UNDERSTANDING SCENES WITH SHADOWS

VISION FLASH 21

by

David L. Waltz

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

Vision Group

November 1971

ABSTRACT

The basic problem of this research is to find methods which will enable a program to construct a three dimensional interpretation from the line drawing of a scene, where the scene may have shadows and various degeneracies. These methods differ from those used in earlier related programs in that they use region information extensively, and include formalisms for eye and lighting position. The eventual result of this research will be a program which should be able to successfully treat scenes with far fewer restrictions than present programs will tolerate.

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

INTRODUCTION

In the body of this paper, I deal primarily with the concrete results I have obtained so far, but I think that the significance of the results may not be immediately apparent. Therefore I believe that it will be useful to say something about how I approached this research, about why the results are of the sort they are, and about what relevance this work has to human perception.

Why does it seem so easy for us to apprehend rapidly the three-dimensional nature of scenes, even when they are presented to us in the form of badly exposed and badly focussed photographs of badly lit and unfamiliar objects? While it is possible to demonstrate a large number of optical illusions, I think that the usual reliability of my perceptions is a far more striking phenomenon. Because we can perform as reliably as we can, I began with the basic assumption that scenes provide many redundant pieces of information. What kinds of information and rules could serve to explain the observed performance?

Some cues and rules had already been demonstrated by Huffman, Clowes and others. Their work has shown first that the sets of edge types and junction types which can occur in a scene made up of objects with planar surfaces and trihedral junctions can be easily enumerated, and then that we can only interpret a line as a particular type if we can label the entire scene from the set of allowable junctions in a manner consistent with this line assignment. Clearly this approach cannot provide a sufficient explanation of visual perception, since in general any scene can be labeled in several different ways, and there is no obvious way to select the interpretation we would call correct. My first idea was to extend the set of allowable junctions to include shadow junctions, and I found that in simple scenes with shadows we can sometimes parse uniquely because shadows in some sense provide additional information. However in more complex scenes, the larger set of labels merely compounded the problem of multiple interpretations. This led me to think more about the nature of our perception of scenes with shadows.

Why does it seem easy to interpret scenes regardless of lighting, and in particular, how are we able to "factor out" shadows, even in cases where they are the most prominent features in terms of contrast? My search for rules began with the observation that if a scene is lit with a single light source we can partition the set of junctions into those which can occur with respect to a particular plane and those which cannot occur. This introduced a new problem because we must know the orientation of a surface and light source with respect to the viewer before we can say for certain whether or not particular junctions can occur. Nevertheless I had found an interesting kind of interrelation between these variables in that if we know orientations we can partition junctions and if we know the junction labels we can tell something about surface, light and eye placement. This suggested that I could use this type of fact both to check for labeling validity (all labelings must be consistent with a single interpretation of light and eye positions) but even more important it provided a potential active tool if I could obtain partial information in a scene. Perhaps most important of all, it pointed out the importance of finding surface orientations; just as a line label constrains the possible labeling of the junctions at both ends of the line, so a region orientation label could constrain the possible junctions and lines which bound it.

As soon as I began thinking in terms of regions I came up with some powerful results. The first region I considered was the illuminated portion of the surface (assumed to be a plane) which supports a scene. Along the boundary of this region and the scene, the only line assignments which are possible are shadow edges, concave edges (when the scene objects rest on the surface) and obscuring edges. Therefore we can (1) partition the set of possible junctions into those which can occur along the boundary and those which cannot, (2) we can say how these junctions must be oriented if they occur on the boundary (no convex edges can occur along the boundary for example), and (3) we can further partition these junctions according to lighting direction, since they all appear in relation to the same surface.

Once I enumerated these junctions another fact emerged: certain junctions on the scene/background boundary must always be partially labeled in a unique manner. What this means is that certain local features always have a unique meaning in a particular context. I have since been able to find a number of similar examples which occur in other contexts. These observations have allowed me to find rules which enable us to partially label virtually any scene; given a partial labeling, we can often find additional "forced choice" labels, use all these to determine lighting position, and use this in turn to do further forced labeling.

As I show in the paper, we can also formalize labels for region orientation and illumination and look for scene cues to label these just as we do lines. Some of the cues include line slopes, junction types, and placement in the scene (e.g. on the scene/background boundary or in the interior). I have also developed some of the rules which relate line labeling, junction labeling, and region labeling to each other and to themselves as well as to lighting and eye positions.

To illustrate the inherent redundancy of the scene features, the types of information I have listed are sufficient to solve a number of shadow scenes, yet I have not even mentioned one of the most obvious features of shadows- their darkness. At one time I had felt that this was perhaps the most important shadow cue, and that it was also possible that we were able to detect a particular kind of microstructure in shadow edges which we could use to distinguish them from other edges. Clearly darkness and edge type are not enough to alone identify shadows, since we do not often confuse painted areas with shadows. In order to recognize shadows we need to know about the consistent relationships which shadows have to lighting and to the objects which "cause" the shadows.

Finally, although I did not set out to solve the problem of degeneracies in scenes I have been able to make some progress toward handling them. Roughly, degeneracies are points at which separated vertices and lines appear to be part of the same vertex, so that the resultant vertex is not in the allowable set of junctions, or where lighting is such that shadow lines are projected on junctions or otherwise line up with edges. If we could move the light source and eye slightly, these degeneracies would disappear (though we might form new ones). However I have been able to find some methods of decomposing or otherwise understanding such junctions without resorting to making physical changes, largely as a byproduct of the use of a larger number of scene cues and better general rules.

Rather than consider any more details, I would now like to return to the general problem. What exactly does this research add to our understanding of the general problems of vision programs and human vision?

First I have shown that we can formalize and make good use of scene information other than lines and junctions.

Next, although I have chosen to use only that information which I know how to extract from scenes, I have provided a far more general framework than existed before. For example, already having a formalism for region orientation and lighting information allows us to add colors or textures of regions as properties.

I have also shown that when we assign values to scene variables we can put explicit conditions on other scene variables. This leads to a greater understanding of the active nature of visual processing, which we can express in the form of theorems in a program. This approach has several advantages over previous labeling procedures. (1) We can check at each step to see whether it is reasonable to proceed in a particular manner, and do not need to defer confirmation or contradiction until later in the operation of the program. (2) We have more conditions which must be satisfied, so that we should get fewer parsings for each scene; in order to obtain these, we also do not have to start with a completely unlabeled line drawing for every parsing. (3) We can handle degeneracies more easily since we do not have to depend on the single observation that all the line labels must match.

In all these senses, I have made a start toward understanding vision in a way which will allow me to write a program using procedures as in Winograd's program rather than blind tree search techniques. Like Winograd I have tried to make the meaning of various scene features clear in terms of the consequences of interpreting them in each possible manner.

Finally I believe that my research will provide a more solid basis for understanding various phenomena in human vision. While I would certainly not contend that the processes in human vision are like the ones I describe, we should at least be able to gain a better idea of the necessary complexity of a theory to explain human vision, and some perspective on what kinds of information such a theory must contain.

I. Ground rules

I will assume that I am given only the following information which comprises a line drawing of a scene:

1. Lines. Lines are joined collections of feature points defined to be on the retina, where a feature point is a retinal point at which the spacial derivative of light intensity is greater than a certain threshold. I will assume that a line drawing has only straight lines, and that we are given no information about which lines are likely to be cracks, shadows, or edges by the line finding programs.

2. Junctions. Points on the retina where two or more lines intersect.

3. Intensities. Relative light incident on any point on the retina.

4. Regions. Areas of the retina bounded by lines. The set of all regions in the line drawing fills the retina.

This information can be coded into the following form which I will assume is directly accessible to the program.

1. For each junction, all the junctions that are connected to it by single line segments.

2. For each junction, a type, such as ell, fork, arrow, etc. I will have more to say about these later.

3. For each junction, all the regions surrounding it.

4. For each junction, its coordinates.

5. For each region, its intensity.

6. For the background region, a special name to distinguish it from all other regions in the line drawing.

The object of my work is to arrive at a single parsing, or assignment of line labels to line segments in the line drawing. There are seven line labels: concave, convex, crack, plus two orientations each for obscure and shadow. I will allow the scene (which corresponds to the line drawing) to have shadows from a single light source, coincidences of line segments and junctions from the point of view of the eye, and objects which have faces with arbitrary reflectivity. Some scenes are essentially ambiguous; by this I mean that humans can arrive at more than one interpretation for such a scene. In these cases my goal will be to find these interpretations. These are considerably more ambitious goals than have been attempted before, but I believe that there is sufficient information in the representations that I have described to come close to these goals.

In the sections which follow I will first discuss the programs which have been suggested to solve similar problems, their limitations and then new methods for overcoming these limitations.

II. Huffman, Clowes, and Dowson's Programs

The programs written by or suggested by these men all share the following assumptions which are built into the methods:

1. Every junction which can appear in a line drawing under particular assumptions (such as only trihedral junctions) is listed; each junction which occurs in a line drawing is assumed to be in this list.

2. The only conditions for labeling a line are (a) that the junctions which result be from the allowable set and (b) that all junctions in the line drawing can be labeled in a manner which is consistent with this label for the line.

3. Thus these programs require only that line labels agree and put no explicit conditions at all on regions.

In general each program of this type will produce several parsings, and indeed, as long as all junctions in the line drawing appear in the set of allowable junctions and as long as there are no degeneracies, the parsing we would call "correct" will be among those produced. However, this still leaves us with the problem of finding the "correct" parsing from among those produced. Even more seriously, these programs produce a correct parsing only if the needed junctions are in the allowable set. There is no way to handle degenerate junctions except to list them in the allowable set. In general doing this will lead to many more parsings from which to select the correct one, since there will be more possibilities foreach junction type. (i.e. 20 FORK junctions instead of 10.) An obvious method one might suggest for this problem would then be to somehow order the junction types so that the ones most likely to be encountered would be tried first. Then we could suggest as the correct parsing the one which had the "most likely" set of consistent junctions. The problem with this approach is that the relative likelihood of various junctions depends on lighting and eye position, on position and orientation within the line drawing, and other factors; in many instances a junction labeling which is extremely likely in one context will be impossible in another context.

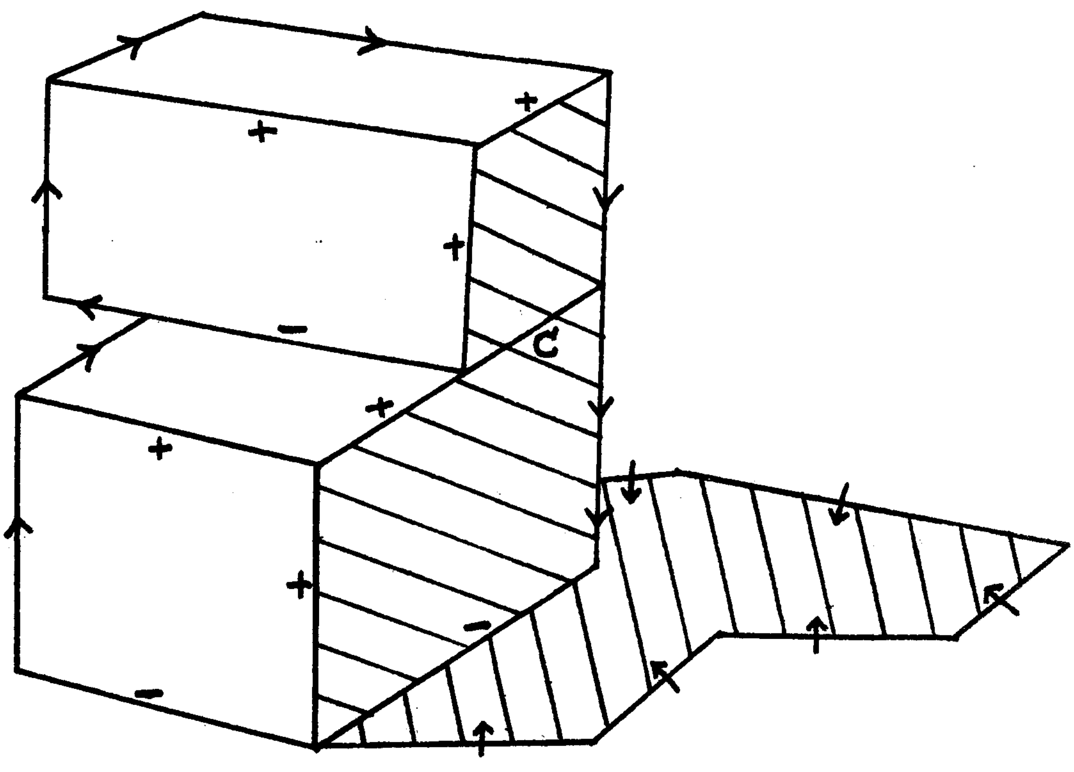
In the methods I suggest, I will try at each point to make the context of each line segment, junction, region, and line drawing as explicit as possible, and label line segments and regions on the basis of this context information. By context I mean all the partial information extracted up to the particular point in the program. As will be shown, partial labelings of lines or regions, eye position, light position, knowledge about the types of objects in the scene, orientation of lines and junction types adjacent to a junction can all function as context information. The bulk of this paper is devoted to showing ways in which context affects the interpretation of local features in a line drawing, and to showing ways to obtain context information from local features.

If the context is uncertain (as at the start of the program) I will demonstrate that we can make initial hypotheses and then work on the basis of these. Later I will show how to generate hypotheses as we proceed through the program. The results of this approach are methods which allow a large portion of the labeling to be done with little or no back-up, greater insight into the interactions of different levels of information, and programs which (I hope to convince you) will be able to successfully find the "correct parsings" for scenes under the limitations I have listed.

Both Huffman and Clowes treat at some length an alternate scene representation, called the dual-graph of a scene. In dual space each region maps into a point in such a way that parallel surfaces map into the same point. For a treatment of this representation, refer to their papers. Their feeling seems to be that it would be useful to develop the dual representation and the normal representation of a scene in parallel, whereas I feel that it is more natural to cast dual-space ideas and results into a form that can be used in the normal representation. However I am still open on the matter and not prepared to defend my view.

Figure 1 lists all the possible trihedral, non-degenerate junctions classified by junction type and within each type by line labels. These junctions were obtained by methods discussed in Dowson and Waltz.

