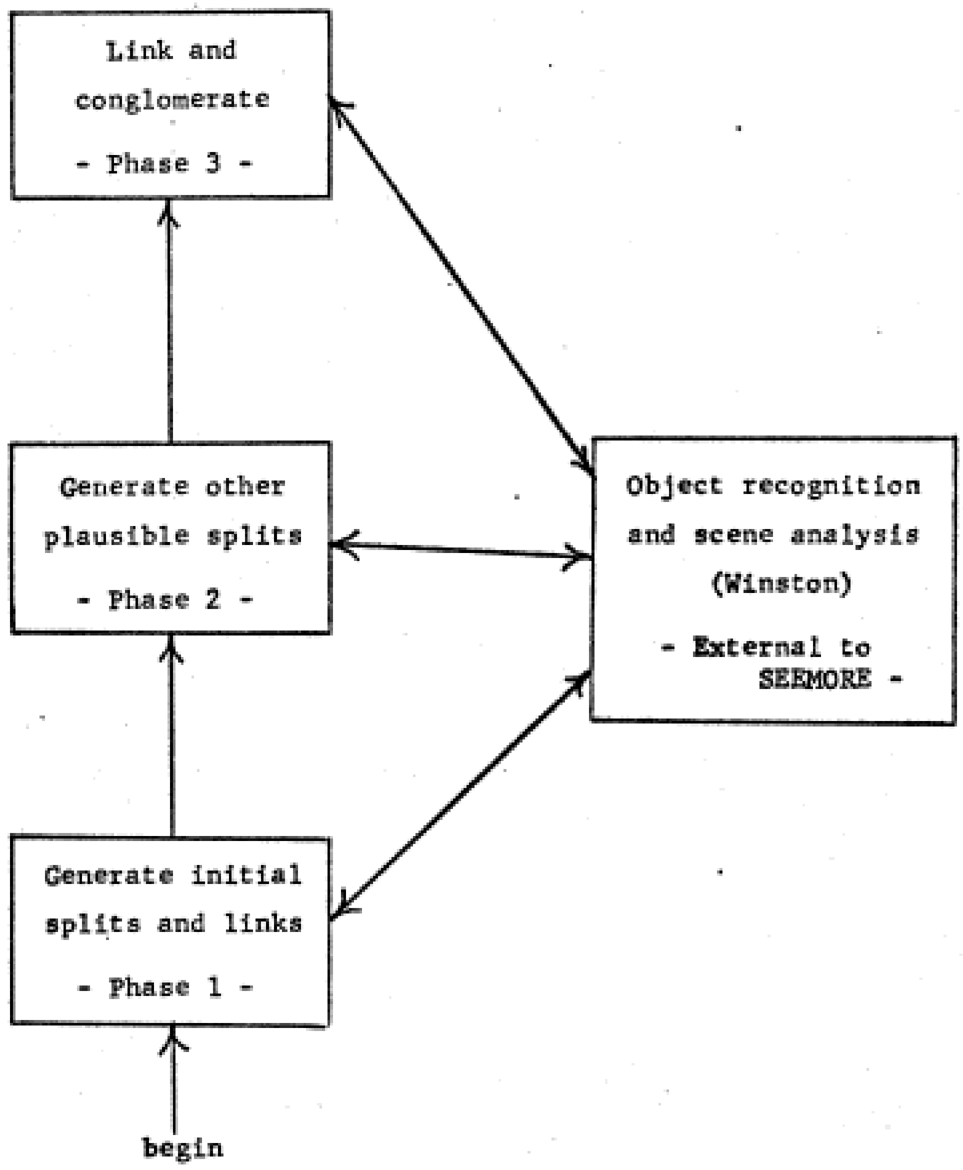

Link and
conglomerate
- Phase 3 -
Generate other
plausible splits
- Phase 2 -
Generate initial
splits and links
- Phase 1 -
begin
Object recognition
and scene analysis
(Winston)
- External to
SEEMORE -

Figure 2-1

enough, as splitting-vertices. A commonly occurring splitting-vertex is the generalized four-line vertex, represented in figure 2-2A. The K and X joints are special cases of four-line vertices. More generally, any vertex of four or more lines is a splitting-vertex. In the example of figure 2-3, SEEMORE marks vertices a, b, c, d, and e as splitting-vertices.

Notice in figure 2-4A that vertices a and b would be classified as splitting-vertices. However, these are false vertices, in the sense that they do not persist under a small change in viewing angle, as shown in figure 2-4B. On the other hand, vertices c and d are invariant under small changes in viewing angle. SEEMORE assumes that all vertices in the input scene are similarly invariant; scenes like figure 2-4A will, in general, not be analyzed properly.

Any split which emanates from a vertex v is terminated at the next vertex encountered on the split line, unless the next vertex is an L. In figure 2-5, for example, a single split initiated at vertex v along the line v-a is extended through vertex a and terminates at vertex b as shown.

As illustrated in figure 2-6, a split propagates through a pair of matching-Ts if the same region R borders on the center lines of both Ts.

2.1.2 Phase 1 heuristics

The phase 1 heuristics are used to initiate splits and links based upon evidence provided by splitting-vertices and special configurations as described below. These heuristics are considered

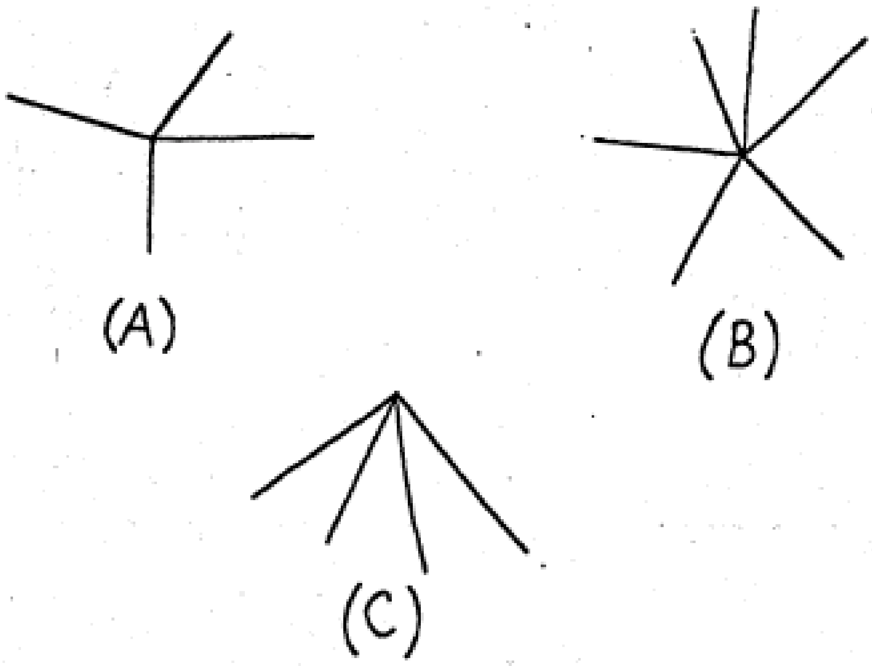


Figure 2-2 Splitting-vertices

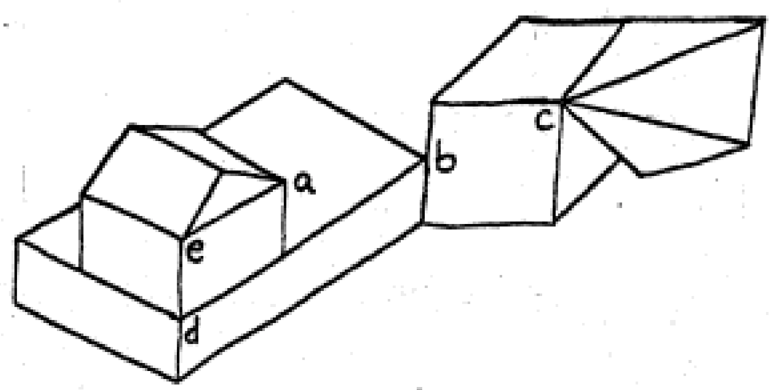


Figure 2-3

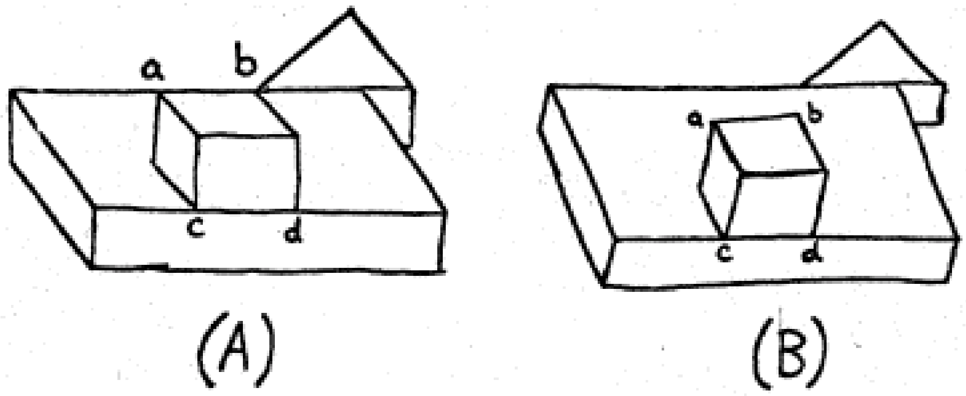


Figure 2-4

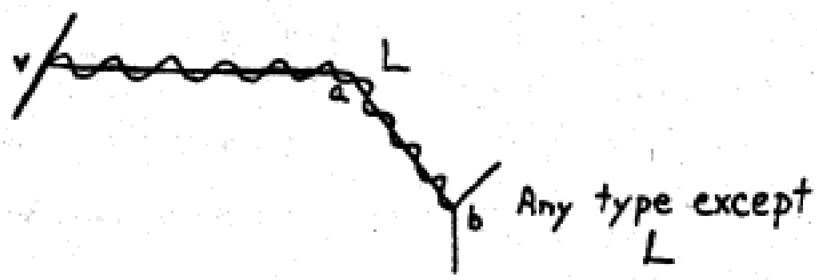


Figure 2-5 v-a-b is a single split

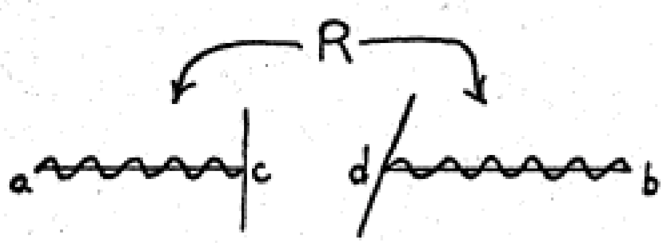


Figure 2-6 a-c, d-b are made as a single split to be more reliable than those which will be applied later in phase 2.

2.1.2.1 Heuristic: split around strongly-conglomerated objects

In order to implement this heuristic, SEEMORE first invokes an _initial-linking_ routine. These initial links are placed much more cautiously than the more general SEE-type links to be used later. Vertices at which initial links are placed are illustrated in figure 2-7. These links are subject to the veto of a link-inhibitor of the sort used in SEE. Situations in which links will be inhibited are illustrated in figure 2-8. A more lenient link-inhibitor, to be described later, will be used to complete the linking in phases 2 and 3.

Any two regions joined by at least two initial links are said to be members of the same _strongly-conglomerated_ _nucleus_. A region is considered to be a member of such a nucleus only if it is doubly linked to another single region within the nucleus; single links to two different regions in the nucleus do not suffice. Each strongly-conglomerated nucleus of two or more regions found in the scene is submitted to a higher-level program capable of recognizing objects. The higher-level program should verify the nucleus only if it makes sense as a complete unoccluded object in the scene. A nucleus so verified is called a _strongly-conglomerated_ _object_.

