WHISPER: A Computer Implementation Using Analogues in Reasoning

by

Brian V. Funt

Artificial Intelligence Laboratory
Stanford University
Stanford, California
U. S. A. 94305

Technical Report 76-9

Based on the author's theses of the same title, submitted on March 31, 1976.

December, 1976

Department of Computer Science
University of British Columbia
Vancouver, B. C.

A I
cancel ✓

ABSTRACT

The use of an analogue as an aid to a problem solving program is investigated. A working system, the advantages of the analogue it uses, the mechanisms required, and the interaction with other forms of knowledge are described.

The program, WHISPER, uses a diagram together with procedures for modifying it, as an analogue of a situation involving a stack of arbitrarily shaped rigid bodies. It determines a stack's stability and predicts the motions of any unstable object by examining the situation's diagram. The analogue is particularly valuable in detecting discontinuities in an object's motion. For example, collisions with other objects or cliffs an object might slide over can be 'seen' in the diagram rather than having to be inferred from a description of the situation.

WHISPER uses a simulated parallel processing 'retina' to look at the diagram which is encoded in a two-dimensional array. It consists of a fixed number of processors operating in parallel and communicating only with their immediate neighbours. WHISPER's retina resembles the human retina in some respects. Its resolution decreases away from its center. It can be moved to fixate on different sections of a diagram.

A set of domain independent features are extracted from WHISPER's diagrams by procedures, called perceptual primitives, which execute on the parallel processing retina. Example features are: symmetry of an object, similarity of two objects, and contacts of an object with other objects. In addition to these primitives, the retina can be used to 'visualize' the rotation of an object without having to move it directly in the diagram.

The advantages of analogues are classified in terms of two categories according to whether a correspondence exists between the behaviour of the analogue and the behaviour of the external situation, or whether a correspondence exists between the static configurations of the analogue and those of the external situation. Some reasons for the effectiveness of analogues are presented.

Table Of Contents

ACKNOWLEDGMENT

I would like to express my appreciation to all the people who have contributed their energy and insight to this work. In particular, Raymond Reiter, my thesis supervisor, has provided essential encouragement and criticism. I especially thank him for the many exciting conversations we have had in which many ideas took shape. Alan Mackworth is a very knowledgeable and constructive critic who on several occasions prevented me from re-inventing the wheel. I am also grateful to him for suggesting WHISPER's 'falling objects' domain. I thank Richard Rosenberg for his support, comments, and criticisms throughout the numerous years since he first introduced me to Artificial Intelligence. I have also benefitted from the criticism of E.W. Elcock of the University of Western Ontario. There are many other people, Gordon McCalla, Peter Rowat, Mike Kuttner, Greg Shannan, Jim Davidson, and Bill Havens, who have been both encouraging and appropriately skeptical. I thank Karen for reading, commenting on, and typing the thesis drafts, and most importantly for being herself.

The financial support of the National Research Council of Canada grant A7642 is gratefully acknowledged.

Chapter I: Introduction

I-1 Analogues In A Problem Solving System

Conceptually simple problems should be answered with conceptually simple solutions. These rarely are obtained by Artificial Intelligence systems; paradoxically, the field's methods are more successfully applied to difficult problems than to ones children can solve.[1] One major advantage that children have in comparison to problem solving systems is their sensory access to the external world. They benefit from experimenting with the environment. It is easier to observe the effects of a change, than to infer them from a description of the environment and a knowledge of its physical laws. Similarly, when direct interaction with a situation is impossible, it is easier to predict the outcome of a proposed change by observing the outcome of an analogous change made to an analogous situation. Diagrams, maps, scale models, and computer simulations are analogues which people routinely use as an aid in reasoning. This paper explores analogues: how they are incorporated into a problem solving system, the way the entire system is thereby simplified, and the solutions they enable the system to discover.

Problem solving progresses simultaneously on several levels. Polya[2] has identified four which he terms: the 'heuristic level', the 'mathematical level', the 'relational level', and the 'image level'. The first three can be related to current Artificial Intelligence approaches: (i) The 'heuristic level' corresponds to the goal-oriented approach. At each stage in the search for a solution the aim is to accomplish a relevant goal or sub-goal. (ii) The 'mathematical level' corresponds to the currently invoked equation, assertion, or procedure. (iii) The 'relational level' corresponds to the complete tree of the search space, the branches which have already been explored, and the branch currently being investigated. Thus, three of these levels have counterparts in problem solving systems. It is the 'image level' which has thus far been ignored. "On the uppermost level, the image level, we see the evolution of the investigated geometric figure in the problem solver's mind. At each stage, the problem solver has a mental picture of the geometric figure he explores, but this picture changes in transition to the next stage; some details may recede into the background, other details come to our attention, new details are added."[3] If a diagram is admitted as well as a 'mental picture', then analogues correspond to this level. The major questions are: Why and in what ways is the image level useful? How is it used? What mechanisms are required to make use of it? How does it interact with the other levels?

WHISPER, a computer program, demonstrates the advantage and feasibility of using analogues in reasoning. It makes hypotheses and draws conclusions based on the state of the diagram. There is continual interaction between WHISPER's knowledge of the problem domain and the diagram as the solution progresses. WHISPER can 'look' at the diagram, making changes and modifications to it as the action unfolds.

WHISPER's task is to determine the stability of a stack of objects, and to predict what happens if it is unstable. Figure I-1 depicts a typical configuration of objects. WHISPER, using a diagram of this situation, determines that object B 'hangs over too far', and will fall. It then envisions B's toppling motion and foresees its collision with D. The diagram is then updated reflecting the resulting situation (figure I-2). This is the first in a sequence of 'snapshots', each portraying a new event in the collapse of the original structure. WHISPER sees from the diagram of this new situation that B upsets the balance of D on C, envisions the rotation of D until it hits the table, and creates a new diagram (corresponding to the situation of figure I-3). The causal connection between B and D is found through the diagram, not though logical inference about the shapes, positions, or objects' loci of motion. With nothing to support B, it continues falling until it hits D again (figure I-4). The three 'snapshot' diagrams (figures I-2 through I-4) constitute WHISPER's description of the solution.

The overall structure and organization of the WHISPER system is shown in figure I-5; its essential components are:

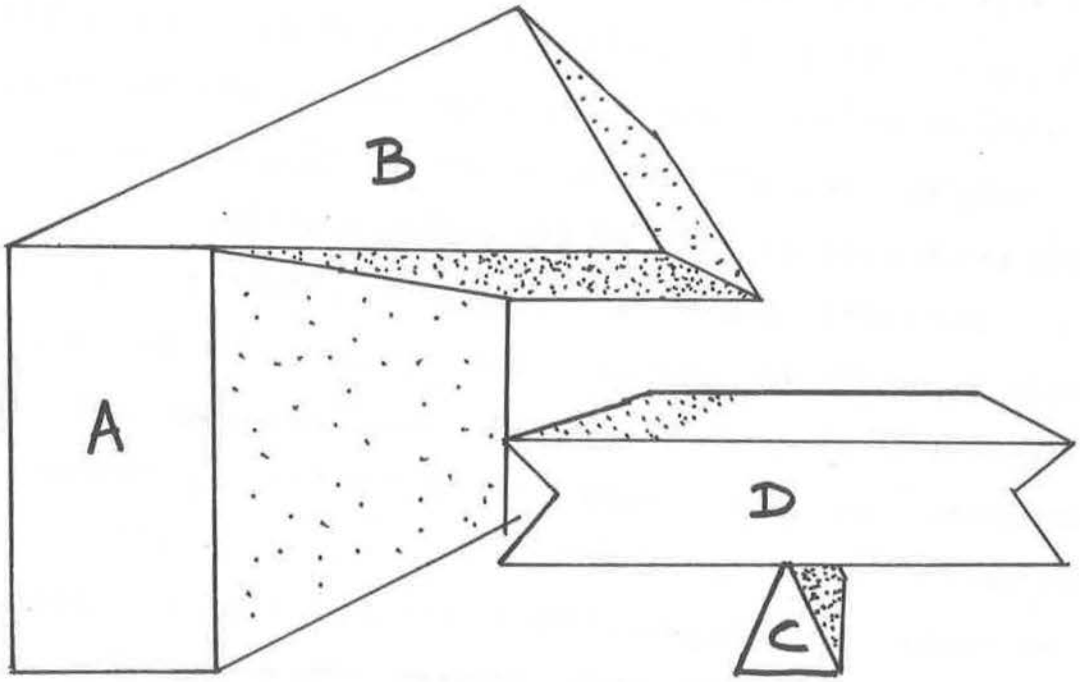
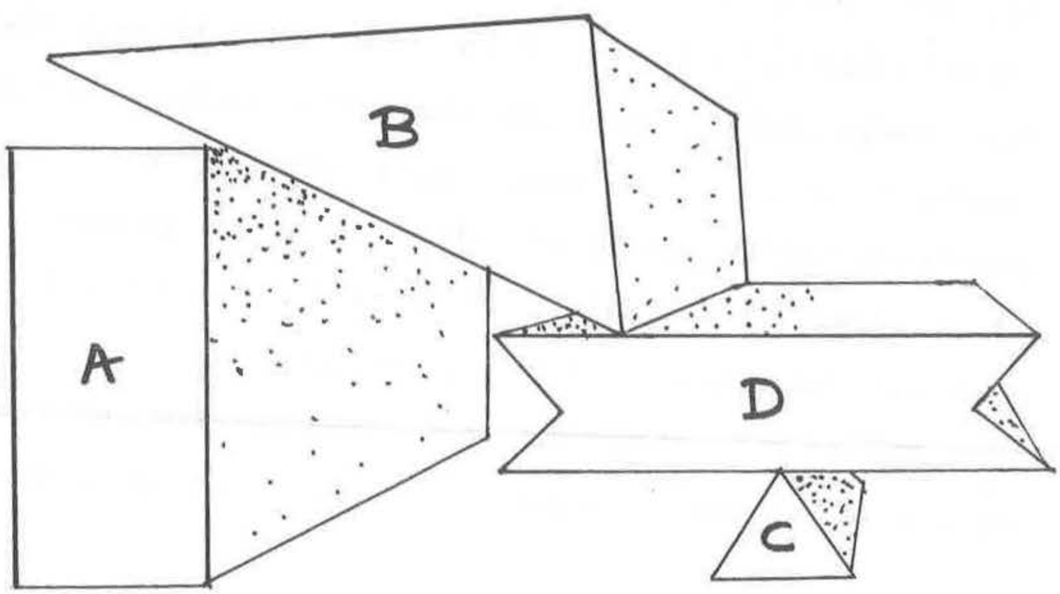