Figure 1



The meaning of line labels

Figure 1

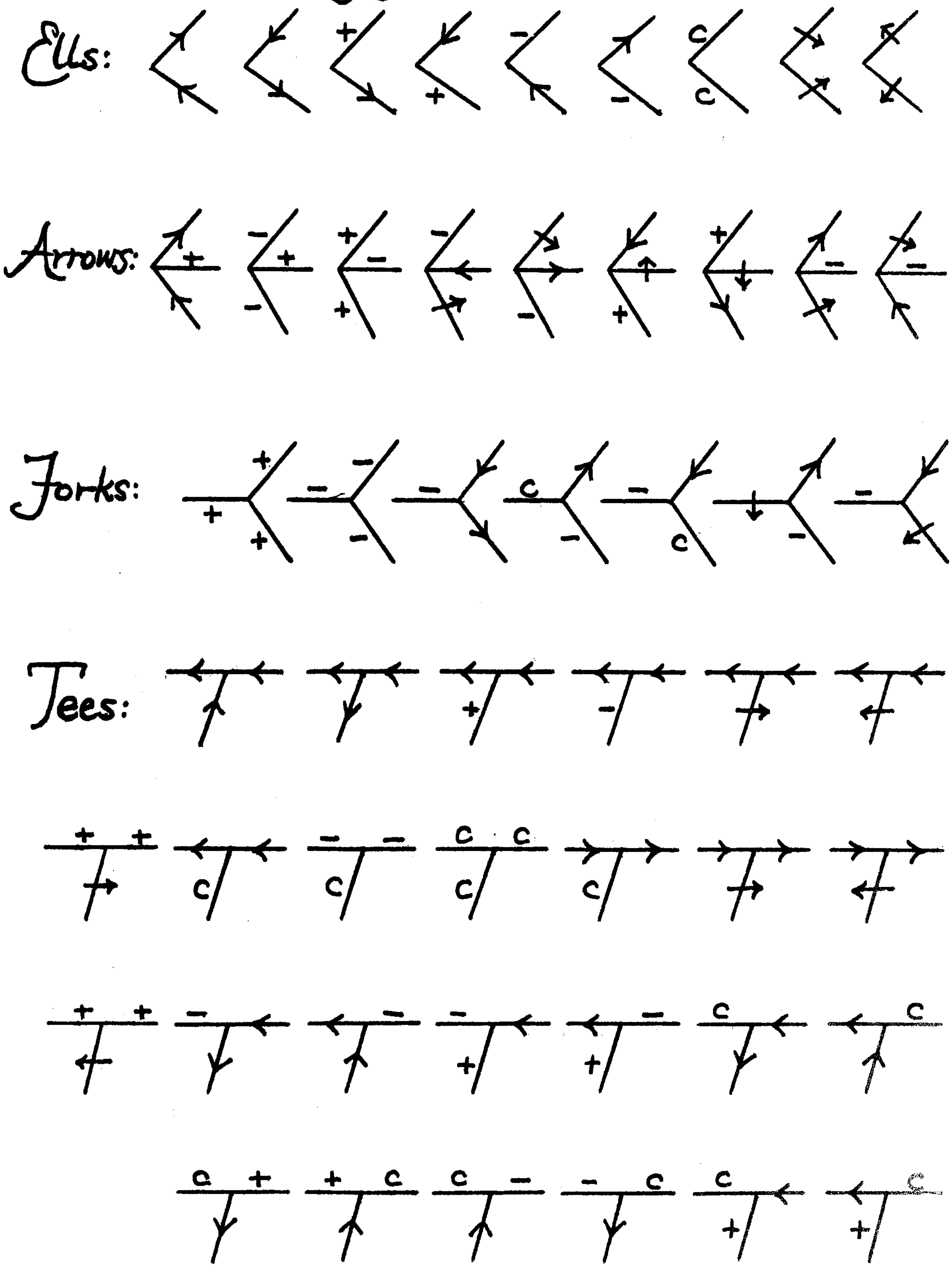


Figure 1

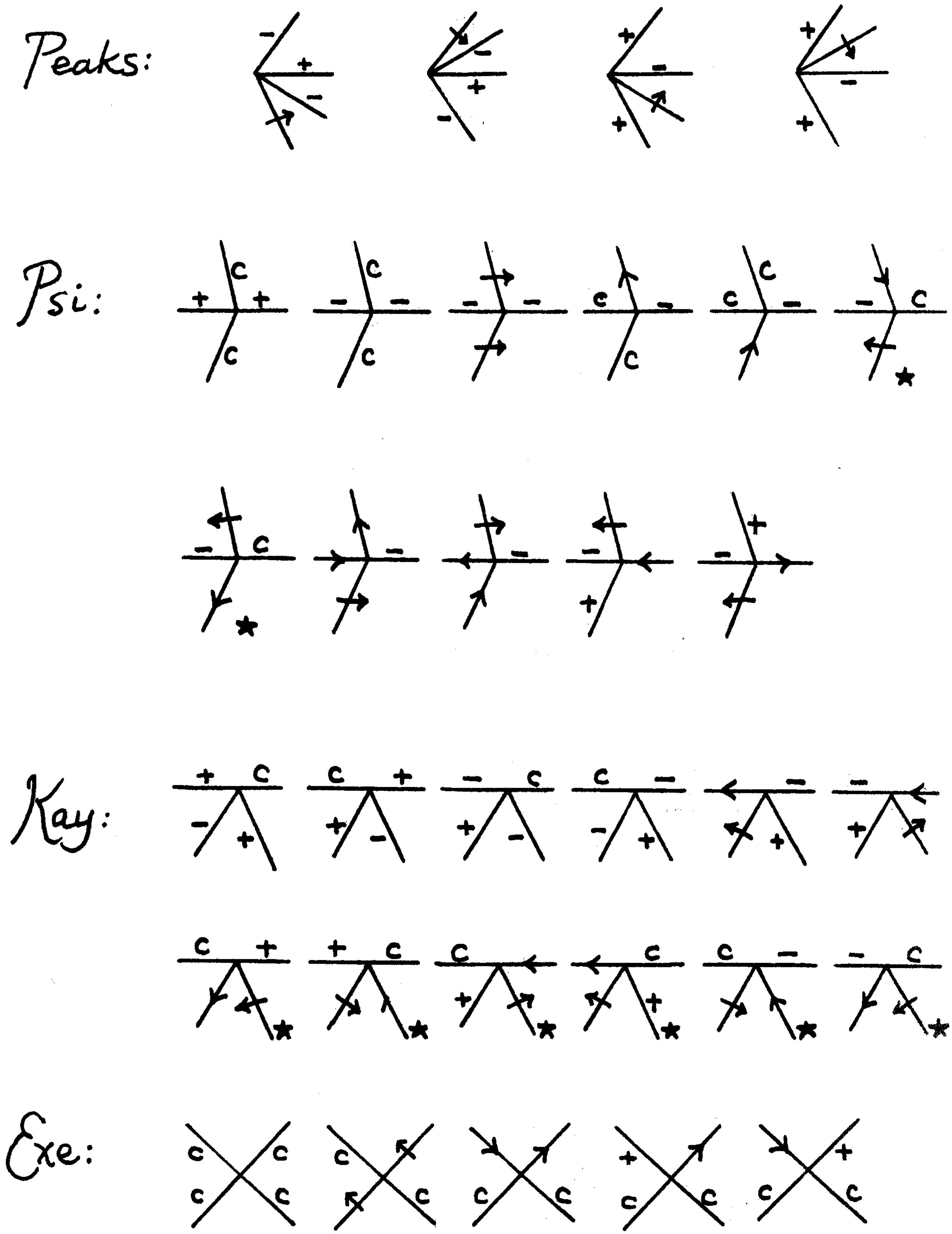
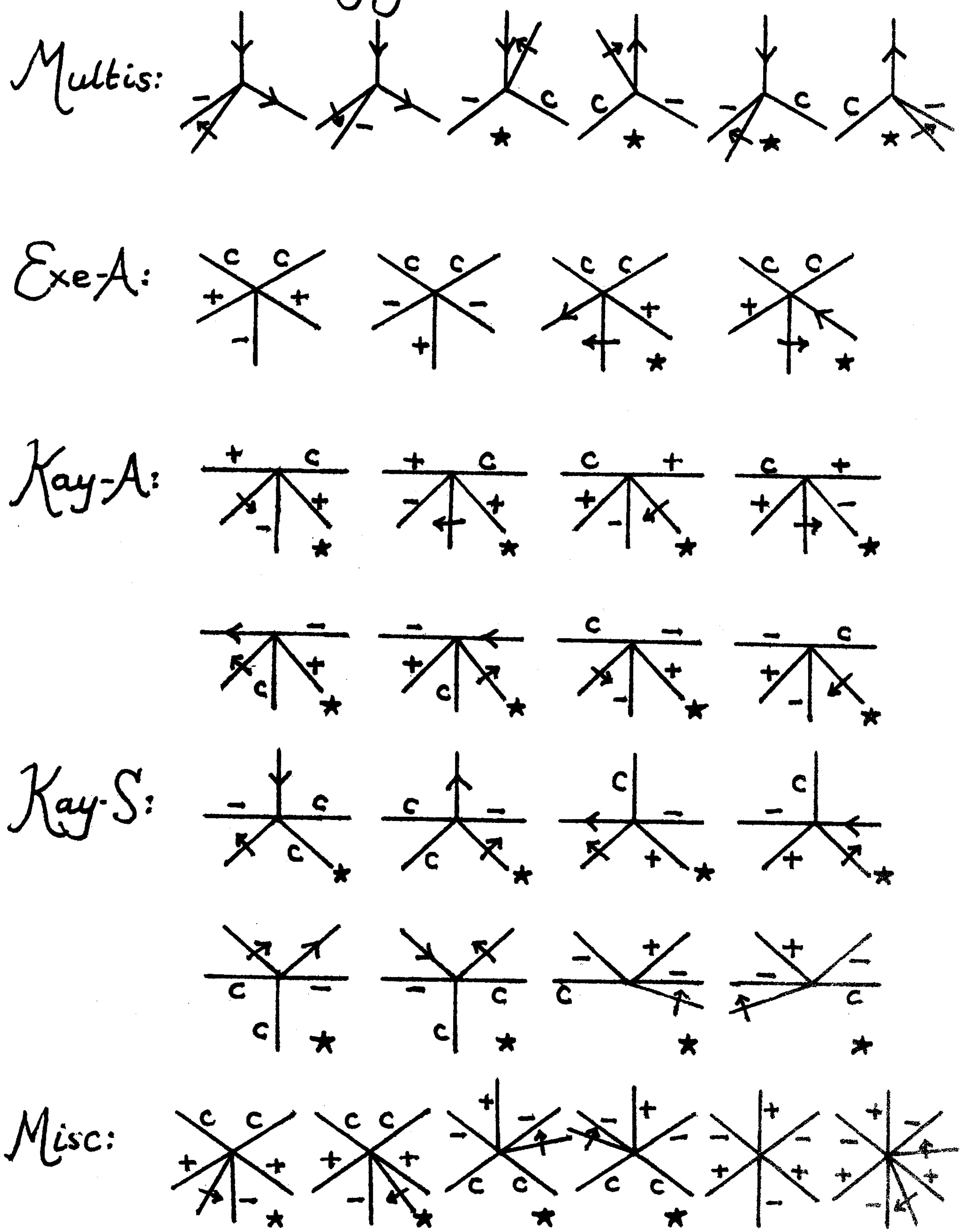


Figure 1



III. General plan for a program

Basically, the methods I will describe depend on the following observations:

1. We can generate a complete set of non-degenerate junction labelings under particular assumptions.

2. For each junction type, we can also specify a finite set of region labelings which give values for orientation and illumination, one and only one of which must always be true.

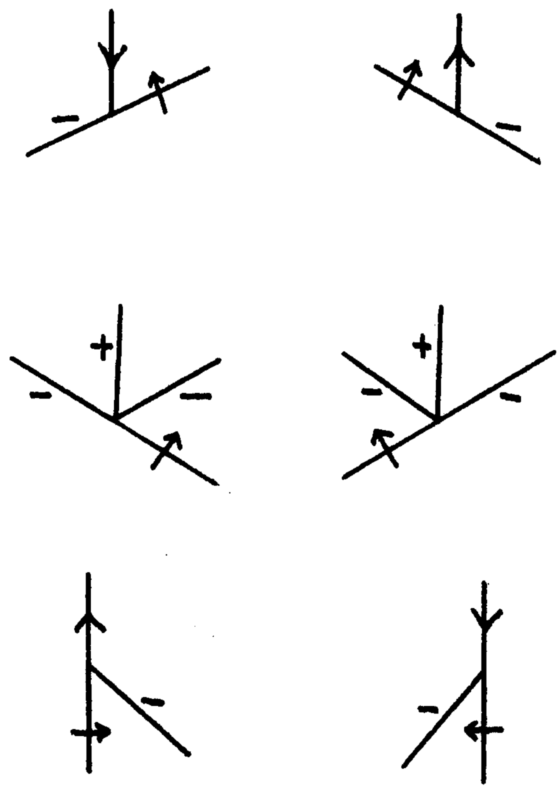
3. Just as two junctions joined by a line segment must be labeled so that the line connecting them has a single value, so must each junction around a region be labeled so that the region has a single orientation and illumination value.

4. Whenever we assign a region a particular value we also place explicit constraints on the possible position of the eye and light source.

5. All regions and junctions within a scene must agree with a single value for light source position and a single value for eye position.

6. Junctions which are degenerate because of light source positioning (usually when the light source lies in the plane of a surface) can only occur under very restrictive conditions on the regions surrounding the junction and on the light and eye;

Figure 2



therefore these junctions can be included in the complete set without danger of causing large numbers of spurious parsings. Figure 2 shows this set of junctions.

7. We can identify other degenerate junctions as those which either (a) do not have the form of a known junction or (b) cannot be interpreted as members of the set of allowable junctions in a manner consistent with the rest of the scene.

8. While it is impractical to list all these other types of degenerate junctions which can occur, as long as these junctions comprise only a small percentage of the total number of junctions in a scene we will generally be able to obtain at least a partial labeling imposed by junctions and regions surrounding them. On the basis of partial labeling and knowledge about the ways that degeneracies can occur, we can suggest ways of separating the true junction components.

9. We can use line orientations to suggest region and junction labels. Given three unknown variables, namely the region types on both sides of a line segment and the line segment label, plus one known, the line orientation, we can tell very little. However if any of the three variables have known values, we may be able to solve for the other two, and if two of the three are known we usually will be able to deduce the third. In either case we can at least limit the possibilities.

10. Region intensity is not always a very good indicator of

region type, since objects can have arbitrary reflectivity. We can however say that whenever we label a line as a shadow line, the region on the side we have labeled as a shadow must be darker than the other side of the line. Similarly, if two regions separated by a line are approximately equal in intensity, it is unlikely that the line separating them is a shadow line. In addition we can say that the darkest regions are likely to be shadow regions, whereas the brightest ones are extremely unlikely to be shadow regions.

IV. Region notation

IV.A. Junction segment notation

As will become obvious later, we will have use for a notation which allows us to name regions around a junction. By conventions shown in Figure 3, we can code a region in the form of a circular list of sequential junctions, listed by convention in a clockwise direction when viewed from inside the region.

IV.B. Illumination

Definition: Causing junctions are ones in which (1) one of the lines of the junction is a shadow line and (2) this shadow line is the projection of one of the other lines in the junction.

A region may be one of the following types:

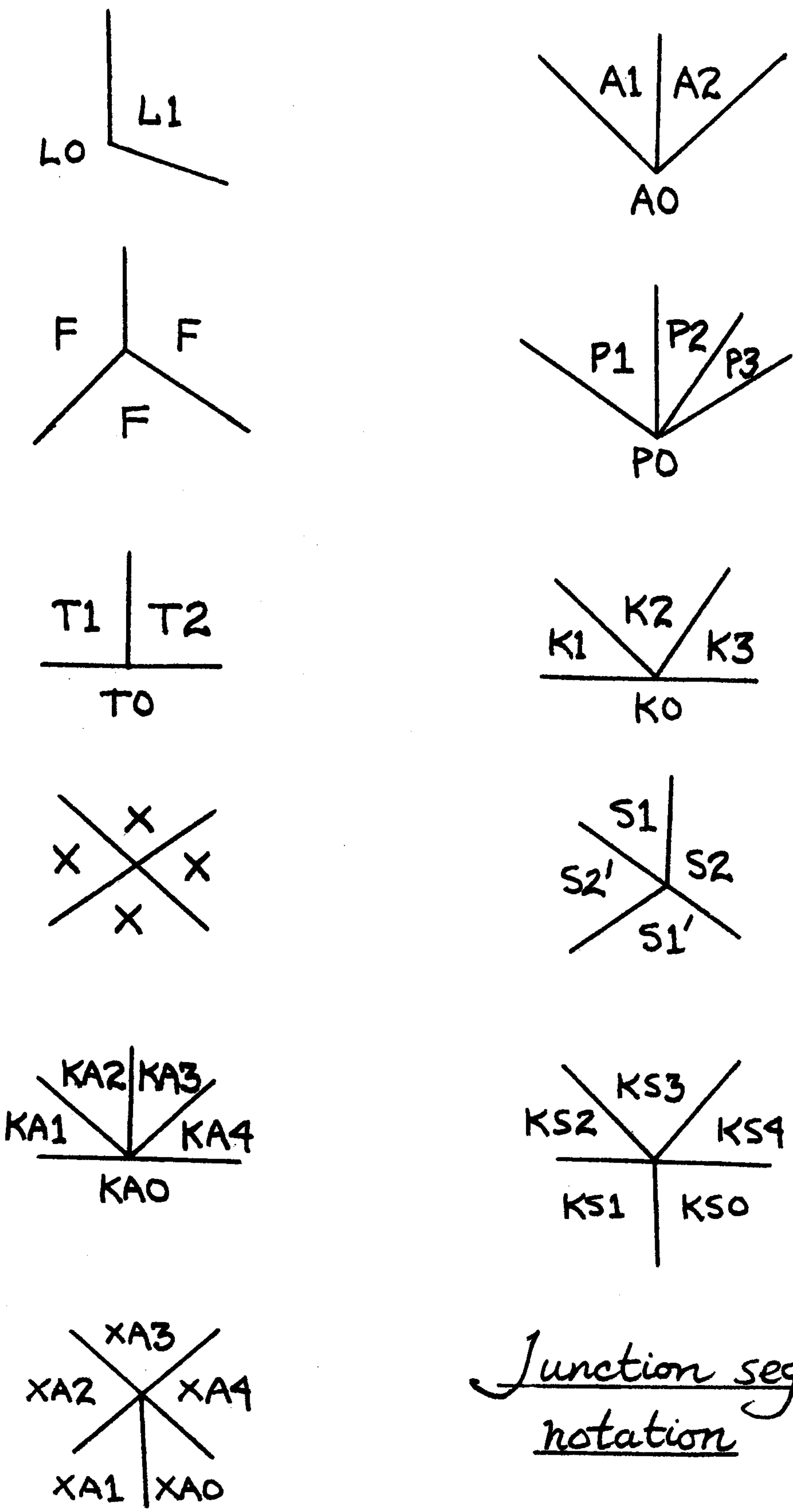
I = illuminated

S = shadowed; if type S then it must be either

SA= a projected shadow region with one or more "causing junctions" on its boundary.

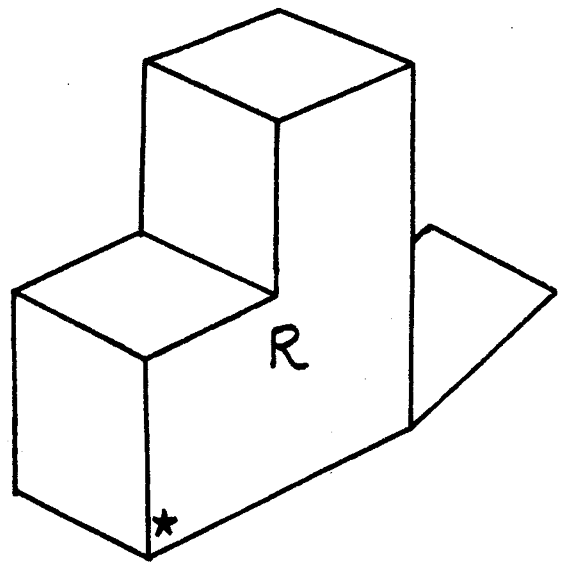
SB= a projected shadow region with no "causing junctions" on its boundary which cause it. (A type SB region can be bounded by causing junctions which obscure it. See figure 4A for the distinction.)

Figure 3



Junction segment notation

Figure 3

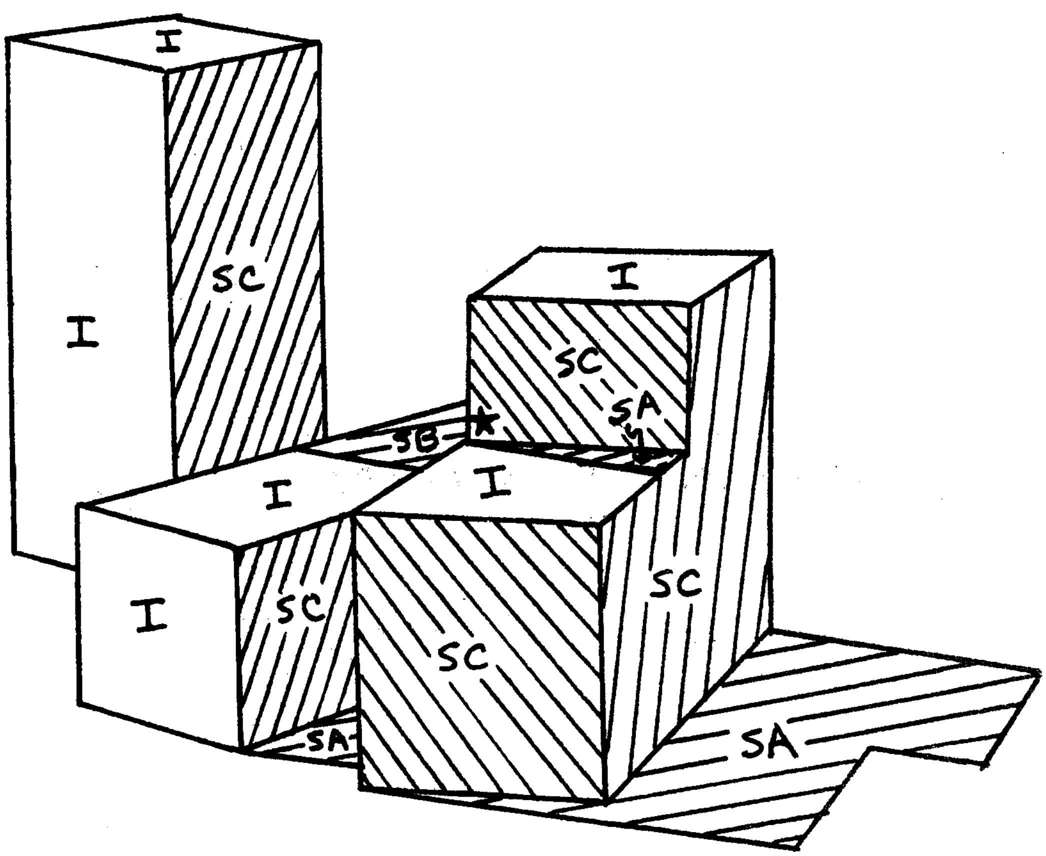


Example of region notation for R, beginning at ★:

R: A2/F/A0/F/A1/T0/A1

Figure 4A

Region illumination notation



★ is a shadow causing junction bounded by an SB region.

SAB = a projected shadow region formed by an overlapping of a type SA and a type SB shadow region.

SC = a region which is shadowed because it is oriented away from the light source.

The background region is defined as that portion of the support surface which is illuminated and which shares a boundary with the edge of the retina. The support surface is the union of the background region, any illuminated regions of the support surface not connected to the background region, and shadows projected onto the support plane. The support surface can thus have regions which are labeled in any manner except as type SC or type SAB.

IV.C. Some facts about regions and illumination

Definition: A shadow causing edge is one which casts a shadow.

I will appeal to your knowledge of the world to state the following facts without proof or with minimal proof.

1. A shadow causing edge must always be convex.

2. Every convex edge must always be labeled as "+" or obscure.

3. Every "+" or obscuring line segment is convex.

4. Every shadow causing edge has a type I region on the side oriented toward the light source and a type SC region on the other side.

Proof: Because an edge illuminated on both sides cannot cause a shadow, nor can an edge shadowed on both sides cause one. If the shadowed side is any other type than type SC, then the shadow is due to the projection of another object where the projected shadow coincides exactly with the edge itself; but then if the object causing the shadow were removed, both sides of the edge would be illuminated, so the edge cannot cast a shadow.

Definition: A region bounded by two edges of a junction which obscure it has no necessary relation to either of its adjoining regions; I will call such a region an obscured region with respect to such a junction.

5.(a) If a shadow causing edge is labeled "+", then one side of the edge is type I and the other is type SC. If the shadow causing edge is labeled as an obscuring edge, and the junction is a shadow causing junction (i.e. the shadow is visible) then (b) one side of the edge is labeled type SA and the other is type I, or else (c) one side is labeled type SC and the other is labeled I, except when the region adjacent to the type SC region is a region obscured with repect to the shadow causing junction.

Proof: (a) is trivial. In (b) and (c) if we can see the edge then we can see either the type I region (b) or the type SC region (c). If we can see the type I region and we can see the type SA shadow region, then these two regions must adjoin one another at the junction- what could be between them? If we can see the type SC region, and the junction itself, then the adjacent region must be I or else the shadow causing edge could not cast a shadow at the junction, except in the case where the adjacent region is an obscured region, with respect to the shadow causing junction.