The use of these objects in splitting is illustrated in figure 2-9. The general splitting rule is shown in figure 2-9A. Strongly-conglomerated object (1 2 3) has a splitting-vertex _v_ on its boundary; splits are made which extend from vertex _v_ to neighboring vertices

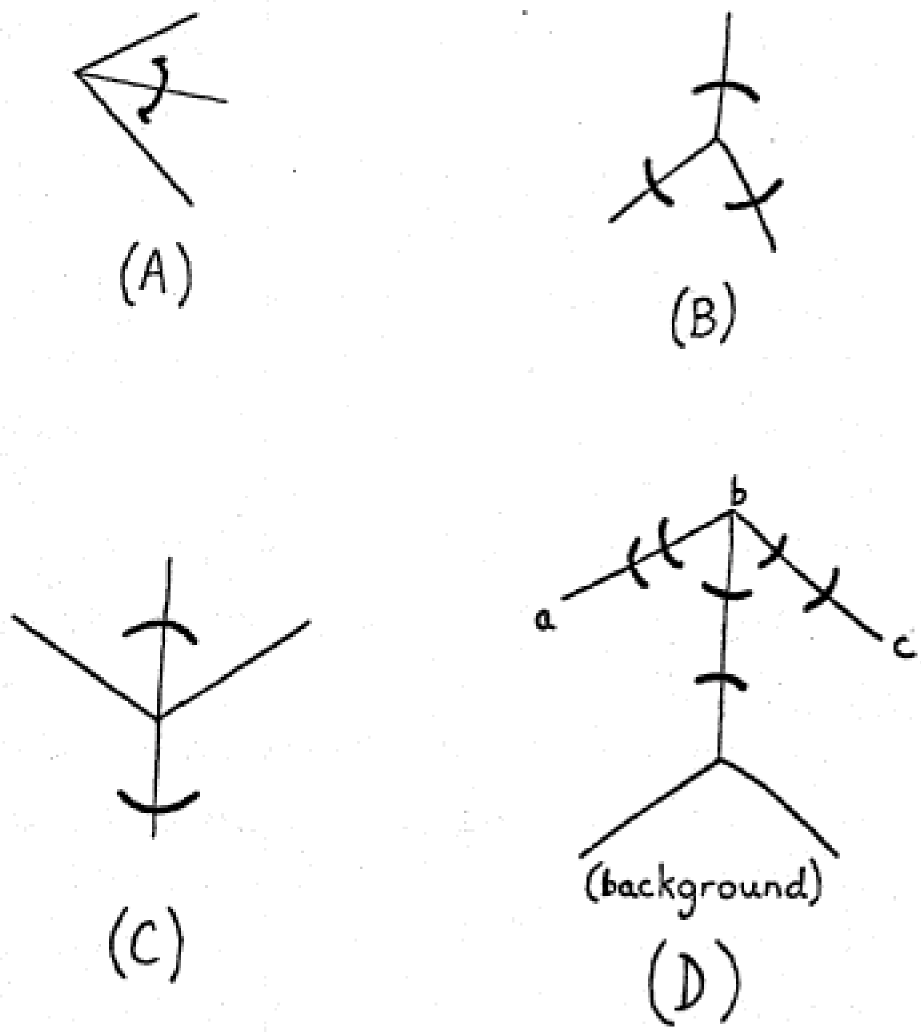


Figure 2-7   Initial links

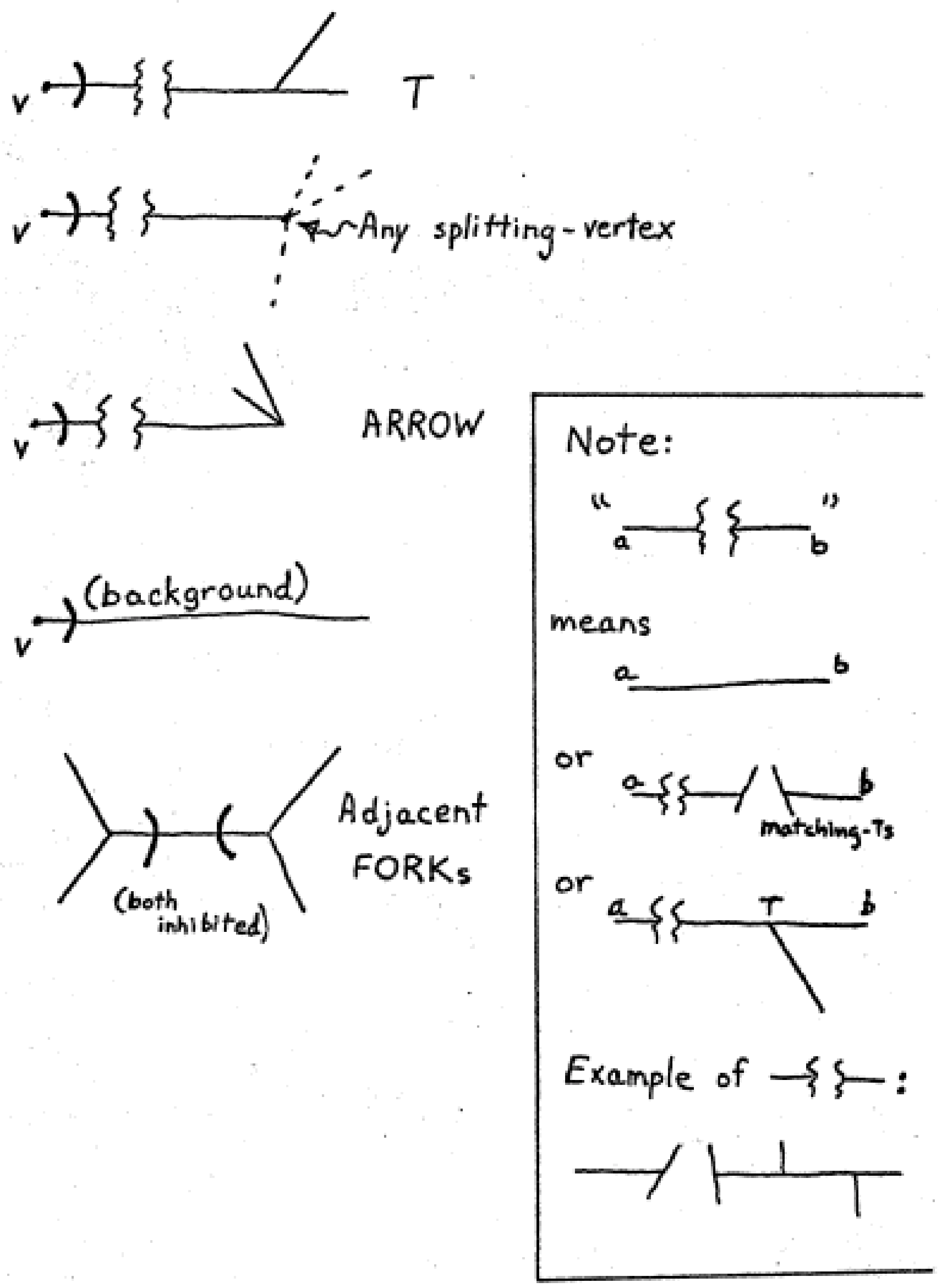


Figure 2-8 Inhibited initial links

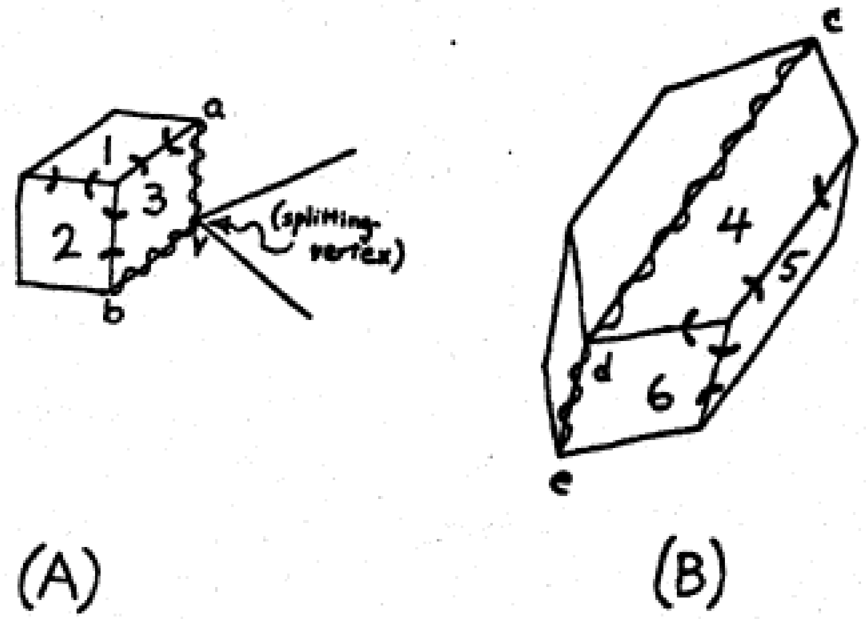


Figure 2-9

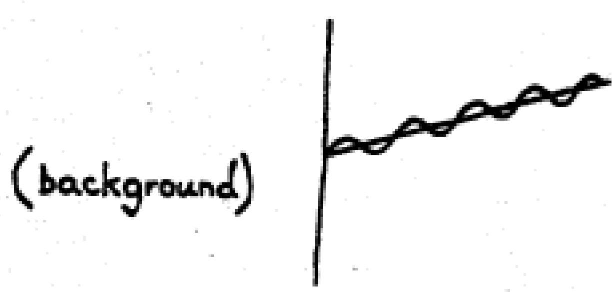


Figure 2-10

a and b along the boundary of object (1 2 3). For example, in figure 2-9B, split c-d-e is made along the boundary of strongly-conglomerated object (4 5 6). It should be noted that the phase 2 split-extending heuristics, to be described later, are not applied to the splits just described.

2.1.2.2 Heuristic: internal-T

When a T-joint borders on background as shown in figure 2-10, a split is placed on the center line of the T.

2.1.2.3 Heuristic: special MULTI

As shown in figure 2-11, if a vertex has four or more lines of which two are collinear, and of which only one falls to one side of the collinear lines, then a split is placed along that single line. In the special case of a four-line vertex, this heuristic reduces to the X-joint heuristic illustrated in figure 2-12. In this case, if either of the links shown in figure 2-12 has been inhibited during the initial linking step, then the splits are inhibited as well.

2.1.2.4 Heuristic: split to external concavities

This heuristic is best explained by an example. In figure 2-13, the FORK vertex a borders on background; the line which does not border the background connects to splitting-vertex v. In this case, a split is made along v-a. Vertex a is referred to as an external concavity.

More generally, this heuristic places splits from splitting-vertex v to external concavity a in either of the situations depicted in figure 2-14.

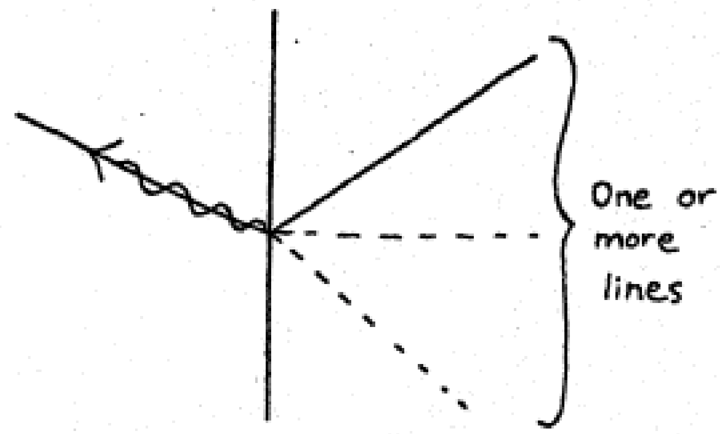


Figure 2-11

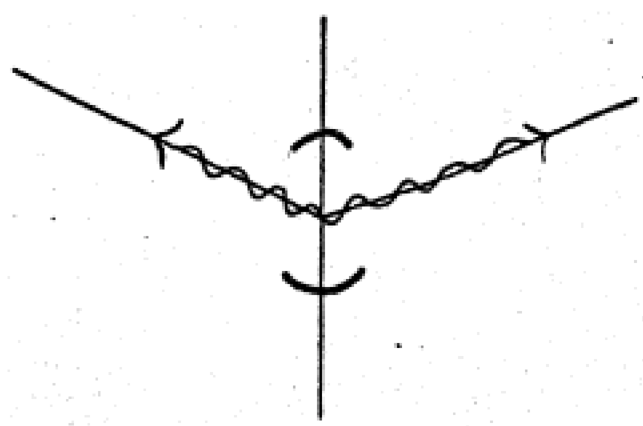


Figure 2-12

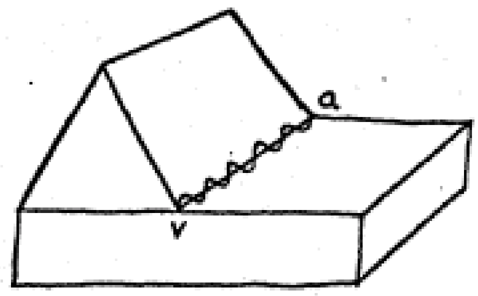


Figure 2-13

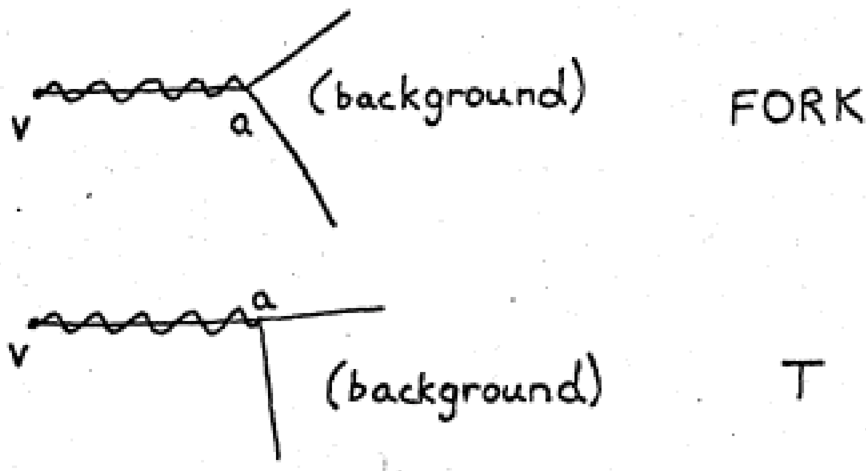


Figure 2-14

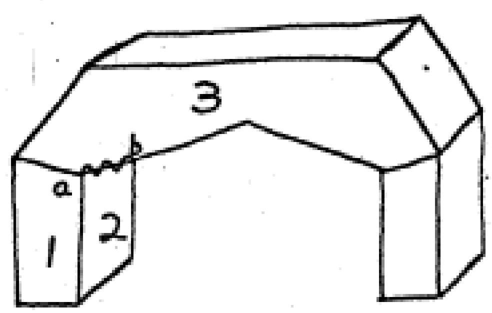


Figure 2-15

It is conjectured that it would also be useful to split between FORKs and external concavities; in other words, to allow the vertex v in figure 2-14 to be a FORK. This additional heuristic would place the important split a-b in figure 2-15, enabling body (1 2) to be separated from region 3. However, this heuristic is not included in the current version of SEEMORE pending further study.

