WORKING PAPER 62

KNOWLEDGE ABOUT INTERFACING DESCRIPTIONS

Michael R. Dunlavey

March, 1974

Abstract

This concentrates on interactions between knowledge stated in diverse representations. It proposes a vision program that classifies any complicated object as an elaborated instance of a simple one it already understands. The resulting global-local connections facilitate evaluation of overall properties, such as visual shape and ability to support other objects.

Flexibility is achieved through simultaneous use of multiple equivalent representations. These are coordinated via interfacing rules for giving hints, constraining choices, and filling in missing detail, making use of the great redundancy in most visual scenes.

An important feature of the system consists of domain-dependent rules for guiding the flow of control and choosing hypotheses.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0005.

1. Introduction

This thesis is concerned with local-global interactions and representation-shift within a vision theory for simple assemblies of blocks, like those illustrated in Figure 1.

Local-global -- This system determines "overall shape" in the process of viewing an assembly, enabling it to follow edges and surfaces constituted in diverse ways. It determines other properties of assemblies, like ability to support one another, concurrently with viewing them.

Representation-Shift -- The system has the ability to see the same thing in more than one set of terms. For example, a square can be seen as four edges joined by angles or four angles joined by edges. This leads to flexibility in recognizing overall shapes constituted in unusual ways, and the inherent redundancy permits incomplete or occluded shapes to be perceived.

1.1 Motivation

Current vision theories, and theories in other channels of perception, have progressed to the point where they have a complexity barrier to break. Visual wholes may be composed of parts in incredibly many different ways, but up until now theories of vision have been confined to certain primitive shapes, such as childrens' blocks.

It is now time to move up from this primitive visual world. Think of the average rectangular chair back. It can be identified visually, that is, the eye can follow its edges, surfaces, and so forth. Its edges can be curved, molded, scalloped, etc. Such an edge cannot possibly be followed by any currently existing edge trackers, because they operate solely on the basis of intensity profile, or, in some cases, texture information.

My approach is to say that one must meet the problem of detail head-on, not try to ignore it statisticaly or to use reason in place of perception. I think the way to recognize a macroscopic feature, like an edge, is to get and use a detailed description of what the feature looks like in terms of microscopic primitives.

I will invent two straw men to illustrate my argument. First, the statistical approach to visual detail either means defocussing or fourier modelling. This will often have the effect of obscuring the information needed. For example, to enter a building one has to distinguish between doors and tall windows, and this often has to be done on the basis of minute semantic evidence such as the presence of a hinge and handle.

The theorem-proving approach, on the other hand, says that you identify the primitive parts and then determine the existence of the macroscopic object by essentially proving that it's there. In the extreme, this becomes quite unworkable when one considers viewing an average brick house, first identifying all umpteen bricks and reasoning up from there. What's more, this completely fails to take advantage of the basic visual similarity among houses, be they brick or wood.

My approach is a moderation of the rational approach. I say that one perceives a brick house basically as an elaborated box, having visual edges. One perceives what the edges look like, namely, a column of bricks. It is in following that column that the eye actually studies the primitive bricks and their primitive edges.

In perceiving the house, it is necessary to retain the connection between local and global, to facilitate reasoning about the object. This would allow, for example, that if one sees a water faucet in the fourth row of bricks, it appears to be in the bottom of the wall. Walls have lots of other features, too: supporting top surfaces, supported bottom surfaces, holes, etc., all of which have specific meaning in terms of bricks and surfaces of bricks.

1.2 Difference Between This Theory and Debugging Theories

I view this work as being complementary to present work on learning, automatic programming, and debugging in the following senses:

Local-Global -- My system is concerned with relating a single package of local information, like a bunch of blocks, to multiple global representations, namely visual and structural wholes, and I am more concerned with recognizing these global properties than with causing them to exist. The connections between local and global concepts formed by my system tend to form tangled orthogonal trees, rather than the relatively simple macro-expansion-plus-patches relationship between a typical program and its plan.

Representation-Shift -- I conjecture that soon the emphasis in automatic programming will switch from designing procedures to designing data structures or representations. It is probably too much to expect that representations can be chosen out of thin air. It is more reasonable to try to adapt representations that already exist for various purposes. That is what my system does when it perceives complex objects as similar to simple ones it already knows. A key to this recognition process is that each simple object has multiple equivalent representations, so it is likely that matching can be accomplished against one of them.

The issues common to this thesis and automatic programming and learning are just the issues that arise whenever new structures of information or knowledge are created - how they can be well chosen, and how they relate to other things that are known.

2. Scenario

The system tries to perceive any complicated object or relation as an elaborated instance of one it already understands. Visual shapes are perceived as simple block-like or panel-shaped bodies. Support relations are perceived in terms of these simple objects. The simple representation is used as a framework to guide more detailed analysis. Figure 2 illustrates the global and local appearance of a lapped-panel(a panel made of boards laid edge-to-edge). The system is trying to see a parallelogram, and when it attempts to verify the edges and ends, it finds a detailed row-like structure of board-ends. It then hypothesizes the structure of the panel, and interface knowledge uses this to figure out what the other edges ought to look like. This mixture of top-down and bottom-up processing is made possible by the Frame Systems paradigm of perception.

Once the system can recognize a lapped-panel and an arch-pair, it can be made to recognize a table. It is important to describe those first, since the vision system would not likely have a rich enough hypothesis repertoire to form and manipulate a description of a table as 10 blocks meaningfully arranged.

In seeing the table, its stability will be questioned, and analyzed as in Figure 3(a). Stability is verified at different levels of approximation until it is verified that the structure is stable. Figure 3(b) shows what happens in the stability analysis when the boards in the top panel are rotated 90 degrees. At the third level of breakdown, it is seen that the top boards do not lie across the arches, so support cannot be assumed. Then the vision component can be asked which boards are visibly supported, as in Figure 3(c).

This scenario is typical of the kind of processing necessary in a visual world where wholes cannot always be reduced to their ultimate parts, but must be understood at some imperfect level of approximation.