6. A type SC region can have no shadow edges bounding it.

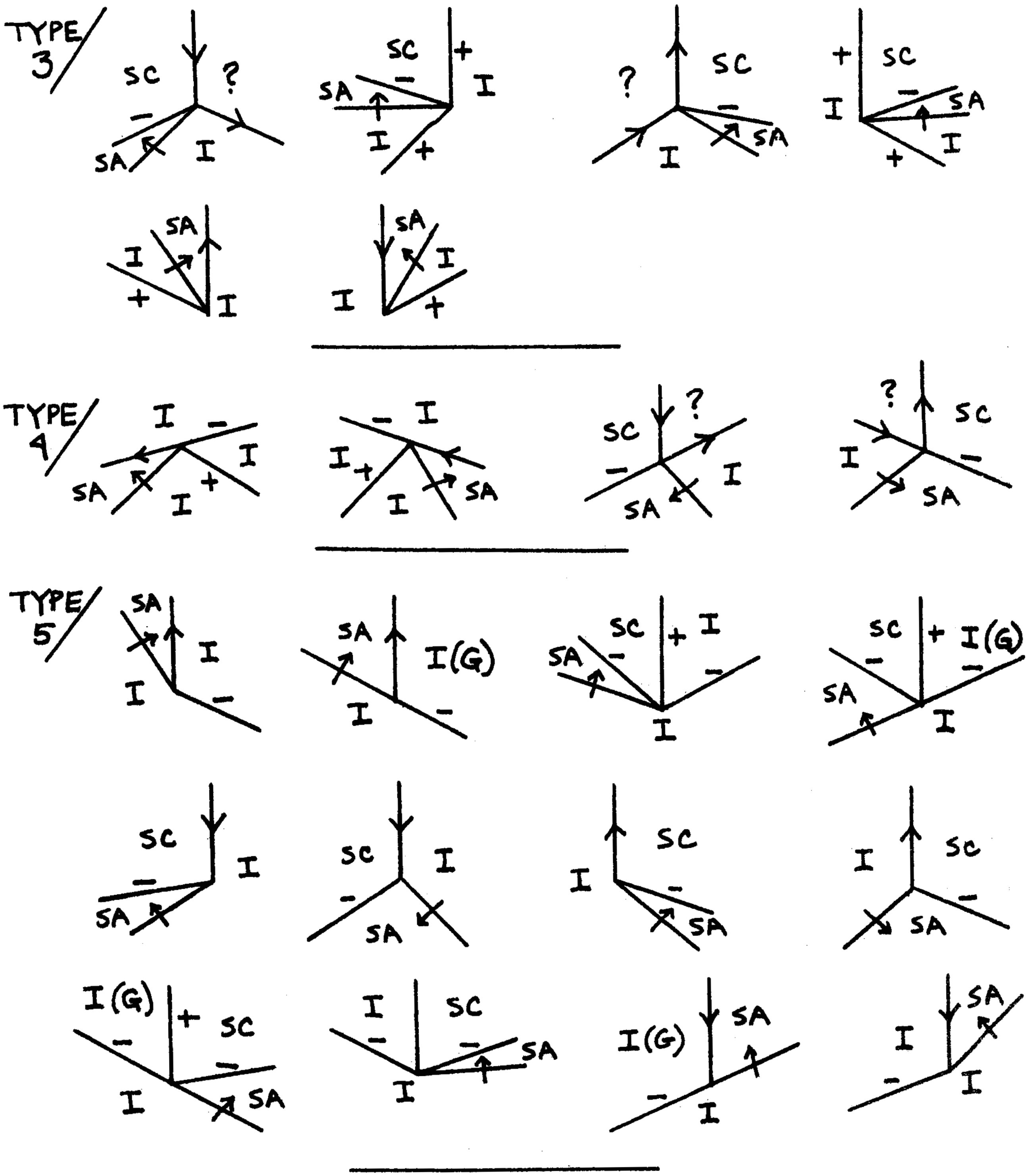
7. In non-degenerate cases a region X separated from a region Y by a crack or concave edge is type I iff region Y is type I; otherwise both are type S.

Proof: The only way this could not be true is if a shadow edge coincided exactly with the concave or crack edge.

8. If we can identify a junction as a shadow causing junction, then we can uniquely label all of its regions for illumination except those which are obscured with respect to the junction, if any.

Proof: Consider the set of all shadow causing junctions (figure 4B). We can always by definition label the regions bounded by the shadow line as types SA on one side and I on the other. If the SA region is bounded on the other side by a concave edge, label the next region SC, otherwise label it type I (using fact 5 above), unless that region is obscured with respect to the

Figure 4B



I(G) indicates that such a surface is either illuminated or has glancing illumination (light source in the plane of the surface)

Each of these junctions has variations with cracks; these are the starred junctions in figure 1.

junction. Using fact 7, we can then label all the other regions.

IV.D. Exact relations between eye, light, and scene; visibility and illumination of surfaces

As a first approximation, consider the light source and eye to be infinitely distant from the scene, the scene to be supported by a plane with no holes, and the light source to be at an elevation great enough so that all shadows cast in the scene are entirely surrounded by illuminated backgound on the retinal projection of the scene.

Define a coordinate system C with respect to the eye. C is a right hand coordinate system, origin in the center of the scene, oriented with its x and z axes on the support plane and y axis normal to the support plane. The z axis coincides with the projection of a line from the origin to the eye onto the support plane, or in other words the eye is in the y-z plane in the quadrant between the +z and +y axes. The eye elevation, ϕ_E, is defined as the angle between the +z axis and the line from the origin to the eye. Thus ϕ_E is between 0 and 90 degrees.

The normal to any surface in the scene can be defined in this coordinate system by two angles, ϕ_S and θ_S. ϕ_S is the angle between the normal to a surface and the projection of the normal onto the support plane. θ_S is the angle between the projection of the normal onto the support plane and the z axis, measured in a counterclockwise direction when viewed from the +y axis.

Using these definitions, the vector from the origin to the eye in C is

$$\vec{V_E} = \sin\phi_E\,\hat{y} + \cos\phi_E\,\hat{z}.$$

The normal vector to a surface is

$$\vec{V_S} = \sin\theta_S\cos\phi_S\,\hat{x} + \sin\phi_S\,\hat{y} + \cos\theta_S\cos\phi_S\,\hat{z}.$$

The condition which must then be satisfied in order that a surface be visible is

$$\vec{V_S}\cdot\vec{V_E} > 0$$

$$\text{OR}\quad \sin\phi_S\sin\phi_E + \cos\theta_S\cos\phi_S\cos\phi_E > 0.$$

Similarly, we can define a coordinate system C', where the light source lies on the y'-z' plane at an elevation ϕ_L. The surface orientations are given in this coordinate system by ϕ'_S and θ'_S. The conditions for visibility in C are then analogous to the conditions for illumination in this coordinate system C'. Therefore let

$$E' = \sin \phi'_S \sin \phi_L + \cos \theta'_S \cos \phi'_S \cos \phi_L$$

then

if $E' > 0$, the surface is illuminated

if $E' = 0$, then the light source lies in the surface plane, i.e. the surface has glancing illumination

if $E' < 0$, then the surface is shadowed, and its edges cast shadows.

We can then define a transformation from C' to C, where C' is related to C by the angle θ_{EL}. θ_{EL} is the angle from the z axis of C to the z axis of C', or equivalently, the angle between the projections of the eye-origin and light-origin lines onto the support surface. In C, the light source normal is

$$\vec{V_L} = \cos\phi_L \sin\Theta_{EL}\hat{x} + \sin\phi_L \hat{y} + \cos\phi_L \cos\Theta_{EL}\hat{z}$$

so the condition for surface illumination is

$$\vec{V_L} \cdot \vec{V_S} > 0$$

$$\text{OR } (\cos\phi_L \sin\theta_{EL} \sin\theta_S \cos\phi_S + \sin\phi_S \sin\phi_L$$
$$+ \cos\phi_L \cos\theta_{EL} \cos\theta_S \cos\phi_S) > 0.$$

In Appendix 1, I show how we can find all these angles under certain assumptions.

IV.E. Quantized region orientation

Rather than use the full treatment developed in the previous section and Appendix 1, I will introduce a quantization of the angles defined there. One reason for doing so is that it will often be difficult to find the eye elevation in cases where there are objects without right angles, and another is that there is no necessity to be this precise for the time being. It is interesting that even a very rough quantization provides a uniform and effective method for eliminating parsings which are physically impossible, but allowable when only line labels are required to match.

Two regions orientations are special. One of these is horizontal, denoted by H, which means parallel to the support surface. The other is B which is used to label any region which is part of the visible support surface.

All other orientations are quantized according to their θ_S value, i.e. according to the angle between the projection of their normals onto the support surface plane and the scene z axis. The quantizations are:

F = front, θ_S within about 10 degrees of the z axis; the exact values will vary depending on the distance of the eye from the scene, but the working definition is that such a surface can have right angle vertices at both left and right extremes such that for each of the vertices only one junction segment out of a possible three is visible. Another way of putting this is that a region labeled F can have edges labeled "convex" at its right and left extrema only if the object does not have rectangular vertices.

R = right, θ_S between the limit of F and 90 degrees.

BR = back right, θ_S between 90 and 180 degrees.

BL = back left, θ_S between 180 and 270 degrees.

L = left, θ_S between 270 degrees and the limit of F.

To this will be appended a suffix to indicate the value of ϕ_S, the angle between the normal to a surface and the projection of the normal onto the support plane. These suffixes are:

V =vertical, ϕ_S = 0 degrees.

D = down, ϕ_S < 0 degrees.

VU = vertical/up, ϕ_S between 0 and 45 degrees.

HU = horizontal/up, ϕ_S between 45 and 90 degrees (90 degrees = horizontal).

Notice that in a scene made up only of objects which have rectangular corners, where none of the objects "leans" on another, the only possible visible surface orientations are B, H, LV, FV, and RV, regardless of eye position. In addition, a surface either belongs to the set of illumination types I, SA, SB, SAB or is type SC, and this is a function only of relative light/eye angle, independent of light elevation.

IV.F. Light position quantization

The relative light/eye angle, θ_{EL}, will be quantized according to Table 1. There are twelve possible values, which can be visualized by considering a vantage point on the scene y

Table 1

Name:	Centered at: (eye @ 6:00)	Angles:
F	6	(-15°, 15°)
FR	5	(15°, 45°)
RF	4	(45°, 75°)
R	3	(75°, 105°)
RB	2	(105°, 135°)
BR	1	(135°, 165°)
B	12	(165°, -165°)
BL	11	(-165°, -135°)
LB	10	(-135°, -105°)
L	9	(-105°, -75°)
LF	8	(-75°, -45°)
FL	7	(-45°, -15°)

axis. From this position, looking down on the scene, F is a 30 degree wide segment centered on the z axis (6 o'clock). FR is then centered around 5 o'clock, B around 12 o'clock, etc.

The light elevation is quantized into three values, according to which angle is nearest the actual value:

L = low, 30 degrees.

M = medium, 45 degrees.

H = high, 60 degrees.

The eye elevation angle is quantized in four values, three of which are exactly the same as for the light source elevation, the additional one being VL = less than 15 degrees.

Given this quantization and the methods of Appendix 1 we can then generate values for each combination of light source position, surface orientation, and eye position which indicate how a given surface is illuminated and whether or not it is visible.

Figure 5A

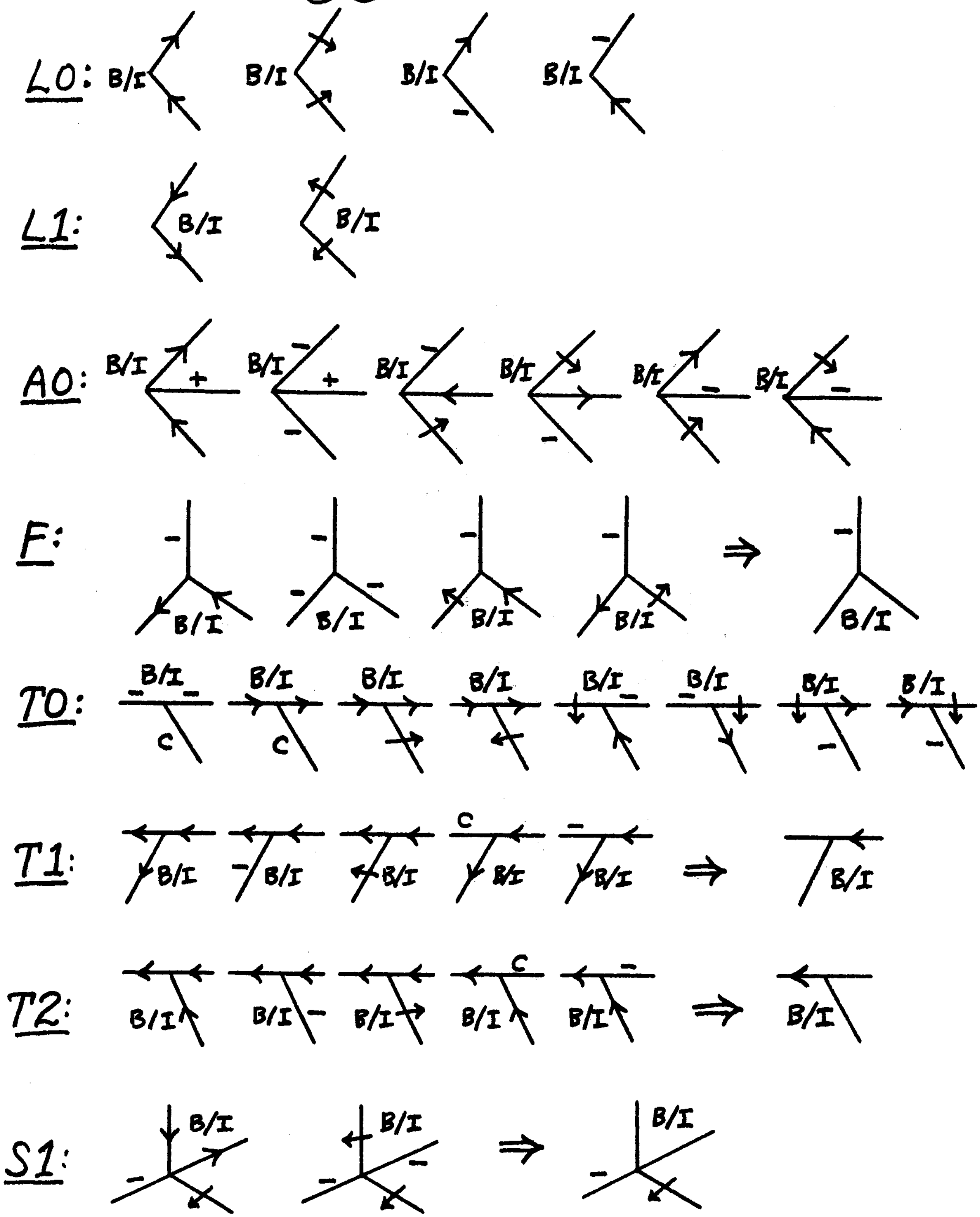
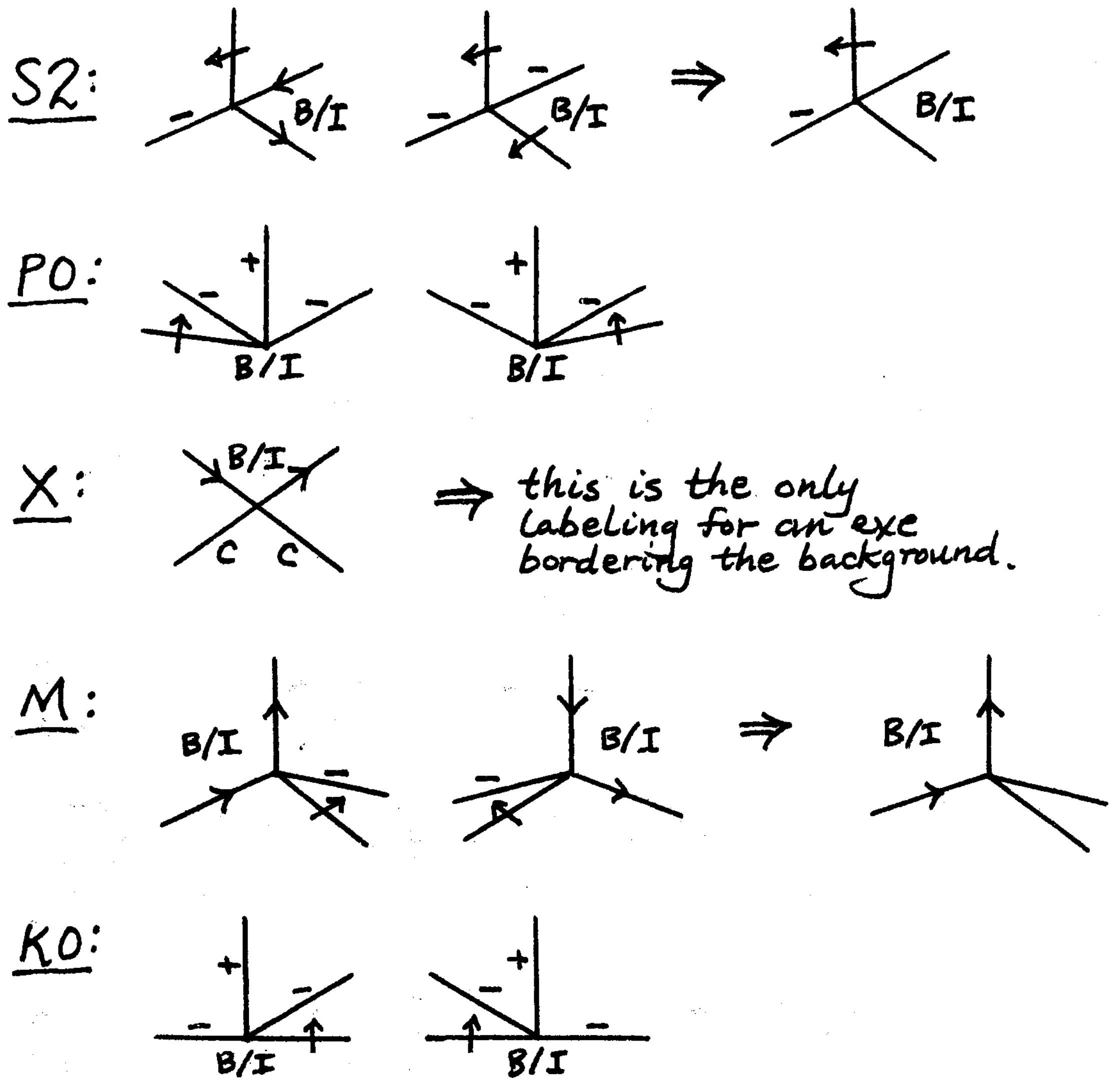


Figure 5A


S2:
B/I
B/I
B/I
PO:
B/I
B/I
X:
B/I
C C
this is the only labeling for an exe bordering the background.
M:
B/I
B/I
B/I
KO:
B/I
B/I

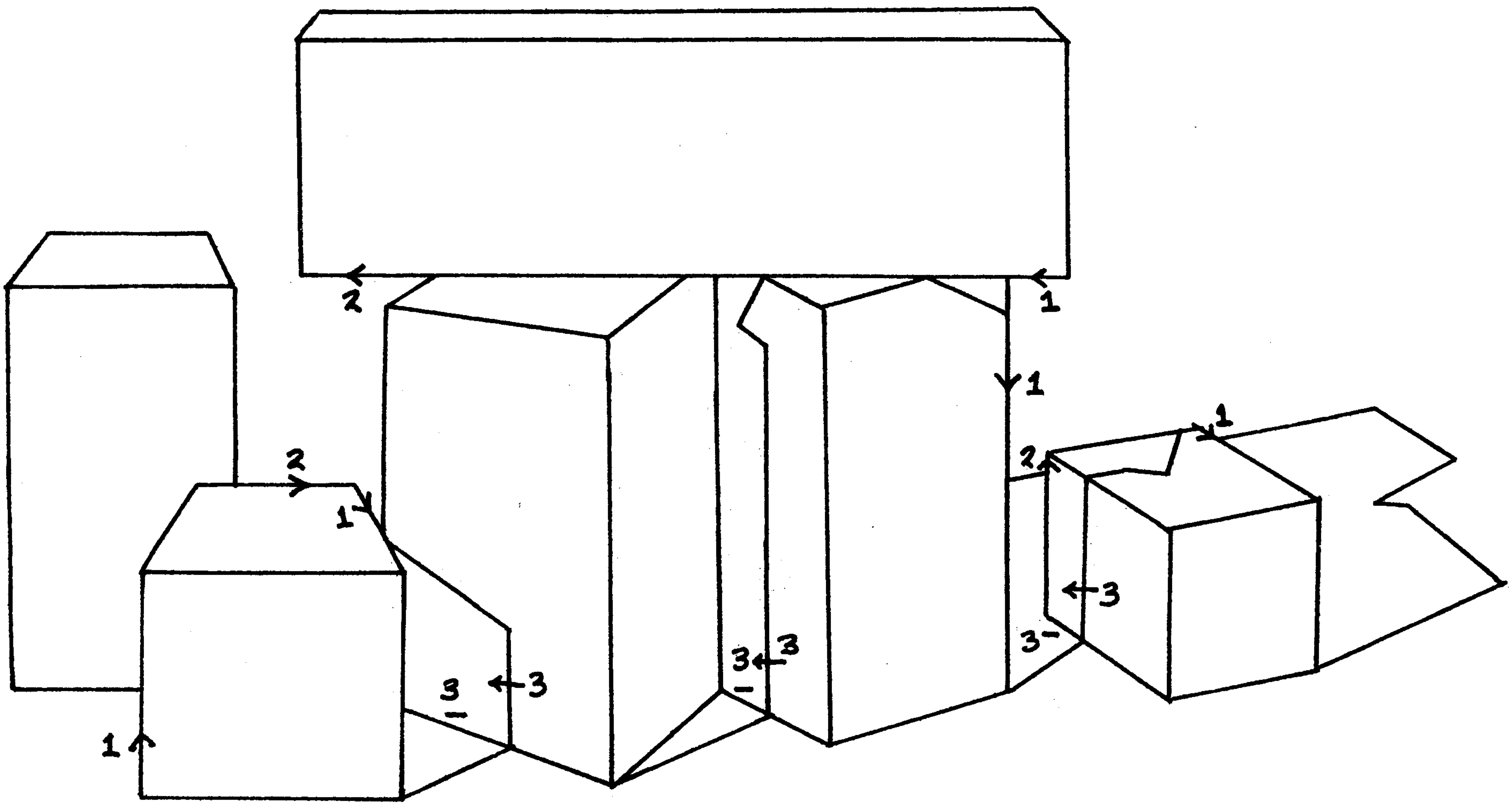
V. The background/scene boundary

While I will talk extensively about the scene/background boundary, I would like to point out that all of the techniques developed to handle this boundary will also be applicable to any isolated subpicture in the scene, or for most portions of the scene around which we can draw a path which consists only of the junctions which characterize the scene/background boundary, plus special treatment of TEE junctions. Since I have not worked out the details of isolating such portions of the scene, I will not make any overoptimistic claims, but I intend to follow up on this approach.