2.1.2.5 Heuristic: split the crossbar of the T-ARROW

The configuration for which this heuristic applies is illustrated in figure 2-16. Splits propagate along the crossbar of the T in both directions. In figure 2-17, this heuristic generates the split a-b-c. A variation of this heuristic which is employed in phase 1 is the T-X configuration depicted in figure 2-18. Under study but not being used is the T-link configuration shown in figure 2-19.

2.2 Phase 2: Generate other plausible splits

2.2.1 Extending splits

This routine uses evidence provided by splitting-vertices and other configurations to extend splits which have been placed in phase 1. The routine applies only a couple of heuristics at present, and may represent an area in which new heuristics and some fine tuning would help significantly the performance of the SEEMORE procedure on difficult scenes.

A very common situation for extending a split is represented in figure 2-20. If exactly one split terminates at a four-line vertex, such as split a-b in the figure, then the split is extended along the opposite line b-c. In addition, links are placed across the two

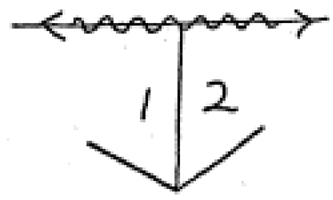


Figure 2-16

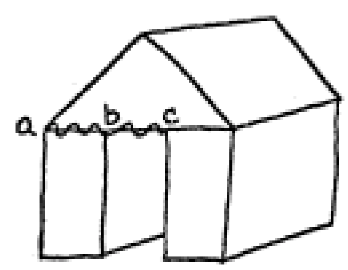


Figure 2-17

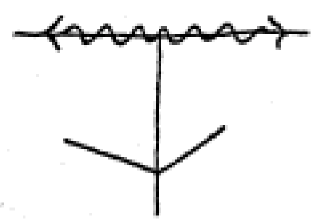
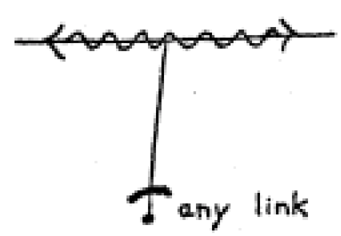


Figure 2-18

Figure 2-19



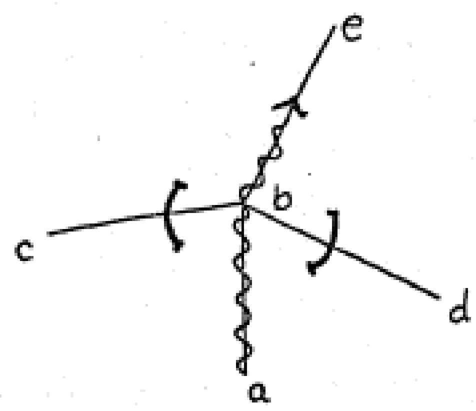


Figure 2-20

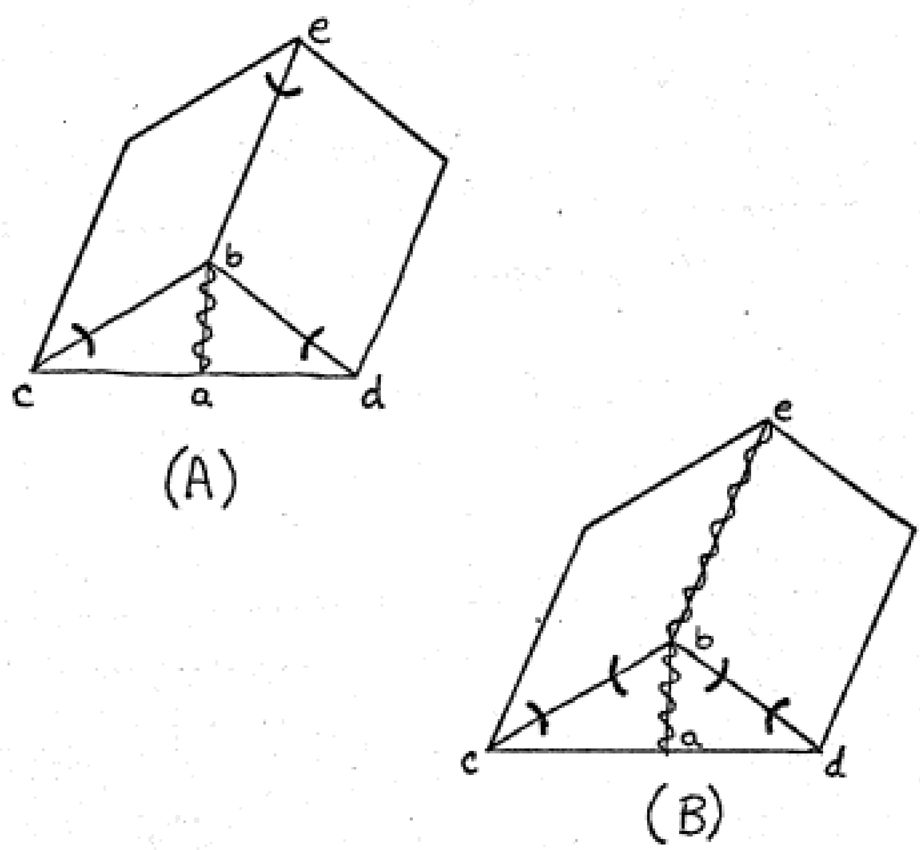


Figure 2-21

unsplit lines as shown. Figure 2-21A represents a scene in which initial splitting and linking has been completed, and the split-extending heuristic applies. The result is shown in figure 2-21B. Notice that the link at vertex e has been broken by the split from b to e. The split-extending would be finished at this point, since no other splitting-vertices remain.

This split-extending heuristic applies to Ks and Xs as special cases of four-line vertices, but does not apply to PEAKs.

Another, rather specialized, split-extending heuristic applies to the five-line MULTI with two collinear lines, illustrated in figure 2-22. Figure 2-23 shows an example of this configuration in a scene.

If either of the links generated by one of the above heuristics is inhibited by the link-inhibitor described below in the discussion of phase 3, then the split is not extended. The other link is still placed if not inhibited.

Finally, another specialized heuristic is depicted in figure 2-24A. Although FORK b and ARROW d have both linked regions 1 and 2 in the initial-linking phase, the split entering the FORK along a-b provides strong enough evidence to break both the links by extending the split along b-d. Of course, the symmetrical split along b-c would also have been sufficient evidence. A split is similarly extended from the FORK to a PEAK as in figure 2-24B. Applied to the ambiguous scene shown in figure 2-25A, this heuristic enables SEEMORE to produce the analysis represented in figure 2-25B.

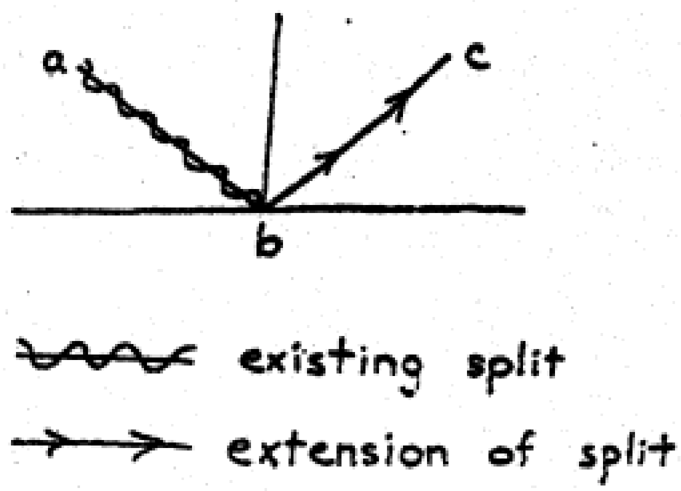


Figure 2-22

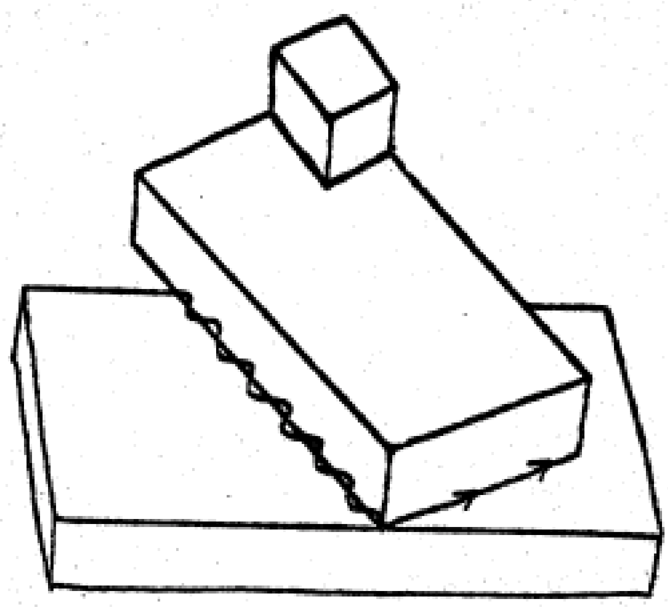


Figure 2-23

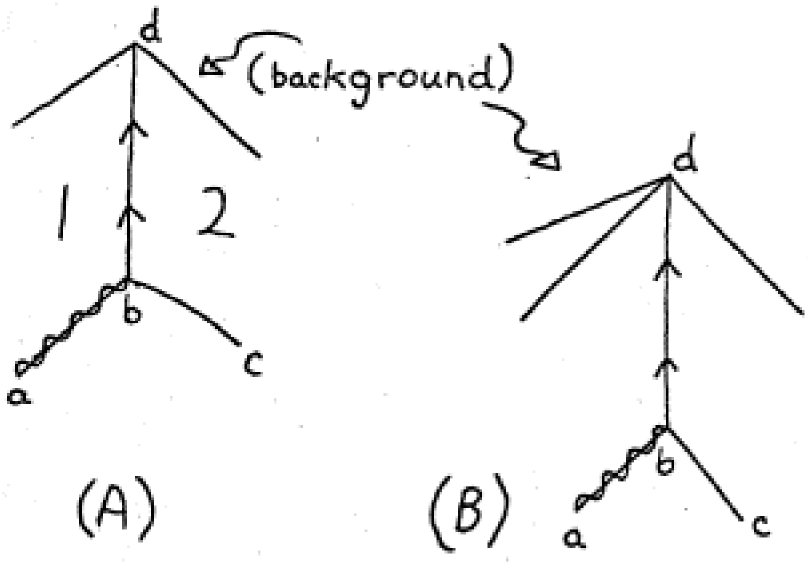


Figure 2-24

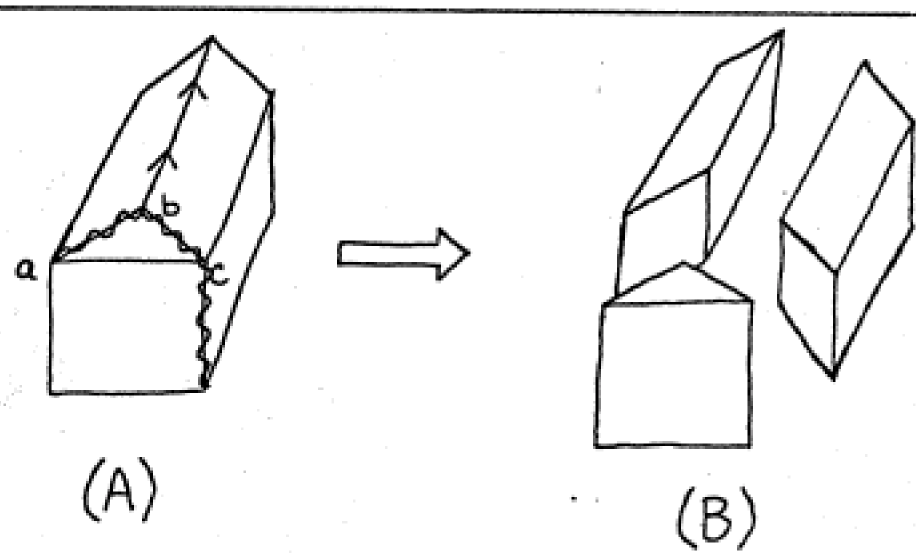


Figure 2-25

The split-extending routine iterates until no further extending can be done. If any strongly-linked nuclei are formed during split extension or at any subsequent point in phase 2, the routines which were employed at the beginning of phase 1 (section 2.1.2.1) are invoked to verify objects and make splits accordingly.

