PROGRESS IN EXTENDING THE VIRGIN PROGRAM

VISION FLASH #20

Mark Dowson

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

Vision Group

September 1971

ABSTRACT

The VIRGIN program will interpret pictures of simple scenes. This paper describes a program, SINNER, which will deal with pictures which contain cracks and shadows. In addition to handling pictures of this richer world, SINNER employs heuristics which use knowledge about the structure of the three dimensional world to reduce the number of interpretations of some pictures and to augment the efficiency of the parsing process.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0002.

Introduction

Vision Flash #14 (1) discusses extensions of the VIRGIN picture parsing program (2), which assigns interpretations to simple straight line drawings, to handle pictures which include "cracks" and "shadows". This proposed extension consists of two parts. First, the set of junction theorems is expanded to include labelings of "crack" and "shadow" junctions. Second, a number of "heuristics" are described which will act to reduce the number of possible interpretations of each picture. Both these types of extension correspond to additional knowledge about the visual world and its representation in pictures. The extra junction theorems carry information about how the local features of a richer three dimensional world are represented in a two dimensional picture; the heuristics embody knowledge of a more global nature about the structure of the three dimensional world so represented.

The SINNER Program

A program has been written, called SINNER, which includes a subset of the additional junction theorems and some of the heuristics described in Flash #14. Even this partial extension raises enough issues to be worth discussion at this stage. Figure 2 shows the junction labelings which are embodied as junction theorems in the original VIRGIN program while figure 3 shows the additional shadow and crack labelings included in

SINNER. In these labelings, an arrow across a line indicates that the line represents a shadow edge with the shadow on the side of the arrow head. A line marked 'CR' represents a crack. Note that, under the assumptions given in Flash #14, the two regions on either side of a "crack" or "shadow" line represent coplanar surfaces. Figure 1 shows a picture of a scene with both cracks and shadows. It is labeled with one possible interpretation to illustrate these conventions.