You will remember that earlier I defined the background region to be that portion of the visible support surface which is illuminated. Every line which bounds this region can have only one of three labels, namely obscure (a line segment obscuring the support surface), concave (or minus), and shadow, where the shadow line can be oriented in only one direction. We can then look at the complete list of possible junctions and pick out those which have segments bounded by any combination of these three labels. Figure 5A shows this set of junctions, where B/I in a junction segment indicates that this segment can be part of the background region.

We can now make the rather remarkable observation that whenever a junction on the scene/background boundary has one of its segments of type T1, T2, S1, S2, M, or F as part of the background region, we can partially label that junction directly. Frequently, having done this, there may be forced choices for other junction labelings; i.e. if we check all the junctions which are partially labeled, we may find that there is only one possibility for labeling some junctions, given that the partial labeling must be preserved.

In any event, merely by knowing that one of the segments of a junction is part of the background region, we can use this restricted set of junctions from which to choose a labeling. Figure 5B shows the results of applying the restricted set of possibilities to a line drawing (taken from a photograph). Figure 5C shows the results of applying forced rules to the partially labeled line drawing of Figure 5B. The steps involved in forced labeling are listed in order of their application beneath Figure 5C.



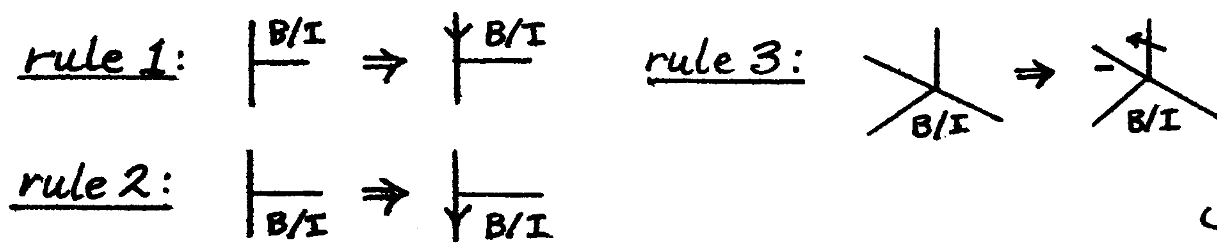
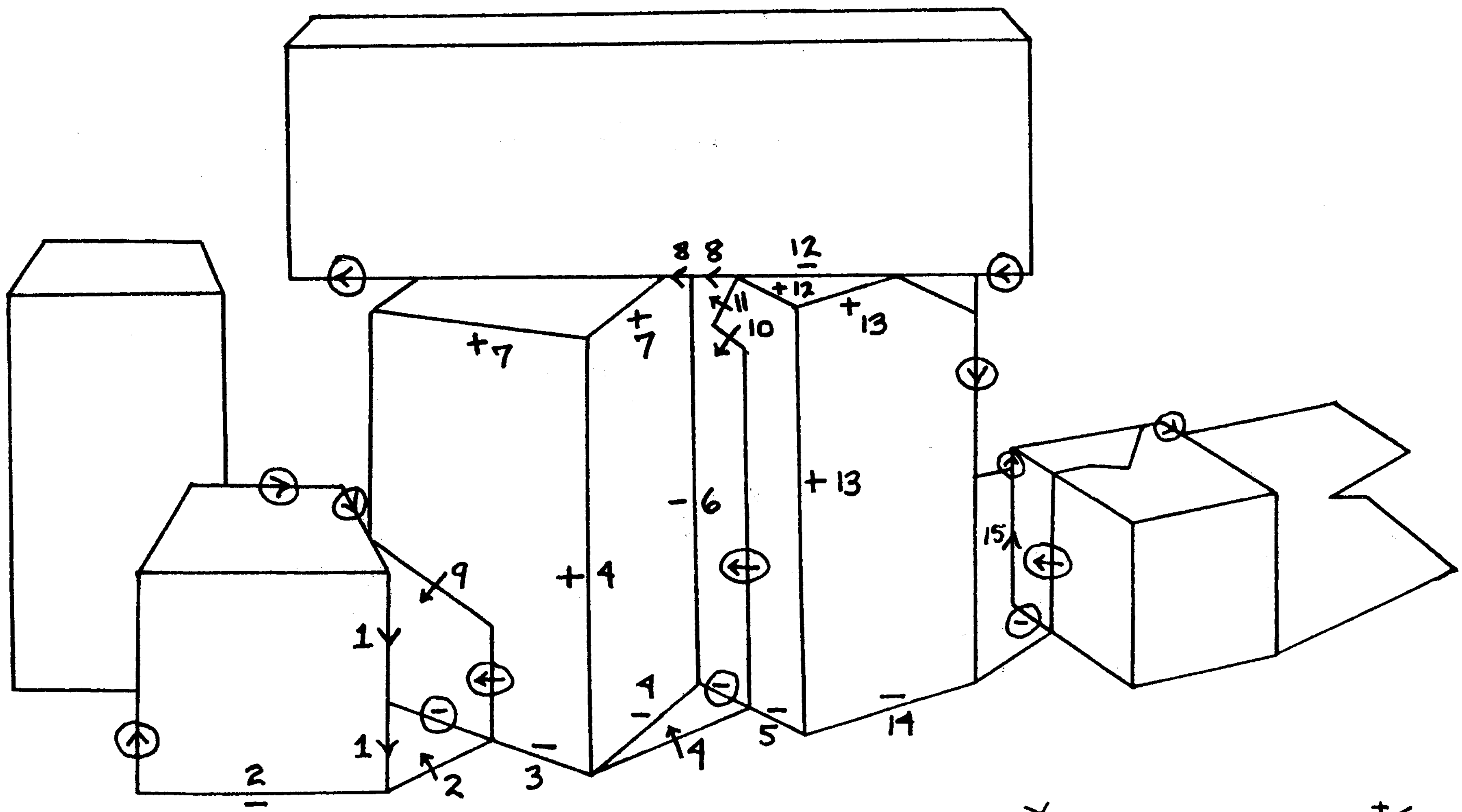


Figure 5B



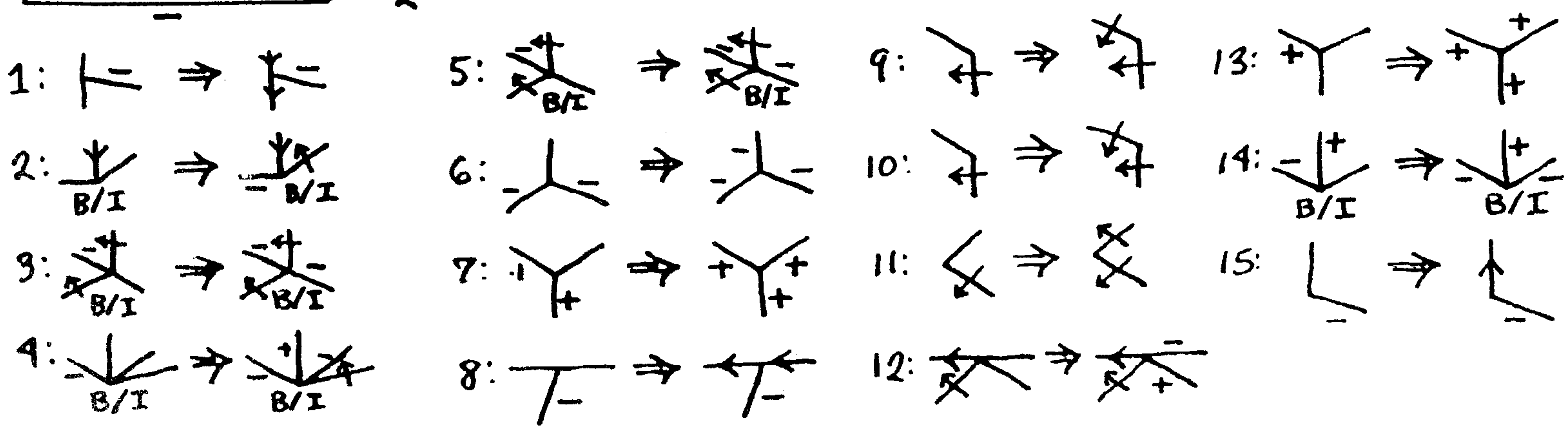


Figure 5C

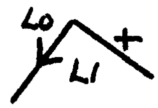
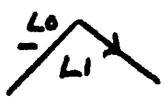
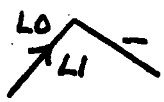
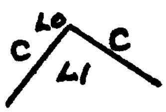
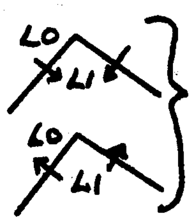
Labels carried over from 5B are circled.

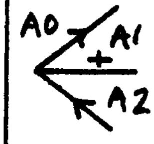
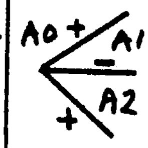
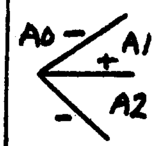
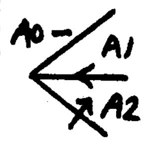
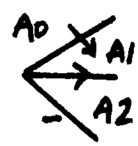
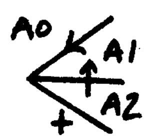
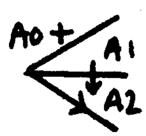
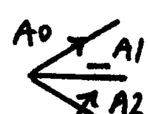
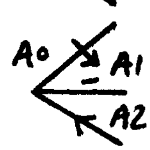
VI. Scenes with rectangular vertices

VI.A. Theory

As I mentioned earlier, the cases where a scene contains only rectangular vertices is a particularly easy one. In this section I will show how we can complete the labeling of the line drawing shown in Figures 5B and 5C assuming rectangular vertices only. First recall that the only surface orientations which are possible in the rectangular vertices case are B, H, RV, FV, and LV. In Figure 6, I show a compilation of all the possible ways in which each region can be labeled for orientation, given a particular line labeling for a junction. I have not listed all the junction types, but have confined myself to those which are needed to complete this example: ell, arrow, fork, tee, peak, psi, and kay junctions. In practice we need not have this entire table in memory since we can use theorems to generate possible region labelings as needed. I will give rules for generating this table in the section on line orientations.

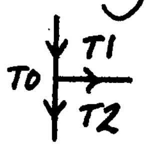
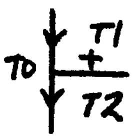
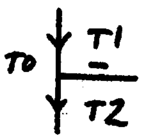
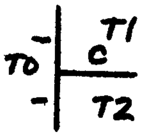
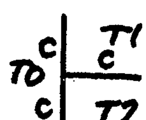
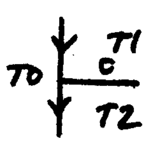
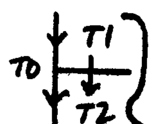
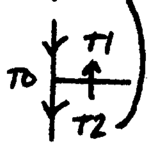
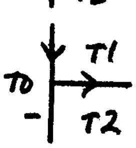
Figure 6

ELL labeling	Allowable surface orientations for: L0	L1
	B,H,L,F,R	H,L,F,R
	H,L,F,R	B,H,F,L,R
	H F,L R	F,R H L
	H F,R L	F,L H R
	H,B F,R	F,L H
	H,B F,L	F,R H
	H L F R	H L F R
	B H F R L	B H F R L

ARROW labeling	Allowable surface orientations for: A0	A1	A2
	B,H,F,L,R B,H,F,L,R B,H,F,L,R	H R,F L	L,F H R
	H L R	L R H	R H L
	H,B L R	L R H	R H L
	B H F L	R,F R,F H H	B H F L
	B H F R	B H F R	L,F L,F H H
	H H F,R L	F L H R	F L H R
	H H F,L R	F R H L	F R H L
	B H	L,F L,F	B H
	B H	B H	R,F R,F

Note: throughout this figure I have omitted the "V" from FV, RV & LV.

Figure 6

TEE labeling	Allowable surface orientations for: T0	T1	T2
	F,H R L	Any Any Any	H,R,F,L H,F,R F,R,L
	F,H R L	H,R,F,L F,L,R F,L,H	Any Any Any
	H F,H,L F,R,L F,H,R	L F,R,L R H	R H L F,L,R
	F,H,L F,H,R F,R,L H	B,H F,R,L L R	F,R,L B,H R L
	B,H B,H,L B,H,R F,R,L	F R L H	F R L H
	F H R L	F H R L	F H R L
	L,R,F,H L,R,F,H L,R,F,H L,R,F,H	H R L F	H R L F
 	L,R,F,H L,R,F,H L,R,F,H L,R,F,H L,R,F,H	B H R L F	B H R L F
	F,R,L H	Any Any	H F,L

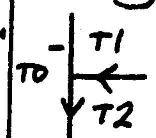
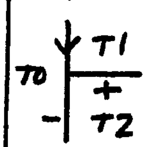
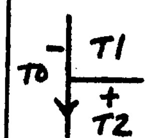
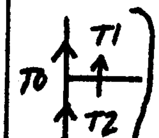
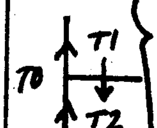
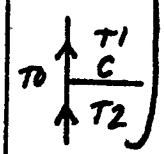
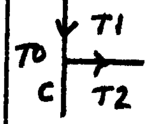
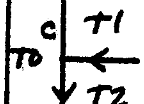
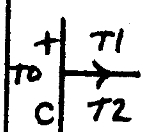
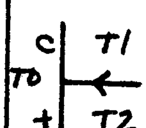
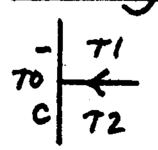
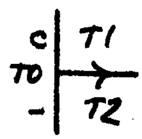
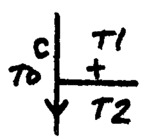
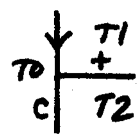
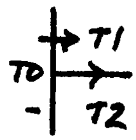
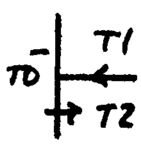
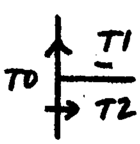
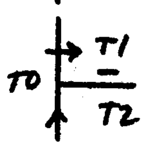
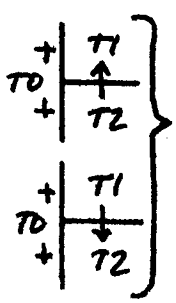
TEE labeling	Allowable surface orientations for: T0	T1	T2
	F,R,L H	H F,R	Any Any
	F,R,L L,F R H	R F H L	H H L R
	F,R,L R,F L H	H H R L	L F H R
  	Any Any Any Any	H R L F	H R L F
	R F H	Any Any Any	R F H
	L F H	L F H	Any Any Any
	L F H	H H R,F	L F H
	R F H	R F H	H H L,F

Figure 6

TEE labeling	Allowable surface orientations for: TO	T1	T2
	H L F R	R,F H H L	H L F R
	H R F L	H R F L	L,F H H R
	L F R H	L F R H	H H L F,R
	R F L H	H H R F,L	R F L H
	B H	B H	F,L F,L
	B H	F,R F,R	B H
	B H	F,L F,L	B H
	B H	B H	F,R F,R
	F,L,R L R	H R L	H R L

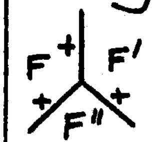
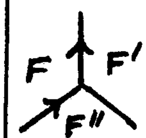
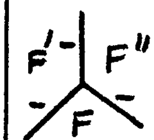
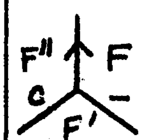
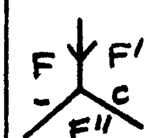
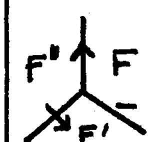
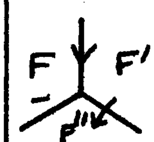
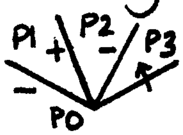
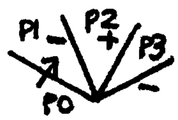
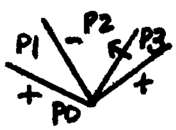
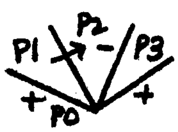
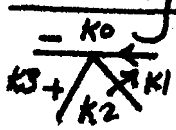
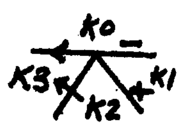
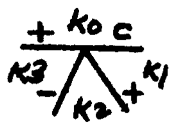
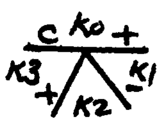
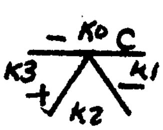
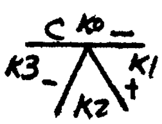
FORK labeling	Allowable surface orientations for: F	F′	F″
	H	R	L
	Any Any Any	H L,F L	R,F H R
	B,H	L	R
	F,L R H H	H L R F	H L R F
	F,R L H H	H R L F	H R L F
	L L	B H	B H
	R R	B H	B H

Figure 6

PEAK labeling	Allowable surface orientations for P0	P1	P2	P3
	B H R	L L H	R R L	B H R
	B H L	B H L	L L R	R R H
	L H	R L	H R	H R
	R H	H L	H L	L R

KAY labeling	Allowable surface orientations for: K0	K1	K2	K3
	F,R,L L,F R H	R F H L	R F H L	H H L R
	F,R,L R,F L H	H H R L	L F H R	L F H R
	H L R	H L R	L R H	R H L
	H L R	L R H	R H L	H L R
	H R L	H R L	L H R	R L H
	H R L	L H R	R L H	H R L

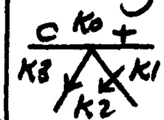
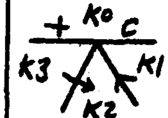
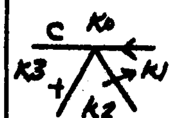
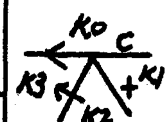
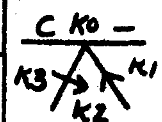
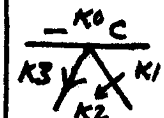
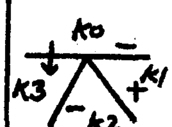
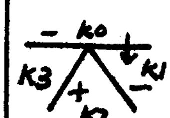
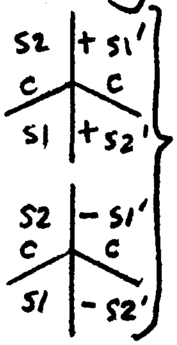
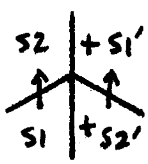
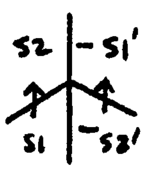
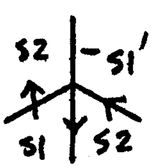
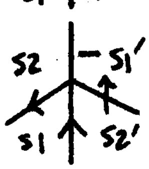
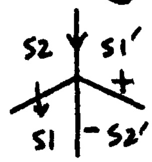
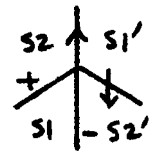
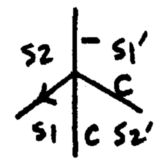
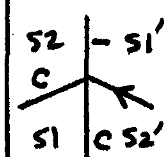
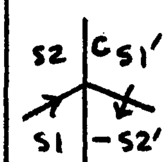
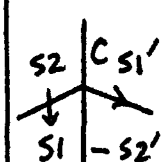
KAY labeling	Allowable surface orientations for: K0	K1	K2	K3
	L F H H	H H R F	H H R F	L F H H
	R F H H	R F H H	H H L F	H H L F
	R H F H	H F H L	H F H L	R H F H
	L F H H	L F H H	H H R F	H H R F
	H H L F R	R F H H L	H H L F R	H H L F R
	H H L R F	H H L R F	H H L R F	L F R H H
	B H	L L	R R	B H
	B H	B H	L L	R R

Figure 6

PSI labeling	Allowable surface orientations for: S1	S2	S1′	S2′
	R L F H H	R L F H H	L H H F R	L H H F R
	H H H R F L R L	H H H R F L R L	R F L H H H L R	R F L H H H L R
	H H H R F L R L B B B R F L	H H H R F L R L B B B R F L	R F L H H H L R R F L B B B	R F L H H H L R R F L B B B
	H	H	R	Any
	Any	L	H	H

PSI Labeling	Allowable surface orientations for: S1	S2	S1′	S2′
	R H	R H	H L	L R
	R L	H R	L H	L H
	H R L H F	L H R F H	H R L H F	H R L H F
	H L R H F	H L R H F	R H L F H	H L R H F
	R F	H H	H H	H H
	H H	H H	H H	L F

In the rectangular vertex case there is a particularly simple relationship between orientation and possible lighting. Table 2 shows this relationship. Using the results of this table and Figure 6, we can now assign lighting possibilities to each of the possible junctions previously classified by line labels and region orientation. Rather than show the complete set, I have chosen to show only the possibilities for junctions which border the background, but the same methods can be used to assign values for all junctions. The results of this process are shown in Figure 7.