2.2.2 Apply weaker splitting heuristics

After the initial splits of phase 1 have been extended, there often remain splits which are incomplete or splitting-vertices through which no splits have yet traversed. If this is the case, SEEMORE employs weaker and more conditional evidence to propose and extend additional splits. As will be seen later, SEEMORE stands ready to generate alternative analyses, if requested by a higher-level program, by backing up and placing alternative splits.

The remainder of this section catalogues several configurations which seem to provide weaker evidence for splitting in many scenes. SEEMORE attempts to apply these heuristics in the order in which they are presented below. The first applicable rule is invoked in each iteration until the list is exhausted; splitting is then complete.

2.2.2.1 Heuristic: split along the non-adjacent lines

This is a generalization of the split-extending rule for four-line vertices. Given the generalized four-line vertex of figure 2-26, this rule indicates the alternatives of splitting along either a-b-c <u>or</u> d-b-e. SEEMORE takes these alternatives into consideration when applying any of the heuristics below to a four-line vertex.

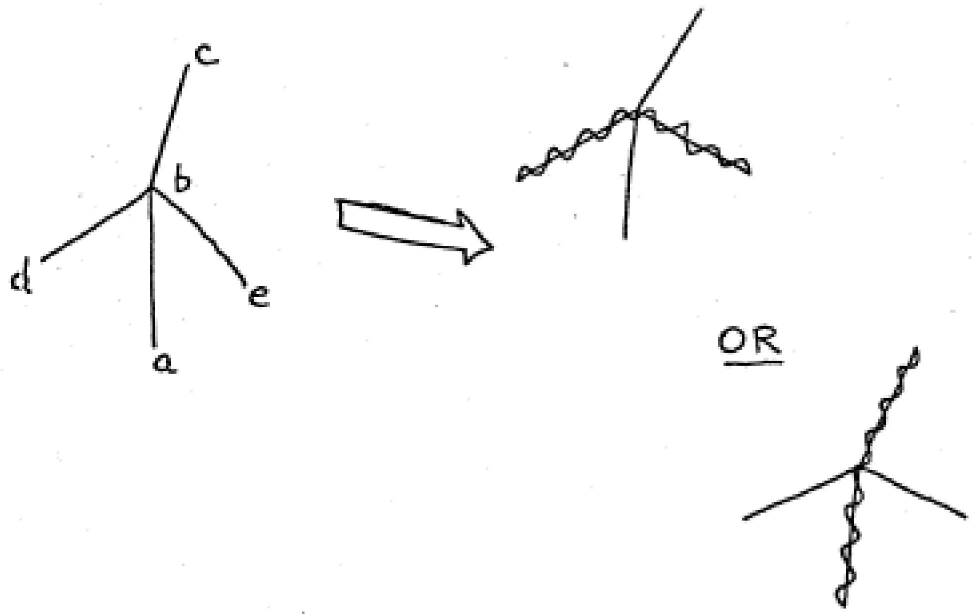


Figure 2-26

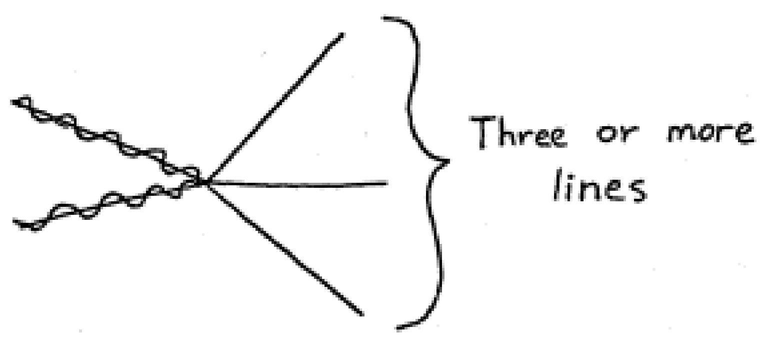


Figure 2-27

2.2.2.2 Heuristic: make the most complete split possible

The above title is actually a general statement of two specific rules which are applied at each four-line vertex which is not a PEAK, has not yet received any splits, and does not border on the background. One of these rules is: extend an unfinished split to the background or to another unfinished split whenever possible. For this purpose, an unfinished split is defined as one which terminates at a vertex not bordering on the background. Additionally, if a split traverses a MULTI but leaves three or more adjacent lines unsplit as shown in figure 2-27, an unfinished split is considered to exist at the MULTI. Look at figure 2-28. Notice that an initial split is made along e-d, then extended along d-c. The present rule favors splitting c-a-b rather than g-a-f, in order to complete the unfinished split ending at vertex c.

The other specific rule, to be tried if the first does not apply, is: split to the background whenever possible. In figure 2-29, for example, we consider splitting either b-a-c or e-a-d. The present rule would prefer the former split, since b and c both border on the background, while e is an internal vertex.

In the example of figure 2-30, the split-to-background rule indicates splits c-a-d and c-b-d. The resulting analysis proposes objects (1 2), (3 4), and (5 6), which could plausibly represent three tetrahedra. Humans seem to prefer the analysis (1 3 5) and (2 4 6), which is produced as an alternative by SEEMORE. Perhaps the appropriate heuristic here involves minimizing the length of splits used or

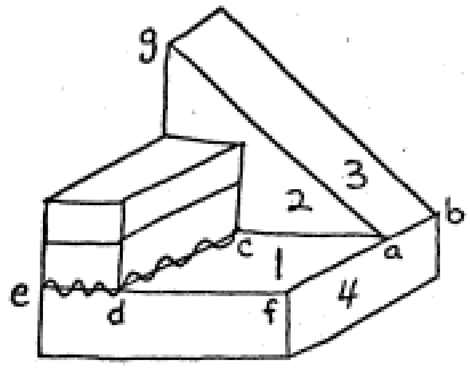


Figure 2-28

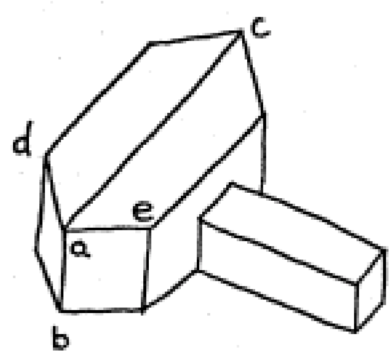


Figure 2-29

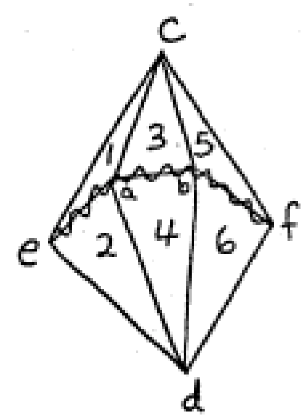


Figure 2-30

the number of bodies found; neither of these factors is taken into account by SEEMORE.

One further example indicates the power of the most-complete-split heuristic. In figure 2-31, SEEMORE uses this heuristic to correctly split along e-d-a, producing the plausible bodies (1 2 3), (4 6), and (5 7). The inherent complexity of this scene is reflected in SEE's answer: bodies (1 2 3), (4 5 7), and (6) are reported.

2.2.2.3 Heuristic: split between a pair of splitting-vertices

This rule applies only to pairs of splitting-vertices which are neighbors on the same line (figure 2-32) or which have a common neighbor (figure 2-33). Each of the vertices a and b in the figures must be either a PEAK, a MULTI of more than four lines, or a four-line vertex which has thusfar received no splits. Furthermore, a and b must be neighboring only each other in this fashion.

If these conditions are met, splits are placed as shown in figures 2-32 and 2-33. The split-extender is used if applicable.

Both forms of this heuristic may simultaneously apply in some cases. An example is figure 2-34, which contains two neighboring PEAKs. The first analysis splits between the PEAKs directly, along a-b. This will generate SEEMORE's initial guess: bodies (1 2 3) and (4) are reported (figure 2-35). If an alternative is requested, the alternative split to a common neighbor of the pair of splitting-vertices is used. In this case, the only common neighbor vertex is c, and the alternative split is a-c-b. The resulting alternative answer reports bodies (1), (2), and (3 4), as shown in figure 2-36. Both

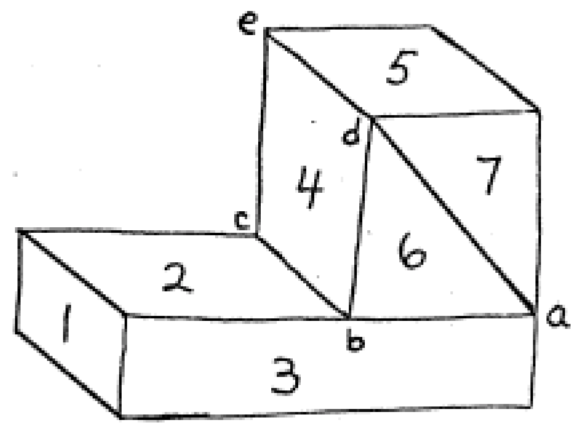


Figure 2-31

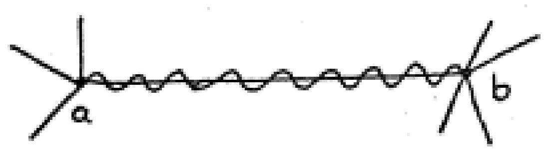


Figure 2-32

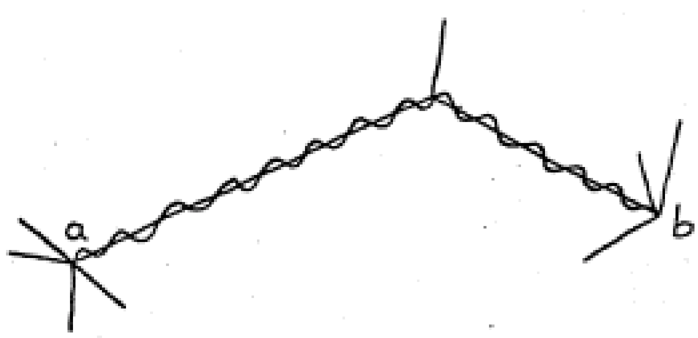


Figure 2-33

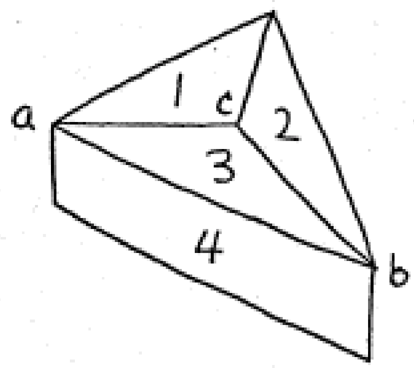


Figure 2-34

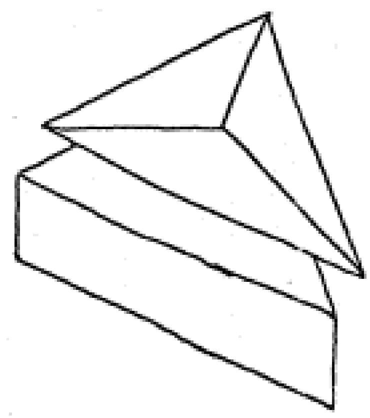


Figure 2-35

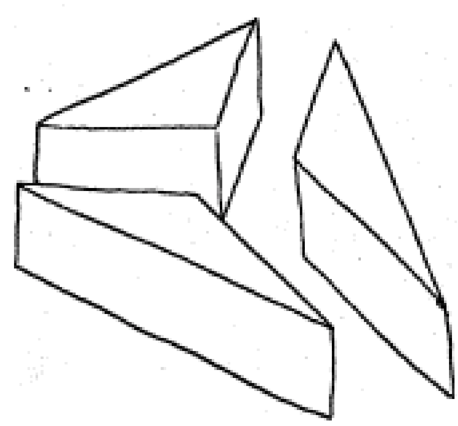


Figure 2-36

these alternatives are plausible analyses of the scene.