FIGURE 1

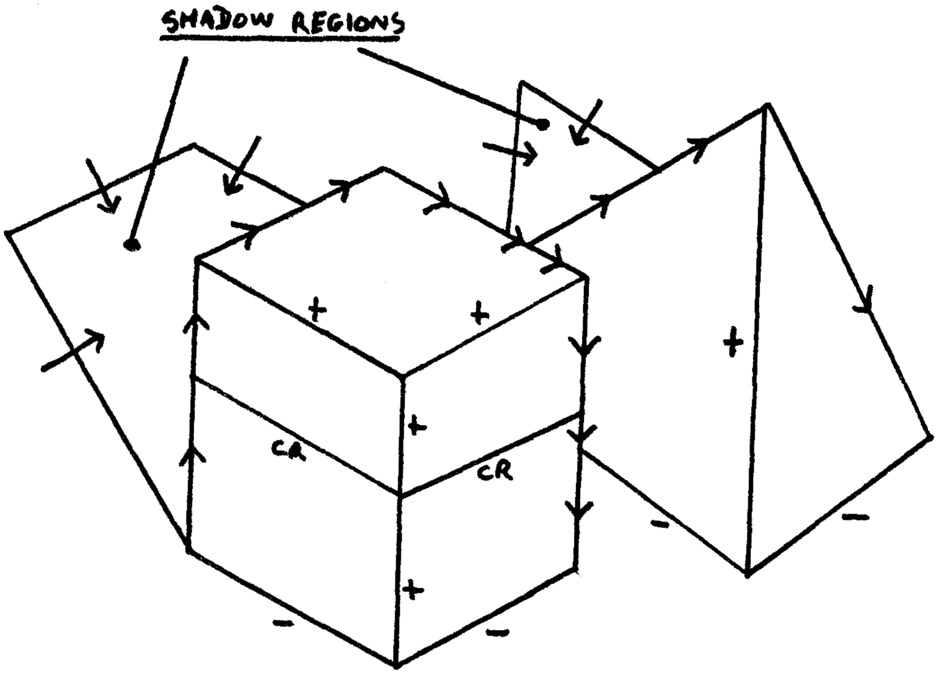


FIGURE 2

The VIRGIN Junction Labelings

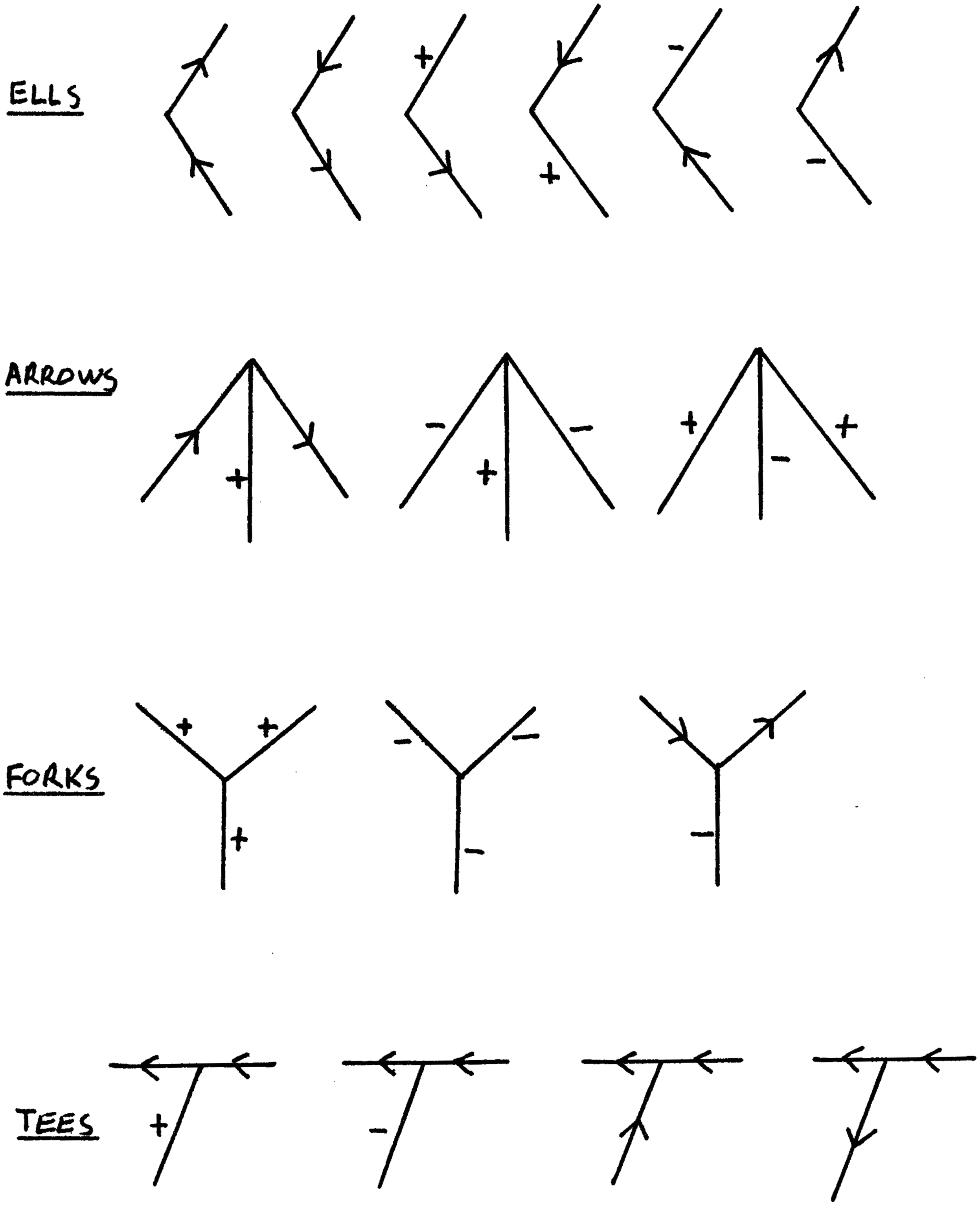
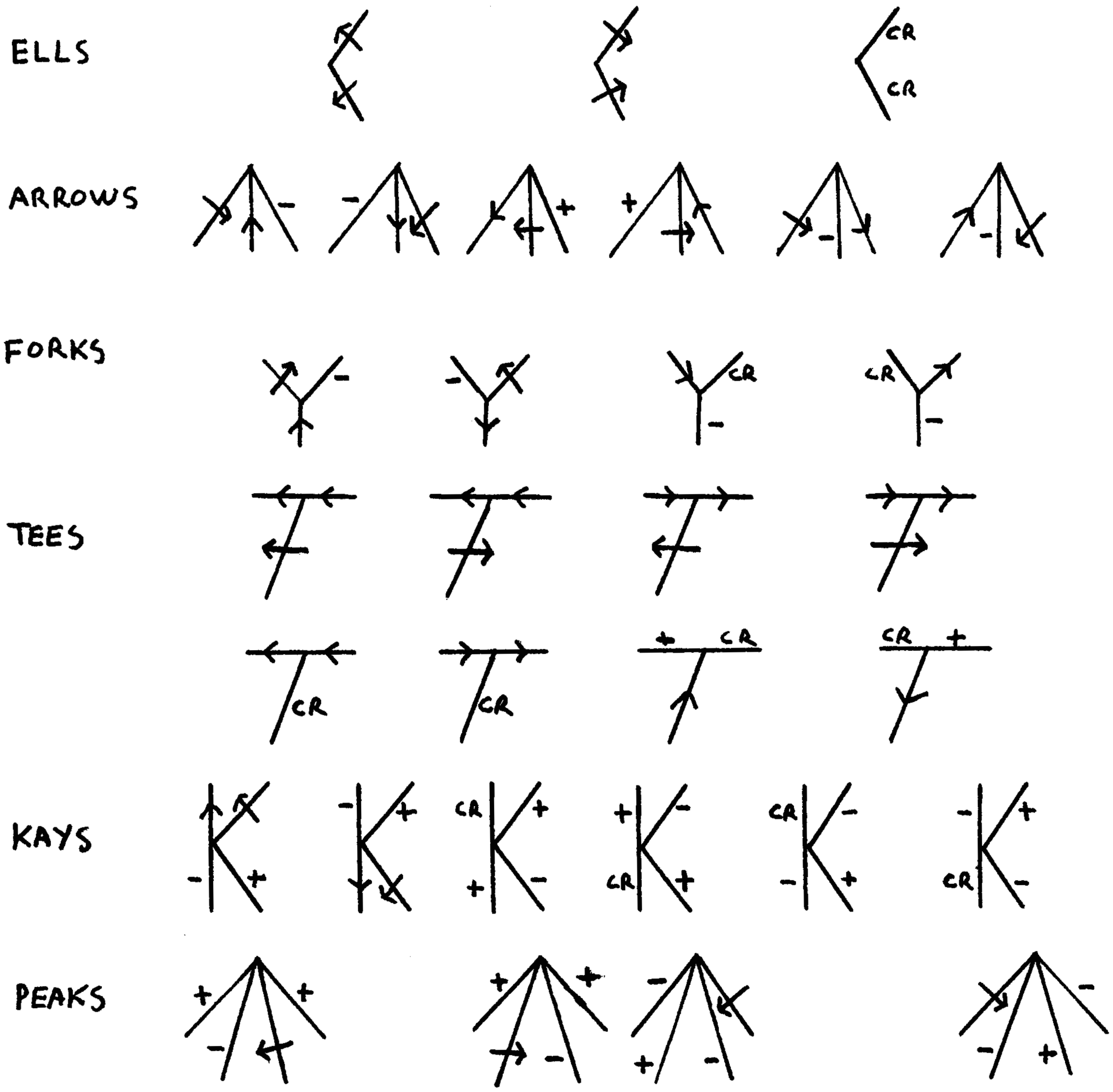