Table 2.

$Q(\theta_S)$ \ $Q(\theta_{EL})$	LV	FV	RV	H
F	I	I	I	I
FL	I	I	G	I
LF	I	I	G	I
L	I	G	S	I
LB	G	S	S	I
BL	G	S	S	I
B	S	S	S	I
BR	S	S	G	I
RB	S	S	G	I
R	S	G	I	I
RF	G	I	I	I
FR	G	I	I	I

I = illuminated, SA, SB or SAB
S = shadowed, SC
G = glancing illumination; may be I or S or degenerate.

Figure 7

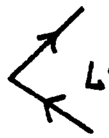
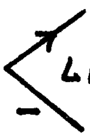
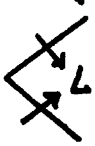
ELL junction labeling	orientation/illumination for surfaces: L0	L1	Lighting must be between:
L0  L1	B/I	H, L, F, R	——
L0  L1	H, L, F, R	B/I	——
L0  L1	B/I B/I	L/I F/I	(F, BL) (F, L)
L0  L1	B/I B/I	R/I F/I	(BR, F) (R, F)
L0  L1	B/I	B/SA, B/SB	——
L0  L1	B/SA, B/SB	B/I	——

Figure 7

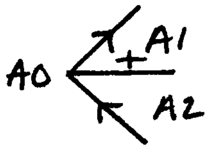
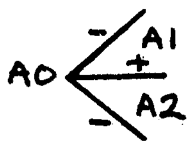
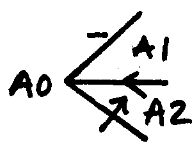
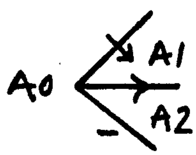
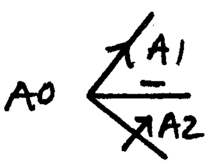
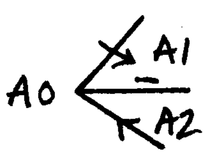
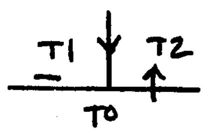
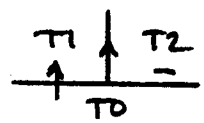
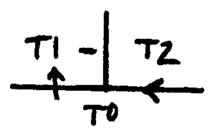
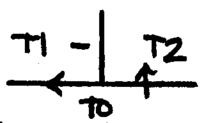
ARROW junction labeling	orientation/illumination for regions: A0	A1	A2	Lighting must be between:
	B/I	H/I	L/I	(RF, BL)
	B/I	H/I	F/I	(R, L)
	B/I	H/I	L/SC	(BL, FR)
	B/I	H/I	F/SC	(L, R)
	B/I	R/I	H/I	(BR, LF)
	B/I	F/I	H/I	(R, L)
	B/I	R/SC	H/I	(FL, RB)
	B/I	F/SC	H/I	(L, R)
	B/I	L/I	R/I	(RF, LF)
	B/I	R/I	B/SA	(F, LF)
	B/I	F/I	B/SA	(F, L)
	B/I	B/SA	L/I	(RF, F)
	B/I	B/SA	F/I	(R, F)
	B/I	L/SC	B/SA	(LB, B)
	B/I	F/SC	B/SA	(L, B)
	B/I	B/SA	R/SC	(B, RB)
	B/I	B/SA	F/SC	(B, R)

Figure 7

TEE junction labeling	orientation/illumination for regions: T0	T1	T2	lighting must be between:
	B/I	R/G	B/SA	(FL, LF)
	B/I	F/G	B/SA	L only
	B/I	B/SA	L/G	(RF, FR)
	B/I	B/SA	F/G	R only
	B/I	B/SA	R/SC	B only
	B/I	B/SA	F/SC	B only
	B/I	L/SC	B/SA	B only
	B/I	F/SC	B/SA	B only

All other TEEs have no information (or very little) with respect to light placement, so I will not list them.

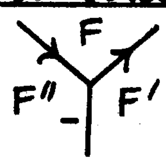
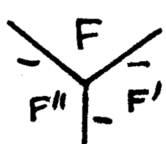
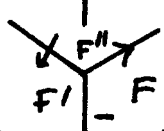
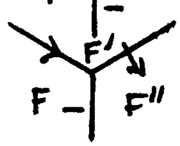
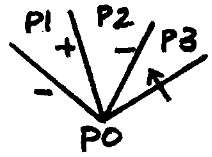
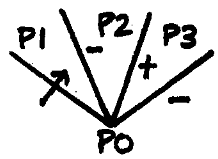
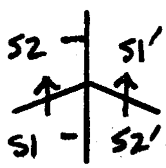
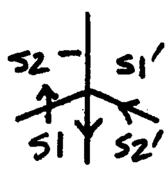
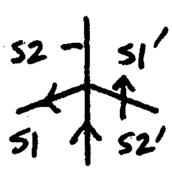
FORK junction labeling	F	F′	F″	lighting must be between:
	B/I	H/I	R/I	(BR, LF)
	B/I	H/SB	R/SC	(FL, RB)
	B/I	H/I	F/I	(R, L)
	B/I	H/SB	F/SC	(L, R)
	B/I	L/I	H/I	(RF, BL)
	B/I	L/SC	H/SB	(LB, FR)
	B/I	F/I	H/I	(R, L)
	B/I	F/SC	H/SB	(L, R)
	B/I	L/I	R/I	(RF, LF)
	B/I	L/SC	R/SC	(LB, RB)
	B/I	L/I	R/I	(RF, LF)
	L/SC	B/SA	B/I	(B, R)
	R/SC	B/I	B/SA	(L, B)

Figure 7

PEAK junction labeling	orientation/illumination for regions: P0	P1	P2	P3	lighting must be between:
	B/I	L/I	R/SC	B/SA	(FL, BL)
	B/I	B/SA	L/SC	R/I	(BR, FR)

PSI junction labeling	S1	S2	S1′	S2′	lighting must be between:
	B/I	B/SA, B/SB	R/SB	R/I	(BR, LF)
	B/I	B/SA, B/SB	F/SB	F/I	(R, L)
	L/I	L/SB	B/SA, B/SB	B/I	(RF, BL)
	F/I	F/SB	B/SA, B/SB	B/I	(R, L)
	B/I	B/SA, B/SB	L/SB	L/I	(RF, BL)
	R/I	R/SB	B/SA, B/SB	B/I	(BR, LF)
	H/I	H/SA	R/SC	B/I	(R, B)
	B/I	L/SC	H/SA	H/I	(B, L)

Let me review exactly what we need to do to arrive at these listings. First we must generate a full set of junction line labelings for each junction type. We can then take each labeling and assign region values for all the possible orientations in which this labeling can occur. (Notice that in those junctions which have cracks, I have included some gravity information, on the basis of the observation that certain orientations can never occur if gravity is assumed to be in the -y direction, i.e. normal to the suppport surface and downward. I will have more to say about this later, along with information about support which can be deduced from local features.) We have thus divided each possible labeling for lines into several labelings, each of which has the same line labels, but different region labels. We can then take each member of this extended set and divide it into several possibilities according to lighting direction.

Notice that in some cases, particularly in those junctions I have called "shadow causing junctions", there is only one possibility for region illumination of the junction. In these cases we can then say that the junction as labeled can only be present in the scene if the lighting is from a particular subset of the set of possible positions. Thus knowing that the lighting is from a particular position we can constrain the possible labelings for junctions in the line drawing and in the opposite

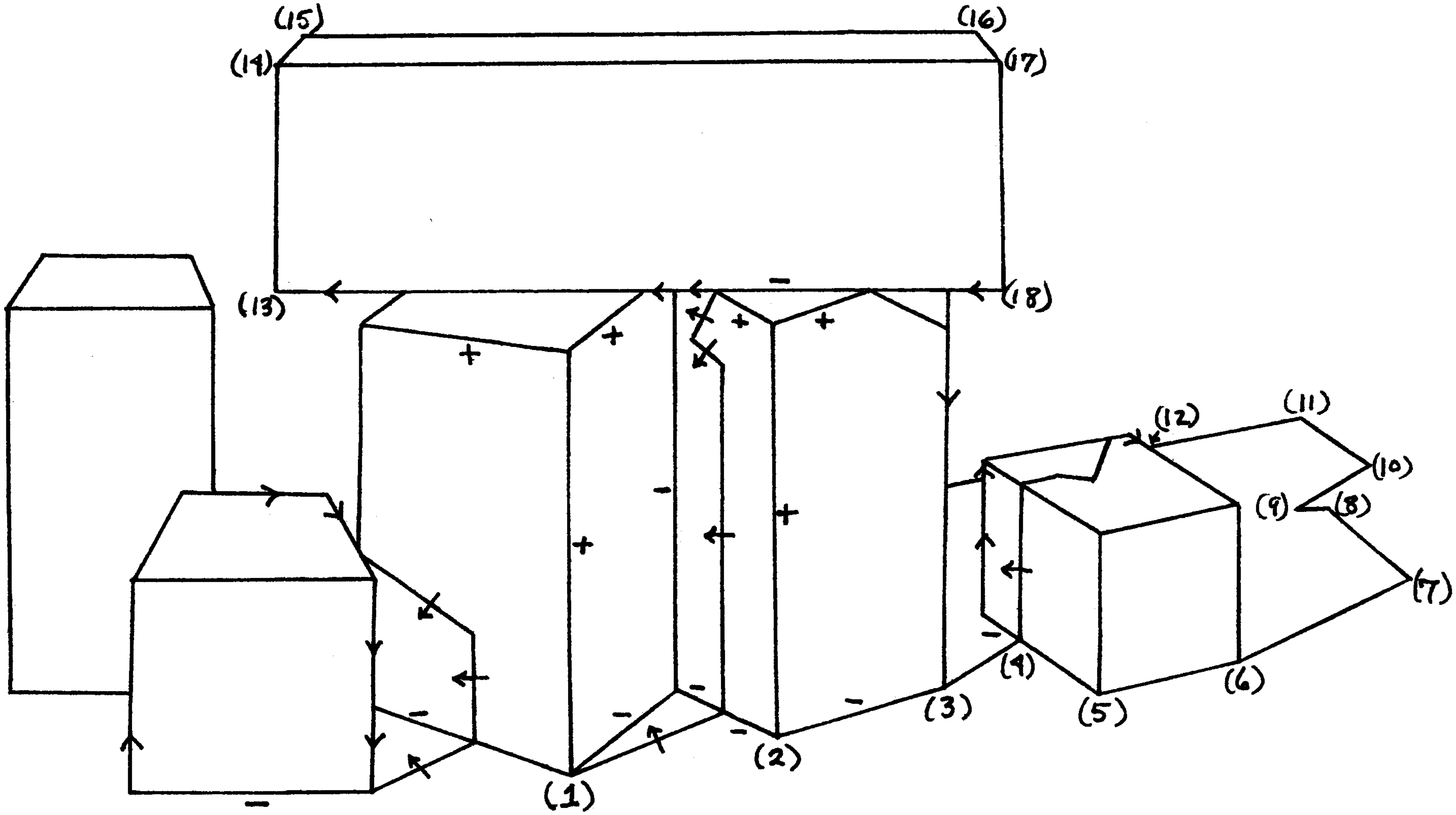


Figure 8

direction of implication, knowing a particular junction labeling we can constrain the possible light positions.

VI.B. An example

To show how these facts help us in labeling a scene, let us now return the example started in the previous section. Given the partial labeling, of 5C and assuming that all its vertices are rectangular with no degeneracies, look first at the junction labeled (1) in Figure 8. Given the line labels and refering to figure 7, there is only one possibility for labeling the regions:

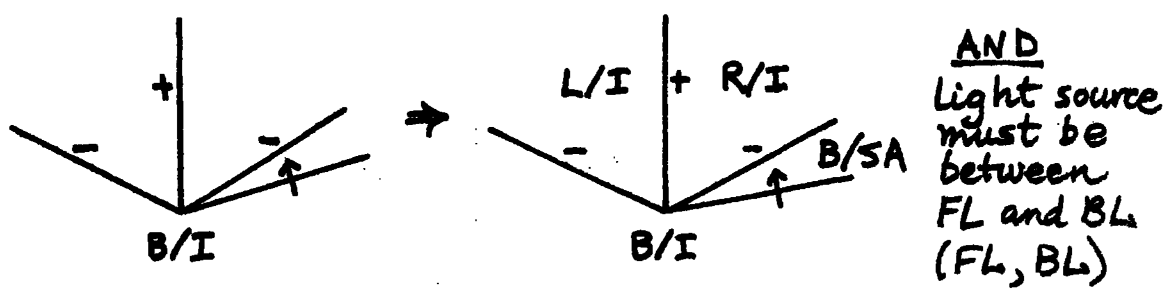


Figure 8A

At junction (2) by the same reasoning we have:

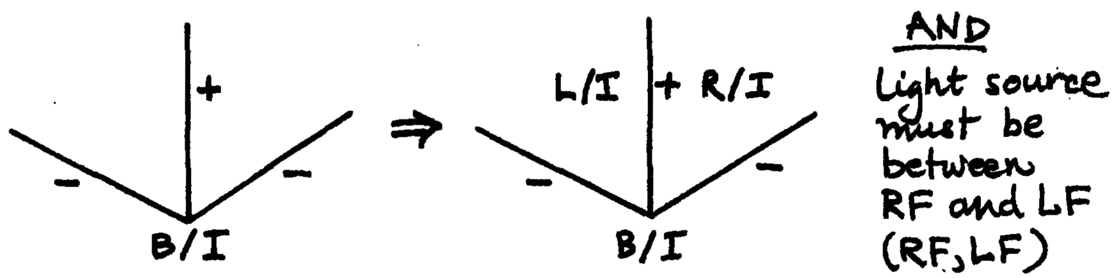


Figure 8B

Thus we may deduce that the lighting must be somewhere in the intersection of the two intervals specified by (1) and (2), namely (RF, LF) $\wedge$ (FL, BL) = (FL, LF).

At (3) we now have:

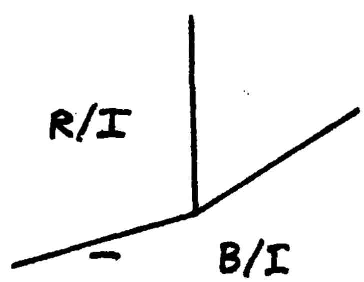


Figure 8C

so we must look for a junction which agrees by region type, line label, and lighting. The only two candidates which agree on line labeling are:

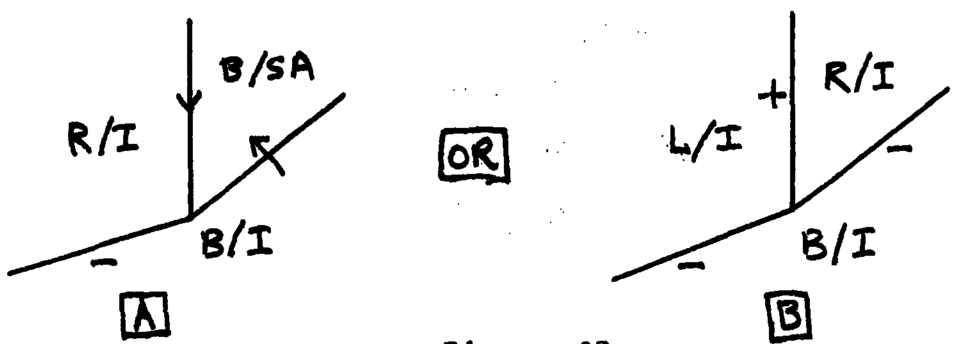


Figure 8D

and clearly only the first is possible when region labels are considered. Hence we can label the other two segments and the other two lines. No further restriction on lighting is implied, since (F, LF) $\bigwedge$ (FL, LF) = (FL, LF).

At (4) we now have:

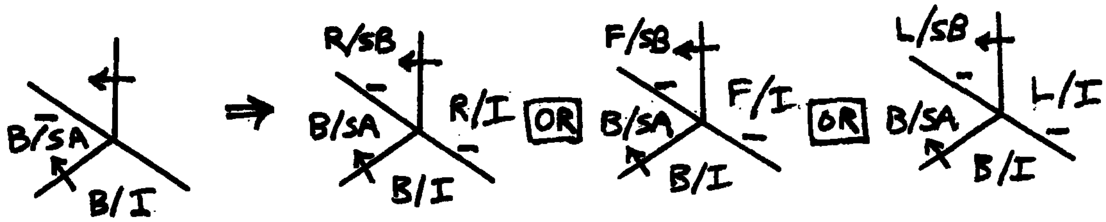


Figure 8E

and all these are possible. However, we can now label all the line segments, even though we are not able to label the regions yet. We do know that the two regions for which we do not yet have an orientation label must be given the same orientation label:

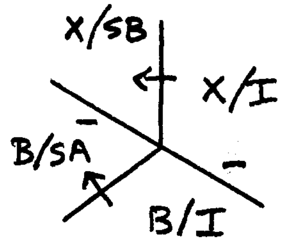


Figure 8F

At (5) we can have either:

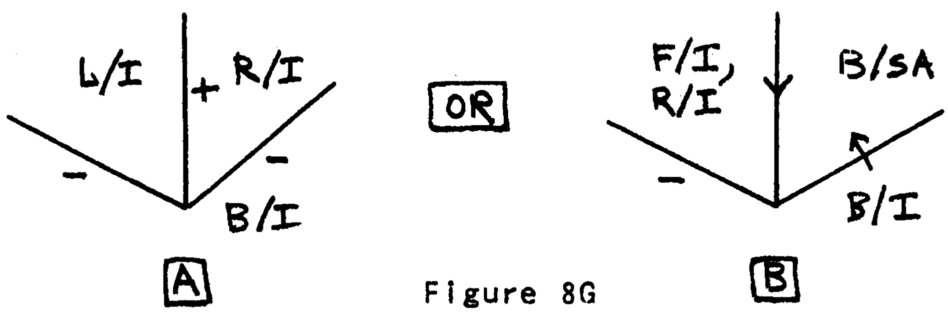


Figure 8G

At (6) we can thus have, if A is the correct labeling, only:

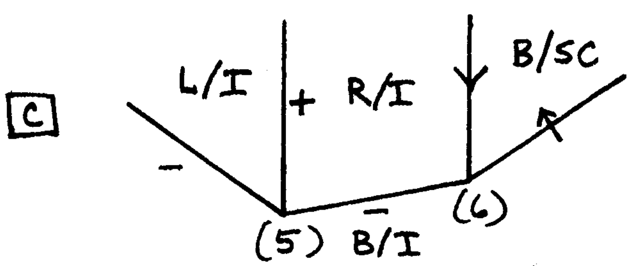


Figure 8H

whereas if B is correct we can have either:

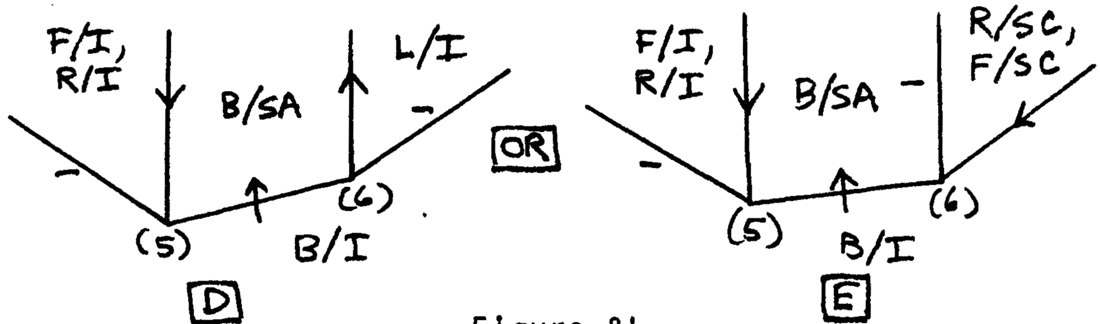


Figure 81

However, if D were true, junction (6) would imply that the lighting was between (R, F) and (R, F) $\cap$ (FL, LF) = nil. Similarly the lighting possibilities given E as the proper labeling have an empty intersection with (FL, LF), so we can eliminate both D and E, leaving only C. This now enables us to go back to (4) and assign X = L.