2.3 Phase 3: link and conglomerate

When phase 2 of SEEMORE cannot start any new splits or extend any existing ones, phase 3 is entered. Phase 3 applies the remaining heuristics of SEE, with a few modifications as outlined below. In this phase the remaining links are placed, subject to the constraints of splits already made. Nuclei are formed and single-region bodies are handled in similar fashion to SEE. Phase 3 generates a tentative analysis, which is presented to a higher-level program for approval.

2.3.1 Completion of linking

The initial linking has already placed links at FORKs, ARROWs, and X-joints as shown in figure 2-37. Some of these will have been inhibited by the phase 1 link-inhibitor. Others will have subsequently been broken by splits. New links may have been added by the phase 2 routines. The linking is now completed by placing strong and weak links exactly as SEE does, <u>except</u> as specified in the paragraphs below. Refer to Guzman's Ph.D. thesis [4], pages 81 through 88, for detailed descriptions of the linking heuristics.

2.3.1.1 Inhibition of links

Figures 2-38 and 2-39 illustrate the situations in which links are inhibited during the final linking stage of SEEMORE. Figure 2-38 depicts inhibited links which are also inhibited by SEE. These include links across lines with passing-Ts; notice, however, that ARROW links are <u>not</u> inhibited by the passing-T. Also inhibited are

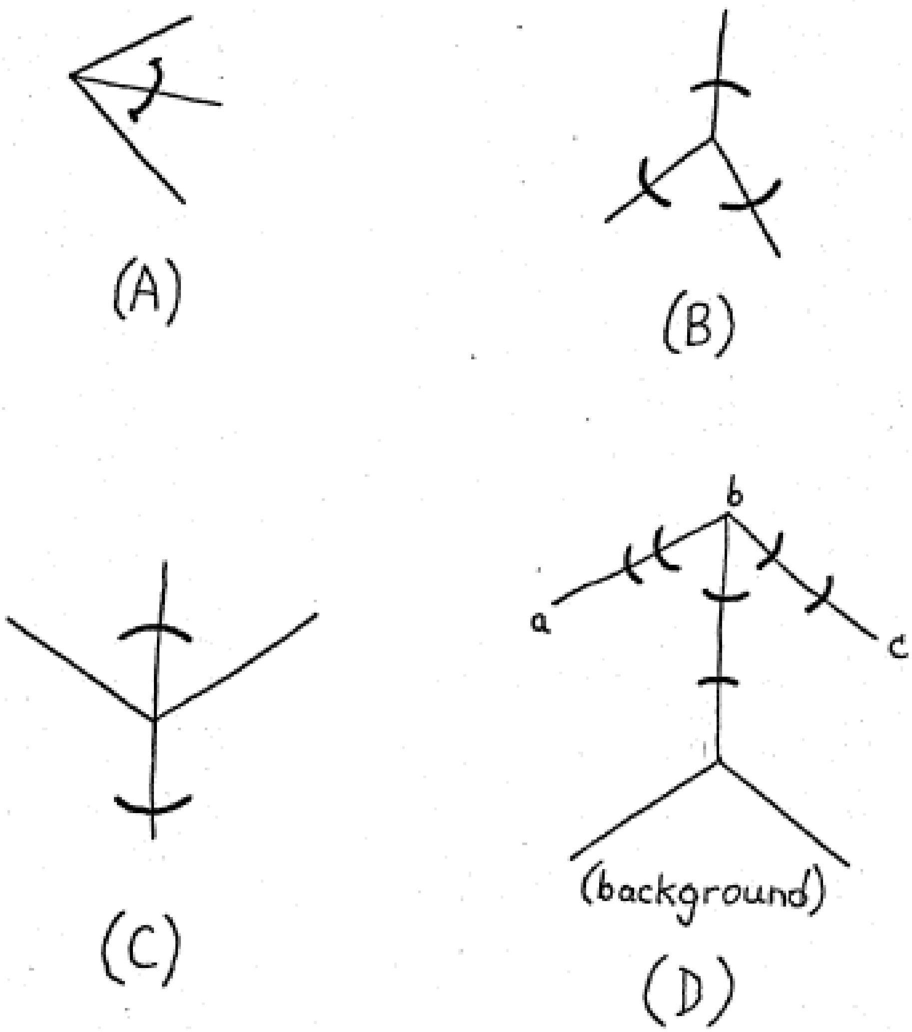


Figure 2-37 Initial links

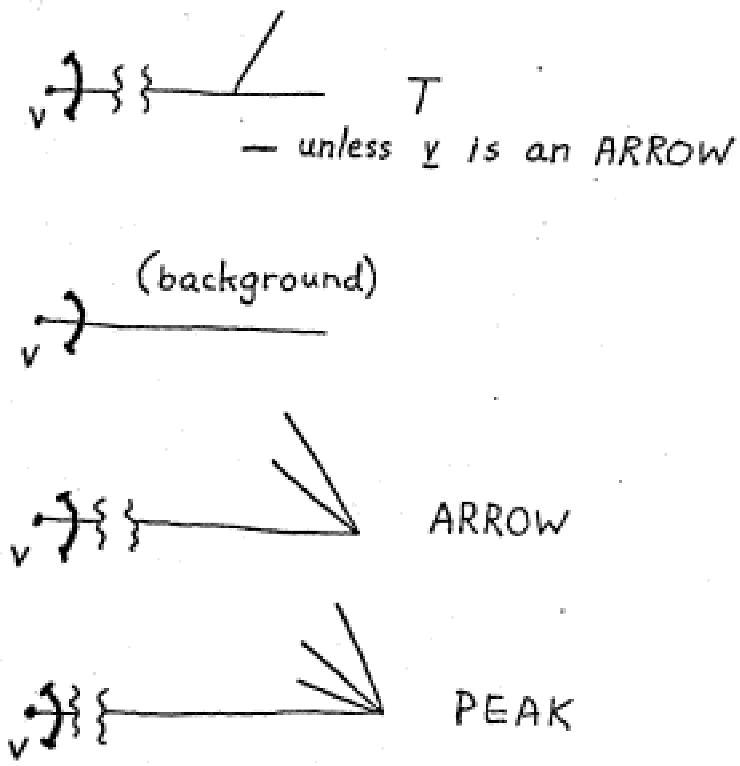


Figure 2-38

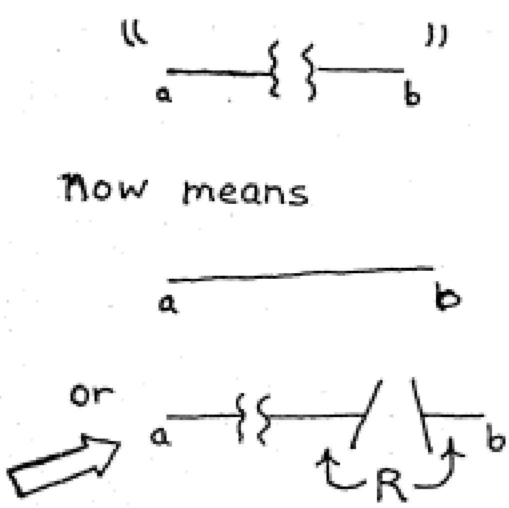


Figure 2-39

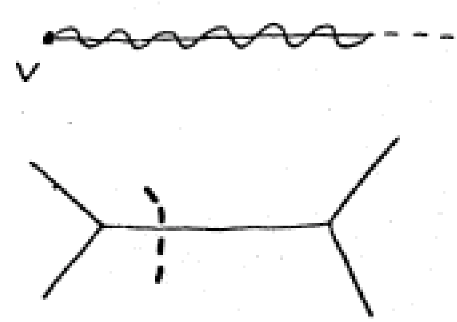


Figure 2-40

links between any object and background; and links across the outside line of a PEAK or an ARROW. Notice that the through-Ts function has been restricted as shown in figure 2-39. In the matching-T situation marked in the figure by a broad arrow, the same region R must border the center lines of both of the T-joints. R in this case may not be background.

Links are also inhibited across split lines (figure 2-40). Also shown in figure 2-40 is the pair of neighboring FORKs, across which strong links are inhibited and a single weak link is placed as shown.

A linking heuristic used in Guzman's program is illustrated in figure 2-41A. It is omitted from SEEMORE because it is inconsistent with the assumption I have made that the type of each vertex is invariant under small perturbations of the viewing angle. Another heuristic used by SEE, illustrated in figure 2-41B, is made unneces sary by other heuristics of SEEMORE.

Unlike SEE, SEEMORE does not place matching-T links at this step. These are treated separately, as discussed below.

2.3.1.2 Formation of nuclei

On the basis of strong and weak links which have been placed up to this point, SEEMORE conglomerates nuclei using the heuristics of Guzman's program. After conglomeration is complete, matching-T links are processed as described in the next section.

2.3.1.3 Matching-Ts

SEEMORE's decision to link or not to link across a pair of matching-Ts is always made provisionally. If the higher-level program

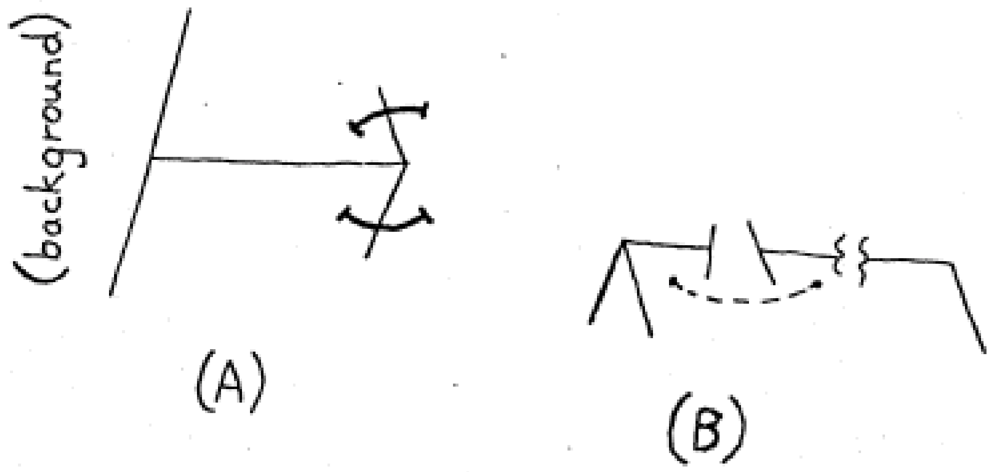


Figure 2-41 Links not used by SEEMORE

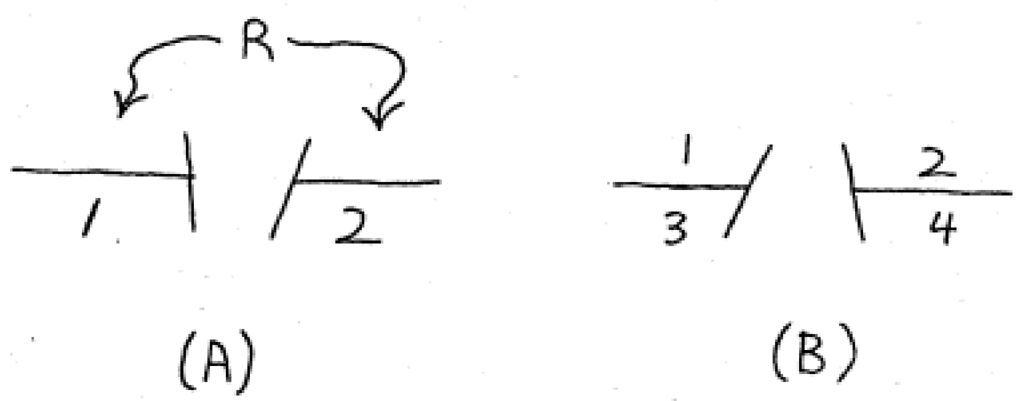


Figure 2-42

registers an appropriate complaint, SEEMORE stands ready to change its mind about the matching-T links involved.

Figure 2-42 illustrates two possible matching-T configurations. In figure 2-42A, the same region R borders on the center lines of both Ts. In this case, SEEMORE will place a link between regions 1 and 2. This heuristic still applies if R is background. Notice that if regions 1 and 2 were also identical, there would be no need to place any links at all. The link between regions 1 and 2 will not be placed if there is a split between any member of the nucleus to which region 1 belongs and any member of the nucleus to which region 2 belongs.

In figure 2-42B, regions 1, 2, 3, and 4 are all distinct. In this case SEEMORE does not place links on the first pass, but plans to link 1 to 2 and 3 to 4 as an alternative. As before, matching-T links between two different nuclei are inhibited if a split exists between members of those nuclei.