2.1 Seeing a Panel

A visual "panel" is an intermediate level concept in the visual world. A panel can be a book, a bench, a table-top, a wall. It is neither high level, like "chair" - nor low level, like "board". Chairs can be made out of panels and panels can be made out of boards. The usefulness of the panel concept is that it helps to organize descriptions. It is doubtful that one could describe furniture without it.

Panels can be constructed in too many ways to enumerate. Although it is appropriate to know a few representative panels, the "essence" of a panel has to do with visual properties like "rectangular, flat", and physical properties like "solid". Such a property is not reducible to a simple conjunction of parts and relations, but rather tends to have a complex theory of its own. For example, the theory of visual rectangles deals in edges, ends, positions, and angles. The theory of visual edges deals with abstractly linear things -- texture-intensity boundaries,

black-on-white lines, rows of objects. How can all this be tied together in perception of the panel shown in Figure 4(a)? A picture of the local-global connections for this visual panel is displayed in Figure 4(b), which says that the edges are edges of boards and the ends are rows of ends of boards.

The function of interface knowledge is to set up this cross-reference. That is, it takes a description of the thing to be seen and relates it to a visual description of the thing so that it can be seen.

Typical processing done in recognition of the rectangular panel would go something like this: The rectangle hypothesis desires to verify four edges and a physical composition. One edge hypothesis moves the fovea to an area of interest, the rightmost area of the outline of the panel, as discovered in a low-resolution scan. Within the area proximate to the fovea, bottom-up processing is performed, and a row of board-ends is found. Since an edge can consist of a row of things, provided they have a skinny aspect ratio, the edge hypothesis assumes that the edge is physically a row of board ends. Then it links up the ends of the edge with the first and last elements of the row, and suggests looking for them. When they are found, the edge is specified, visually and physically.

Now constraints are satisfied within the hypothesis structure, so it is hypothesized that the visual panel is physically a row of boards, because interface knowledge knows that the end of a row of boards is often a row of board ends. By now the panel is largely determined, although we don't yet know that it's rectangular. However, the three remaining edge hypotheses are rapidly checked out, since they are pretty well determined, and the panel is "seen". Next, a category of interface rules is applied to the object, and it is "appreciated" as being flat, solid, and other nice properties.

2.2 Descriptions for a Table

This example outlines the representations needed for understanding a table consisting of a panel supported by two parallel arches as in Figure 5(a). Prior to this example, the system has already acquired and appreciated the concepts of a lapped-panel and an arch-pair. The program perceives the panel and the arch-pair, perceives and checks the support relationship. The defective table in Figure 5(b) will fool the system until it has reason to question its stability. When it discovers why it is not stable (the boards in the top panel do not lie across the arches), it can modify the table description to check for that in the future. It perceives the overall shape as that of a large block, and appreciates that the top is a rectangular support surface. Figure 6. depicts the top-level representations for the table and its constituents.

An important part of this process is classification of the object's overall three-dimensional shape. The cylindrical representation schemes of Agin and Hollerbach [2, 3] are well suited since they are convenient for extracting support and touch information as well as visual outline.

Once an aggregate body has been seen, some questions about its non-

visual properties can be answered by referring to the visual representation, particularly questions about support, stability, and touch.

The need for interface knowledge comes when we try to define non-primitive aggregate objects such as the tower, fence, arch, etc. in Figure 1. It becomes necessary to appreciate how these objects should participate in old relations like support, left, right, at-location, inside, bigger, adjacent, standing, lying, touching, rectangular and stable, or new relations like disjoint and similar. The following are some possible interface rules for determining these relations:

(This does not yet consider plans for assembly of these objects - which entails another domain of properties about typical subassembly strategies. That domain is necessarily more complicated because it has to do with side effects. [4, 5])

SUPPORT-SURFACE -- an upward facing clear surface of a rigid, stable part of an object, such that there are no higher surfaces immediately nearby.

SUPPORTABLE-SURFACE -- is a downward-facing surface of a solid part of the object such that there are no parts of the object obstructing the surface and no lower surfaces immediately nearby.

SUPPORTS -- one object is supported by another if
1) the second one is rigid and has support-surfaces, and the first is rigid and has supportable surfaces, and one or more support surfaces of the first touch support surfaces of the second.
2) the second one is not rigid, but every rigid subgroup of it is stably supported by the first.

STABLE -- A block is stable if its CG is above the touch area of its supportable surface. An object is stable if each supportable surface is supported and each physical part is stable.

SURFACE -- a simple fragment of a face of an object, a face of an object, a proximate group of coplanar surfaces facing the same way, or a row of coplanar surfaces facing the same way.

PROJECTING -- a surface projects if it is "higher" than any of its nearby neighbors.

INDENTED -- opposite of projecting.

TOUCH -- holds between one surface and another if the first is facing opposite the second, and they are coplanar and overlapping. If one is indented, the other must be projecting, have sufficient height of projection, and be totally inside the first.

INSIDE -- one object is inside another if the convex hull of the first is inside the convex hull of the second.

BIGGER -- on object is bigger than another if its smallest enclosing parallelepiped has more volume.

DISJOINT -- one object is disjoint from another if they have no shared principal parts.

PRINCIPAL-PART -- a imperatively present part or relationship.

ALIKE -- comparison reveals unimportant differences.

ALIGNED -- two objects are aligned if they touch and if the convex outlines of their touching surfaces are coextensive.

STANDING -- an object is standing if it is solid, has proximate parts,

and either
1) it is semirigid and either its long or intermediate dimension is vertical, or
2) it is not semirigid but every semirigid subgroup is standing.

SEMIRIGID -- a group of parts is semirigid if each part is stably supported by other members of the group or by the supporter of the group.

Of course, this is not a complete list.

2.3 Frame System Operation - Low Level Vision

A frame is a conceptual chunk of knowledge - it behaves as both hypothesis and fact depending on its state of verification. It has terminals, which are like variables - to which it attaches subframes. The primary activity of a frame is to try to assign satisfactory subframes to its terminals. The linguistic analogy of a frame is like a node on a parse tree, except that the terminals of a frame are clustered into semi-independent subgroups, each of which are capable of fully satisfying the frame, such as the angle and line subgroups of a polygon.