FIGURE I-1

FIGURE I-2



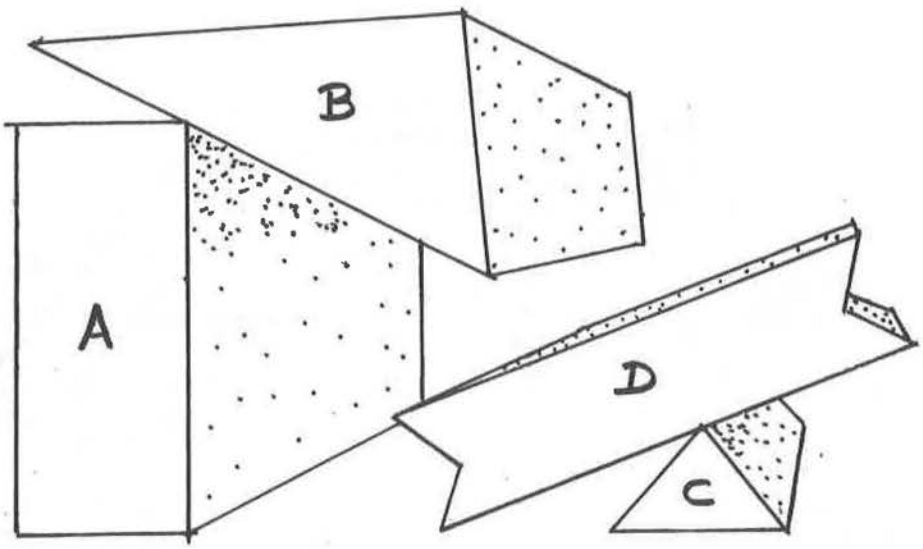
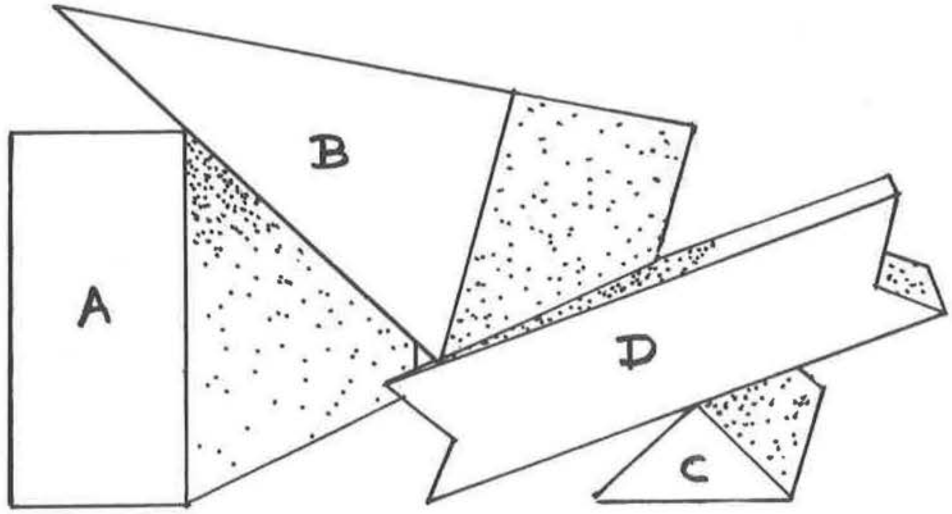


FIGURE I-3

FIGURE I-4



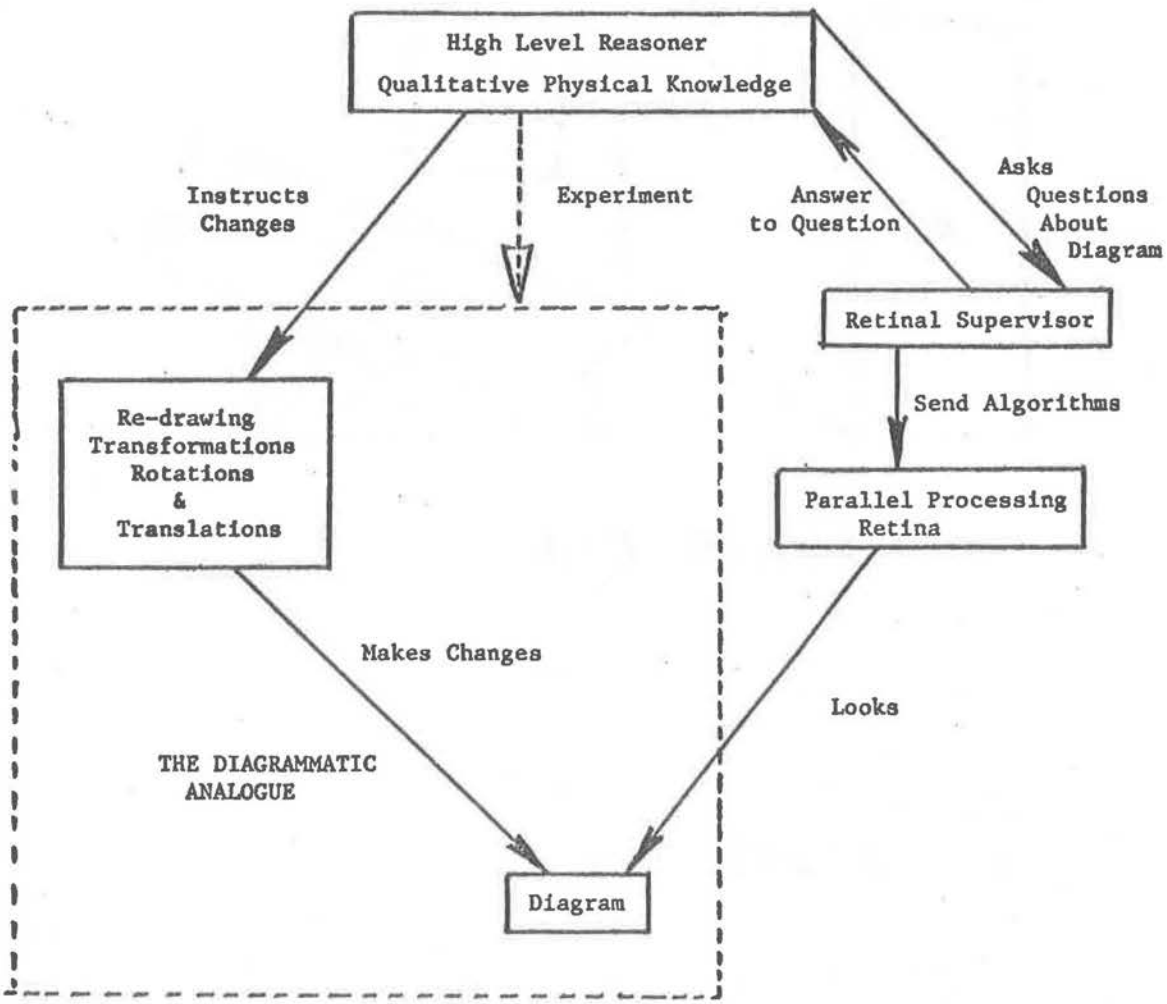

High Level Reasoner
Qualitative Physical Knowledge
Instructs Changes
Experiment
Answer to Question
Asks Questions About Diagram
Retinal Supervisor
Send Algorithms
Parallel Processing Retina
Re-drawing Transformations Rotations & Translations
Makes Changes
Looks
THE DIAGRAMMATIC ANALOGUE
Diagram

FIGURE I-5

the qualitative physical knowledge, the retina, the redrawing transformation procedures, and the diagram. The qualitative physical knowledge is the domain dependent part of the system, consisting of 'specialist' procedures expressing elements of the behaviour of rigid bodies when acted upon by gravity. The retina is a specially structured parallel processor which 'looks' at the diagrams. It follows instructions from the qualitative physical knowledge 'specialists'. Changes are made to the diagram by the redrawing transformations. They also are under the command of the qualitative knowledge specialists. The diagram functions as the system's chief representation of the problem situation. Together the diagram and redrawing transformations which modify it are an analogue (dotted box) of WHISPER's problem situations. The interaction with the analogue is by experimentation.

Knowledge of physics is represented procedurally, each specialist encapsulating a qualitative piece of knowledge such as: 'If the center of gravity of an object does not have supports to both its left and right, then it hangs over too far', or 'If an object hangs over too far, then it will topple, rotating about the nearest support point to the center of gravity'. The qualitative physical knowledge is the top level of the WHISPER system. In contrast to Fahlman's[4] BUILD system, WHISPER's understanding of Physics is closer to a child's than an engineer's.

When a 'specialist' requires information about the state of the world in deciding the applicability of its knowledge to the current situation, it sends a request to the retina to examine the diagram for the presence of a specific feature. The 'specialist' interprets the feature relative to the current domain. For example, a 'specialist' which needs to know if object X supports object Y, asks the retina to see if Y is above X and Y touches X in the diagram. If the qualitative knowledge discovers that a change of state, an action, will occur in the world, then it calls the redrawing transformation to modify the diagram to reflect the effects of this action.

The purpose of the retina is to extract information from the diagram in response to queries from the qualitative physical knowledge specialists. Its role parallels the human eye and its early perceptual processing stages. The retina is basically a parallel processor, and algorithms, called perceptual primitives, have been designed to execute on it. Due to parallelism, their execution times are of the same order of magnitude as more conventional operations. Each perceptual primitive determines whether a particular feature exists in the diagram as seen from the current location of the retina.

The diagram the retina 'looks' at is the pattern formed by values in a two-dimensional array. The combination of WHISPER's retina and array diagrams parallels human use of diagrams represented on paper, not human visual imagery. Paper is simulated by the array. The diagram of the scene of figure I-1 which WHISPER uses is shown in figure II-1. A problem is stated to WHISPER as a diagram of this type. They are constructed so that objects' shapes and positions are represented by corresponding shapes and positions in the diagram. The diagram allows WHISPER to work with both convex and concave irregularly shaped objects without added difficulty. For easy recognition, each object is shaded a different colour, and contours of objects are shaded a colour related to the colour of their interiors.

The combination of the diagram and transformations applied to it is an analogue of a situation involving a stack of physical objects. An analogy exists both between the static states of the diagram and the static states of the physical situation, and between the dynamic behaviour of objects in the diagram and the behaviour of objects in the world. Clearly, the behaviour in the diagram and behaviour in the world are not identical. Objects in the diagram do not automatically begin to move as do objects in the real world. However, many aspects of an object's dynamic behaviour are properly portrayed when it moves in the diagram. If an object moving in the diagram collides with another object, then a collision will also occur in the world. If a path is clear in the diagram, then it is also clear in the world. Moving an object also causes its support and contact relationships to change. The modified diagram automatically reflects these changed relationships. The side effects of an action can be simply observed in the diagram. This results from the representation of spatial relationships in the world by analogous spatial relationships in the diagram, and the representation of action in the world by analogous action in the diagram.