Figure 2-43 illustrates some typical applications. When presented with the scene of figure 2-43A, SEEMORE generates a split and links as shown. The split inhibits all of the possible matching-T links; SEEMORE finds bodies (4 6 7), (1 2 3), and (5 8). In figure 2-43B there are no splits. Using the matching-T heuristic, SEEMORE places a single strong link between regions 4 and 5, and another single link between regions 7 and 8. These are sufficient to conglomerate body (4 5 6 7 8). If the higher-level program indicates that (4 5 6 7 8) must be split apart, SEEMORE will back up and remove one or both of the offending links, depending on the exact form of the

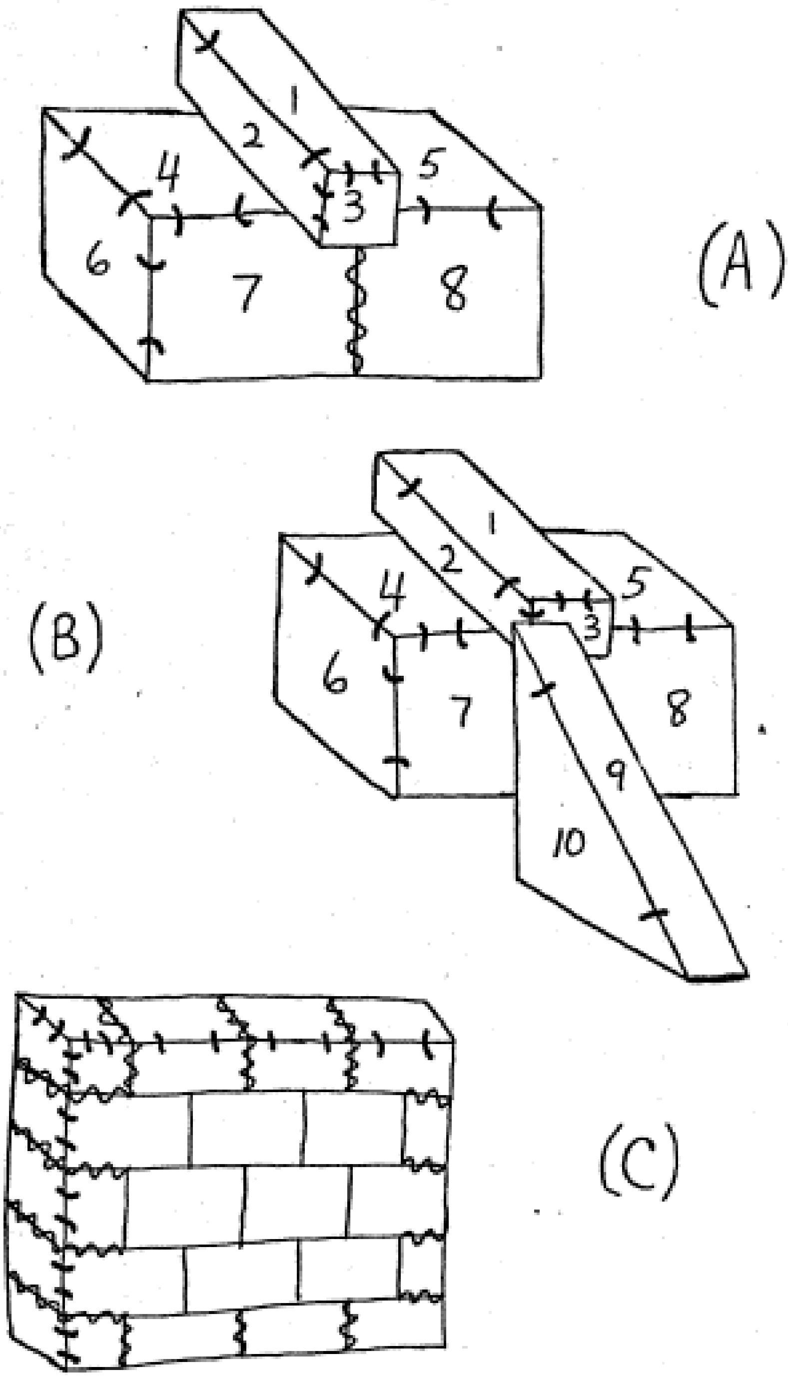


Figure 2-43   Scenes with matching-Ts

complaint. The result will be two separate bodies, (4 6 7) and (5 8).

The scene of figure 2-43C would present problems for SEE due to the large number of spurious matching-T links. Using the modified matching-T heuristic, SEEMORE does not place any of the spurious links; all of the separate bricks are correctly identified.

3 Producing alternative answers

3.1 The nature of a complaint

If a higher-level procedure is dissatisfied with SEEMORE's first attempt, it should ideally return to SEEMORE with the following information:

1) a list of the regions which have not been satisfactorily conglomerated

2) the nature of the complaint: body has too many regions/ is overlinked; body has too few regions/ is underlinked; body (or bodies) unrecognizable / does not make sense; analysis of the scene is plausible, but alternative(s) desired. Specific regions should be related to each complaint.

Of course, it may not always be possible for a higher-level program to provide a complaint in such detail. It is conceivable that a number of objects, or even the whole scene might be incomprehensible to the higher procedure. If there are not many splitting vertices in the scene, the number of alternatives which SEEMORE can generate is small and could be exhaustively enumerated if necessary. Usually, among these alternatives would be the original SEE's analysis of the scene.

3.2 The alternative-list

Each time a splitting heuristic is employed in phase 1 or phase 2 of SEEMORE, a notation is made at the top of a special list called the alternative-list. Each entry in the alternative-list identifies the regions for which the given heuristic has made a splitting decision, called reference-regions, and indicates what step(s) should be taken to generate an alternative splitting analysis for the specified regions. The alternative-list entry also indicates what sort of change can be expected if the given alternative is employed: specifically, the entry may indicate more-splitting, less-splitting, or different-splitting. A more-splitting alternative is expected to split apart two or more currently-conglomerated regions; a less-splitting alternative is likely to conglomerate currently separated regions; while a different-splitting alternative will help to generate an alternative partitioning of the indicated regions. The splitting and linking which is indicated on the alternative-list is described, but not actually carried out unless a complaint is being acted upon.

Notice that the alternative-list is ordered according to a heuristic measure of strength, with alternatives produced by less reliable rules appearing nearer to the top of the list. Later, when processing a complaint, SEEMORE will try these alternatives in order starting from the top of the list, since the alternatives appearing there were produced by heuristics more likely to make mistakes.

3.2.1 Producing alternative-list entries -- phase 1

There is only one alternative to applying a given splitting heuristic in phase 1: not applying it. This will be entered at the top of the alternative-list as a less-splitting alternative, along with the regions between which the given split passes.

3.2.2 Producing alternative-list entries -- phase 2

Let us first consider splits which are extended by one of the split-extending heuristics. In all such cases, the alternative is not to extend the split. This is entered on the alternative-list as a less-splitting alternative, with the bordering regions of the extension split as reference-regions.

For any split generated in phase 2 by the most-complete-split heuristic, two entries are made in the alternative-list. The first of these, a less-splitting alternative, inhibits the split entirely. The other entry, representing a different-splitting alternative, indicates both omitting the original split, and placing an alternative split along the pair of lines originally unsplit.

Figure 3-1 illustrates the rule just stated for a four-line vertex. In figure 3-1A, the most-complete-split heuristic has been applied to vertex a to produce the splits and links shown. At the same time, two alternatives are indicated on the alternative-list. The less-splitting alternative, with no splits or links, is shown in figure 3-1B. The different-splitting alternative is depicted in figure 3-1C. Here the original split has been abandoned, while alternative split c-a-b is made and links are produced as shown.

If the split-between pairs-of-splitting-vertices heuristic is applied, one indicated alternative is to split to a common neighbor vertex, as illustrated in figure 3-2. This is a different-splitting alternative. Additionally, the alternative of no split at all is indicated as a less-splitting alternative.

Throughout phases 1 and 2, whenever a split is inhibited, a more-splitting alternative is recorded which makes the inhibited split.