A frame for a parallelogram would consist of two groups of terminals, four lines and four angles, connected by constraint links. The opposite lines would be linked to be parallel, and the lines and angles would link to each other to express the connectivity. Figure 7. depicts a parallelogram frame with constraint links.

A parallelogram is recognized by something like the following process: First the parallelogram frame is created from a template. Since it requires angles and lines, a class of angle and line subframes are created and allowed to find a number of lines and angles in the foveal area. Now the parallelogram frame can begin to assign its terminals to them.

Each terminal is associated with a description of how tightly it is constrained, to be used in choosing which terminal to assign next. In this example, a simple number will suffice. Line-angle connectedness will have unit weight, while line-parallelism will have weight 0.5. Each time a terminal is assigned to a line in the scene, constraints propogate from it to the angles to which it is linked, and its opposite line terminal. Then the terminal which is most constrained is chosen for next assignment. When the figure is almost complete, the constraints on remaining terminals may be so strong that their subframes are simply assumed instead of searched for. Figure 8. graphically depicts the process of recognizing a parallelogram.

A central feature of frames is the ability to replace a bad frame by a better one without losing information. A bad frame is manifested by difficulty in assigning terminals. For example, if the parallelogram frame is actually looking at a triangle, it will only be able to find three lines and angles. Associated with the frame are "error expert" programs which classify the configuration of errors and suggest which alternate kind of frame would be more appropriate. The parallelogram

frame is replaced by the triangle frame, old terminal assignments are spliced into the new, and processing procedes. Frame-shift may occur several times and at different levels in recognizing a complex scene. This leads to great efficiency compared to classic search techniques, because information is not necessarily discarded upon changing a hypothesis.

Frames should also be able to capture, where needed, some of the flavor of context-free grammar, in which subframes are not shared by more than one parent, and no subframes can be left over. No global conventions about these can be sufficient. For example, as shown in Figure 9., the parallelogram faces of a cube must share certain boundary lines, but they must not share any corner angles, and if there are any leftover lines or angles in the scene, these could very well indicate the presence of another object.

Another mechanism present in frames is the ability to "excuse" the absence of certain terminals under appropriate conditions, such as occlusion. The example depicted in Figure 10. is of two parallelograms, one of which occludes the other. Two angles and a line cannot be seen because of the occlusion, so they are excused.

The occlusion example illustrates that the line and angle frames have to be somewhat smart. That is, a line knows which side is inside, it takes note of auxiliary lines branching off of it, and it is sensitive to being partially occluded in the middle or at the ends. An angle is sensitive to being part of a vertex, especially of a type which suggests occlusion.

All of these features should apply upward to more complex objects like blocks, hopefully allowing comparable performance to existing systems, as in Figure 11., except for holes and shadows.

References

1. Minsky, M., "Frame Systems: a Theory for Representation of Knowledge." (in preparation), Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

2. Agin, G., "Representation and Description of Curved Objects." Memo AIM-173, Stanford Artificial Intelligence Project, October 1972.

3. Hollerbach, J., "Heirarchical Shape Description of Objects by Selection and Modification of Prototypes." M. S. Thesis, Massachusetts Institute of Technology, 1974.

4. Fahlman, S., "A Planning System for Robot Construction Tasks." AI TR-283, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, may 1973.

5. Goldstein, I., "Understanding Fixed Instruction Turtle Programs." Ph.D. Thesis, Massachusetts Institutes of Technology, 1973.

6. Winston, P., "Learning Structural Descriptions From Examples." Ph.D. Thesis, Massachusetts Institute of Technology, 1970.

7. Sussman, G., "A Computational Model of Skill Acquisition." AI-TR-297, MIT Artificial Intelligence Laboratory, 1973.

8. Goldstein, I., "Understanding Fixed Instruction Turtle Programs." Ph.D. Thesis, Massachusetts Institutes of Technology, 1973.

9. Freiling, M., "Functions and Frames in the Learning of Structures." Working Paper 58, MIT Artificial Intelligence Laboratory, 1974.

10. Winston, P. (Ed.), "Progress in Vision and Robotics." AI TR-281, Massachusetts Institute of Technology Artificial Intelligence Laboratory, May 1973.

11. Waltz, D., "Generating Semantic Descriptions From Drawings of Scenes With Shadows." Ph.D. Thesis, Massachusetts Institute of Technology, 1972., also ibid., pp. 81-148.