FIGURE 3

Additional Junction Labelings Included in SINNER



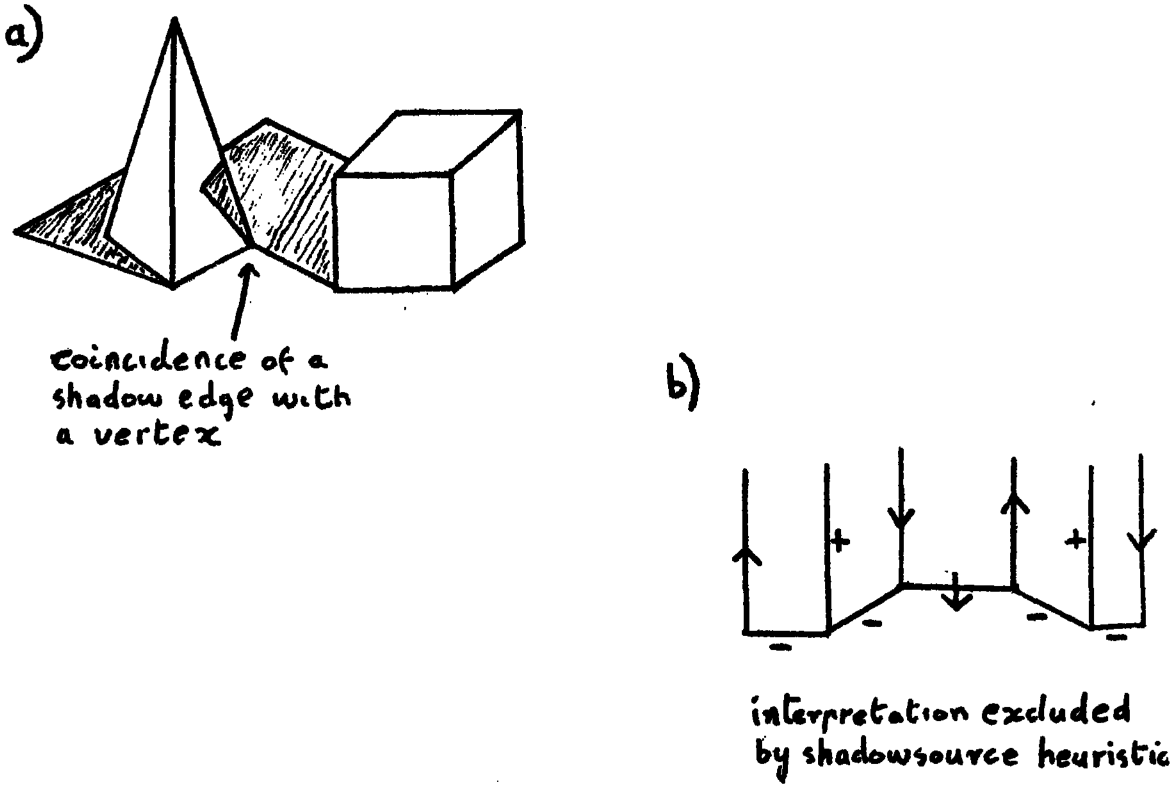
The SINNER Heuristics

The term "heuristic" carries a connotation which is unfortunate in this context; a trick which may or may not work but is useful if it does. In general heuristics are, in a disguised form, the programmer's unarticulated knowledge about the probable structure of the world his program lives in. The heuristics described below are quite different. They are intended to articulate, explicitly, knowledge about the structure of the domain of the program. They can be thought of as the procedural analogues of theorems in projective geometry, optics and mechanics.

The "Shadow Source" Heuristic

One constraint on the scenes and viewing positions which yield pictures that SINNER will interpret is that small changes in the viewing position or in the position of the single light source illuminating the scene will not alter the topology of the resulting picture. This, in particular, excludes pictures like that shown in figure 4a where one body in the scene casts a shadow which falls precisely on a vertex of another body.

FIGURE 4



coincidence of a shadow edge with a vertex

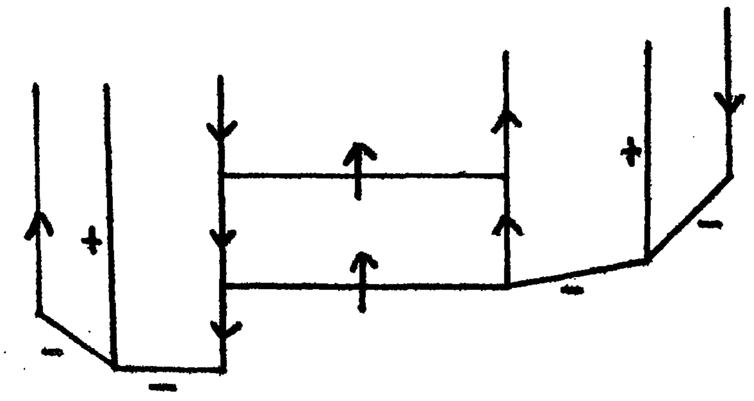
interpretation excluded by shadowsource heuristic

Thus, no line joining a pair of junctions which must represent real, physical, vertices may be interpreted as a shadow edge. When such a junction (Any ARROW, FORK, KAY , PEAK or MULTI) receives an interpretation which labels one of its lines as a shadow , an assertion is made which marks the junction at the far end of the line as a "SHADOWEND". In addition, each such junction theorem tests that the junction it is about to label is not so marked. Figure 4b shows an otherwise acceptable labeling of a fragment of a picture excluded by this heuristic.

The SHADOWREGION Heuristic

Two adjacent regions may represent the shadowed portions of a pair of surfaces. If there is but a single light source the two surfaces must be physically distinct -- which is to say that the line separating the regions cannot be a shadow line. This constraint is embodied in SINNER by marking as a "SHADOWREGION" the region on the appropriate side of each line labeled as a shadow edge; then, as each junction labeling is completed, a test is made of any new SHADOWREGION to make sure that it does not share a shadow line with any other SHADOWREGION. Figure 5 shows a labeling of a picture fragment excluded by this test. The procedure which marks regions as SHADOWREGIONS is an antecedant theorem, one of a family invoked by each labeling assertion within each junction theorem. Other members of this family of antecedant theorems make assertions which are used in the heuristics next described.

FIGURE 5



Region Predicate Heuristics

Assigning an interpretation, as an edge, to a line which associates a pair of regions articulates the relative inclination of the two surfaces that the regions represent; that is, it tells whether the surfaces are coplanar or meet at a relative angle of greater or less than 180 degrees.