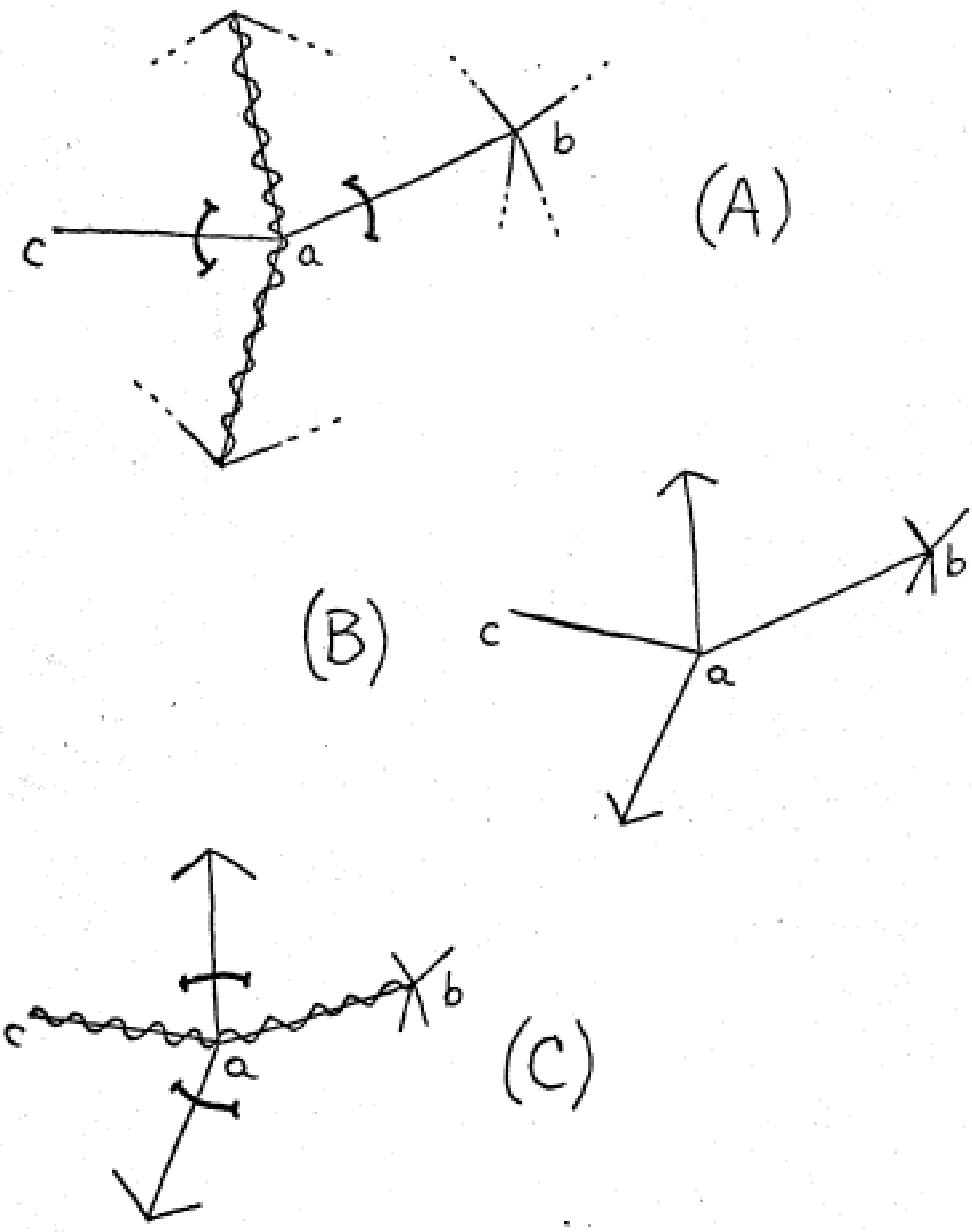


Figure 3-1 Alternatives

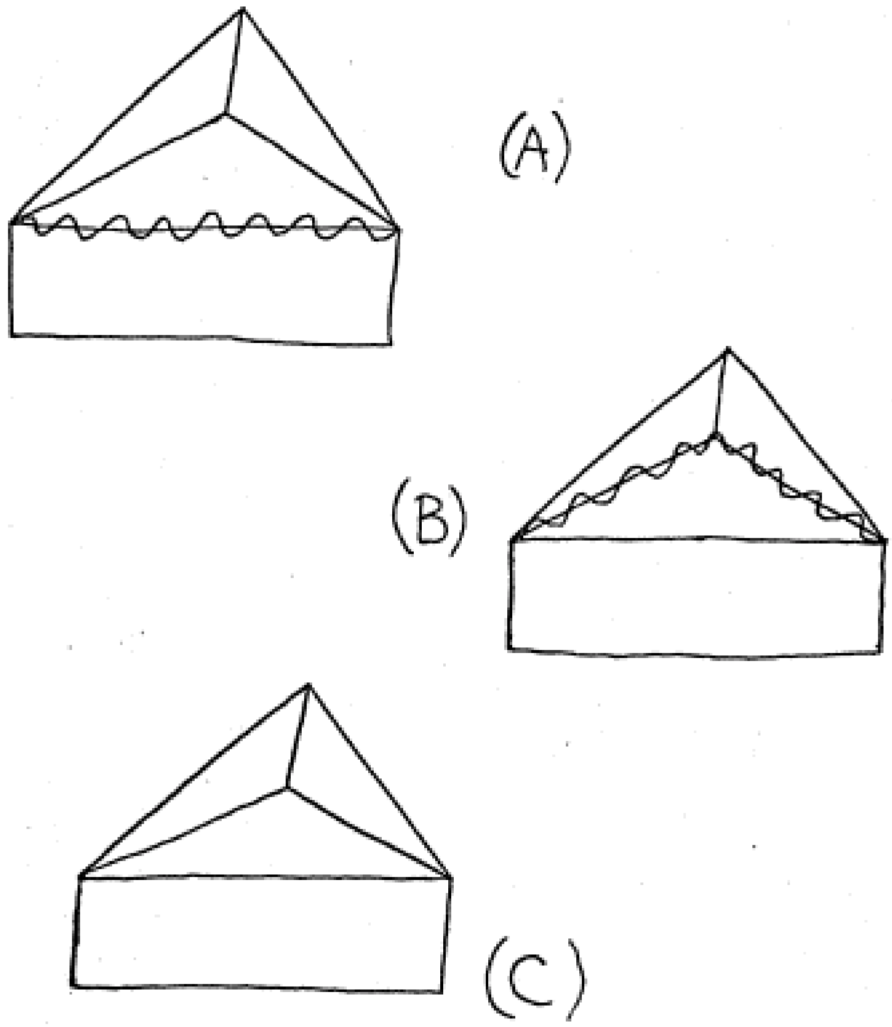


Figure 3-2

3.3 The processing of a complaint

When a complaint is issued by a higher-level program, SEEMORE expects to be provided, as mentioned earlier, with a list of regions, and the nature of the complaint associated with the given regions. Starting at the top of the alternative-list, SEEMORE examines each entry in the list until it finds one whose reference-regions match well with the list of troublesome regions, and which is of a type which would tend to correct the complaint. For example, if the complaint says that a body has too many regions, SEEMORE will look for more-splitting entries with reference-regions from the given body. If an appropriate entry is found, SEEMORE takes the steps indicated in the entry. Then the phase 3 routines are used to complete the linking and see if an alternative analysis has actually been generated. If there are no more-splitting entries in the alternative-list with appropriate reference-vertices, or if no alternative is actually produced by the entries considered, then SEEMORE makes one last effort by searching for different-splitting alternatives with the proper reference-vertices. If no alternatives are ultimately produced, SEEMORE would have to report its failure to generate an alternative analysis. If an alternative is produced, it is submitted to the higher-level program for approval. Additional alternatives could be produced in the same manner if necessary.

If the complaint had been that a body was composed of too few regions, the above steps would be followed, except that less-splitting alternatives would be sought instead of more-splitting ones.

If the higher-level program found the original analysis plausible, but desired an alternative if possible, then different-splitting alternatives would be sought first, followed by more-splitting and less-splitting alternatives if necessary.

Finally, if the higher-level program complained that it could make no sense of SEEMORE's analysis, any type of alternative would be considered. In this case, it is of great help to SEEMORE if the higher-level program specifies regions which it suspects of causing difficulty.

SEEMORE's alternative-generating procedure is successful in producing the alternatives shown for the simple ambiguous scene of figure 3-3, and the more complicated scene of figure 3-4.

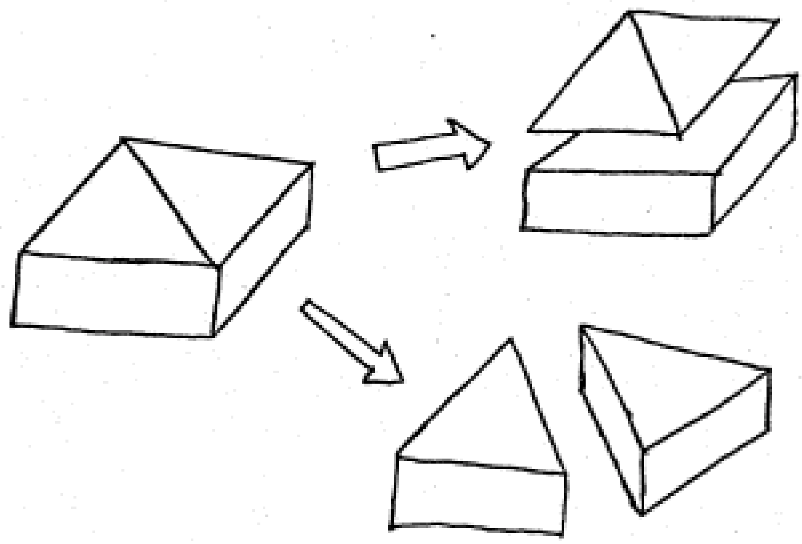


Figure 3-3

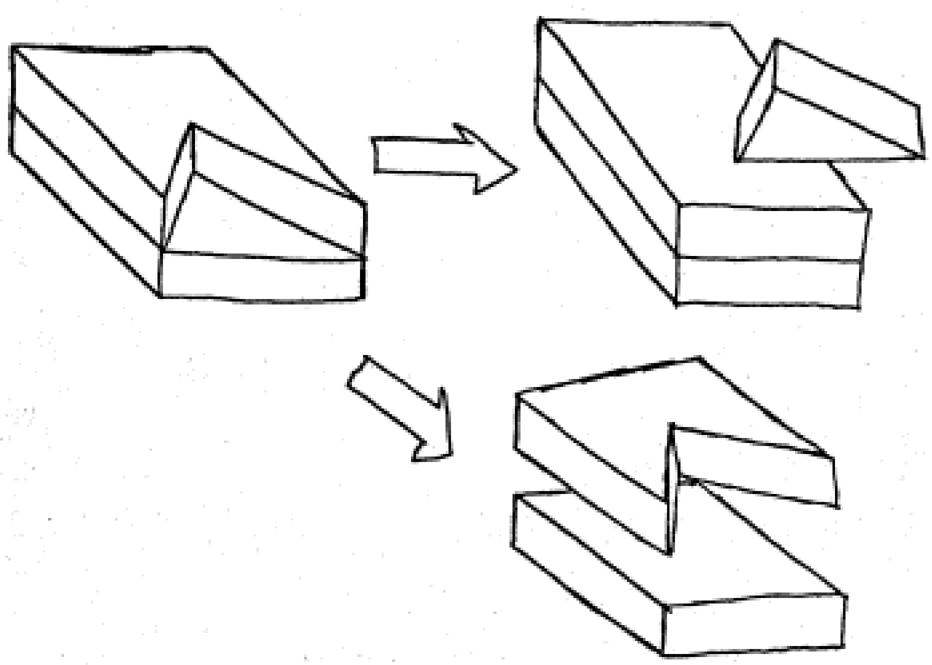


Figure 3-4

4 Some examples

4.1 A detailed example

I can now explain in detail the operations performed by SEEMORE in analyzing the example presented earlier, which is reproduced in figure 4-1.

Links made by the initial linking routine are depicted in figure 4-2A. Nuclei (2 3) and (9 10) are proposed as unoccluded bodies which, presumably, are rejected by the higher-level procedure. Figure 4-2B shows the link diagram at this point. As shown in figure 4-3, the initial splitting routine initiates splits along a-f, b-c, and j-e. This completes the initial splitting and linking phase (phase 1).

In the additional-splits phase (phase 2), further splitting operations are performed. First, the split along b-c is extended to vertex h, which also links regions (5 6) and (9 8) at vertex c. Similarly, split a-f is extended to g, and split j-e is extended along e-g. Then split h-d-k is generated at splitting-vertex d by the most-complete-split heuristic. The split h-d-k also produces the links (7 11) and (8 10) at vertex d. Additional-splits phase 2 has now been completed, as illustrated in figure 4-4.

Finally, phase 3 is invoked. The remaining strong and weak links are placed, as illustrated in figure 4-5. The corresponding link diagram is shown in figure 4-6. Next, procedures similar to the routines of SEE are used to conglomerate regions on the basis of the strong and weak links which have been placed. The final nuclei are shown in figure 4-7. SEEMORE reports the four bodies (2 3 4), (5 6),