12. Freuder, E., op. cit., pp. 174-201.

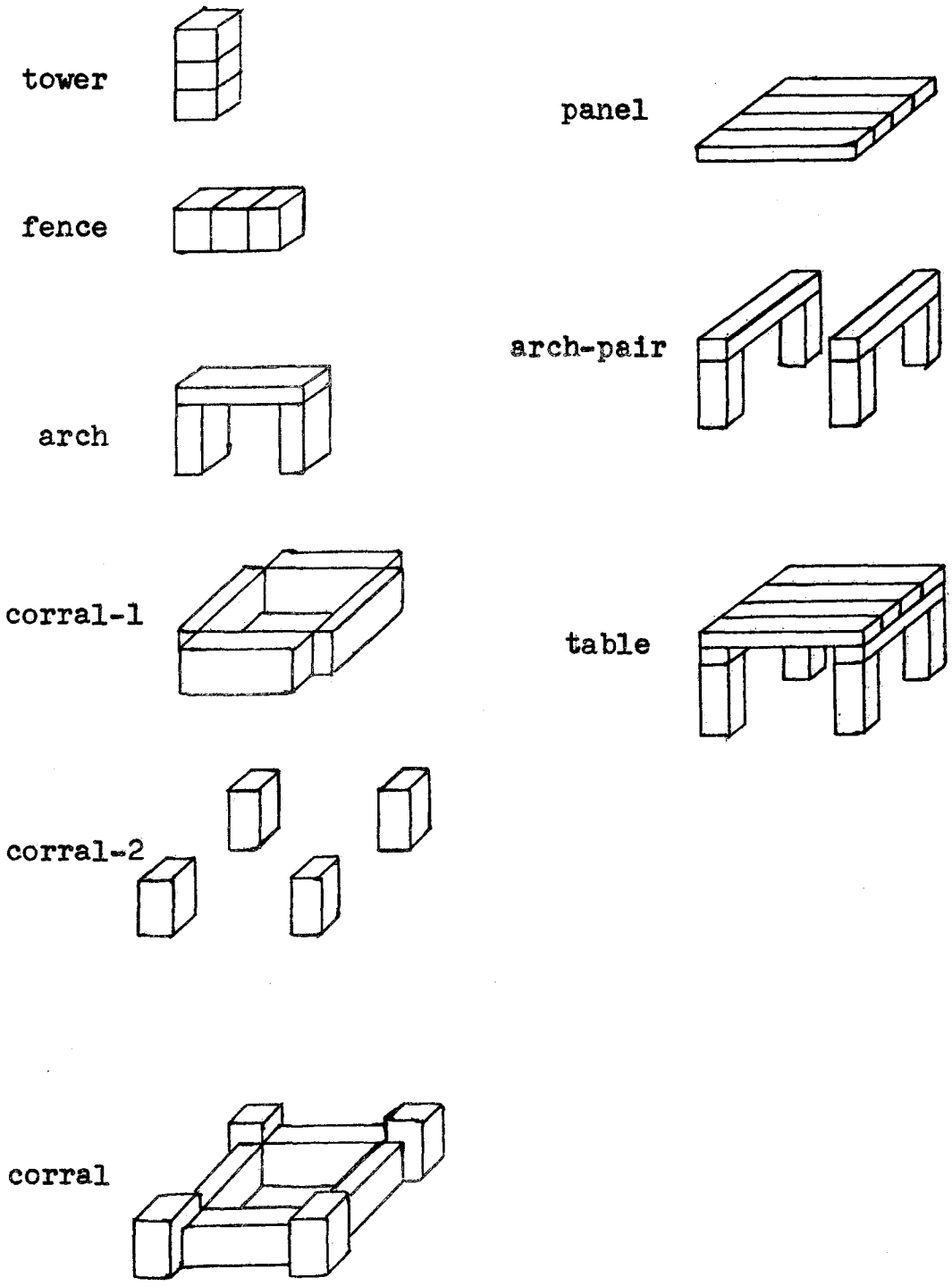


Figure 1. Simple Assemblies of Blocks

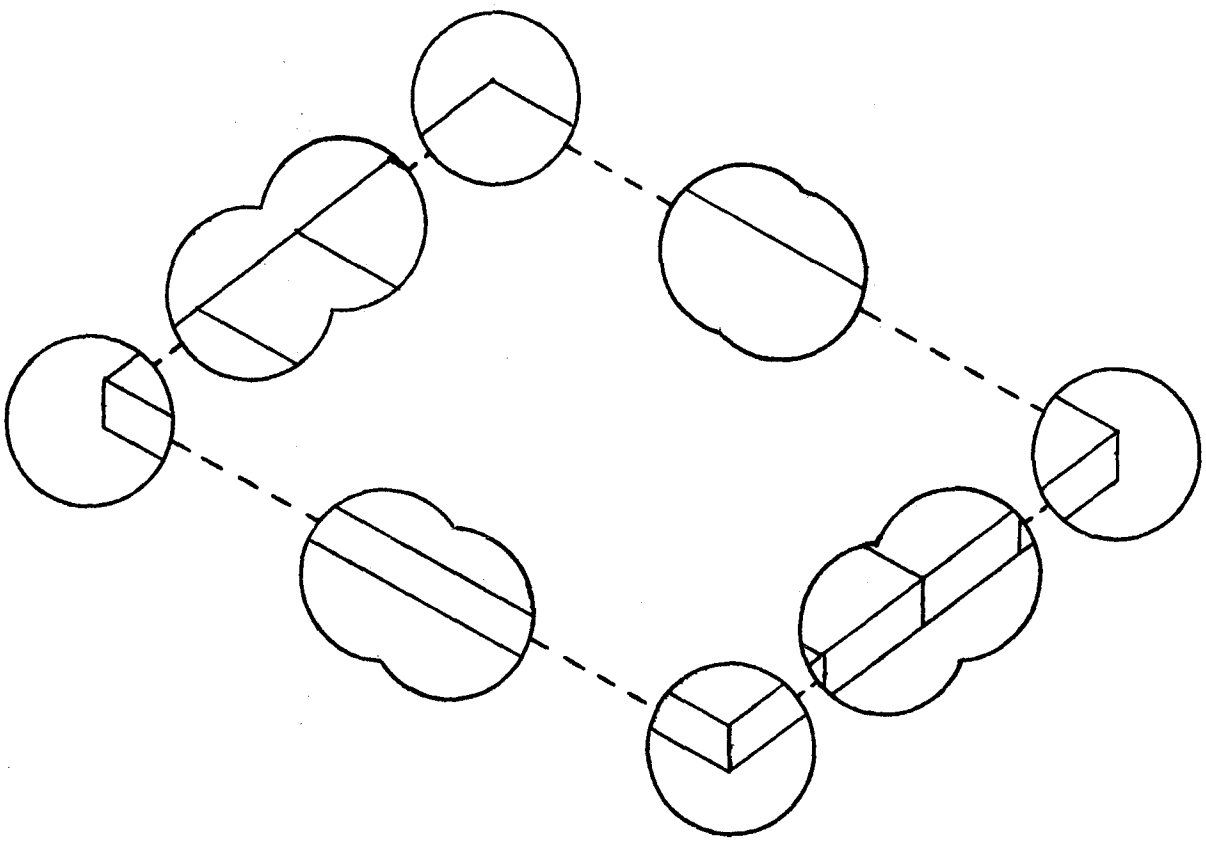