I-2 The Internal/External Question

To the machine, there is no sharp distinction between a diagram represented externally on a piece of paper and a diagram represented internally as a two-dimensional array. Such a distinction is dependent upon a central question: Where does the computer end and the rest of the world begin? What is external to the machine and what is internal? To understand that there is no straightforward answer, consider the example of a movable head disk drive. It is generally considered that the information stored on the disk is internal to the machine. Is this information any more internal than the marks on a piece of paper to the human brain when it is scanned by the human eye?

Portability has been the primary consideration in deciding what is and what is not part of the computational structure of a machine or of ourselves as human beings. Roughly, an entity's portable computational structure is the minimal part of it which must be transported in order that it compute the same results at a new location. For a human the portable computational structure consists of his/her body. Whether this is the minimal computational structure is another question; certainly we cannot think without a brain and enough bodily structure to support it. A thought process dependent upon counting one's fingers should not be ruled out as invalid; peoples' fingers are part of their portable structure. The computational structure of a computer consists of a processor, memory, and perhaps an input/output mechanism. In this framework the disk drive is simply considered to be a form of memory, and an eye to be an input device. Classifying the computational structure of a computer in this way is possibly too narrow and confining.

Space, time and mass may also legitimately be considered as part of the portable computational structure of a machine, because they are omnipresent. Wherever the machine is moved they will be present, so in a sense they are an integral part of any machine. To what extent can space, time and mass be exploited computationally? Many of the advantages of using analogues derive from using space, time and mass directly rather than attempting to model them with symbolic descriptions. In particular, with reference to diagrammatic analogues, there is no need to model two-dimensional space when it can rightfully be considered to be a part of the machine itself. The only problem is to have a device with which to look at and access this space; this is the function of the eye. By adding an eye as an extra piece of hardware, the 'hardware' of space itself becomes available as a medium for representing and manipulating information. This does not obviate the need for some other representation of spatial information; it does eliminate the need for a model of space itself. An example might be in the use of a map to plan a route from one location to another. If a map of an area that a person knew well were not available then he likely could construct one from memory. It seems unlikely that he would have a copy of the map in his memory which he then redraws on a piece of paper, but rather that he constructs it from a set of assertions describing the relevant spatial relationships. The content and structuring of this information does not matter for our current purposes. The important thing is that he can construct at least a rough approximation to a proper map. The two-dimensional topolcgical structure of the paper provides a context in which the facts in his memory are to be interpreted. Rather than having to have a model of two-dimensional space he can use the already available space of the paper. A great many more assertions about spatial relationships can be extracted from the map than were used in constructing it because of the context provided by the paper. This and other advantages of such a re-representation are part of what WHISPER is intended to demonstrate, and they will be discussed in more detail as they arise. For the moment, the point is that _in order to use diagrams it is not necessary to store images of them in memory;_ therefore, if a gain can be made by re-representing the spatial information stored in memory in diagrammatic form, then this might as well be done since two-dimensional space can be considered as a part of the hardware of the machine.

The two-dimensional structure of an array is provided computationally. It is a function of conventions for accessing a one-dimensicnal structure, namely linearly ordered computer memory. Since there is no sharp distinction to be made between diagrams stored in arrays and on paper, WHISPER's use of diagrams can be considered analogous to human use of diagrams. That the diagrams are modeled internally is purely a convenience in that it was easier to provide a software simulation of the eye and paper combination than to provide the actual hardware. The array WHISPER uses is not to be interpreted as a model for human visual imagery. A proposal for using WHISPER's parallel processing eye without an array, and its relationship to imagery is presented in Chapter V.

Chapter II: WHISPER: A System Employing Analogues In Reasoning

WHISPER is a working program. It serves as an instantiation of the general ideas discussed in subsequent sections, and establishes the utility and feasibility of incorporating analogues into problem solving systems. WHISPER is not a study in the specialized domain dependent heuristics pertaining to a particular class of problem. Many of the mechanisms required in interpreting and modifying analogues in WHISPER's domain will also be required when analogues are utilized in systems reasoning on other domains.

WHISPER's reasoning is entirely qualitative in nature. I believe that it is necessary to obtain qualitative solutions to problems before attempting quantitative or precise solutions. A qualitative solution provides a framework on which planning for a quantitative solution can be based. DeKleer[5] has investigated some ways in which this can be accomplished. Analogues are particularly important in reducing the conceptual complexity involved in obtaining qualitative solutions. The effectiveness of analogues in curbing complexity is evidenced by the conceptual simplicity of the qualitative knowledge of Physics which WHISPER employs in solving its problems.

II-1 The Problem Domain

Given a stack of physical objects, WHISPER establishes its stability or instability, and the resulting sequence of events if it is unstable. A typical example of the configurations that WHISPER can handle is shown in figure II-1. The usual assumptions about 'ideal' environments common to introductory Physics texts have been made. The objects are perfectly rigid, of uniform density and thickness, and have frictionless surfaces. They are otherwise of arbitrary shape, not restricted to cubes, wedges, or other simple polyhedra. One further restriction is that the faces of the objects must be aligned. Although this gives the problems a basically two-dimensional character, it is a well precedented and frequently unstated assumption, prevalent in Physics texts and other Artificial Intelligence systems. In particular, although all the problems handled by Fahlman's BUILD system are sketched as 2-D projections of three-dimensional scenes, they all conform to these restrictions and have the same two-dimensionality about them.

Problems are input to the system as an array encoding of a two-dimensional cross-sectional view of the scene. The array can be generated by drawing with a lightpen at a graphics terminal. Nonetheless, it is the array, not the light pen coordinates, that forms the final input to the system.

This class of problem was chosen because it provides a

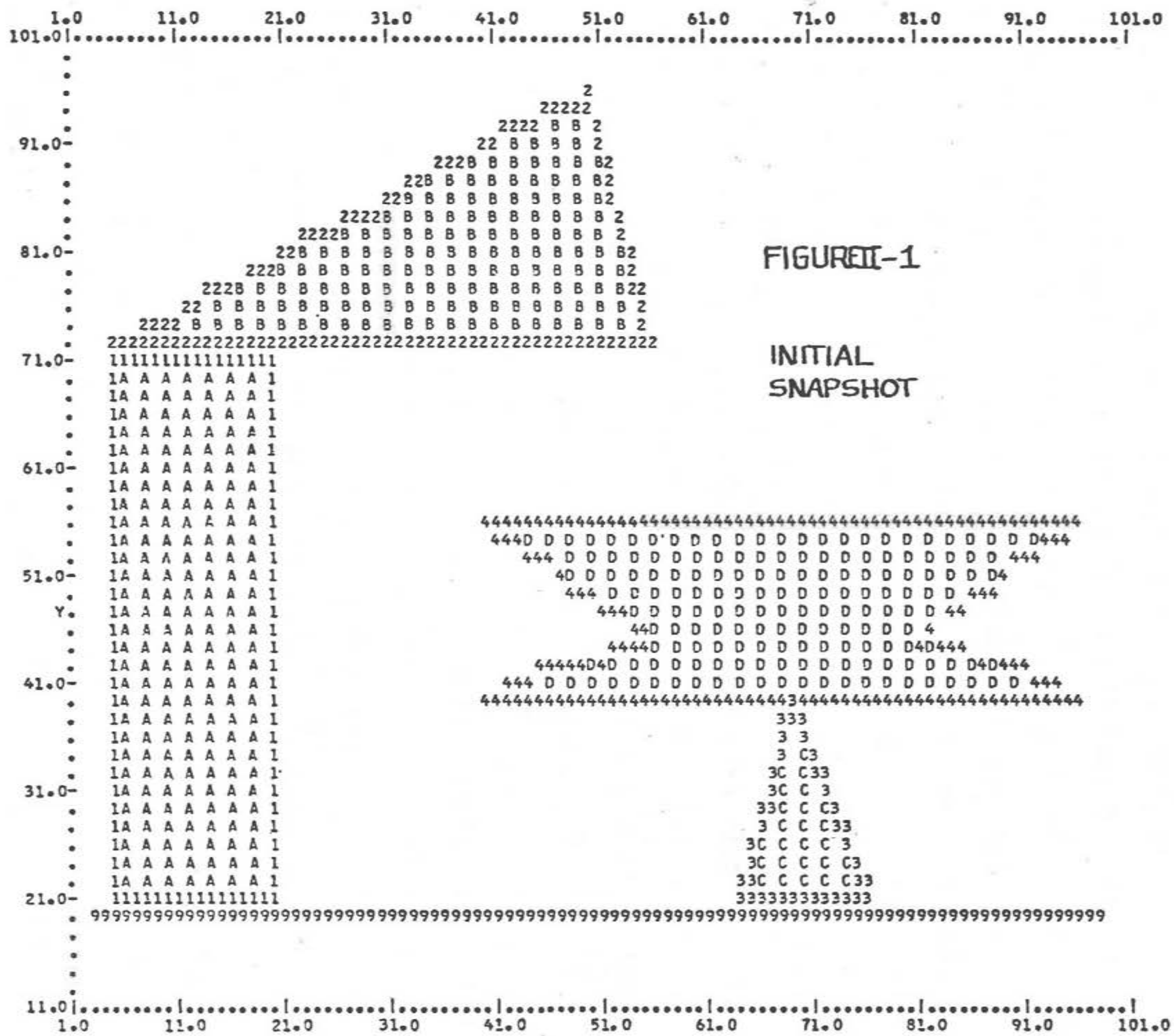


FIGURE II-1

INITIAL SNAPSHOT

non-trivial domain in which to demonstrate many of the thoughts and ideas I had about the utilization of analogues in reasoning and the advantages which a system incorporating analogues would derive from them.[6] These ideas will be discussed as they arise in the description of WHISPER in this chapter, and in more general terms in the following chapter. Problems in this domain are of interest because they involve action and the discovery of causal chains of events. They are everyday, real world problems which people learn to solve at an early age, in contrast to highly intellectual and formal domains such as chess or Mathematics. Surprisingly, problems of this sort have eluded satisfactory solution by other methods, one of the main reasons being the presence of the 'frame' problem (discussed in section II-7.1), another being the lack of an adequate method of representing and manipulating spatial relationships. The physics is simple enough so that in implementing a system one is unlikely to become distracted from the main question at hand - the utilization of analogues in reasoning - and bogged down in a study of irrelevant aspects of the problem domain. Another feature of this domain is that diagrams provide an obvious and commonly used analogue of 'blocks' world situations. These problems also provide an opportunity to study the type of interaction which must take place between propositional knowledge of qualitative aspects of Physics and the analogue. The analogue is repeatedly examined to draw first conclusions, modified to reflect the ramifications of these conclusions, and re-examined to draw further conclusions.