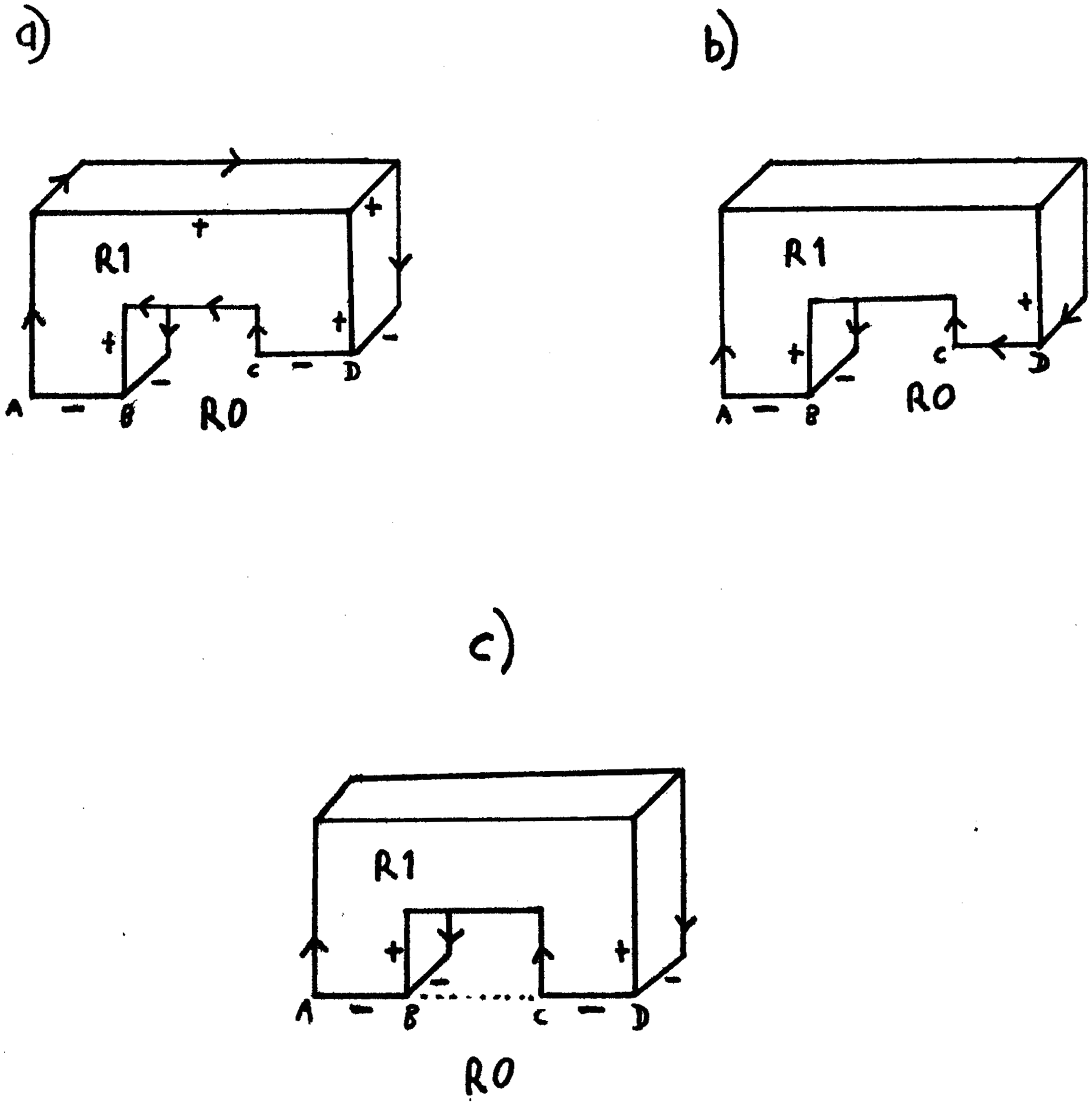
Whenever a line associating a pair of regions (as do all lines) is labeled, one of the family of antecedant theorems mentioned above asserts a predicate of the form:

(PRED J RA <predicate> RB)

Where RA and RB are the names of the two regions, J the name of the junction being labeled and <predicate> is COPLANAR, VX or CV appropriately. Lines labeled "crack" or "shadow" give rise to a COPLANAR predicate, lines labeled "VX" or "CV" to VX or CV predicates respectively.

Now should a pair of regions have more than one line in common, the interpretations that the lines may receive are constrained. No two regions may be associated by more than one of the predicates above at a time and, if two or more lines all give rise to VX predicates or all give rise to CV predicates, the lines must be colinear. Figure 6 illustrates these constraints. The labeling given in fig. 6a is unacceptable as the lines AB and CD are both common to the pair of regions R0 and R1 so at least one of them must take an 'occluding edge' interpretation as in fig. 6b unless they are colinear as in fig. 6c.

FIGURE 6



A simple extension of these constraints is to make the predicate COPLANAR transitive and treat any pair of adjacent, coplanar, surfaces as a single surface for tests based on the above constraints. More complex extensions are possible. For instance (representing the full predicate as (RA <predicate> RB) for clarity)

$$(\mathrm{RA\ VX\ RB}).(\mathrm{RB\ VX\ RC}) \supset \sim(\mathrm{RA\ COPLANAR\ RC})$$

and so on. These extensions have not yet been investigated systematically.

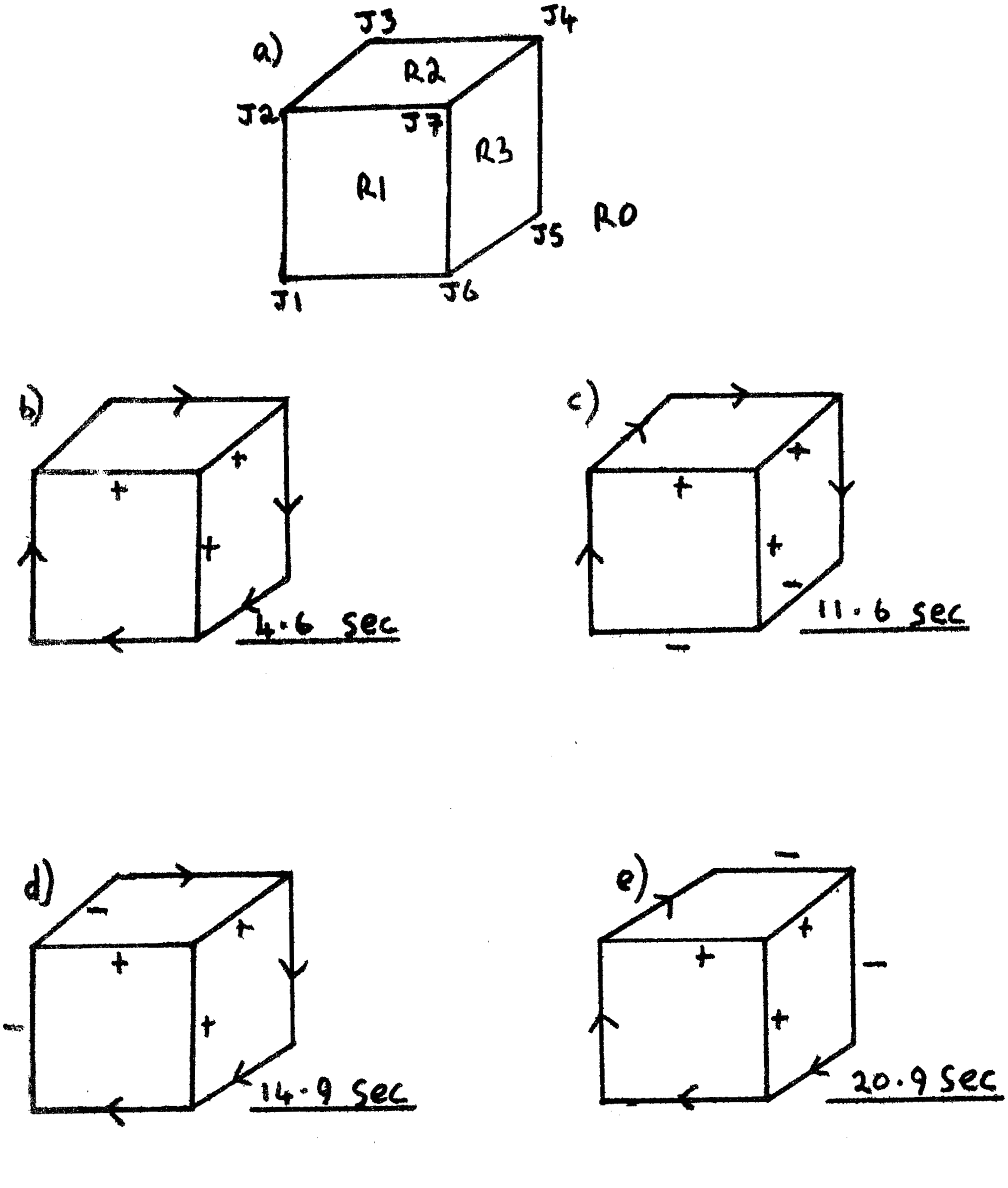
A test to see whether any region predicate constraints have been violated is made after each junction has been labeled. The present version of SINNER merely tests to see that no more than one of the lines common to the same pair of regions has received a VX or CV label, but extending this to include the whole range of region predicate constraints will be a simple matter.