Figure 2. Global and Local Appearance of Lapped-Panel

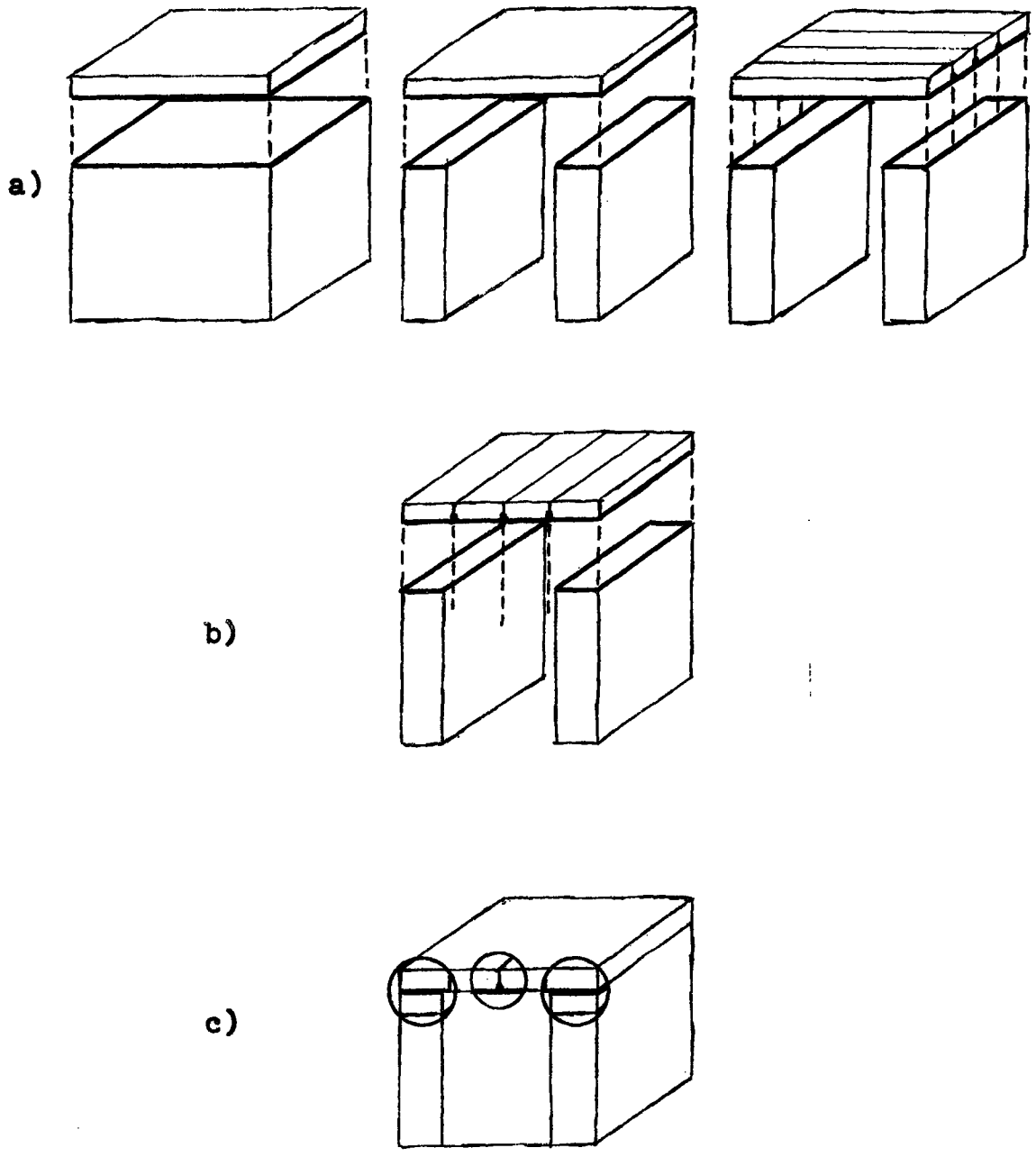
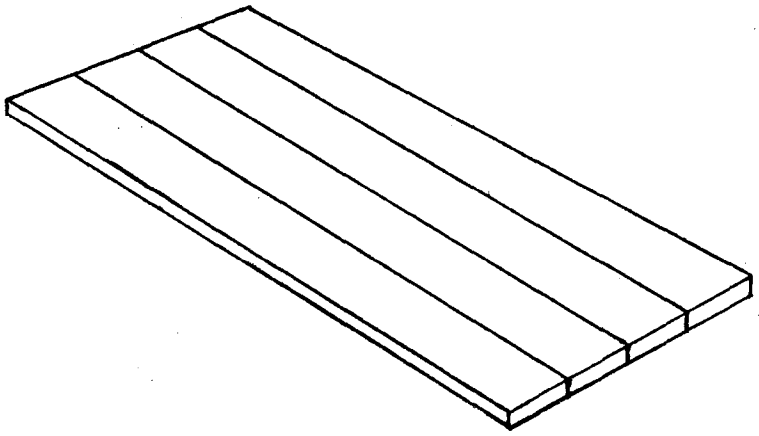


Figure 3. Successive Approximations for Verifying Stability of Tables.

a)



b)