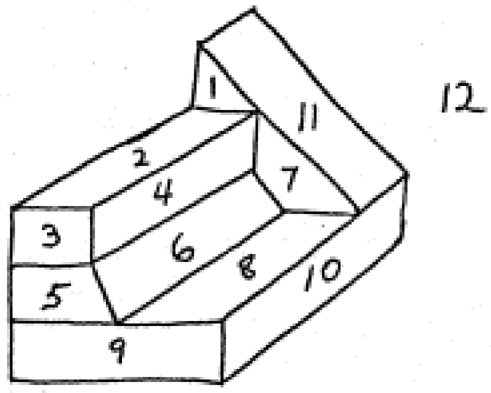


Figure 4-1

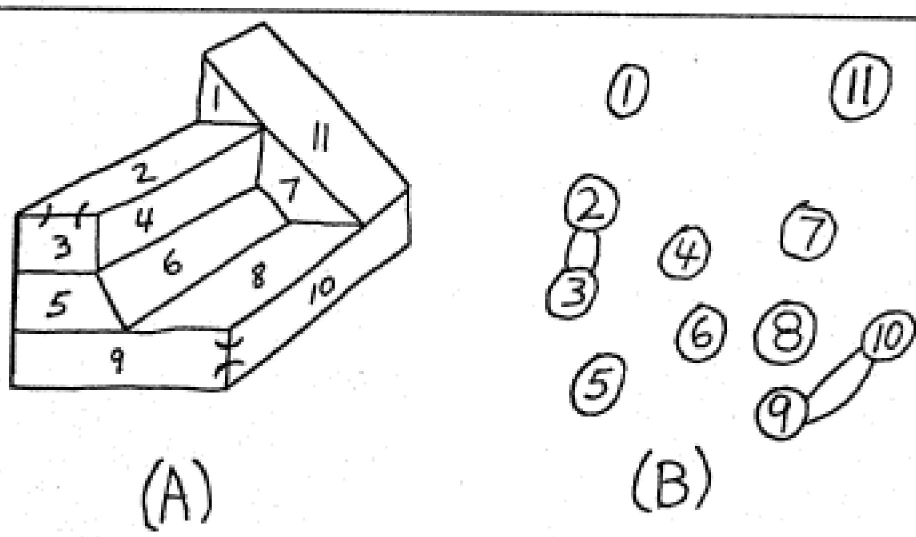


Figure 4-2

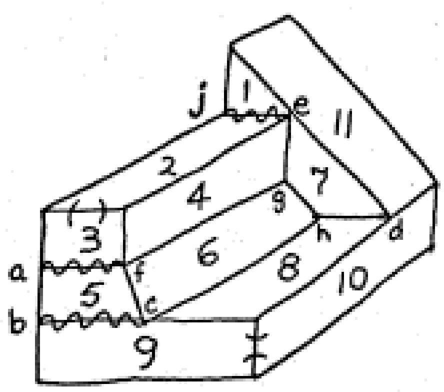


Figure 4-3

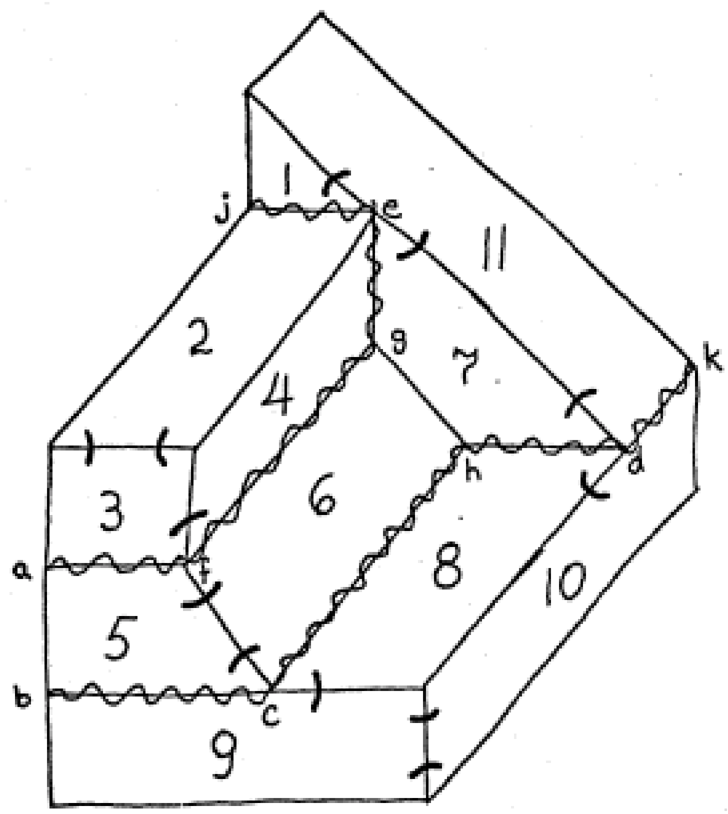


Figure 4-4

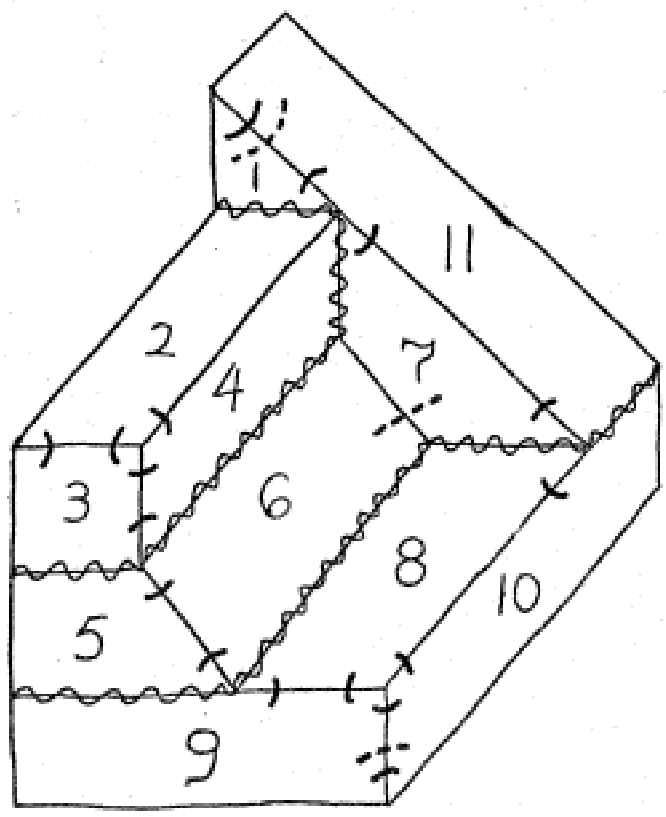


Figure 4-5

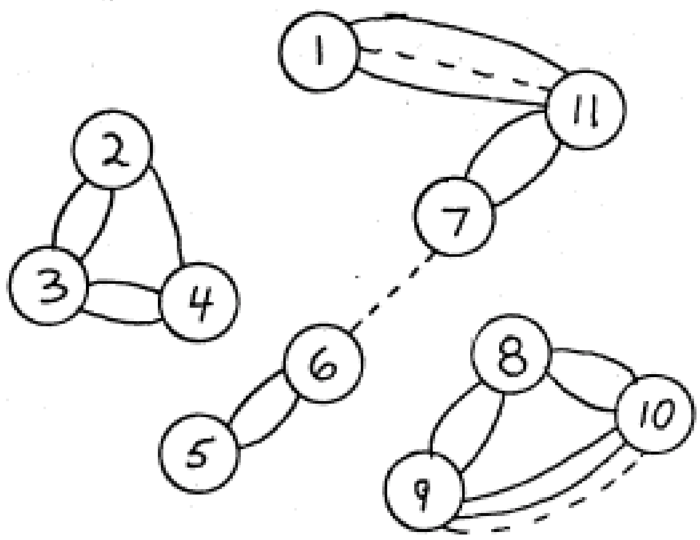


Figure 4-6

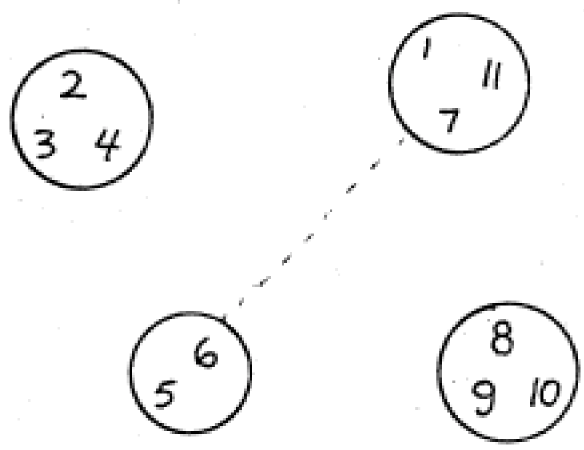


Figure 4-7

(8 9 10), and (1 7 11).

4.2 Other examples

Figures 4-8 through 4-12 present several additional examples in abbreviated form. For each example the scene is presented with all splits and links indicated, and the bodies found by SEEMORE are listed. In the examples of figures 4-9 and 4-12, an alternative analysis generated by SEEMORE is also shown.

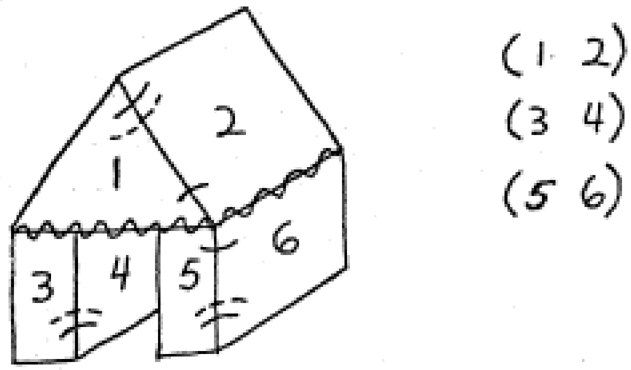


Figure 4-8

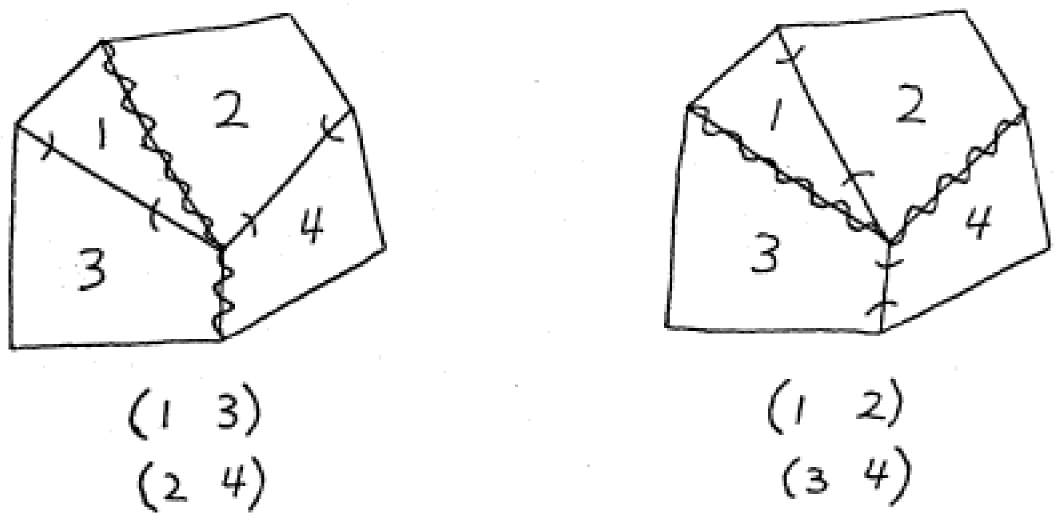
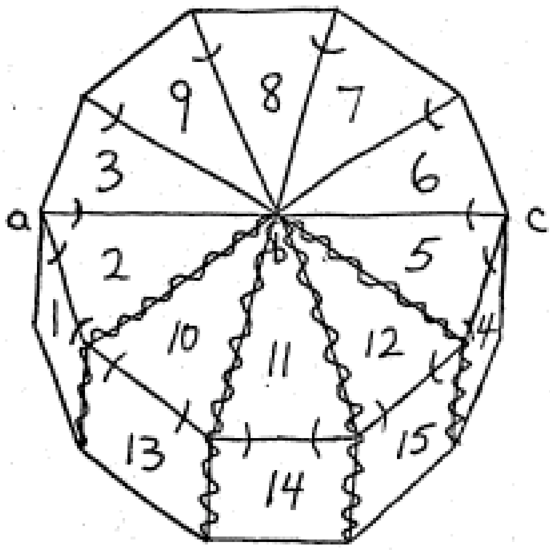


Figure 4-9

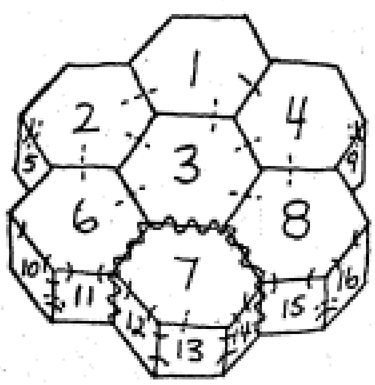


(3) (7) (10 13) (5 4)

(9) (6) (11 14)

(8) (1 2) (12 15)

Figure 4-10



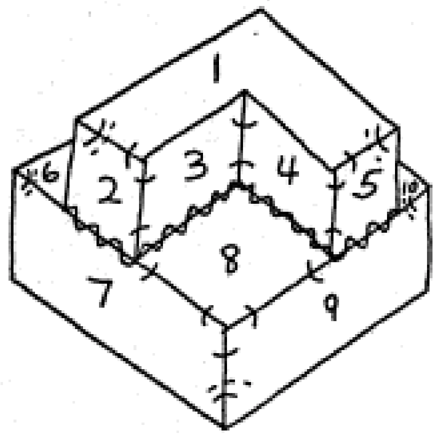
(1) (6 10 11)

(3) (8 15 16)

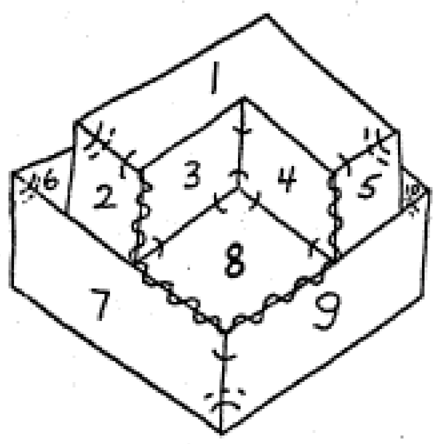
(2 5) (7 12 13 14)

(4 9)

Figure 4-11



(1 2 3 4 5)
(6 7 8 9 10)



(1 2 5)
(3 4 8)
(6 7 9 10)

Figure 4-12

5 Failures of SEEMORE and recommendations

SEEMORE often relies heavily for success upon the technique of conglomerating and recognizing unobscured objects early in its analysis. For example, in figure 5-1, SEEMORE correctly splits along c-a-b-e by first verifying object (1 2). Since the corresponding object in figure 5-2 is not fully visible, splits c-a-d and e-b-d will be generated. Consequently, SEEMORE reports objects (1 2 3), (4 6), and (7 5 8 9). Notice that figure 5-2 is a difficult scene for humans, too.

Although SEEMORE does provide for interaction with a higher-level routine, perhaps this interaction has not been carried far enough. For example, an extended version of SEEMORE might be aware of the kind of body it was looking for, say a rectilinear brick. It might recognize a FORK as the possible corner of a brick, then extend the lines of the FORK looking for the other corners. Or perhaps, having recognized an object in the scene, it would be able to hypothesize what the scene would look like if that object were removed. It could then proceed to analyze the simpler scene. Procedures with such capabilities could only result from a major overhaul of the present vision system.

Another kind of problem results from certain scenes in which crucial splitting evidence is obscured. An example of such a scene is figure 5-3. What is needed here is a line-extender, whose use might be indicated by the T-joints at a, b, c, and d.

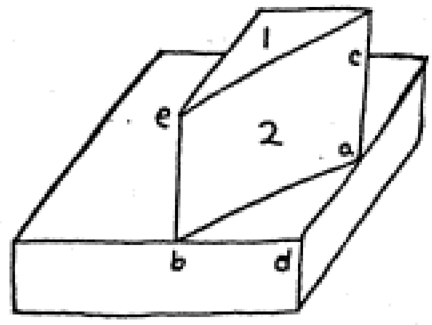


Figure 5-1

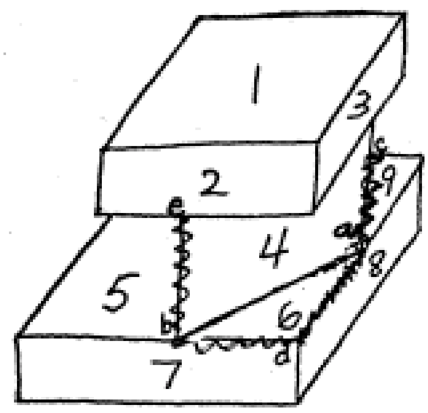


Figure 5-2

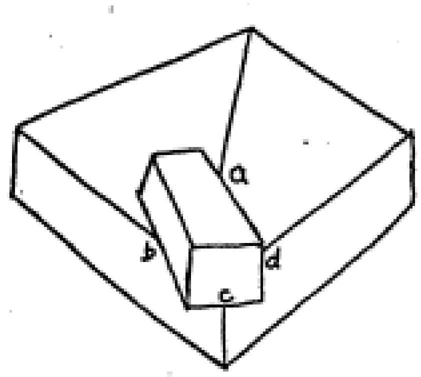


Figure 5-3