Because (6) must be labeled as in C, we can label all on the lines in junctions (7) through (11) as shadow lines, since these are the only possible continuations, regardless of lighting position.

Junction (12) must be labeled as:

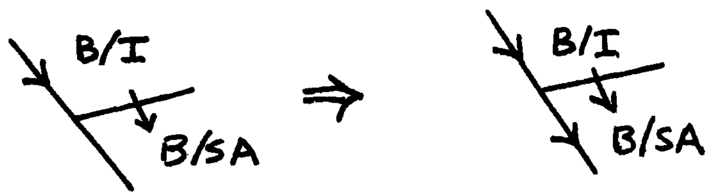


Figure 8J

since again this is the only possibility.

Junction (13) must be labeled as:

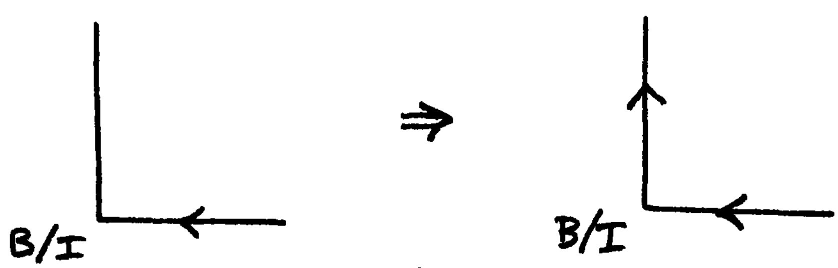


Figure 8K

since there is no other labeling which has one line labeled as shown and which also is possible given the lighting direction. This then allows us to label (14) through (18) as forced choices on a line labeling basis alone. Notice that this forces region R1 to be labeled "FV", since this is the only region orientation which can be bounded by both (14) and (18).

I will not continue to show each step, but will leave the rest as an exercise, if you are interested enough to complete it. The final results are shown in Figure 9. There are three ambiguous line segments, and part of my work to be done in the future will deal with ways of resolving these ambiguities.

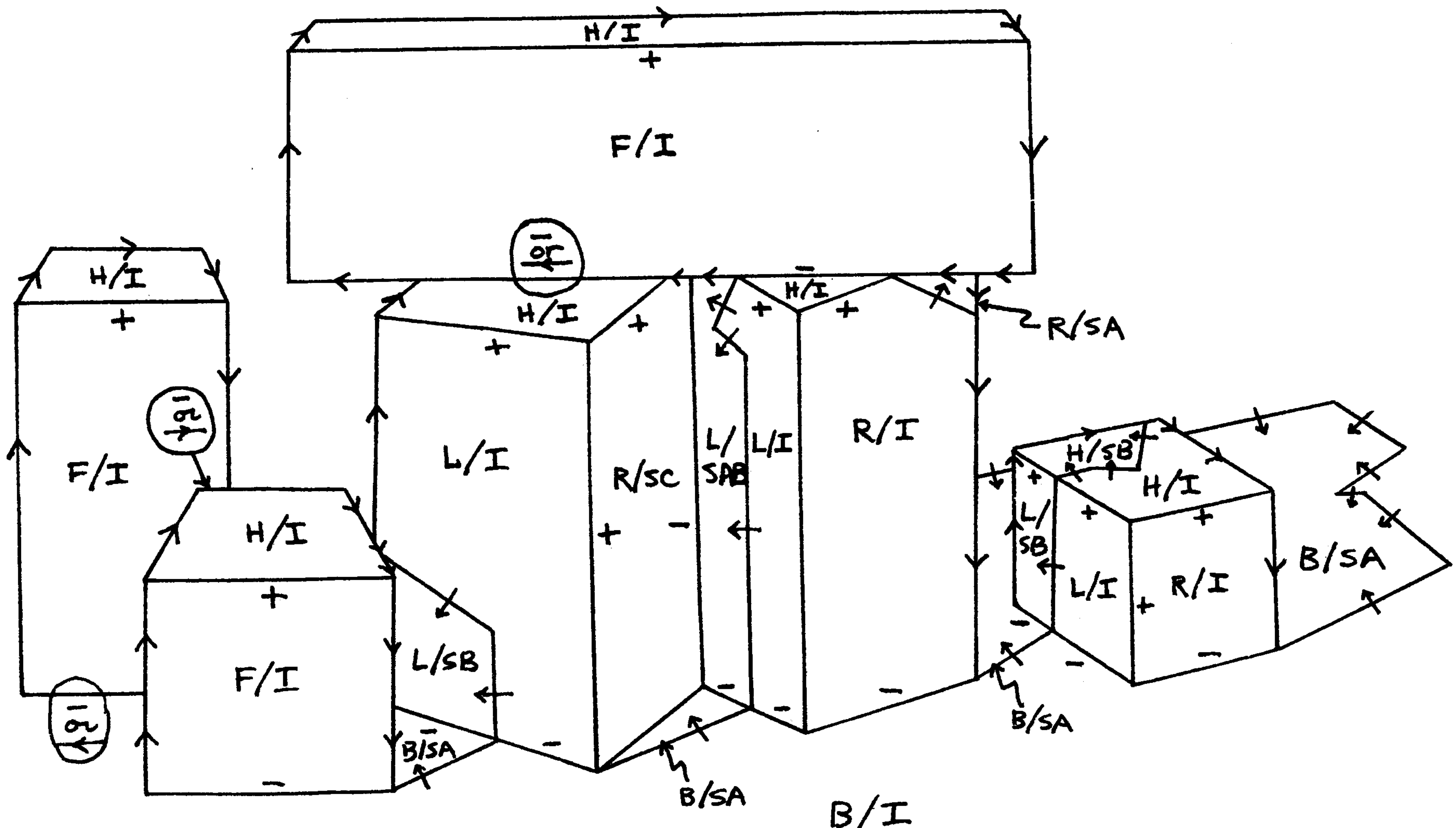

H/I
+
F/I
H/I
+
F/I
H/I
+
F/I
−
L/SB
B/SA
H/I
+
L/I
R/SC
+
−
L/
SAB
L/I
H/I
R/I
−
R/SA
B/SA
B/I
H/SB
H/I
L/
SB
L/I
R/I
B/SA
B/SA

Figure 9

VII. Line segment orientations

In general we will not be able to totally label a line drawing with only the information we used in the preceeding example. Fortunately, there are ways of extracting information which we have not yet exploited.

VII.A. Background/scene boundary revisited

We can assign quantized orientation values to each of the line segments we encounter while moving around the scene in our normal (clockwise as seen from the background) direction. We already know that each line segment on this boundary must be labeled obscure, concave, or shadow, and that in the restricted case of blocks only, which I will continue to use for ease of exposition, each region separated from the background by one of these line segments must be H, FV, RV, LV, or B/SA. I will assign line orientation values from a set of eight compass points, so that for instance if we move directly upward along a line segment, I will label that line N, and so on. N, E, S, and W will be defined to include lines which fall within about 5 degrees of vertical or horizontal, and the angles in between will then be denoted by NE, SE, SW, and NW. (The width of the regions is a function of eye distance; as the distance goes to infinity, the

N, S, E, and W sectors can become arbitrarily small.)

We can then state that the following facts are true by exhaustively considering all possibilities:

1. A shadow line may have any orientation.

2. If the line L is an obscuring line, and X denotes the region separated from the background by L, and O(L) = the orientation of L, then:

O(L) = N ⇒ X = RV, FV, or H.

O(L) = NE ⇒ X = RV (*)

O(L) = E ⇒ X = FV (*)

O(L) = SE ⇒ X = LV (*)

O(L) = S ⇒ X = LV, FV, or H.

O(L) = {SW, W, NW} ⇒ X = H.

(*) indicates that the object of which this edge is a part is supported off the surface or else is concave.

3. If L is a concave edge then:

O(L) = {N, S, SW, W, NW} ⇒ there is no such X with a concave edge.

O(L) = NE ⇒ X = RV/I.

O(L) = E ⇒ X = FV/I.

O(L) = SE ⇛ X = LV/I.

Using the same approach with other line labels, we can generate table 3, which gives the possible region labels for lines with any label and orientation. This table can then be used to generate figure 6, which I presented in section VI.A. with no justification.

VII.B. Interior forks

Rather than give a full treatment of interior junctions, I will treat only one particularly useful junction. If in a scene with only rectangular vertices we encounter a fork in the interior of a scene (i.e. none of its segments is part of the illuminated background) then the corresponding vertex is type I (all line segments convex) if and only if one line segment points "south" from the junction; if there is a type I junction of this sort, then we can also label its regions:

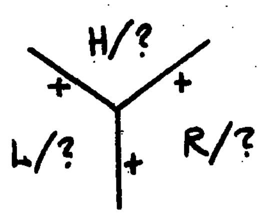


Figure 17A

Table 3

		•→	•←	•=	•+	•c	•↓	•↑
N	X	Any	R,F,H	none	L	X=Y ≠ B	X=Y =Any	X=Y =Any
	Y	L,F,H	Any	none	R			
NE	X	Any	R	R	H	X=Y ≠ {B, L, F}	"	"
	Y	H	Any	B,H	R			
E	X	Any	F	F	H	X=Y ≠ {B, R, L}	"	"
	Y	H	Any	B,H	F			
SE	X	Any	L	L	H	X=Y ≠ {B, R, F}	"	"
	Y	H	Any	B,H	L			
S	X	Any	L,F,H	none	R	X=Y ≠ B	"	"
	Y	R,F,H	Any	none	L			
SW	X	Any	H	B,H	R	X=Y ≠ {B, L, F}	"	"
	Y	R	Any	R	H			
W	X	Any	H	B,H	F	X=Y ≠ {B, R, L}	"	"
	Y	F	Any	F	H			
NW	X	Any	H	B,H	L	X=Y ≠ {B, R, F}	"	"
	Y	L	Any	L	H			

Rough proof:

The entire set of possible forks is:

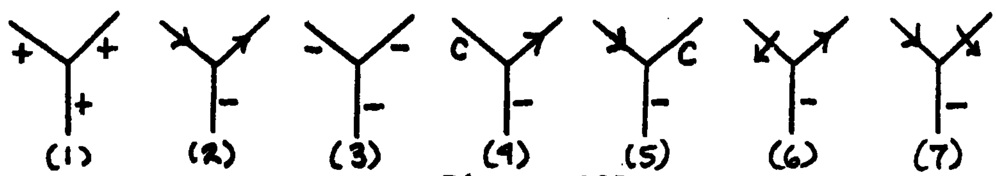


Figure 16B

Clearly (1) can be labeled as shown.

All of the others have at least one concave edge. Suppose a concave edge points "south"; then the region to its right must be LV and the region to its left must be RV. Then in (2) the obscured region must be H, but we have assumed from the beginning that the eye elevation is sufficient so that any horizontal surface with normal upward is visible unless obscured. Likewise (3) is impossible, since it must also have an H region, but the normal of this region points downward, so the surface could not be visible. Similarly (4) through (7) are impossible with their concave edges downward, and similar reasoning can be used to eliminate (2), (4), (5), and (7) in their other orientations.

VII.C. An example with degeneracies

I have picked the scene in figure 10A to illustrate not only the use of line orientations, but also the parsing of a scene with several degeneracies. We can start by labeling the line

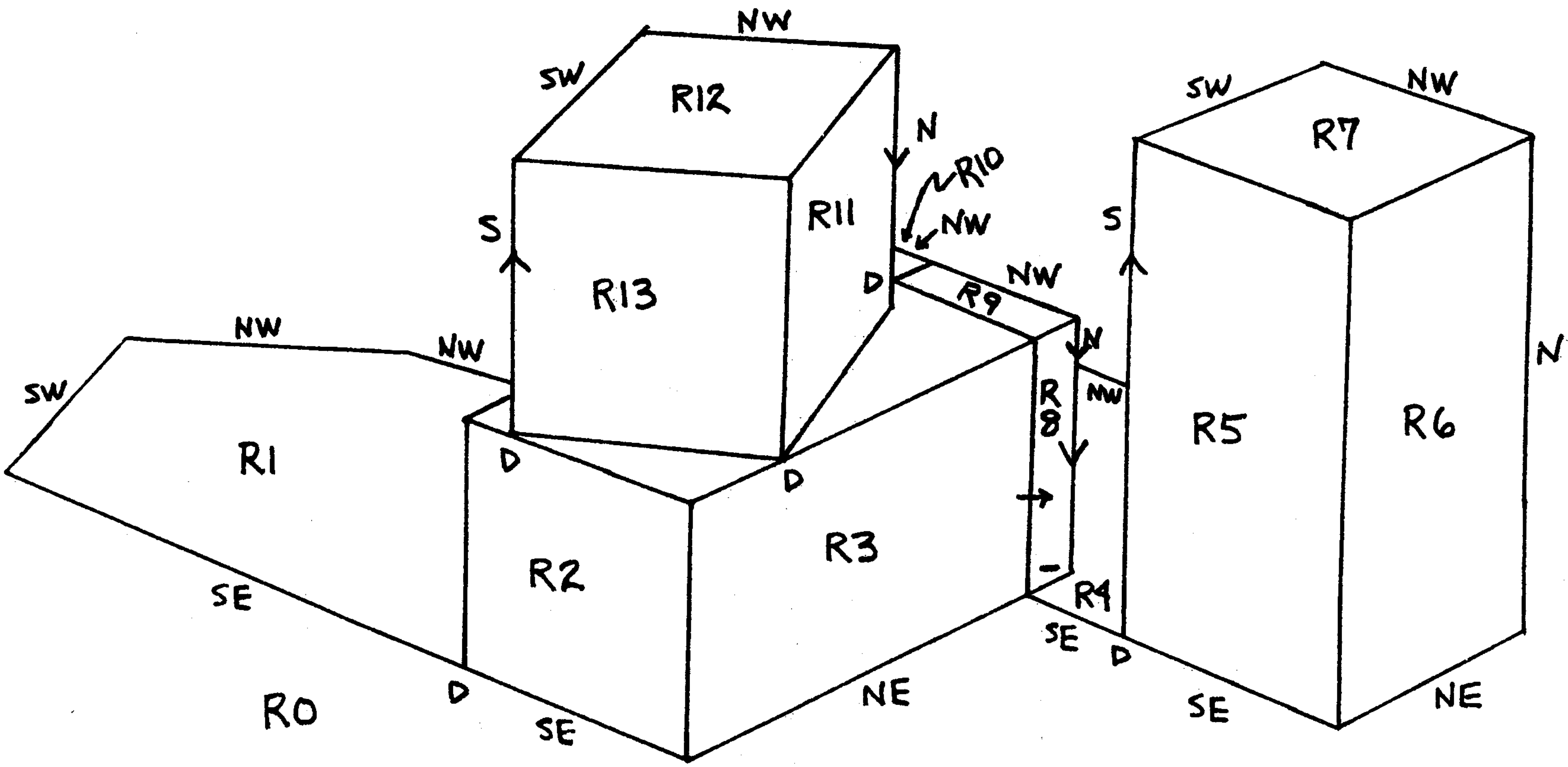


Figure 10A

Drawn to scale from a photograph; perhaps significantly I was not trying especially to photograph a scene with degeneracies, but merely recording scenes with shadows.

segments of juctions with scene/background segments of T1, T2, S1, S2, M, and F as we did in section VI.B, and then label resulting "forced" choices. Figure 10A contains all these labelings, plus orientations for each line segment around the scene/background boundary. Each degenerate junction is marked "D".

Ignoring the interior forks in this example, we can construct possible labelings for R1 through R13. Notice that for the line segments already labeled "obscure" we need not consider shadow and concave edge region possibilities. The possible labelings for each region are shown in table form in figure 10B. From the results of the application of the line orientation rules we can unambiguously label three regions and six line segments directly.

We can then proceed with "forced choices" as we did in the previous example. This process is shown step by step in Figure 10C. As the result of the application of these line orientation rules we are left with six junctions not totally labeled, three of which are degenerate. These are shown in Figure 10D.

Figure 10B

Line segment identified by region and line direction:	Region type if line segment is: concave —	obscure ←	shadow ↑	Possible region labels (intersection of possibilities for each line segment)
R1 – NW	none	H	B/SA	V(R1) = B/SA
R1 – NW	none	H	B/SA	
R1 – SW	none	H	B/SA	
R1 – SE	L/I	L	B/SA	
R2 – SE	L/I	L	B/SA	V(R2) = L or B/SA
R3 - NE	R/I	R	B/SA	V(R3) = R or B/SA
R4 - SE	L/I	L	B/SA	V(R4) = B/SA
R4 - NW	none	H	B/SA	
R5 – SE	L/I	L	B/SA	V(R5) = L
R5 – S	*	L, F, H	*	
R6 - NE	R/I	R	B/SA	V(R6) = R or B/SA
R6 - N	none	R, F, H	B/SA	
R7 - NW	none	H	B/SA	V(R7) = H or B/SA
R7 - SW	none	H	B/SA	
R8 – N	*	R, F, H	*	V(R8) = R or F or H
R9 – NW	none	H	B/SA	V(R9) = H or B/SA
R10 - NW	none	H	B/SA	V(R10) = H or B/SA
R11 - N	*	R, F, H	*	V(R11) = R or F or H
R12 - NW	none	H	B/SA	V(R12) = H or B/SA
R12 - SW	none	H	B/SA	
R13 - S	*	L, F, H	*	V(R13) = L or F or H

R3 and R8 are coplanar; therefore O(R3) = O(R8) = {R, B/SA} ∩ {R, F, H} = R. V(R3) = R/I, V(R8) = R/S(?).

* means that regions cannot have these values since the line segment is already labeled.