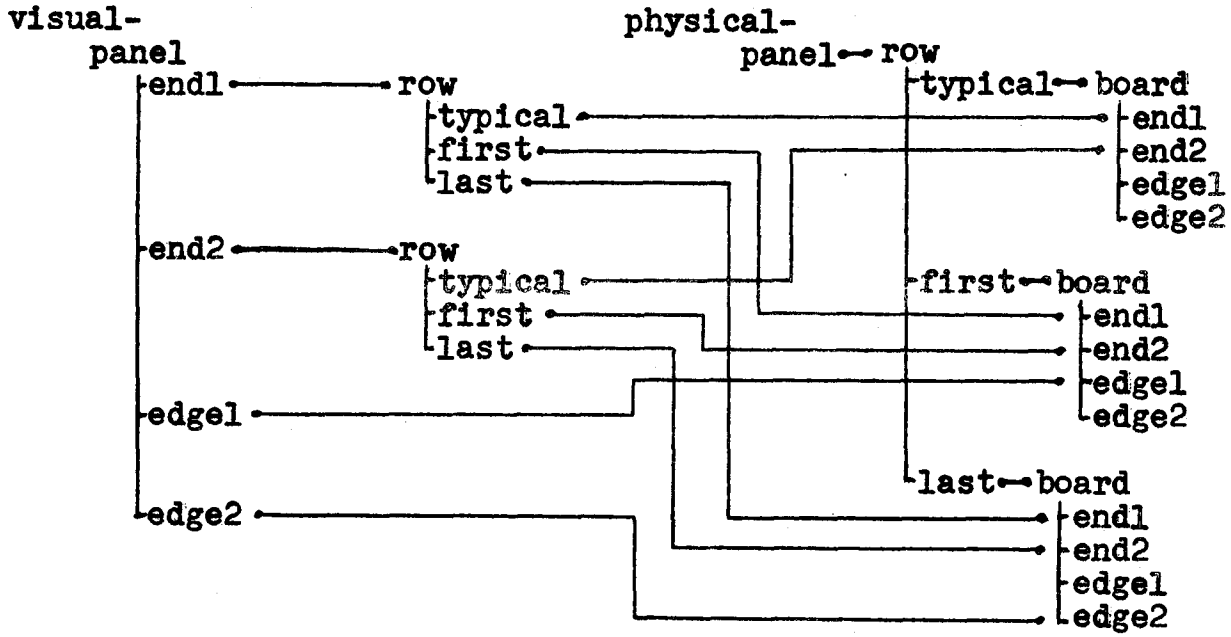
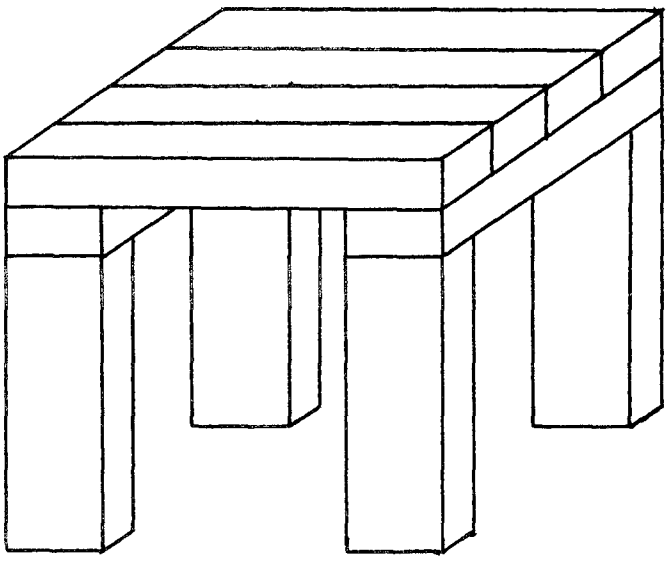


Figure 4. Interfacing of Visual and Physical Panel

a)



b)