II-2 System Overview

I will attempt to put the whole WHISPER system and the question of analogues in perspective before going into greater detail. The significant feature of WHISPER is that it uses a diagrammatic analogue of the situations it reasons about in addition to a descriptive representation of these situations, such as that which could be provided by a set of assertions, a set of procedures, or a network. It relies on the analogy between diagrams of these situations and the situations themselves, and manipulates the analogue during the problem solving process.

In the diagrams which WHISPER uses there are some simple and well-defined correspondences or similarities between the topological structure of the configurations in the diagram and those in the problem domain. Shapes and positions of the configurations in the diagrams are analogous to the shapes and positions of the objects in the real world: the contours in the diagram are identical (except for scaling) to the shapes of the objects (viewed head-on); and the positions relative to one another of the shapes in the diagram and the objects in the world is the same. Of course it is possible to create non-analogical diagrams, ones for which there is no simple correspondence between the configurations of marks in the diagram and the external reality, but these would be of little value.

There is also a correspondence between the changes which occur in the real world and the changes which WHISPER makes to its diagrams. Since the objects in the problem environment are rigid bodies, only linear transformations are applied. It is because of these correspondences between both the static configurations of the diagram and the static physical situations, and between the dynamic actions occurring in these situations, that the combination of the diagram and the procedures which modify it together constitute an analogue of real world situations involving stacks of physical objects. To distinguish this analogue from different types of analogues of other real world situations (e.g. a scale model airplane in a wind tunnel) it will be termed a <u>diagrammatic analogue.</u>

The medium in which the marks of the diagram are stored is that of values in a two-dimensional array (presently 101 x 101). A more common medium is, of course, pencil marks on a piece of paper. Each object has a unique array value for points on its contour and a related value for points in its interior.

For a problem solving system to make effective use of diagrammatic analogues it must have a method of examining and understanding them and a method of altering their configurations of marks. Human problem solvers use their eyes for the examination of diagrammatic analogues. WHISPER has been endowed with an 'eye' also. This 'eye' is a software simulation of some of the dominant features of the human eye.

The simulated eye looks at simulated paper, namely a two-dimensional array.

The software retina has some of the basic characteristics of the human retina. It is movable and can fixate anywhere in the diagram; the acuity varies across it with the center having the highest resolution and the periphery having the lowest; it is composed of many 'receptors' which operate in parallel; and communication between 'receptors' is constrained to message passing between neighbours. These features provide a new framework, partly a data structure and partly a computational structure, in which primitive perceptual operations can be expressed and implemented. Although I will use the terms 'eye' and 'retina' when discussing this framework, there is no direct correspondence between it and any particular physical part of the human eye. The analogy holds only with respect to the gross organization of some of the preliminary processing stages of the human perceptual system.

Obviously, the problem of extracting information from diagrams is related to the questions of visual perception. However, WHISPER's perception of diagrams is simpler than that of human perception of real world scenes because objects in the diagrams are 'colour' coded, and because the objects are portrayed in draftsman's two-dimensional views. WHISPER relies on a number of primitive percepts which are provided by routines relying on the organization of the software retina and its parallel computational capabilities. Recognition of symmetries, similarities, scalings, rotations, and contact points between objects are some of the primitive percepts which are implemented. WHISPER's use of diagrammatic analogues demonstrates that it is not necessary to solve all the problems of visual perception before using a perceptually oriented system. There are some perceptual operations which are both useful to a system such as WHISPER and primitive enough so as not to require a more sophisticated understanding of the world than that required to solve the problem at hand.

There must be a mechanism whereby changes can be made to diagrams to reflect the changed position of objects in the real world. The only transformations which need be considered in the current domain are those of rigid translation and rotation. Of course, other non-linear transformations would be necessary in other domains containing non-rigid entities. There is a correspondence between the transformations which are made to objects in the diagrams and those which occur for rigid objects in the real world. Change in the world is represented by analogous change in the diagram. Rotating or translating an object in the diagram is a simple matter of redrawing the object at its new location by computing the new coordinates of every point in the object, and blanking out the original location.

Transformations can also be 'visualized' on the retina rather than being carried out directly in the diagram. The software retina is endowed with a one-level memory with which it can hold the pattern imposed on it by the object to be moved. The 'image' of that object can then be temporarily translated by simply fixating the retina at a new location and superimposing the stored pattern on the new pattern created on the retina. Similarly, the pattern of an object can be rotated on the retina and re-imposed on the input of the current fixation. This type of tentative transformation is very useful in determining the likely effects resulting from the motion of an object, and in estimating the appropriate parameters to pass to the redrawing transformations just discussed.

WHISPER procedurally represents its qualitative knowledge of physics. This knowledge is qualitative in that WHISPER reasons in terms like: 'If a block is hanging over too far it will topple' and 'If a block is on a slant then it will slide', rather than in terms of moments of inertia and vector components of forces. The qualitative knowledge is intended to reflect what a 'naive' person would use in solving these problems.

To solve a stability problem, the qualitative knowledge procedures direct the eye to focus on various parts of the diagram to extract information required for a decision on the stability of the objects. These procedures question the eye about the features it sees in the diagrammatic analogue and assigns meanings to the primitive percepts it provides. A typical question might be 'Where does object x touch other objects?'

If the situation depicted by the diagram is stable then the problem is solved. If any object is found to be unstable, then the eye is questioned further to establish what the object's motion will be. Whatever the motion - sliding, falling, or toppling - it will not continue indefinitely. WHISPER uses a retinal 'visualization' process (to be described in detail in section III-3.4) to perform a very rough simulation of an object's motion while watching for a collision discontinuity to arise.

Once the type of motion and its discontinuity points are known, then a change can be made to the diagrammatic analogue to reflect the state resulting from the completion of the motion. On a piece of paper this change is made by erasing the marks representing the moving object and redrawing them at the new location; the array equivalent involves the application of a translation or rotation transformation. The diagrammatic analogue, now in a new state, is ready for further consideration almost as if it were an original starting state. Much of the information extracted from the original diagram by directing the eye is now out of date and of little use, but the analogue is in a consistent state and can be freshly re-examined in response to questions posed by the qualitative knowledge procedures. _Determining the discontinuity point of an object's motion, and mapping easily from one state to another are two of the principal benefits which WHISPER reaps_

<u>from a diagrammatic analogue.</u>

II-3 WHISPER's Qualitative Physical Knowledge

The qualitative knowledge divides naturally into two parts. One concerns the stability of objects, the other the motions of objects as they fall. At the top level (figure II-2) WHISPER loops between stability testing and moving unstable objects. The stability test considers all the objects in the structure, and notes all the instabilities it finds in an associative data base. When it is complete, the dominant instability is determined, and the objects affected by it are moved. Only the effect of this one dominant instability is dealt with at this time. WHISPER then outputs the updated diagram as its first solution 'snapshot', and passes it back to the stability tester.

The description of the system will be approached in a top-down fashion and will center on some of the solutions obtained by WHISPER. The first problem to be considered is the 'chain reaction' problem of figure II-1.

II-3.1 Stability Testing

The central idea in the stability test is to divide the initial structure into smaller sub-structures which are tested for stability as if they were single objects. The sub-structures are either individual objects, or conglomerates consisting of two or more objects glued together. The stability test is therefore a two part process - subdivision

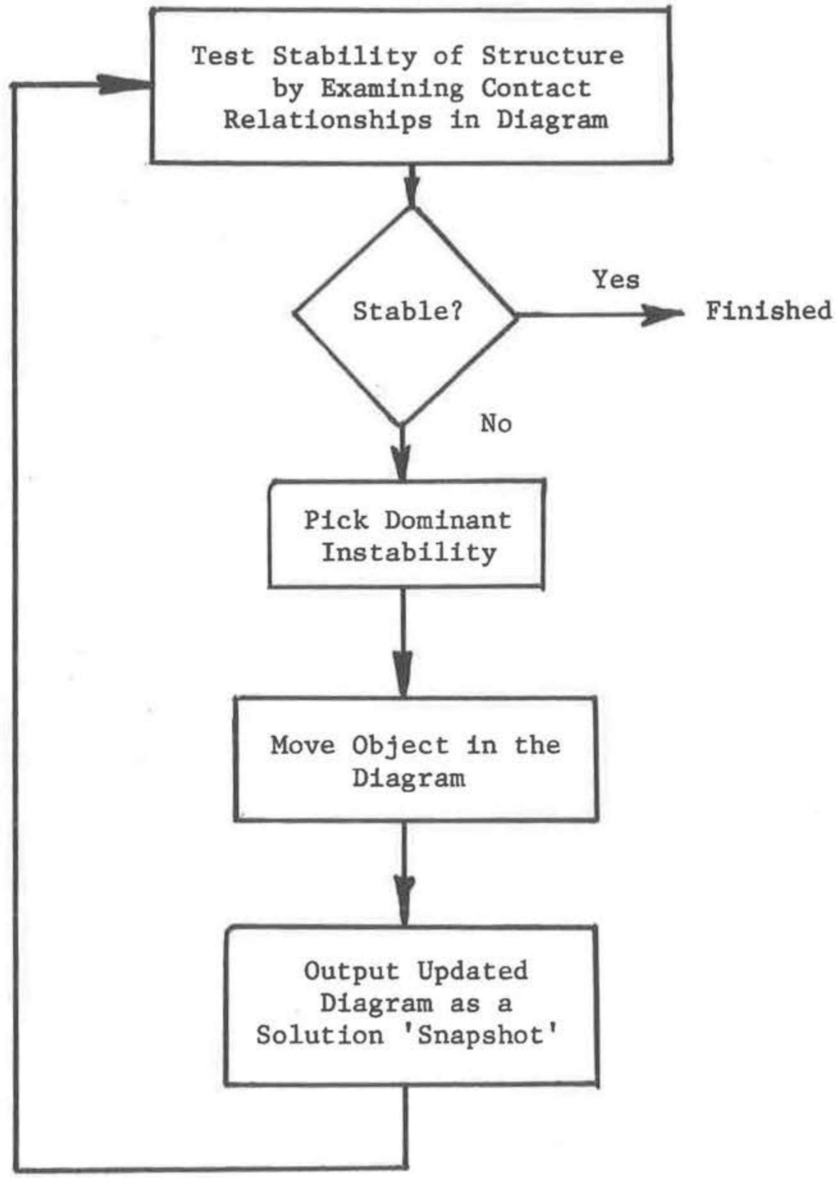

Test Stability of Structure by Examining Contact Relationships in Diagram
Stable?
Yes
Finished
No
Pick Dominant Instability
Move Object in the Diagram
Output Updated Diagram as a Solution 'Snapshot'

FIGURE II-2

of a structure and stability testing of individual objects.