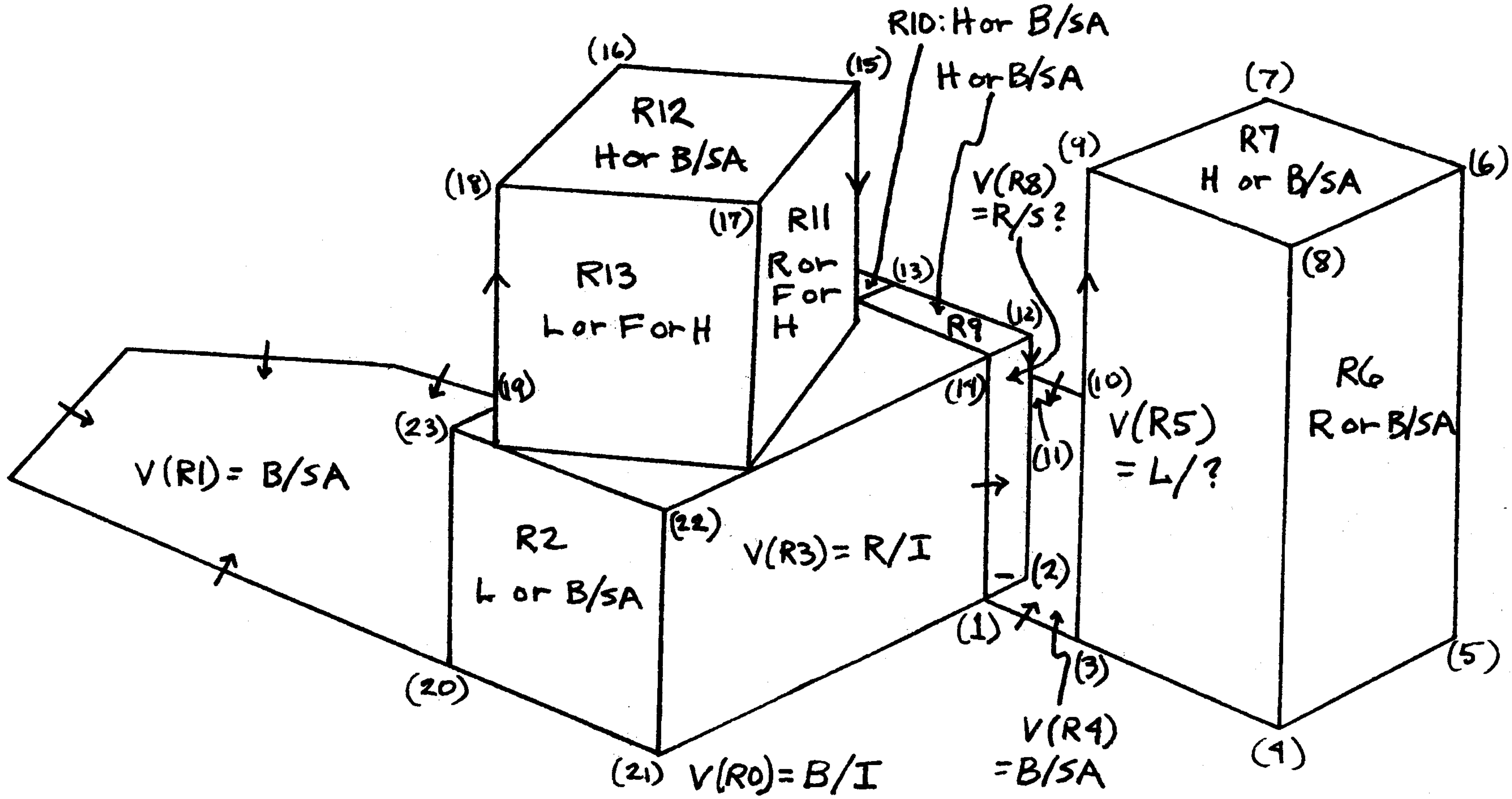
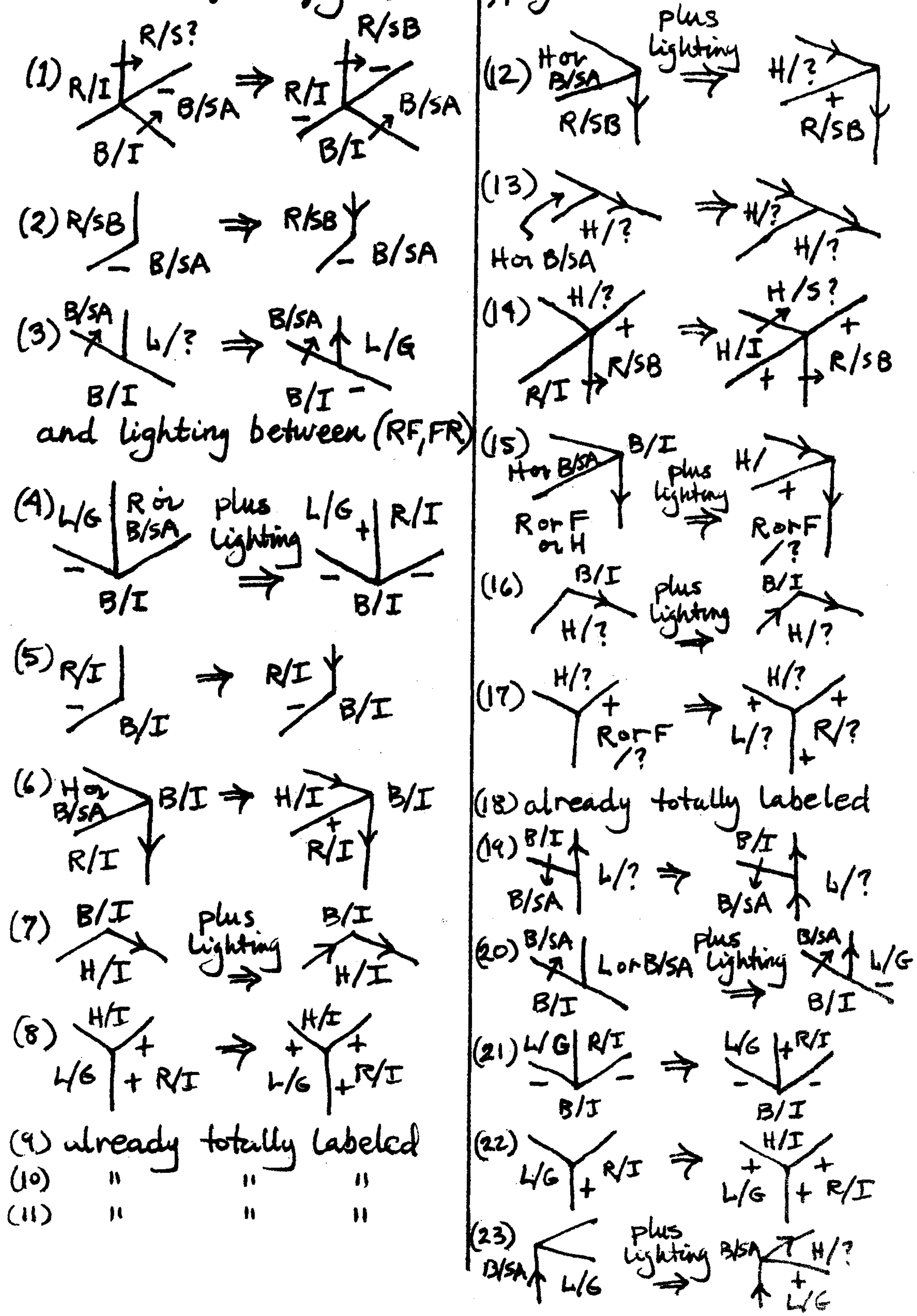

(16)
(15)
R10: H or B/SA
H or B/SA
(7)
R12
H or B/SA
(9)
R7
H or B/SA
(6)
(18)
V(R8)
=R/S?
(17)
R11
R or F or H
R13
L or F or H
(13)
(8)
(12)
R9
(19)
(10)
R6
R or B/SA
(23)
(14)
V(R5)
=L/?
(11)
V(R1) = B/SA
(22)
R2
L or B/SA
V(R3) = R/I
(2)
(1)
(3)
(5)
(20)
V(R4)
=B/SA
(4)
(21)
V(R0) = B/I

Figure 10C

Figure 10C

numbers refer to figure 10C, page i.



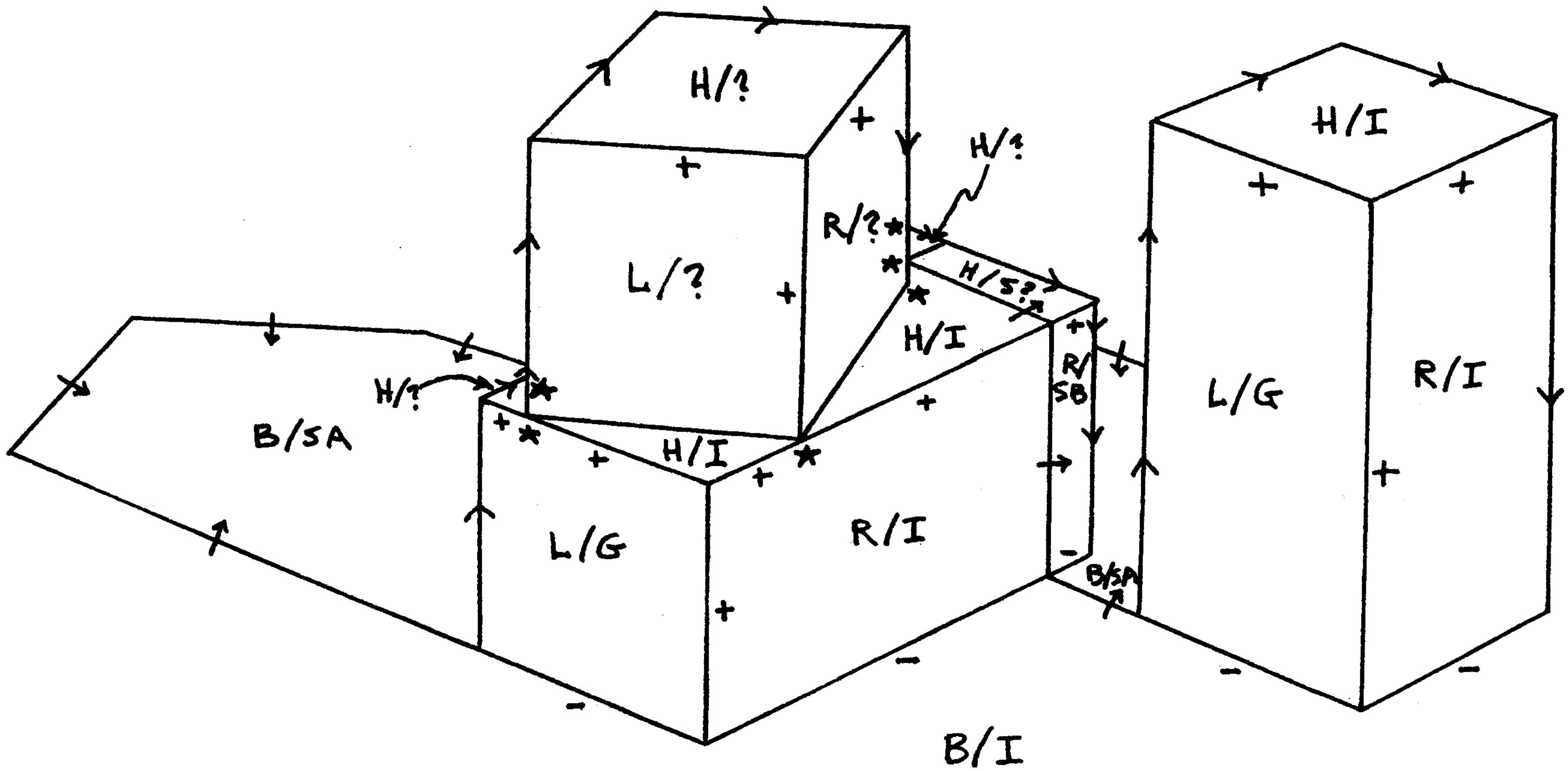


Figure 10D

Results of "forced choice" labeling; starred junctions are not fully labeled. Note however that each has at least one line and all region orientations assigned, and that each degenerate junction cannot be labeled from the allowable set of junctions.

About degenerate junctions we can say:

1. Certain junctions can never be degenerate, namely ell and tee, and possibly others in particular orientations. If ever we are left with such a junction which we cannot label, then there is a mislabeling somewhere else in the scene.

2. Any junction which does not fall into one of the allowable junction categories must be a degenerate junction.

3. Any junction which is degenerate can be broken up into:

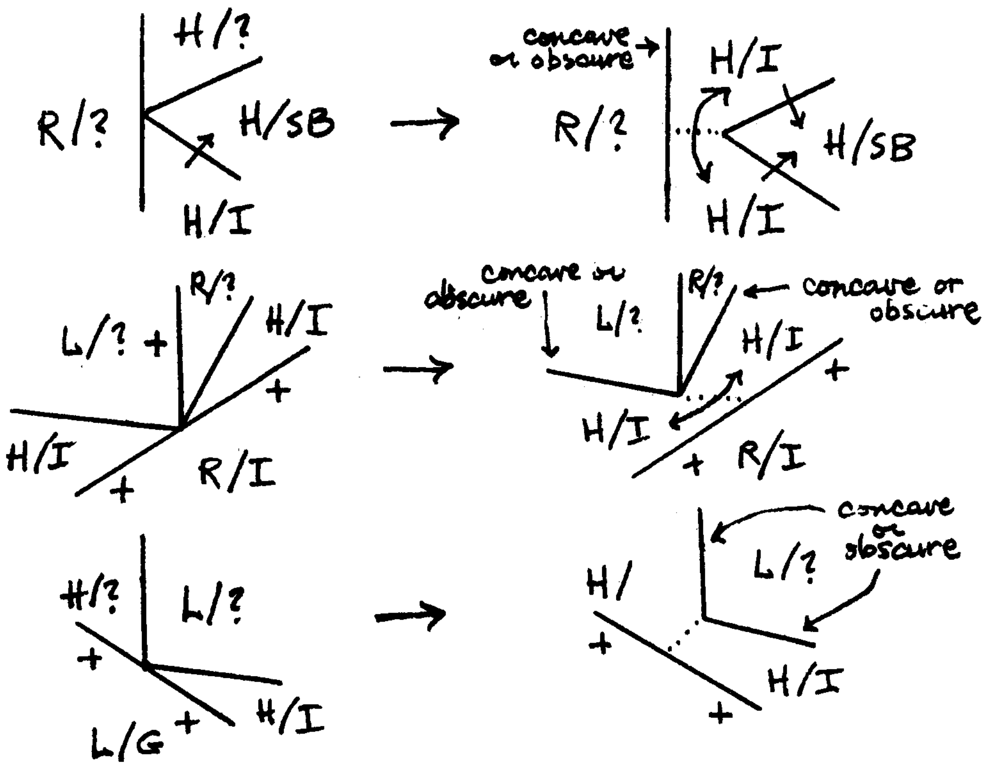
(a) two real junctions, one of which must separate along lines labeled either both obscuring or both concave, or

(b) a junction with a coincidentally aligned shadow edge.

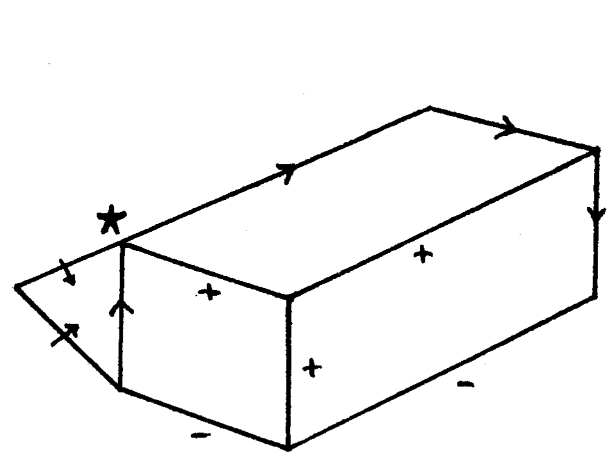
A good heuristic to note is that any straight lines in a degenerate junction which appear to be collinear are likely to be truly connected. Some cases which follow this pattern are shown in figure 10E along with some cases which do not. In any event, I have not yet devoted sufficient attention to degenerate junctions to say any more than that I can show methods to identify which junctions are degenerate, and that I can suggest some promising consistencies between various possible degenerate junctions.

Figure 10E

All three degenerate junctions left unlabeled in 10D can be decomposed by assuming as an heuristic that the straight line in the junction is a true edge, and that all the other lines make up another junction:



An example which cannot be broken up in this manner is:



VIII. Some miscellaneous points of interest

VIII.A. Support

If we ever encounter any junctions such that the regions and lines are labeled as in figure 11, we can draw conclusions about support between the objects which correspond to the regions listed. I have shown configurations which imply support, but by the same token, certain other configurations can be used to ascertain cases where support between objects is precluded.

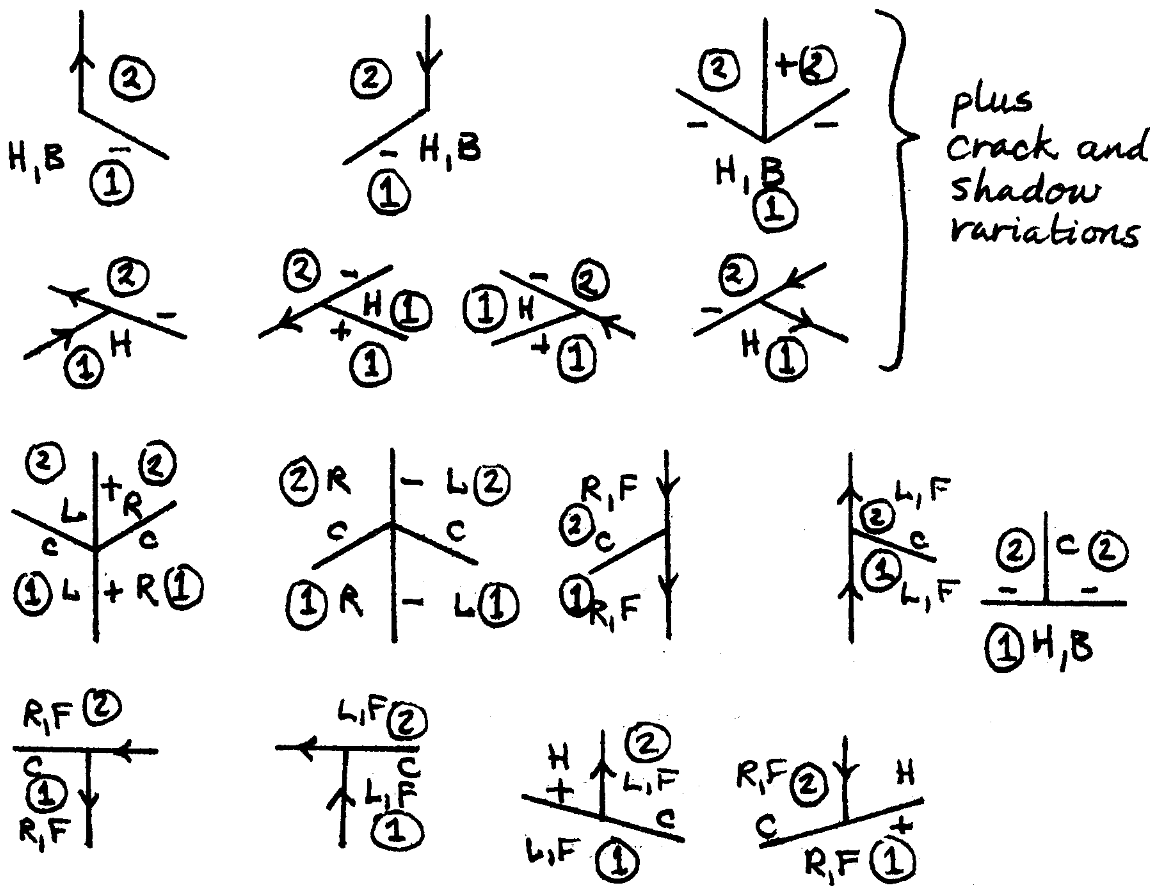
Note that if there are no shadows in the scene we will usually not be able to tell whether one block supports another or whether it merely obscures it, since shadows are the prime means of distinguishing between obscuring and concave edges.

VIII.B. Finding isolated subscenes

I mentioned at the beginning of section V that we can isolate portions of a scene which it may be possible to treat as though they were surrounded by background. To do this we need only find paths around all sets of regions in the line drawing which consist only of junctions which can occur on the background/scene boundary plus TEEs which are oriented so that T0 is in the interior of the region we are isolating. I believe that

Figure 11

① supports ② in all the following cases:



(This is not a complete list)

this is one of the most promising lines of research to be followed up.

VIII.C. Non-rectangular scenes

In this area I still have a great deal of work to do. However, from preliminary observation I believe that it may be profitable to consider every scene initially as though it were a scene with only rectangular vertices. This will accomplish at least two ends: any subportions of the scene which consist of rectangular junctions or in which non-rectangular junctions follow the same types of rules as do rectangular junctions can be labeled, and we can effectively isolate junctions which do not follow this pattern, these junctions will then be either non-rectangular or degenerate.