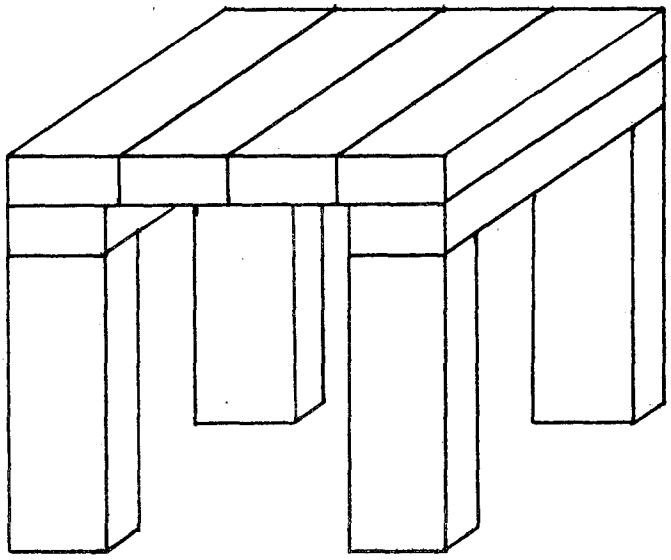


Figure 5. Visual Tables

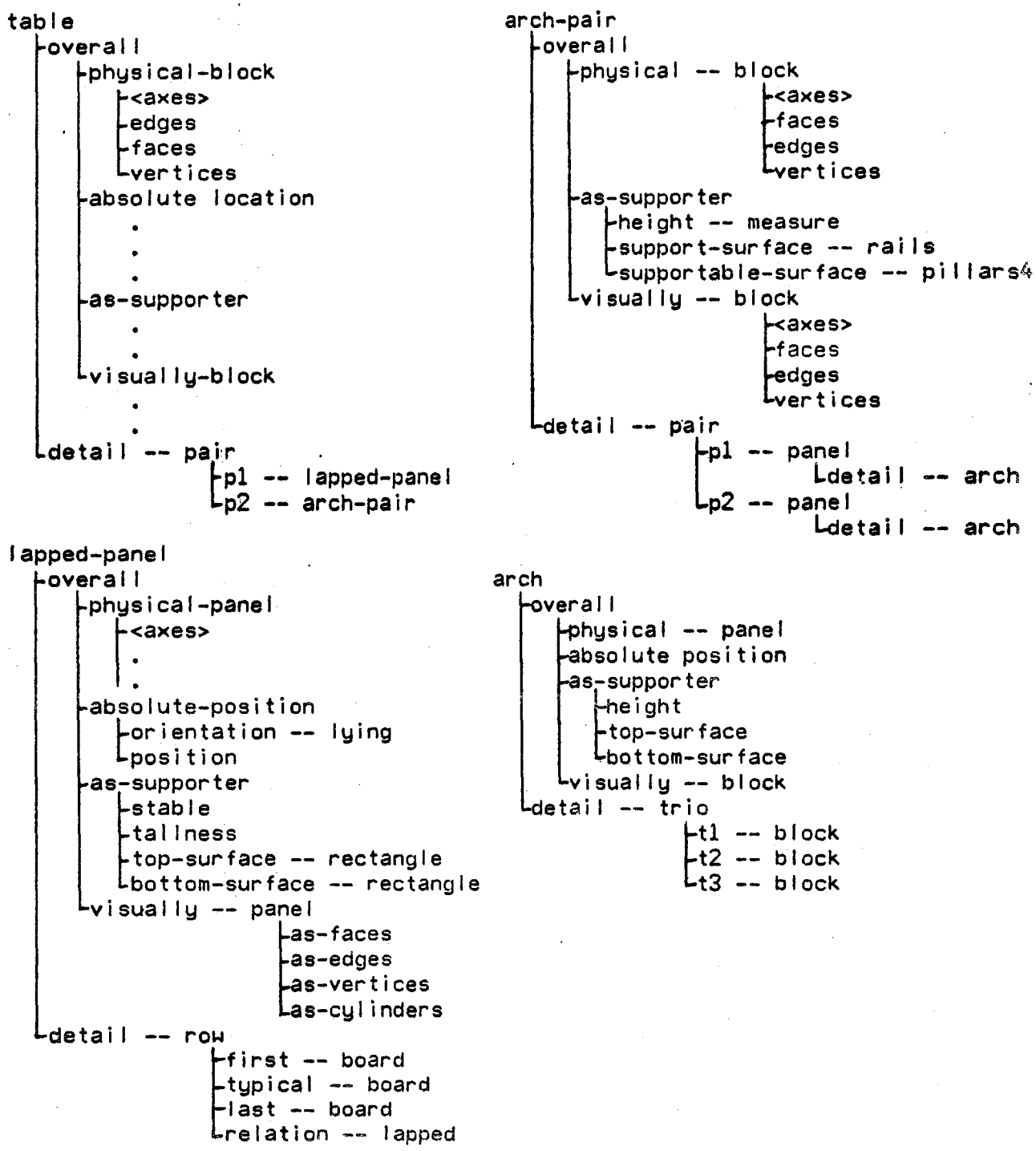


Figure 6. Description Frames for a Table
(Minus Constraint Links)

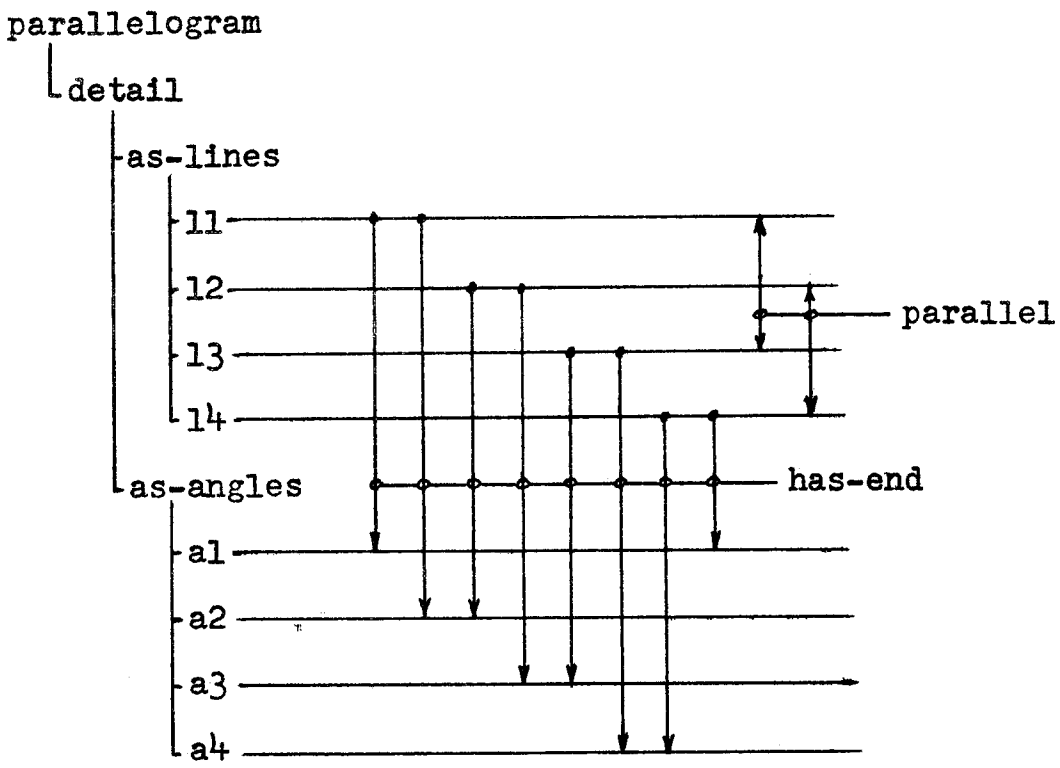


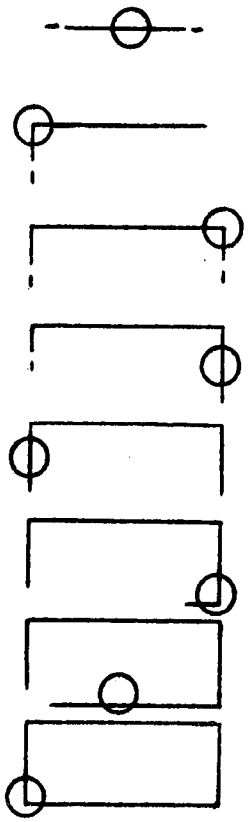
Figure 7. Parallelogram Frame With Line and Angle Subgroups

parallelogram

terminals

lines — angles

Step	l1	l2	l3	l4	a1	a2	a3	a4
1	A		.5		1	1		
2				1	A			
3		1				A		
4		A		1.5			1	
5				A				1
6			1.5				A	
7			A					2
8								A



Meaning:
At each step, a terminal is assigned - "A", and relational constraints propogate from it to other unassigned terminals,
0.5 for "parallel",
1 for "has-end".
In the next step, a most-constrained terminal is chosen for assignment.