The subdivision is directed by the top level routine, STABLE-STRUCTURE, which first constructs a list of the names of all the objects in the scene. (the diagram is in this case, as in all other cases, accessed through the eye and never directly cell by cell.) STABLE-STRUCTURE then tests each object to see if it is UPWARDS-STABLE. If every object is UPWARDS-STABLE then the complete structure will be stable.

An object, O, is UPWARDS-STABLE if the conglomerate object, C, formed by gluing together O and everything O supports (including what O's supportees support), is stable. During the stability testing of C, O's supporters are assumed to be stable.

UPWARDS-STABLE forms the conglomerate object C of O by the following steps:

(1) Let C=O

(2) Use the retina to find the set S of immediate supportees of C

(3) If S is empty then return C

(4) Let C = object formed by gluing C and all members of S together into one object

(5) Go to (1).

In the case of the current example, WHISPER happened to choose object A as the first object to be tested for upwards stability, and so when B is found to be a supportee of A, the point where they touch is considered to be glued, resulting in a new object AB. (Finding points of contact and support between objects is one of the perceptual primitives which will be discussed later, but obviously it need not involve the complex touch test mechanism described by Fahlman, since contact points need only be recognized not computed.) If there were another object on top of AB, then the process would be repeated until either an object with no supportees or an object which acts as a cosupporter of a third object is reached. The dotted curves in figure II-3 encircle the sub-structures UPWARDS-STABLE finds. In (c), Q and RS are cosupporters of X, so the subdivision stops at X.

II-3.2 Object Amalgamation

Combining the descriptions of two objects is particularly simple in the framework of a diagrammatic analogue. Creating a new description from two other descriptions is an instance of the amalgamation problem. In the case at hand the creation of a new object from two other objects is merely a matter of not distinguishing between the 'colour' values (in this case A and B) designating the original objects. A red object combined with a blue object is described as the red-blue object. All the features or properties that can be seen in the original objects can also be seen in the combined object. Some of the properties which the combined object inherits are: its shape, its center of gravity, its mass, its position relative to other

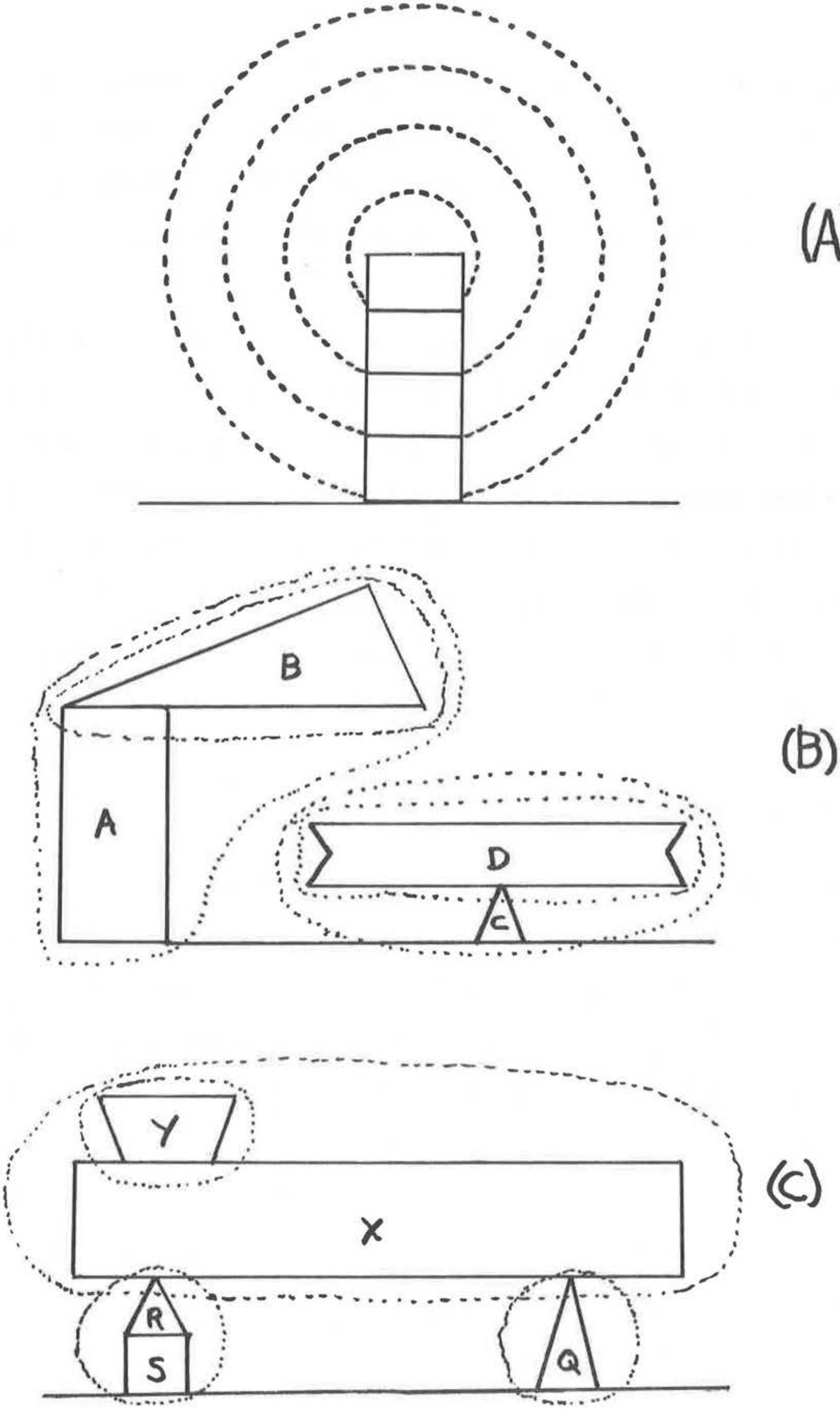


FIGURE II-3

objects, and its support relationships. Some of these features might not be so hard to compute in a descriptive formalism, but there still exists the need to compute them and more importantly, the need for procedures which know how to compute them.

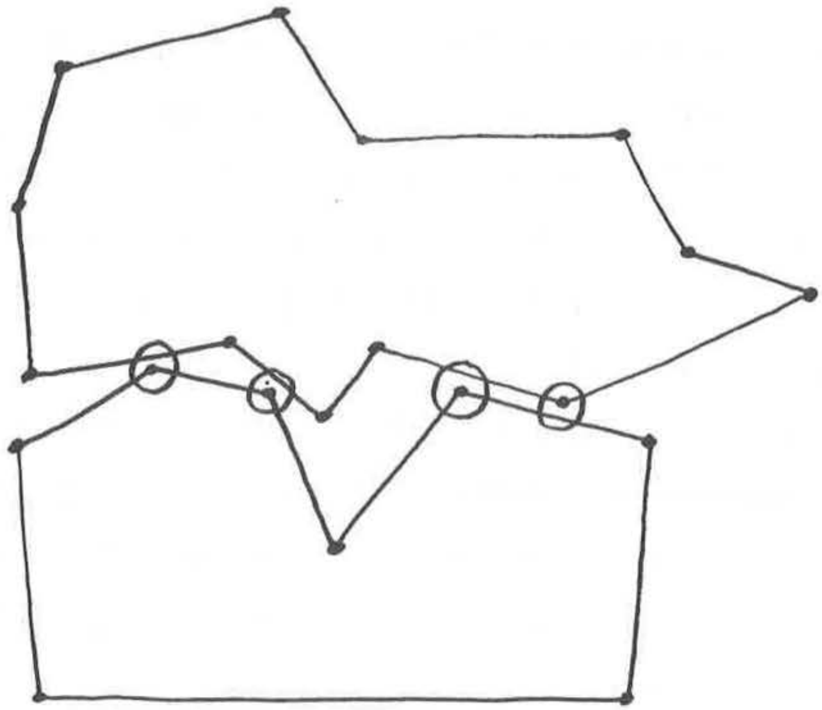
One property which would give the most difficulty in a non - diagrammatic representation would be shape. If, for example, the objects were described as polygons (figure II-4) then there must be some way in which the descriptions of the two objects can be edited to remove the segments corresponding to the contact, and to join the segments which lead from one object into the other even though these may not necessarily occur at the endpoints of the original segments.

II-3.3 Single Object Stability

In the upwards stability test, once the object has been glued to everything it supports the stability of this combined object is tested independently. Thus the problem of determining the stability of a whole structure is reduced at each stage to the determination of the stability of a single object.

For a single object there are only three basic types of instabilities that can arise. An object can either rotate about some support point, it can slide along some surface, or it can simply fall freely. If the center of gravity of an

AMALGAMATION OF THE POLYGONAL DESCRIPTION OF TWO OBJECTS.



POINTS WHERE LINE SEGMENTS MUST BE BROKEN ARE CIRCLED O.

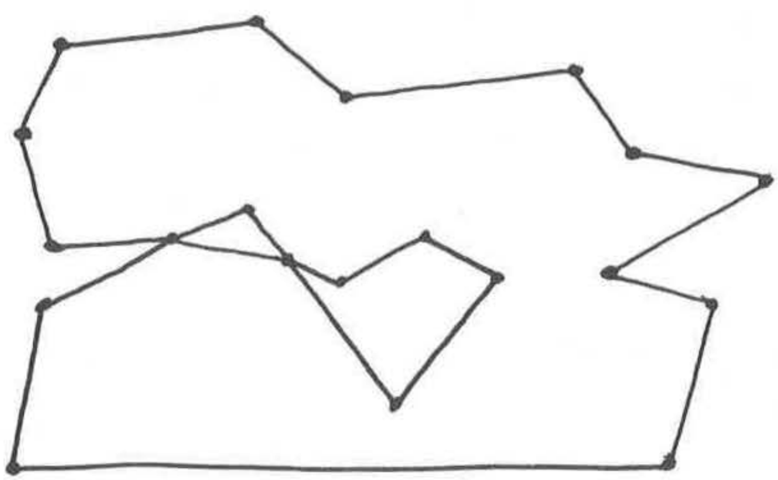


FIGURE II-4

object has a support vertically below it or if there are supports of the object on both sides of the vertical through the center of gravity, then the object will not rotate. WHISPER thinks an object 'hangs over too far' if its perceived center of area falls outside its supports. Because of the restrictions of uniform density and thickness imposed upon the class of objects WHISPER handles, an object's diagrammatic center of area and its physical center of mass are at corresponding locations. Center of area determination is a perceptual primitive whose inclusion in the set of primitives is justified by its importance in the implementation of the similarity and symmetry primitives in addition to its utility in the current domain. In the current problem (figure II-1), WHISPER sees that the center of area of the combined object AB is to the right of the support provided by the table so it notes that AB will rotate and continues with an analysis of the other objects in the scene. (Eventually the stability of B alone will be considered in the upwards stability testing, and it too will be noted as being rotationally unstable because its center of area lies to the right of the support provided by A.)