Results

Although the primary intention of the heuristics described above is to exclude some interpretations of pictures which would otherwise be admissable, they augment the efficiency of the parsing algorithm in other ways. Some junction interpretations which would only be discovered to be incorrect later in the parsing are rejected immediately by the heuristics. In the current implementation this provides only a marginal saving in computation time since the mechanisms for making the tests are relatively "expensive" and only a subset of the possible heuristics are actually applied; but it is expected that, with a fuller set of heuristics, the saving in time will be more significant, particularly with more complicated pictures.

For comparison, figure 7 illustrates the performance of VIRGIN on a simple picture. Figure 7a is the picture presented as data to VIRGIN (in the form of a list of angle and junction type assertions) and figures 7b to 7e show the resulting parsings. The time given beside each interpretation is the processor time taken, after starting the program, to produce the interpretation. These times are given for comparison only since the absolute times depend entirely on the implementation. The discrepancy between the last of these and the ' total time' represents the time taken, after producing the last possible interpretation, to discover that no other interpretation is possible. Time taken up in garbage collection has been subtracted throughout.

FIGURE 7

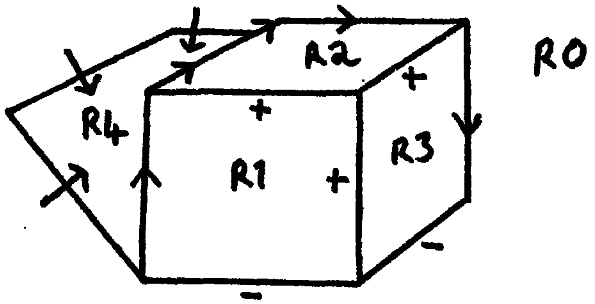


total time = 24·6 sec

The four interpretations correspond to either a cube invisibly supported clear of the surface represented by the background of the picture or stuck to it along some pair of edges of the cube.

The picture of figure 8 has only a single interpretation with the region R4 representing a shadow cast by the cube on its supporting surface. SINNER takes 15 seconds to produce this interpretation and a further 65 seconds to discover that no further interpretations are possible.

FIGURE 8



Note that no explicit information tells SINNER that R4 is a shadow; there is no other possible interpretation under the constraints we have set up. The picture of figure 9 is more ambiguous. Figures 10a through 10f give the interpretations that SINNER returns. Here there are four different interpretations of the junction J27, two of the junction J26 and two of J25. Not all of the 16 combinations are possible, however, as interdependencies amongst the interpretations of the junctions reduce the number of possible interpretations to six.