VIII.D. Curved surfaces

Using the results of Appendix 1, we can find conditions on the boundaries of curved surfaces, since at these boundaries the points must all satisfy the conditions V   V   = 0, where V   is the origin-eye vector and V   is the surface normal. Consequently we can also characterize the quantized region value for a curved surface at each point along its boundary, and perhaps can usefully treat such surfaces according to the range of values through which the surface normal must pass.

VIII.E. A grammar of regions

An interesting corollary to the observation that each region can have only one label is that we can describe a generative grammar for surfaces of particular types. For example, every surface which is bounded by all type 1 vertices (type 1 vertices are ones which occupy only one octant of space at the vertex, or alternately have three convex edges) can be generated by the transition rules:

A1 $\rightarrow$ T0 or L1 or A2 (label —>—)

L1 $\rightarrow$ T0 or L1 or A2 (label —>—)

T0 $\rightarrow$ T0 or L1 or A2 (label —>—)

A2 $\rightarrow$ A1 or F (label +)

F $\rightarrow$ A1 or F (label +) where any "sentence" which contains at least three junctions from the set $\{$L1, A1, A2, F$\}$ represents a real surface of this kind. Such surfaces include the faces of all convex polyhedra such as prisms, cubes, pyramids, etc. which have only trihedral vertices.

We can also list transition rules for all type SA shadow regions and regions bounded only by convex edges, as well as each of the preceeding types as it appears when partially obscured by other objects. We may be able to use these observations to recognize some regions directly, rather than arriving at labeling for regions on the basis of the lines surrounding the region which have already been labeled.

VIII.F. A finer classification of junction types

In all the work so far, I have assumed that we are unable to tell the difference between angles less than or greater than 90 degrees. A glance at the arrow and tee labelings of figure 1 should convince youthat if we have more precise information about angles, we can further partition the possible junction labelings and thus make the problem of assigning labels easier.

IX. Summary and plans for research

I have demonstrated the use of a large number of rather isolated facts and methods, and would now like to say a little about how these can all be tied together in a conceptual framework. As a first approximation to a total picture of this system, we can order the various scene variables according to how widely any particular piece of information can directly influence other scene variables.

1. Line labels affect (directly) only the junctions at both ends of the line segment.

2. Line labels plus orientation of the lines affect the junctions at both ends plus the regions on both sides of the line segment.

3. Junction labels affect all lines which are part of the junction and may in some cases be interrelated with light and eye position.

4. Junction labels plus orientation affect all lines which are part of the junction, frequently are interrelated with light and eye position, and also affect all regions which bound the junction.

5. Region labels affect all junctions and lines bounding the region, are interrelated with eye and light position, and may affect adjacent regions.

6. Region labels plus a line label definitely affect the adjacent regions as well as the other variables from (5).

7. Path labels affect all regions inside the path (as in finding an isolated subscene or moving around the scene/background boundary).

8. Light source and eye positions can potentially affect all junctions and regions.

This is a very rough hierarchy, but I think it gives some idea of the amount and types of interdependence between the various scene variables.

My plans for research include as a top priority the gaining of greater insight into just what type of conceptual framework we need to be able to lucidly descibe the nature of the problem and the methods for solving it. I also intend to follow up all the topics mentioned in the previous section, and finally I plan to write a program embodying the results of my work.

Bibliography

(1) Clowes M.B. On seeing things
AI Journal, Spring 1971

(2) Dowson M. What corners look like
MIT/AI Vision Flash 13, June 1971

(3) Dowson M. & Waltz D.L.
Shadows and cracks
MIT/AI Vision Flash 14, June 1971

(4) Finin T. Finding the skeleton of a brick
MIT/AI Vision Flash 19, August 1971

(5) Guzman A. Computer recognition of three-dimensional objects in a visual scene
Project MAC Technical report MAC-TR-59
December 1968

(6) Huffman D.A. Impossible objects as nonesense sentences
Machine Intelligence 6, 1970

Appendix 1. Exact relations between scene features and light-eye placement

The first problem is to find ways of obtaining the light-eye angles with respect to the scene. Using the coordinate system C defined in section IV.C., and assuming that the projections of the scene x and y axes define the retinal x and y axes, we then know that any vector in the scene:

$$a\hat{x}_s + b\hat{y}_s + c\hat{z}_s$$

appears in the retinal projection as

$$a\hat{x}_r + (b - c)\hat{y}_r$$

where $\hat{x}_s$, $\hat{y}_s$, $\hat{z}_s$ are unit vectors in the scene coordinates and $\hat{x}_r$ and $\hat{y}_r$ are unit vectors in the retinal coordinates.

Appendix 1.A. Eye elevation

If there are rectangular blocks in the scene which rest on the support surface, then we can extract the eye elevation by looking at the appparent angles of the corner of a block. Figure A1 shows such a corner.

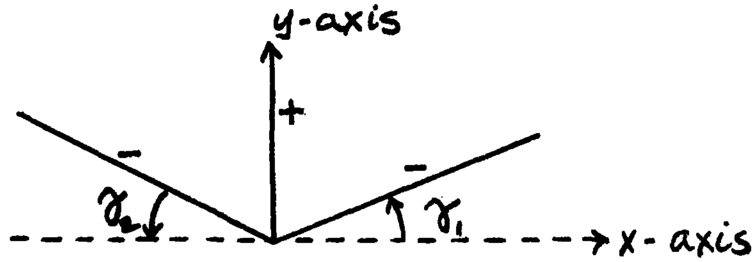


Figure A1

The three visible edges define an orthogonal coordinate system C", rotated by an angle θ_{EB} with respect to the eye coordinate system C as defined in the previous section. This is pictured in Figure A2, where the corner is drawn as it would appear from a point on the scene y-axis. We have

$$\gamma_1'' = \theta_{EB}$$
$$\gamma_2'' = 90^\circ - \theta_{EB}$$

where γ_1'' and γ_2'' are defined with respect to C", not the retina.

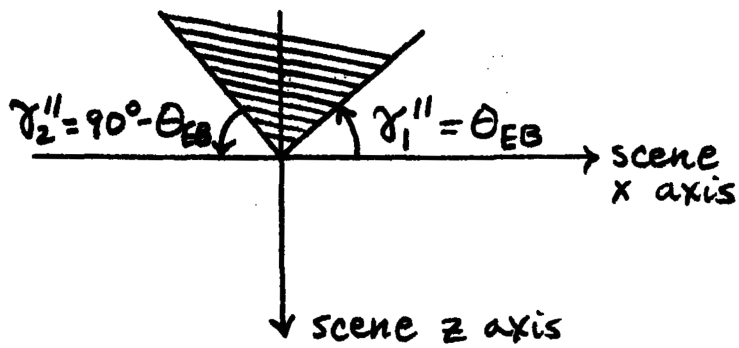


Figure A2

In the scene coordinates

$$\tan \gamma_1'' = \tan^{-1} \gamma_2'' \quad (1)$$

Furthermore we know from elementary geometry that these angles are related to those on the retinal projection by

$$\tan \gamma_1 = \tan \gamma_1'' \sin \phi_E \quad (2)$$

and $$\tan \gamma_2 = \tan \gamma_2'' \sin \phi_E \quad (3)$$

and plugging into (1) we get

$$\tan \gamma_1 / \sin \phi_E = \sin \phi_E / \tan \gamma_2 \quad (1')$$

and since ϕ_E can only be between 0 and 90 degrees, we can solve for sin ϕ_E:

$$\sin \phi_E = (\tan \gamma_1 \tan \gamma_2)^{1/2}.$$

Of course if there are no rectangular corners in the scene, we cannot use this method.

Appendix 1.B. Light placement

Whenever there is a vertical edge, visible in its entirety, which casts a shadow, visible to its first junction, we can find the light placement as a function of the eye elevation. Such a junction is shown in Figure A3.

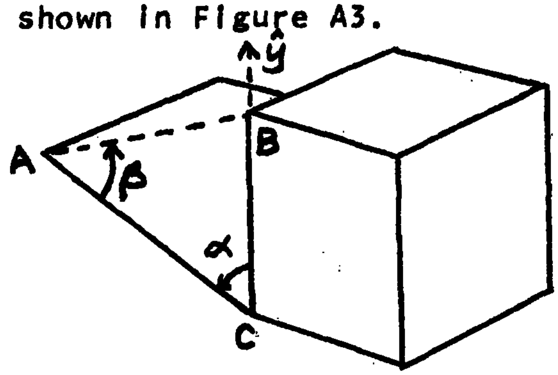


Figure A3

The angle α ($\angle$ACB) is related to the angle θ_{EL}, or angular difference between the projections of the light source and eye onto the support plane, and is independent of ϕ_L, the light source elevation. As above,

$$\tan \theta_{EL} = \tan \alpha \sin \phi_E$$

To find the light elevation is a little more complicated. First consider the vector $\overrightarrow{A'B'}$ in the scene cooordinates. This points toward the light source from the origin. If the light source elevation is ϕ_L and its angle with respect to the eye is θ_{EL}, then

$$\overrightarrow{A'B'} = (\cos \phi_L \sin \theta_{EL})\, \hat{x} + (\sin \phi_L)\, \hat{y} + (\cos \phi_L \cos \theta_{EL})\, \hat{z}.$$

If the eye elevation is given by ϕ_E, then this transforms onto the retina as

$$\overrightarrow{AB} = (\cos \phi_L \sin \theta_{EL})\, \hat{x}$$
$$+ (\sin \phi_L \cos \phi_E - \cos \phi_L \cos \theta_{EL} \sin \phi_E)\, \hat{y}.$$

Assuming α and β are both between 0 and 180 degrees,

$$\beta =$$
$$(\arctan(\sin \phi_L \cos \phi_E - \cos \phi_L \cos \theta_{EL} \sin \phi_E)/\cos \phi_L \sin \theta_{EL}))$$
$$+ (90 - \alpha).$$

Solving for $\tan \phi_L$ gives

$$\tan \phi_L = \cos \theta_{EL} \tan \phi_E + (\sin \theta_{EL} / \cos \phi_E) \tan (\beta + \alpha - 90).$$

Appendix 1.C. Finding surface normals

Whenever a region borders the suppport surface and shares an edge labeled "concave" with it, we can use its angle with respect to the horizontal to find θ_S, the angle of the projection of the surface normal with respect to the z axis of the scene. From the retina we can obtain γ_1, between -180 and 180 degrees, where γ_1 is measured from the x axis (horizontal) to the edge, with respect to the junction encountered first when moving clockwise from inside the background region around the scene/background boundary. This is illustrated in Figure A4.

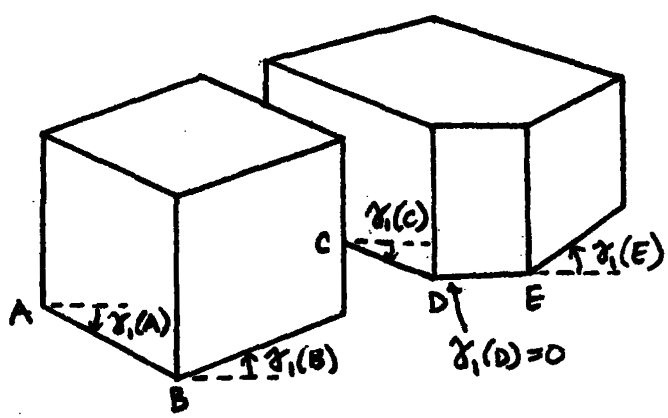


Figure A4

γ_1 and θ_S are related by the equation

$$\tan \theta_S = \tan \gamma_1 / \sin \phi_E .$$

It is not always possible to find ϕ_S , the angle between the support surface and the normal to another surface. Whenever the surface is vertical and rectangular, then its edges will appear as vertical in the projection on the retina. Obviously in these cases $\phi_S = 0$. If we assume that the surface is rectangular, then we can obtain the angle ϕ_S as a function of θ_S and ϕ_E.

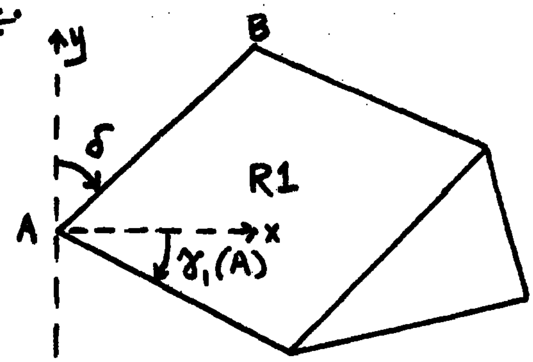


Figure A5

(Assume that R1 is rectangular)

In the scene

$$\overrightarrow{A'B'} = (-\sin \phi_S \sin \theta_S)\,\hat{x} + (\cos \phi_S)\,\hat{y} + (-\sin \phi_S \cos \theta_S)\,\hat{z}.$$

On the retina

$$\overrightarrow{AB} = -(\sin\phi_S \sin\theta_S)\,\hat{x}$$
$$+ (\cos\phi_S \cos\phi_E + \sin\phi_S \cos\theta_S \sin\phi_E)\,\hat{y},$$

so $\tan\delta =$

$$-(\sin\phi_S \sin\theta_S)/(\cos\phi_S \cos\phi_E + \sin\phi_S \cos\theta_S \sin\phi_E),$$

and solving for $\tan\phi_S$ we get

$$\tan\phi_S = (-(\sin\theta_S/\cos\phi_E)\tan^{-1}\delta - \cos\theta_S \tan\phi_E)^{-1}.$$

Appendix 1.D. An example

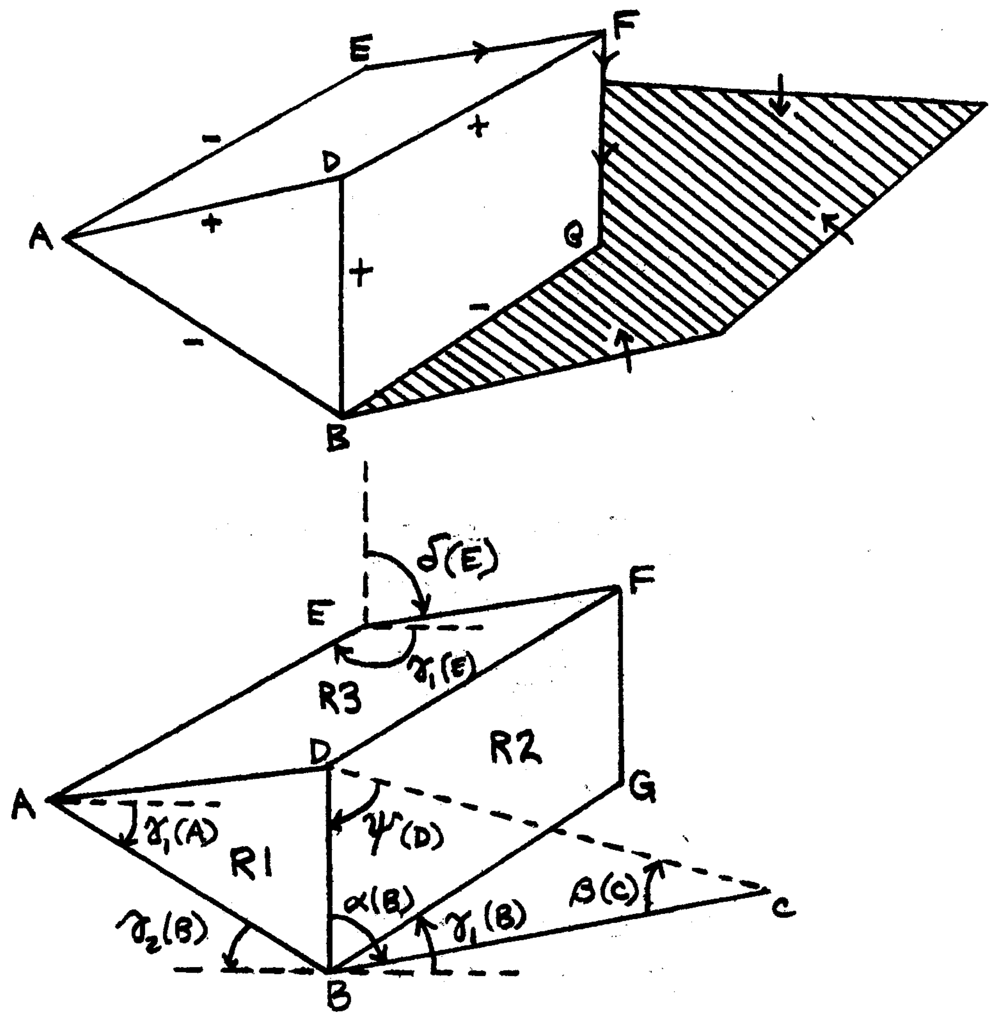
Suppose that we encounter the scene portion shown in Figure A6, and suppose further that we have beeen able to label it as shown. I will show how to find the various scene values.

Eye elevation

$$\sin \phi_E = (\tan \gamma_1 (B) \tan \gamma_2 (B))^{1/2} = ((\tan 30^\circ)(\tan 30^\circ))^{1/2}$$
$$= ((.577)(.577))^{1/2} = .577$$
$$\phi_E = 35.3^\circ$$

Light-eye angle

$$\tan \theta_{EL} = (\tan \alpha (B)) \sin \phi_E$$
$$= (\tan (-75^\circ))(.577) = (-3.73)(.577) = -2.16$$
$$\theta_{EL} = -65.2^\circ = 294.8^\circ$$



$\gamma_1(A) = -30^\circ$

$\gamma_2(B) = 30^\circ$

$\gamma_1(B) = 30^\circ$

$\gamma_1(E) = -150^\circ$

$\delta(E) = 75^\circ$

$\left\{ \begin{array}{l} \alpha(B) = 75^\circ \\ \psi(D) = 75^\circ \end{array} \right\} \Rightarrow \beta = 30^\circ$

Figure A6

Light elevation

$$\tan \phi_L = \cos \theta_{EL} \tan \phi_E + (\sin \theta_{EL} / \cos \phi_E) \tan (\beta + \alpha - 90^\circ)$$

$$= (.420)(.708) + (-.909 \ / \ .816) \tan (30^\circ + 75^\circ - 90^\circ)$$

$$= .298 + (-1.11)(\tan (-15^\circ)) = .298 - (1.11)(-.268) = .596$$

$$\phi_L = 30.8^\circ$$

Assuming that angle FEA is a right angle in the scene, we can then find the surface orientations as well.

Orientation of R1

We assume that $\phi_S(R1) = 0$, i.e. R1 is vertical, since BD is vertical. Whether or not this is so, we can find θ_S.

$$\tan \theta_S(R1) = \tan \gamma_1(A) \ / \ \sin \phi_E(A)$$

and assuming $\sin \phi_E(A) = \sin \phi_E(B)$ we get

$$\tan \theta_S(R1) =$$

$$(\tan (-30^\circ)) \ / \ (\sin (35.3^\circ)) = (-.577) \ / \ (.577) = -1.$$

$$\theta_S(R1) = -45^\circ = 315^\circ.$$

Orientation of R2

Again assume that since BD and GF are vertical, $\phi_S(R2) = 0$.

$$\tan \theta_S(R2) = \tan \gamma_1(B) \,/\, \sin \phi_E(B)$$
$$= \tan 30^\circ / \sin 35.3^\circ = (.577) / (.577) = 1.$$
$$\theta_S(R2) = 45^\circ$$

(This is really a double-check. If we assumed for finding that angle ABG was a right angle in the scene, then $\theta_S(R1) = -45^\circ \rightarrow \theta_S(R2) = 45^\circ$.)

Orientation of R3

Here the surface is not vertical, so we will find $\theta_S(R3)$ first.

$$\tan \theta_S(R3) = \tan \gamma_1(E) \,/\, \sin \phi_E(E)$$

and again assuming that $\phi_E(E) = \phi_E(B)$

$$\tan \theta_S(R3) = \tan(-150^\circ) / (\sin 35.3^\circ) = 1.$$

$$\theta_S(R3) = 225^\circ.$$

(Notice that here we have to use some additional information to tell that this angle is not 45°.)

$$\tan \phi_S(R3)$$

$$= ((-\sin \theta_S(E) / \cos \phi_E(E)) \tan^{-1} \delta(E) - \cos \theta_S(E) \tan \phi_E(E))$$

$$= (-(-.707) / (.816)(3.73) - (-.707)(.707))$$

$$= (.232 + .5) = 1 / .732 = 1.37$$

$$\phi_S(R3) = 53.8^\circ$$