II-3.3.1 Balancing Objects

Equilibrium situations such as that of object D in figure II-1 or of M and N in figure II-5 provide a good example of the qualitative nature of WHISPER's reasoning. The

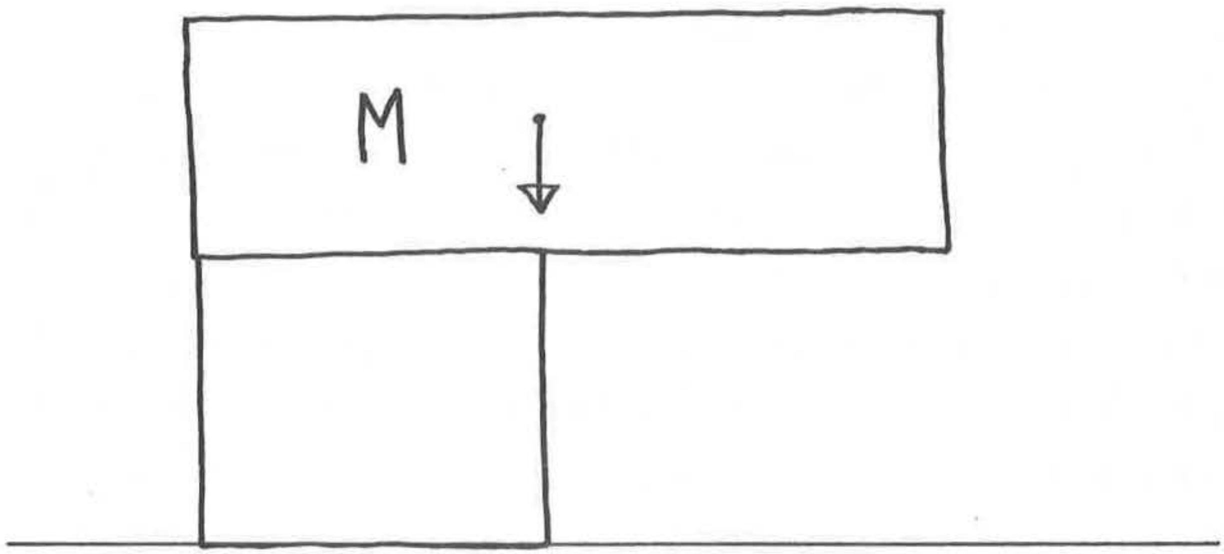
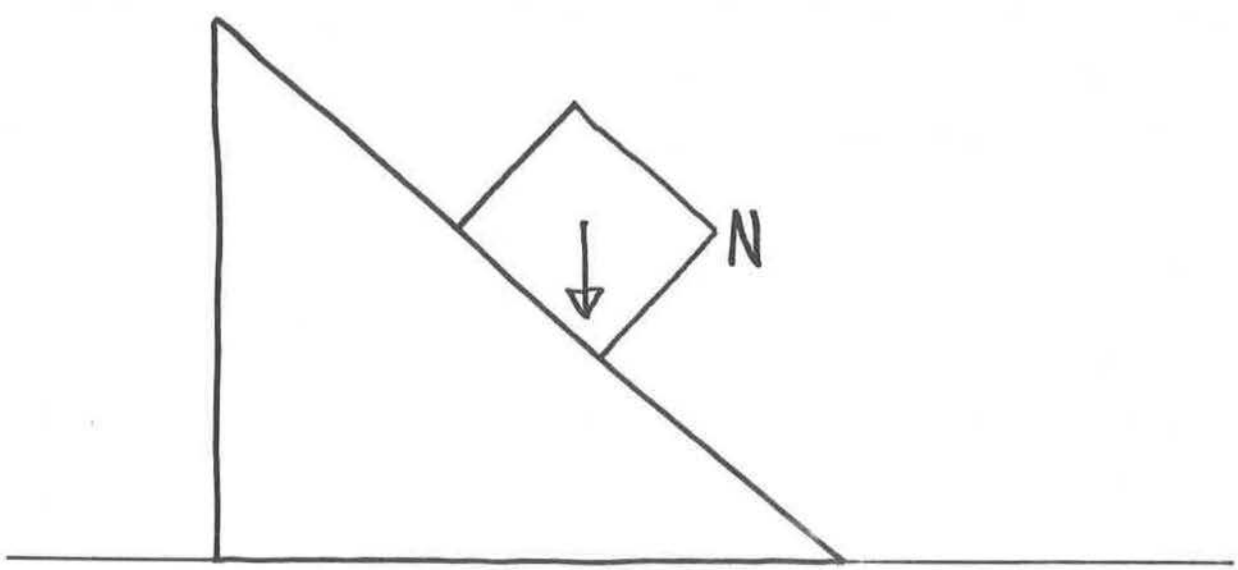


FIGURE II-5



approximate center of gravity of the balancing object is found by the perceptual routines, but this is insufficient for determining the stability of the situation. Since the slightest shift in the center of gravity would upset the balance, its precise location must be known in order to establish that the object is in a state of equilibrium. The center of gravity can be established as being directly above the support point if the balancing object is symmetrical (symmetry is another perceptual primitive) about a vertical axis through the support. If the object is not symmetrical about that axis then WHISPER may have to report that it cannot decide the stability of the configuration. In a case such as that of figure II-6, however, WHISPER determines that although the combined object PQ is unsymmetrical, P itself is symmetrical, and so Q, no matter how small it is, will tip the balance to the right. It is only in the case where Q is small that the need for the symmetry testing arises, since if it were large enough it would have had a significant enough effect on the original approximation to the center of gravity that PQ would have been clearly unstable.

II-3.3.2 Forces On Cosupporters

When two supporting objects participate together in supporting an object then they share the force from that object. This force may be enough to cause one of the

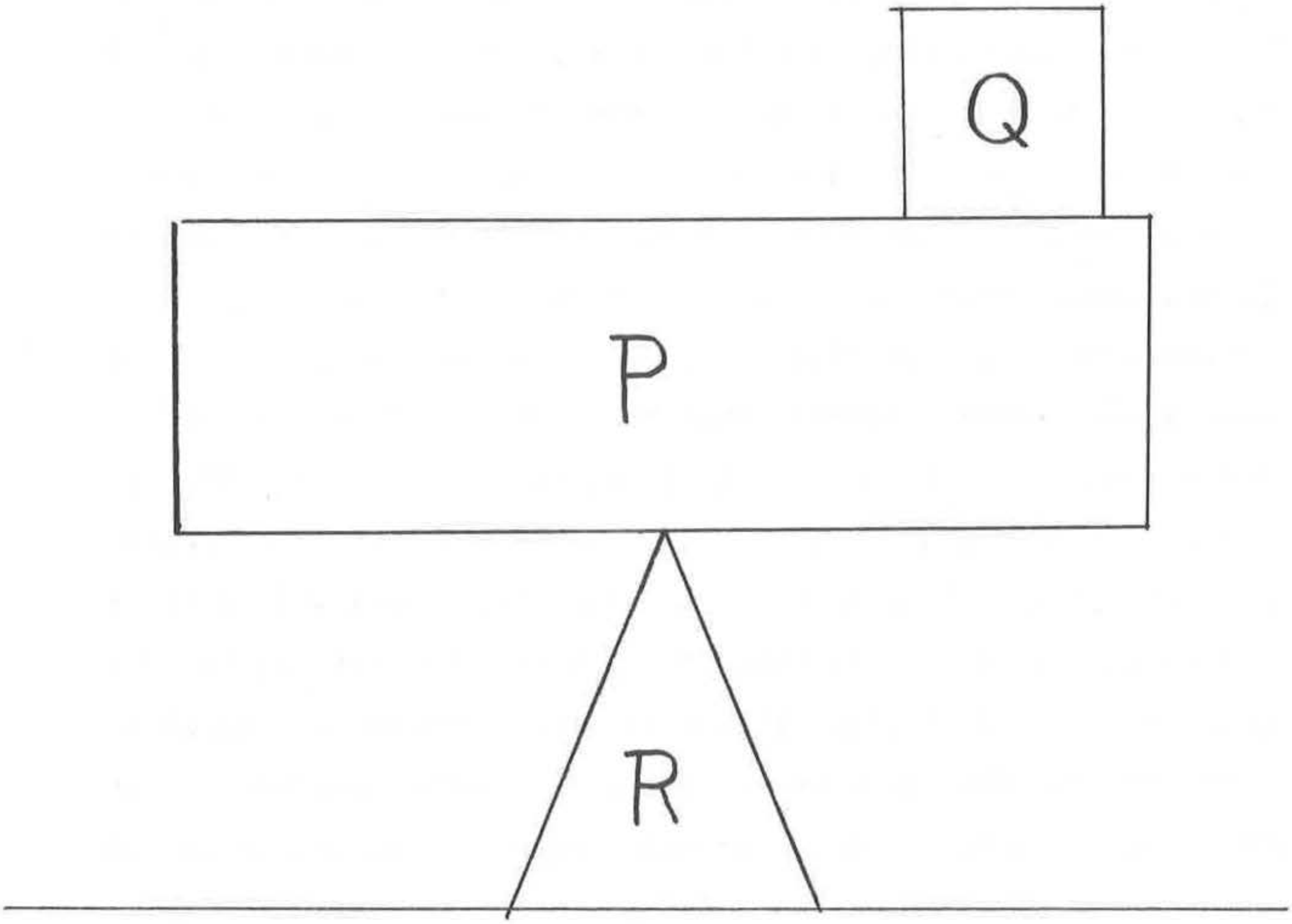


FIGURE II-6

cosupporting objects to rotate, depending on where the supportee makes contact. As mentioned earlier, the process of gluing objects together continues until either a top block (one with no supportees), or a cosupporting object is reached. In the latter case an extra force is noted as being applied to the cosupporters from the supported object. Thus in the second frame, figure II-7, of the current problem WHISPER will proceed to determine the stability of object D in the same manner as in the first frame, except that the extra force from B will be noted. WHISPER does not consider the exact magnitude of the force, but simply notes it as a force greater than zero applied at the point of contact. The effect this force will have is determined by taking the contact point as the new center of gravity of the object receiving it. This is equivalent to assuming that the force is of arbitrarily large magnitude. If the object with this new virtual center of gravity is stable then the extra force has no effect. Thus the stability of R and T is not affected by S in figure II-8, whereas the stability of D is affected by B in figure II-7. Counterbalancing forces such as that provided by V in figure II-9 require a quantitative solution and have not been considered. In general, people when asked about such situations reply that they are not sure about the stability, pointing out that it depends on the exact weight of the counterbalancing object.