FIGURE 9

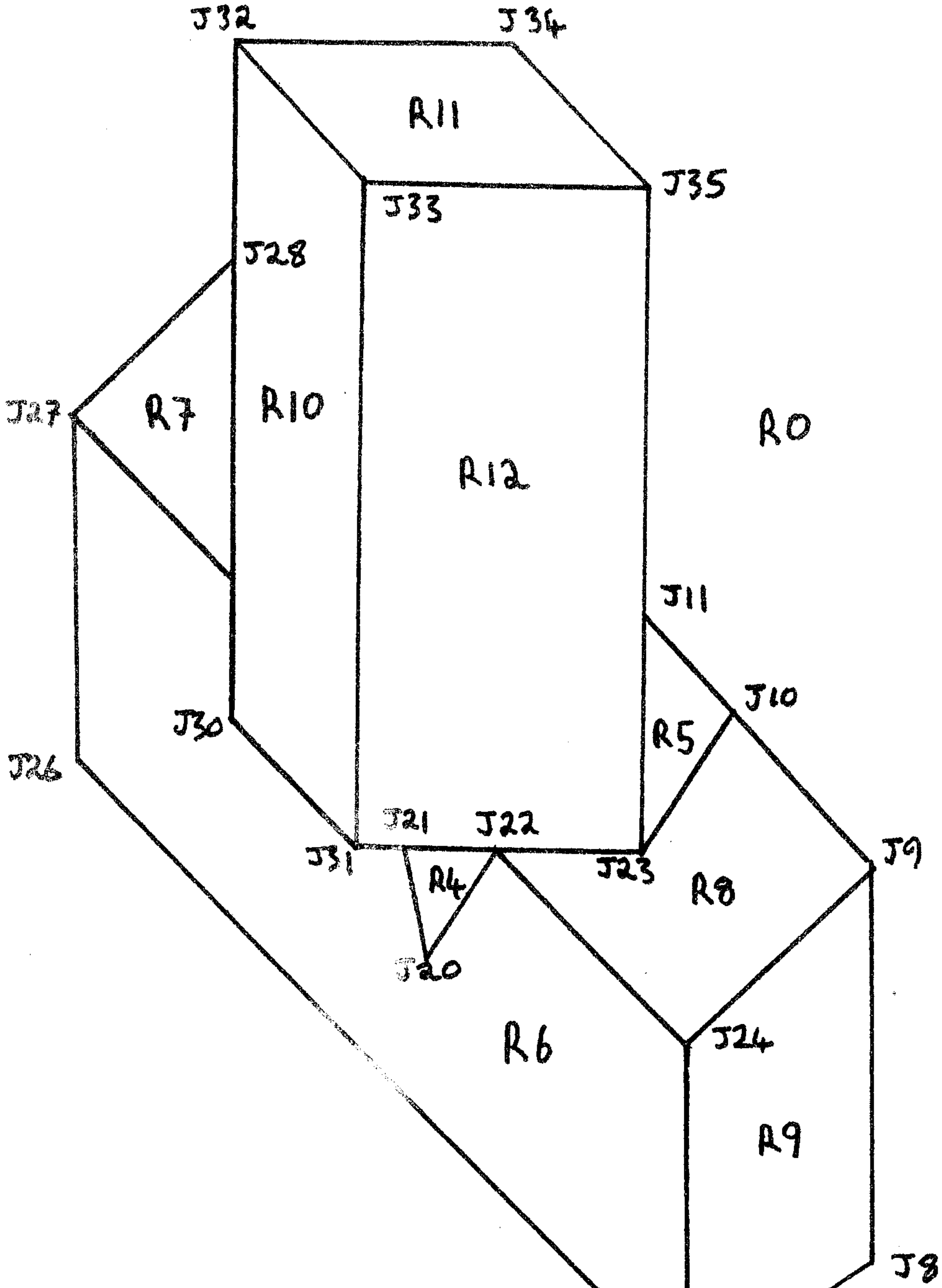

J32
J34
R11
J35
J33
J28
J27
R7
R10
R12
R0
J11
J10
J30
R5
J26
J21
J22
J31
J23
J9
R4
R8
J20
R6
J24
R9
J8
J25

FIGURE 10a

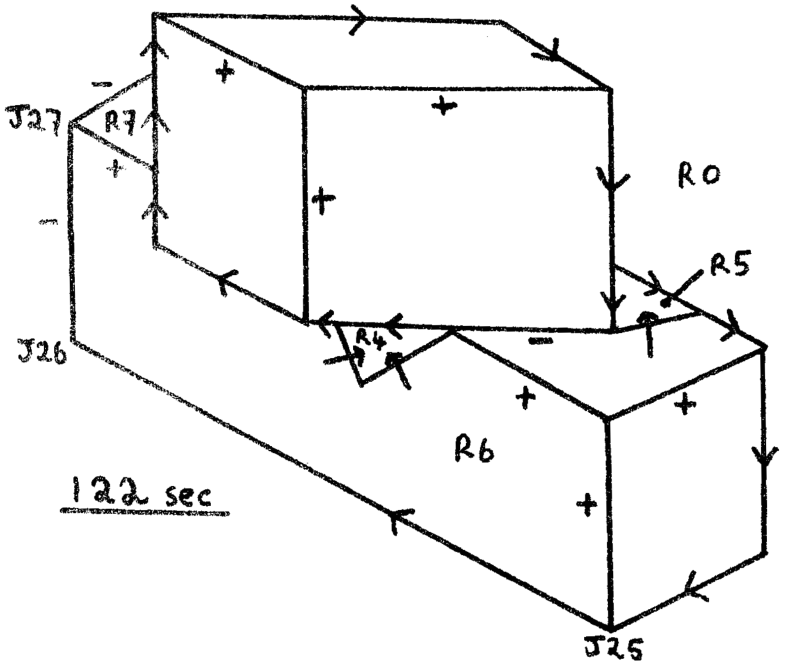
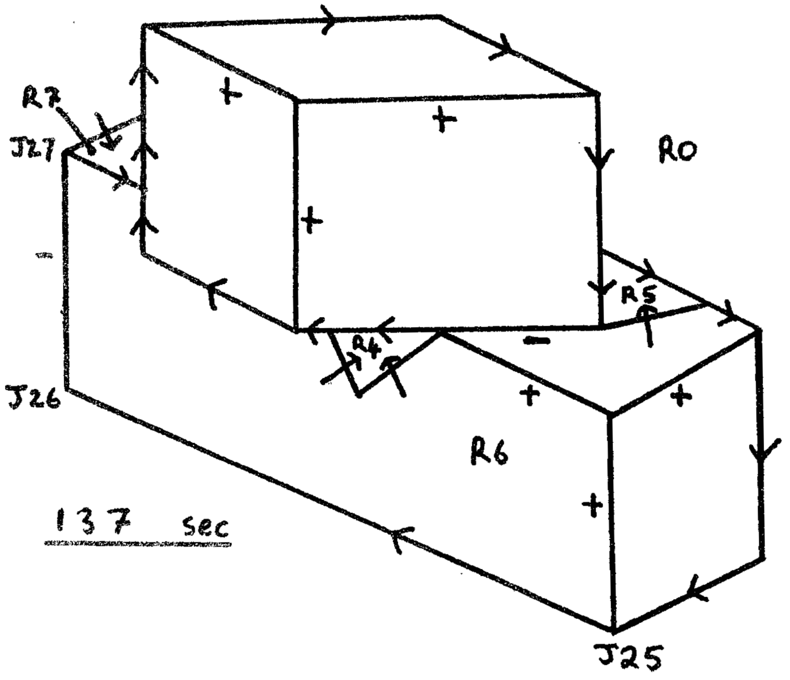


FIGURE 10b



a 'shadow direction' heuristic would exclude this interpretation

FIGURE 10c

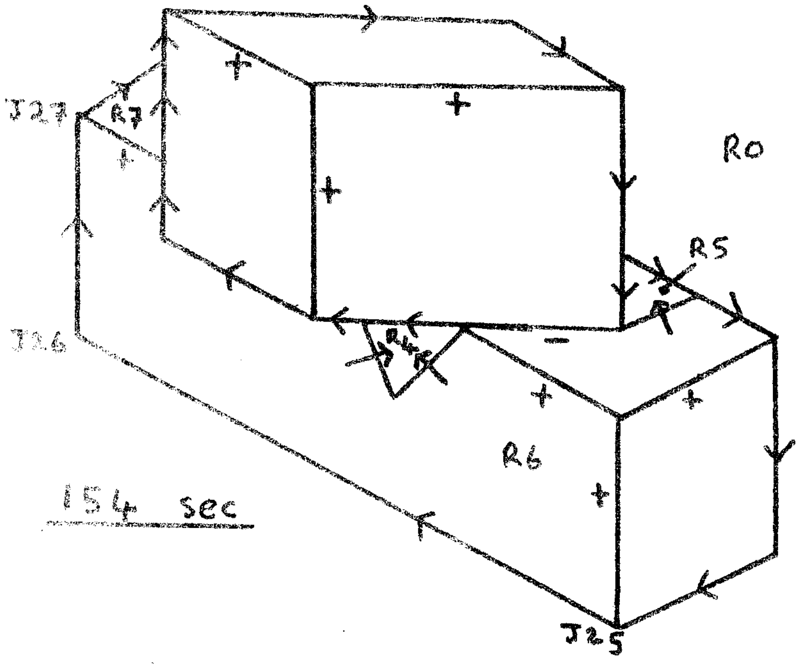
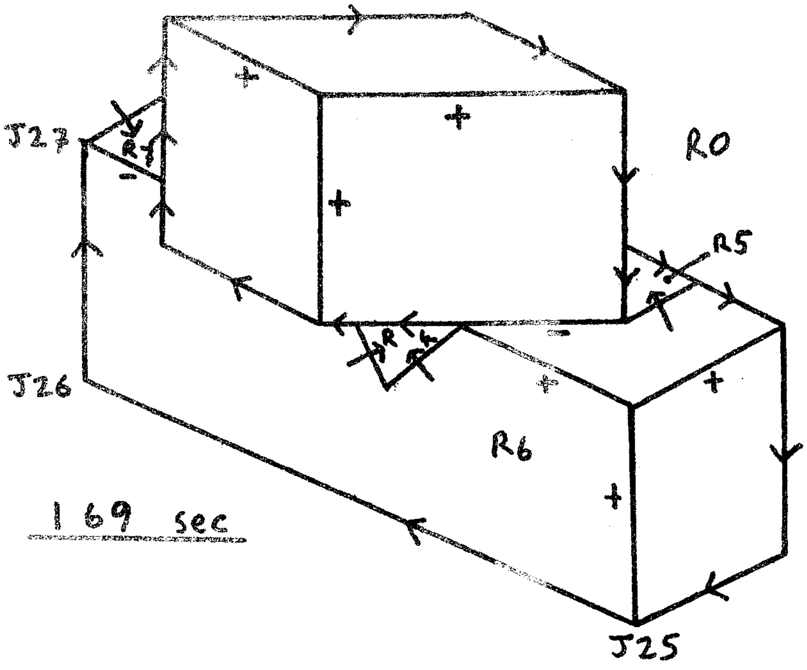