Figure 8. Recognizing a Parallelogram

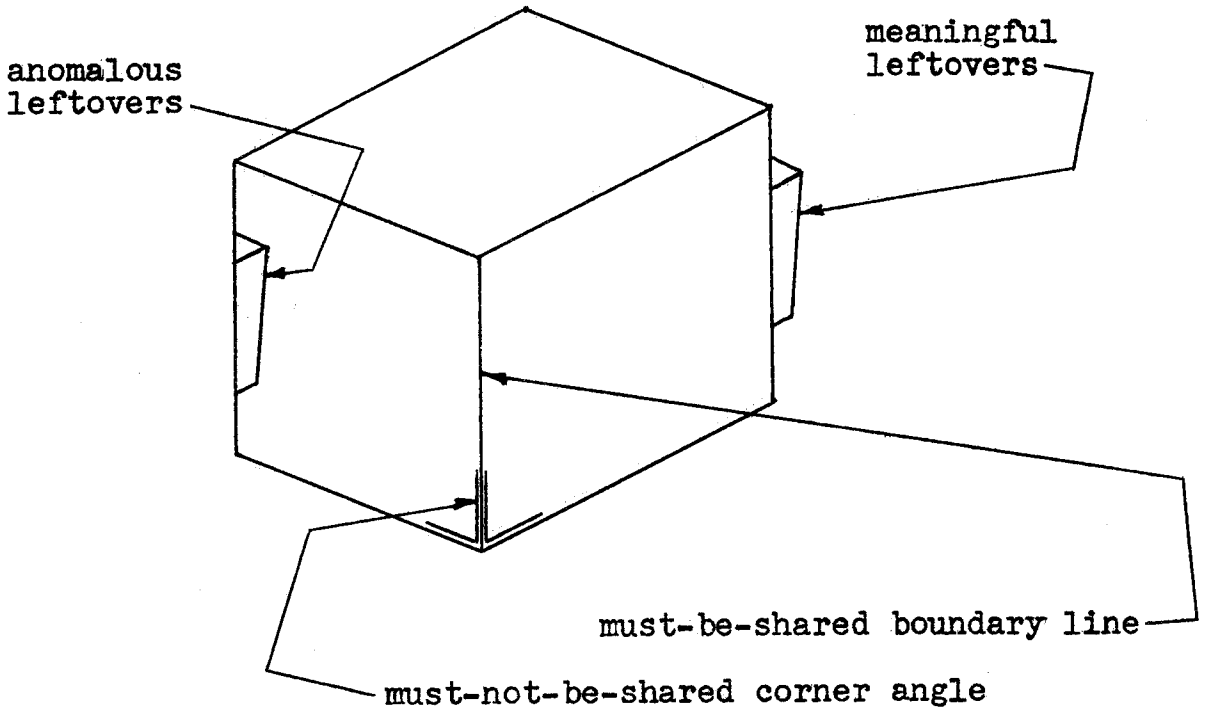


Figure 9. Imperative Sharing, Non-sharing, and Leftovers in a Polyhedral Scene.

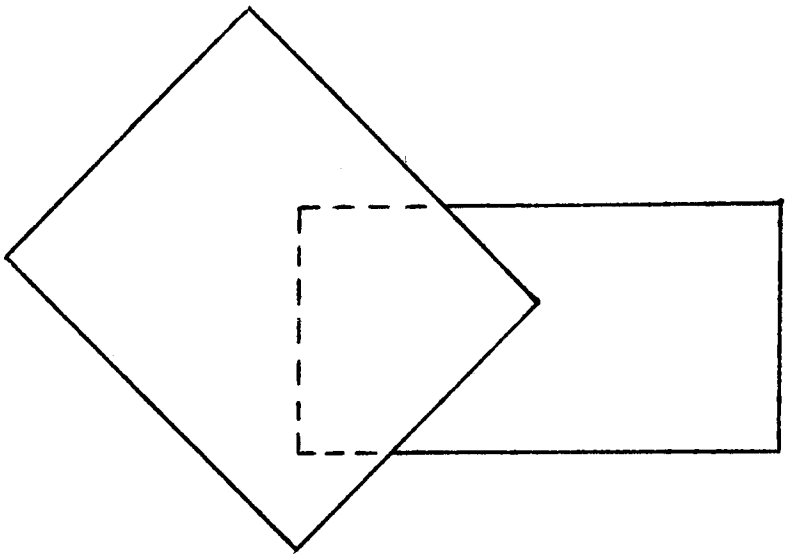


Figure 10. Occlusion Between Parallelograms

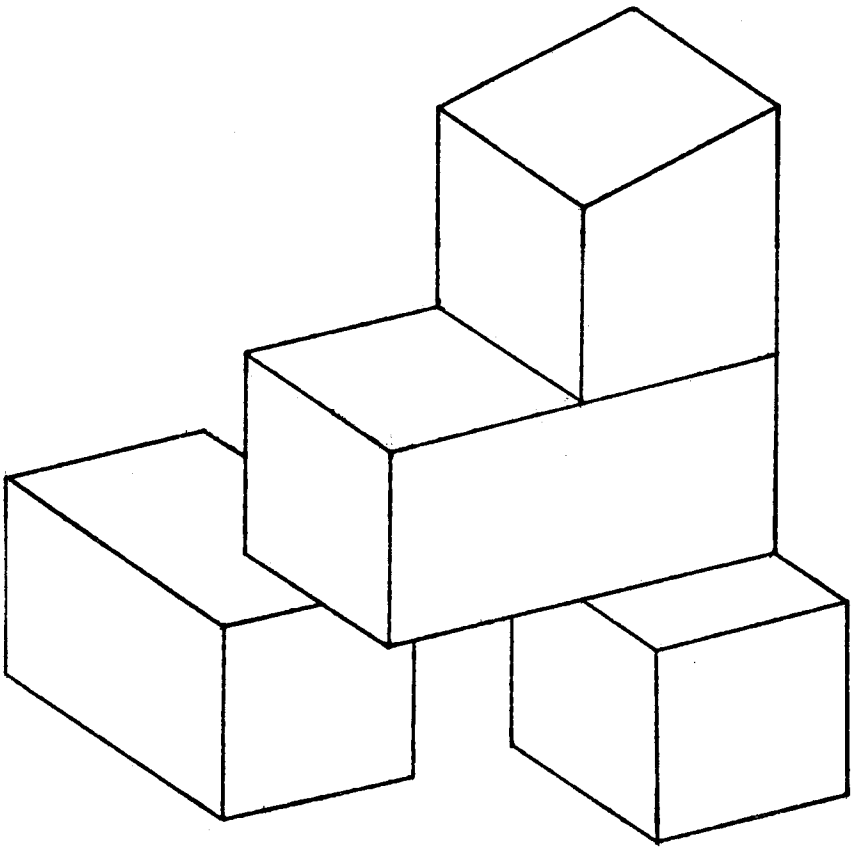


Figure 11. Typical Blocks-World Scene