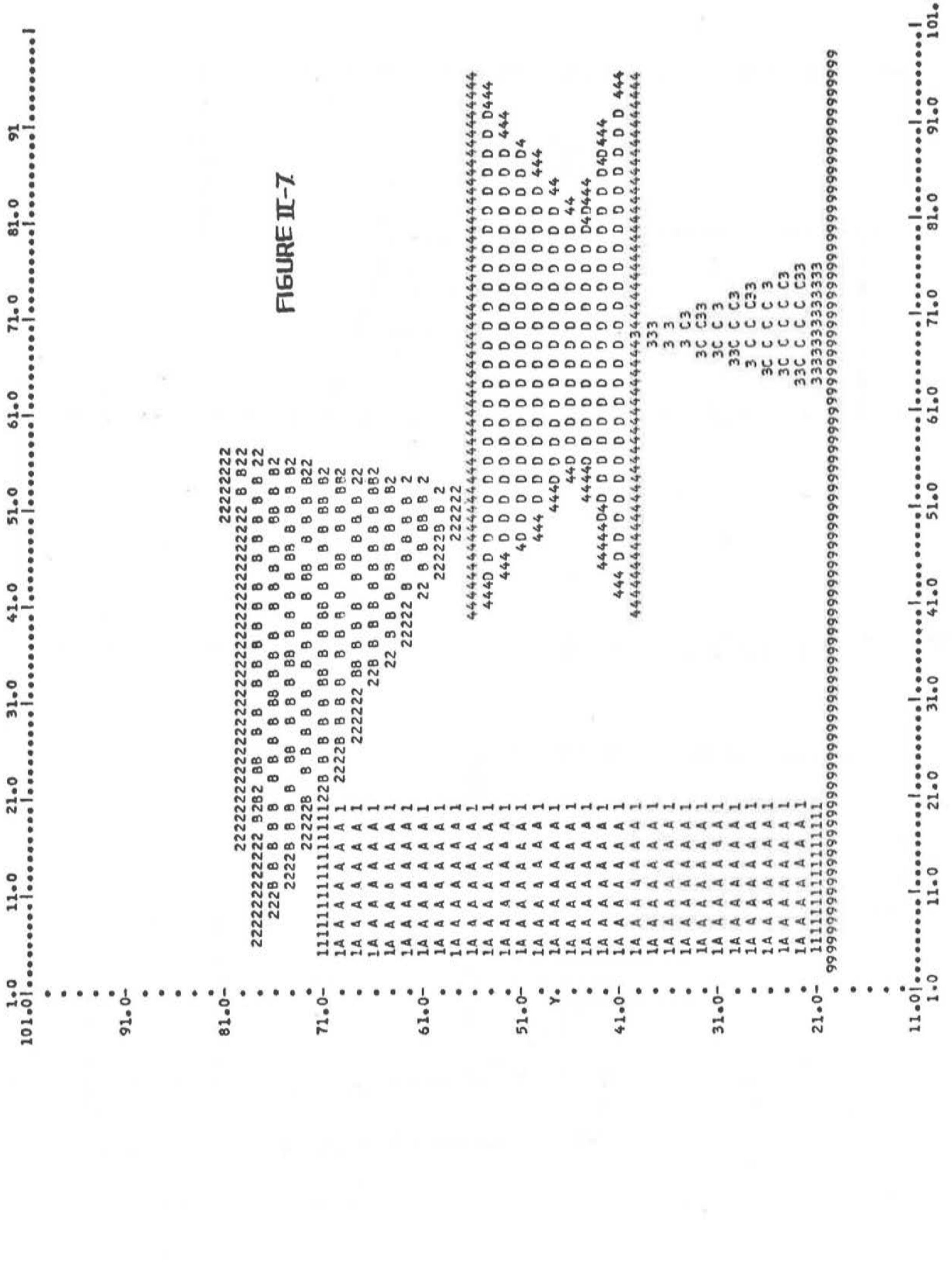


FIGURE II-7

FIGURE II-8

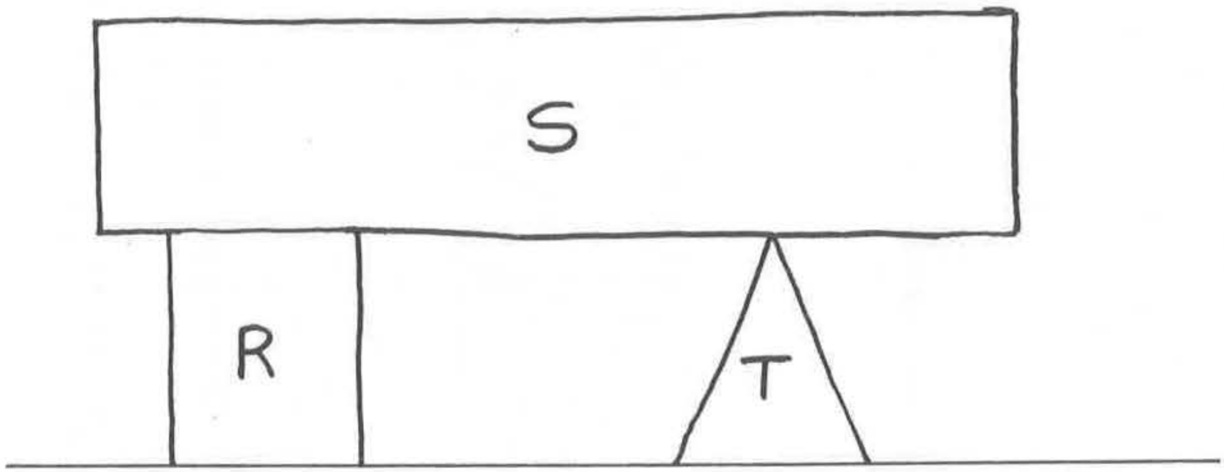
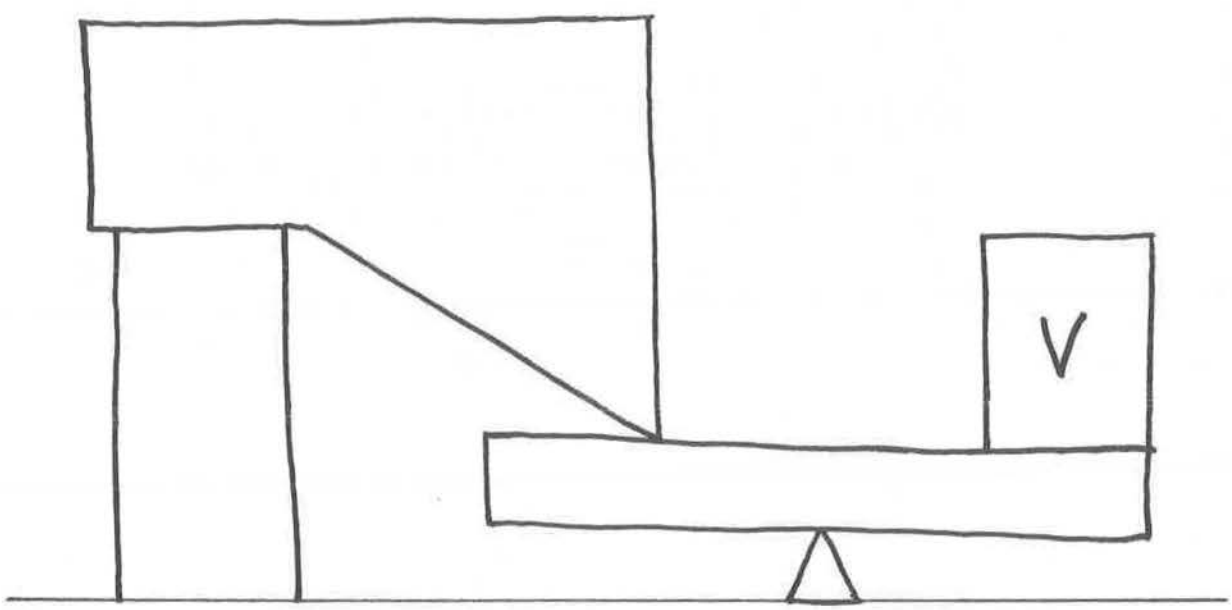


FIGURE II-9



II-4 Rotation Of Objects

If the stability test discovers that a single or conglomerate object will topple then the diagram must be updated to reflect the resulting situation. To do this the angle of rotation must be determined. A toppling object will rotate either until it hits something or until it begins to fall freely. WHISPER visualizes an object's rotation with the retina to determine the point at which its swing ends, and calls the redrawing transformations to rotate the object to that point in the diagram. Figure II-10 shows the overall organization of the rotational motion procedures.