FIGURE 10d



a 'shadow direction' heuristic would exclude this interpretation

FIGURE 10e

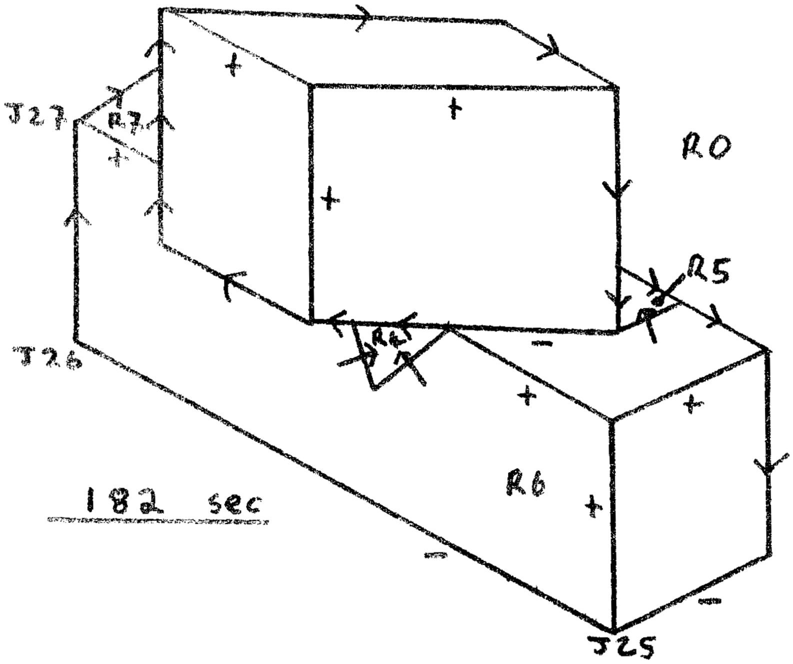
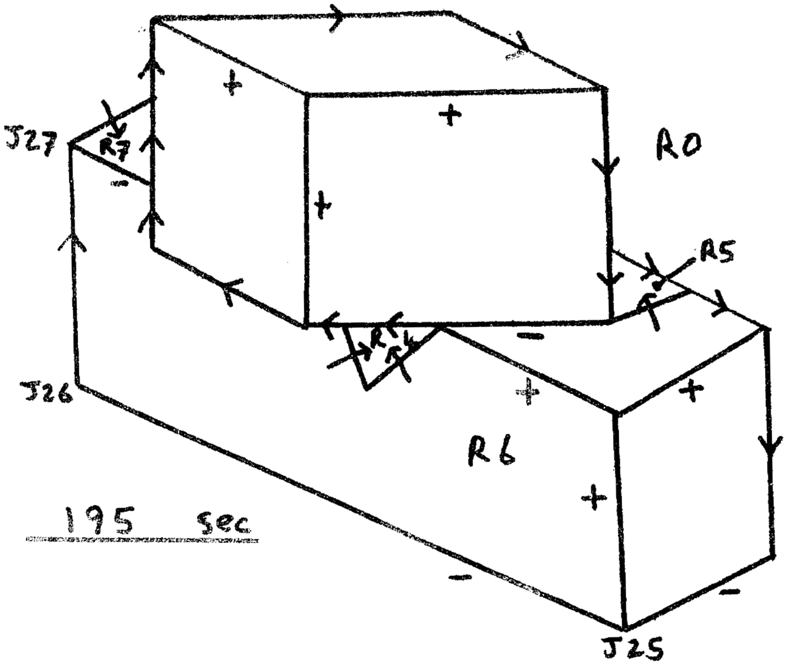


FIGURE 10f



transitive region
predicate heuristics
would exclude this
interpretation

'total time' = 1360 sec

It is now possible to see how more knowledge about the world can reduce the number of interpretations still further. The interpretation given in 10f would not appear if the region predicates were made transitive as discussed above in the section on Region Predicate heuristics. R7 would be treated as coplanar with R0 both of which are concavely related by non-colinear lines to R6. The two other interpretations which give R7 as a shadow region, 10b and 10d, could be excluded by an heuristic which took into account directions of shadows since if R4 and R5 are shadow regions the direction of the light source must be such that there is no object in the scene to cast a shadow in the position of R7. This leaves three interpretations which correspond to the intuitively acceptable ones of the bottom block being attached to a wall (10a), invisibly supported (10c) or attached to the floor (10e).

The picture of figure 11 has only two interpretations shown as 11a and 11b, and it is clear that the interpretation of 11b can be excluded by the same heuristic as would exclude 10f above. However, the very long time that SINNER takes (of the order of forty minutes) to produce these interpretations exposes a major deficiency in the basic structure of the SINNER program.

FIGURE 11a

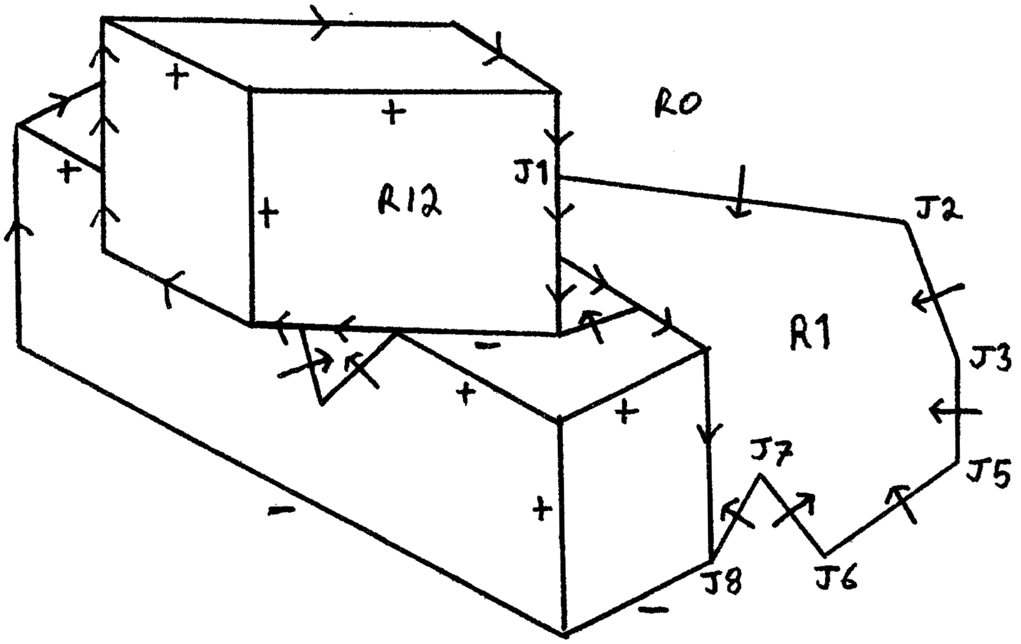
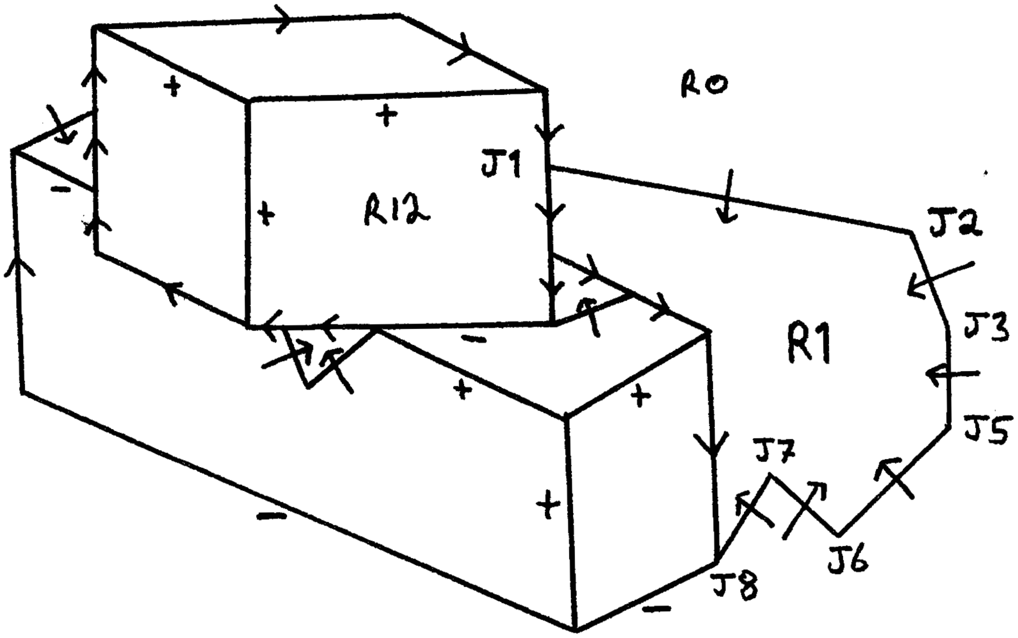


FIGURE 11b



What happens is that, as one of the first regions to be tackled is R12, the T-junction J1 is one of the first to be labeled. Due to the order of junction theorems in the database the shaft of the T gets labeled as an occluding edge. Now this is incorrect as the only possible interpretation of R1 is as a shadow region. However, since R1 is the last region to be dealt with, this error can only be corrected after backing up through almost the whole interpretation tree trying different side branches, unsuccessfully, at each level. What is worse is that when backup finally reaches J1, the interpretation next chosen for the shaft of the TEE is not the right one, but another wrong interpretation. In all this process is repeated five times before the right interpretation surfaces. Clearly this is a stupid way to interpret this picture. For this particular picture the trouble could be cured by reordering the theorems in the data-base, but then the same trouble would be encountered on some different picture. There are other possibilities. One might be to defer the interpretation of the shafts of TEE's until the end of an interpretation. Although this would eliminate some of the worst trouble, it would destroy the uniformity with which the top level program handles various types of picture fragment.

A more general approach would be for the program to take some overall view of the picture in advance of trying to make a detailed interpretation so as to formulate some strategic plan about how to conduct the interpretation -- which region to tackle first, and so on. The final approach is likely to use some combination of this and other methods, but as yet the problems involved are not even well formulated, much less clearly understood.

Conclusions

Extending the VIRGIN program to deal with a richer world and to deal with that world less naively has answered some questions about this approach to picture interpretation and has opened many avenues for future work. We have seen that the approach first delineated by Clowes (3) and Huffman (4) is still applicable in a richer visual world. The effort required to parse a picture grows with the increased variety of picture features and with the complexity of the picture, but not at a rate which makes the approach impracticable. Knowledge about the world less local than that carried by a table of permissable junction labelings will reduce the ambiguity otherwise inherent in some pictures and decrease the effort required to interpret others.

Work in the immediate future will concentrate on increasing the amount of such knowledge available to the SINNER program. This will include some knowledge of geometry; here the theoretical framework provided by Clowes et al (5) will be of use. Before long, however, the more challenging and difficult problem will have to be faced of getting the program to formulate some strategic plan in advance of actually making an interpretation of the picture. This, together with other very global methods of improving the performance of the parsing algorithm, corresponds to a more 'general' kind of intelligence. It seems likely that the results obtained in the course of this exercise will be of wider applicability than to just this single vision program and will open up many more avenues for future research.

REFERENCES

(1) Dowson, Mark & Waltz, David 'Cracks and Shadows.'
A.I. Lab. Vision Flash 14. 1971

(2) Dowson, Mark 'VIRGIN: a program which interprets line drawings.'
Artificial Intelligence Lab. M.I.T.
Memo No.244. (In progress)

(3) Clowes, M.B. 'On Seeing Things'
Artificial Intelligence Vol.2, 1971

(4) Huffman D.A. 'Impossible Objects as Nonsense Sentences'
Machine Intelligence 6. Ed. Collins & Michie 1970.

(5) Clowes M.B.
Mackworth A.K.
Stanton R.B. 'Picture interpretation as a problem-solving process'
Lab. of Experimental Psychology
University of Sussex June 1971.