The transformations of the objects in the diagram are carried out after all of the primitive objects in the scene have been tested by UPWARDS-STABLE. In the current example this requires testing the independent stability of objects AB, B, CD, and D. When this is complete WHISPER will have noted two rotational instabilities, objects AB and B, and no sliding or freefall instabilities (these will be discussed in section II-8 with reference to another example). Since B is a part of AB, WHISPER rotates B rather than AB. (To see that combined objects need to be considered at all look at figure II-11 in which the object which is rotationally unstable is RS alone, so it is RS which must be rotated.) Object B will pivot around the point at which it contacts A nearest the center of gravity of B. WHISPER uses its eye to examine the contact surface between

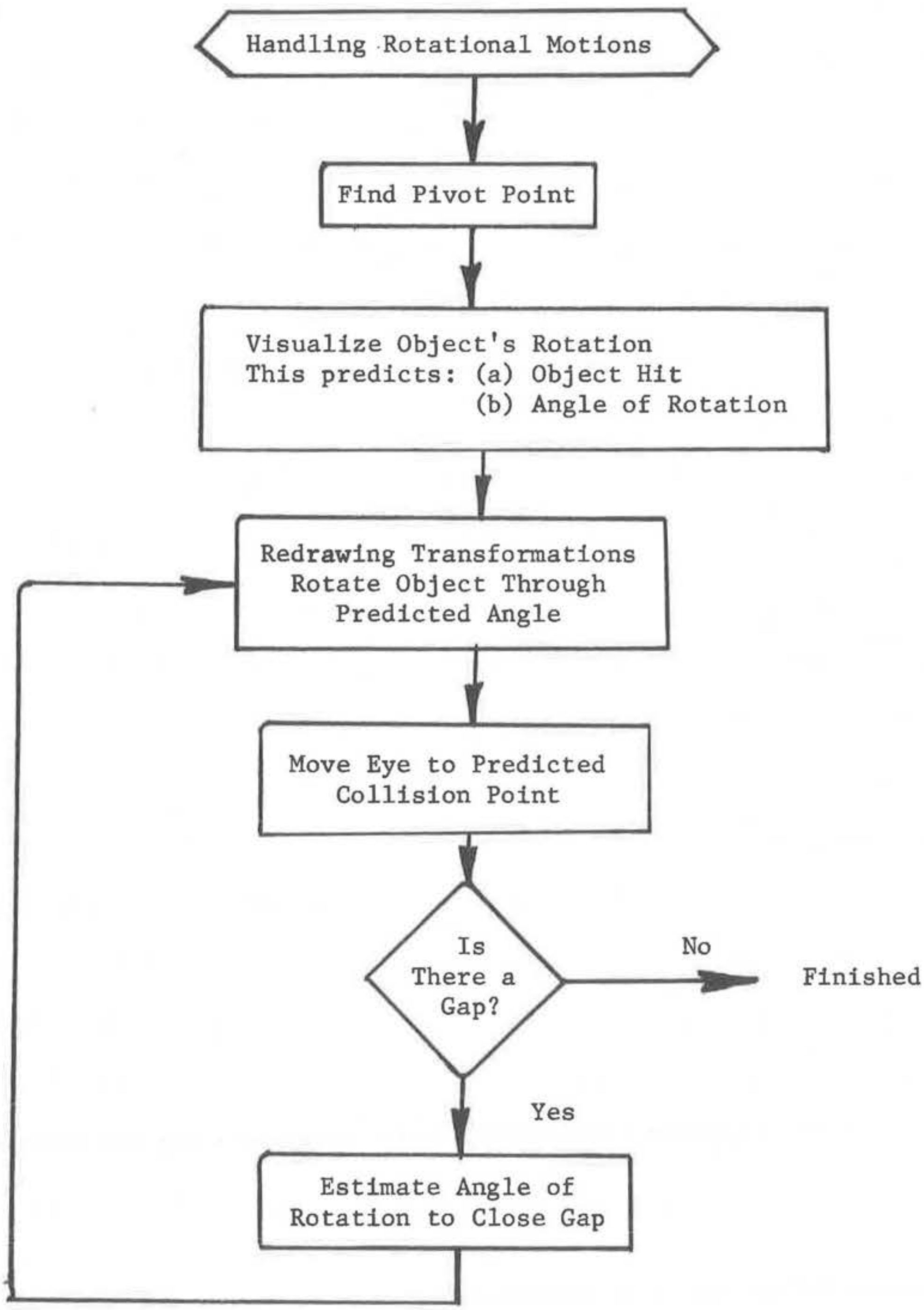
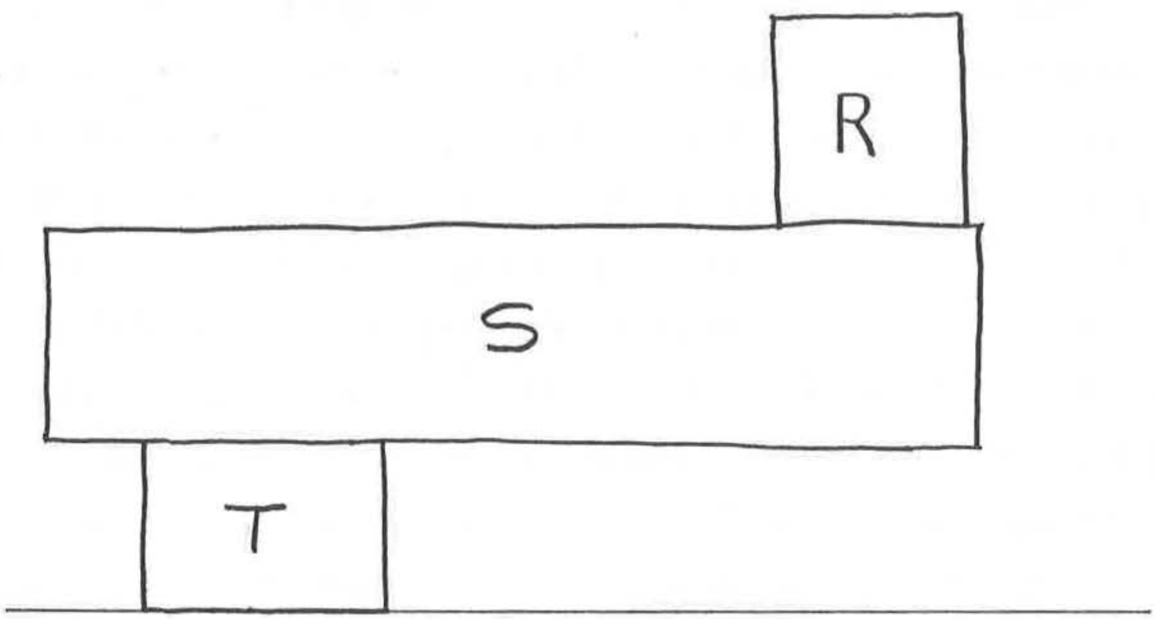

Handling Rotational Motions
Find Pivot Point
Visualize Object's Rotation
This predicts: (a) Object Hit
(b) Angle of Rotation
Redrawing Transformations
Rotate Object Through
Predicted Angle
Move Eye to Predicted
Collision Point
Is
There a
Gap?
No
Finished
Yes
Estimate Angle of
Rotation to Close Gap

FIGURE II-10

FIGURE II-11



the two objects to find the extremities of the contact. Since the contact is to the left of B's center of gravity, it is the right contact extremity which is used as the pivot point.

II-4.1 Finding Discontinuity Points Of Rotational Motions

Given the pivot point, it is simple to start the rotation of B. There is one serious problem remaining, however. When should the rotation be terminated? I term this problem serious only because it or a variation thereof has managed to escape any satisfying and reasonable solution in other problem solving systems. The essential element of the problem is the anticipation or detection of collisions between a moving object and other elements of the environment. Winograd's[7] SHRDLU ignored the problem; Fahlman also appears to have basically ignored it. The closest he has come is with his Findspace Proposer which puts an object into an arbitrary position and then checks whether it touches anything using his rather complex touch test algorithm. Fahlman suggests that it would be hopeless to use a variant of this algorithm for finding paths in 3-space.[8]

II-4.1.1 The Empty Space Problem

The source of the difficulty is in handling negative questions. The approach of most current systems is to describe the position of each object by the coordinates of its origin - some arbitrary point on it. No mention is made of where objects are not located. Thus empty space must be found through 'proof', either computational or deductive, of the statement 'there does not exist an object at location P'. Although it need not necessarily be the case, this has been effected by using the equivalent statement 'for each object O, O is not at P', and individually testing all the objects in the universe. The result is unmanageable growth of computational requirements.

The feature that WHISPER exploits in the diagrammatic analogue is that empty space in the problem environment is explicitly represented by empty space in the diagrammatic analogue. While the proposal that physical space be represented by array space is not new[9], it seems never to have been regarded as viable. Some reasons are:

(i) It would appear to be very inefficient and expensive in terms of memory usage. This objection can be countered in several ways. In view of the discussion on the internal/external question (section I-2), 2-space and 3-space can be viewed as part of the 'hardware' of any machine, so it is a matter of harnessing this space as opposed to allocating more in the form of core storage. In addition, current technology promises vast quantities of cheap computer memory. The limits of processor speed seem much closer to being reached than the limits of storage capacity. In WHISPER's domain there is not only a trade-off between space and time, but one between space and conceptual simplicity as well. It is reasonable to trade a fixed, but large amount of storage for these two factors of time and simplicity.

(ii) The empty space comes in too many pieces to be dealt with by a sequential process. The answer to this objection is to use a parallel process. The human eye is a strong precedent for this suggestion. Normally in AI applications, the linear reduction in elapsed computation time acquired through parallelism is not significant because of the frequently exponential character of the growth of computational requirements; however, in this case the amount of computation is a function of the fixed array size. The number of processors can be made large enough, but still fixed, so that what would be an impractical and inefficient sequential solution becomes practical and efficient in terms of parallel computation. Thus the proposal to explicitly represent physical space with array space attains a certain viability through a re-examination and redefinition of efficiency criteria for both storage and processing.

II-4.1.2 How WHISPER Finds Discontinuity Points Of Rotations

Since the diagram explicitly represents the empty space of the problem situation, WHISPER could determine what happens to moving objects by watching them. As long as an object passes through unoccupied space in the diagram, then it will not be involved in a collision in the real world. It would be a prohibitively expensive computation to simulate the motions by incremental movements, so instead of actually watching objects move in the diagram their motions are visualized with the retina.

There are two types of rotational motion discontinuities. An object will stop rotating either when it collides with another object or when its center of gravity reaches a position directly below its pivot point. At that point the object begins falling freely. The check for these two conditions is accomplished by centering the eye on the pivot point, and 'visualizing' the rotation of the object from its initial position until a collision occurs, or until the object reaches the freefall point.

II-4.2 Characteristics Of Visualization

I emphasize that the visualization process occurs on the software retina not on the diagram itself. For the full 360 degree retinal field there are only a small number of directions (in the case of the current retinal implementation, thirty-six) at which the object is 'visualized' during the rotation. Since an object will never rotate more than half a turn before it falls off, at most half of these need be considered. The mapping from the diagram to the retina ensures that nothing present in the diagram is absent on the retina, with the implication that any empty space on the retina is also empty in the diagram. Thus if a collision is not detected during the visualization process no collision would occur in the diagram, further implying that no collision would have occurred in the real world. This is the case even though only a fixed number of different orientations are tested. The only disadvantage is that some false alarms may arise, because objects are expanded slightly in the mapping from the diagram to the retina. The shift from one orientation to the next, and the test for any colliding contour segments are both parallel computations, so the net serial time required to test for collision by visualization is small. In addition, the number of processes, and hence the number of processors, is fixed as the number of cells (not to be confused with receptors since there might well be more receptors than processors) composing the software retina. In the current implementation this number is 540. A collision is detected if the contour of the rotating object crosses a cell whose current input content as seen directly from the diagram is non-empty and different from that of the contour itself.

II-4.2.1 Surprise Collisions

A nice feature of WHISPER's use of visualization for the detection of collisions is that it is never surprised by the presence of objects such as C in figure II-12. Strategies

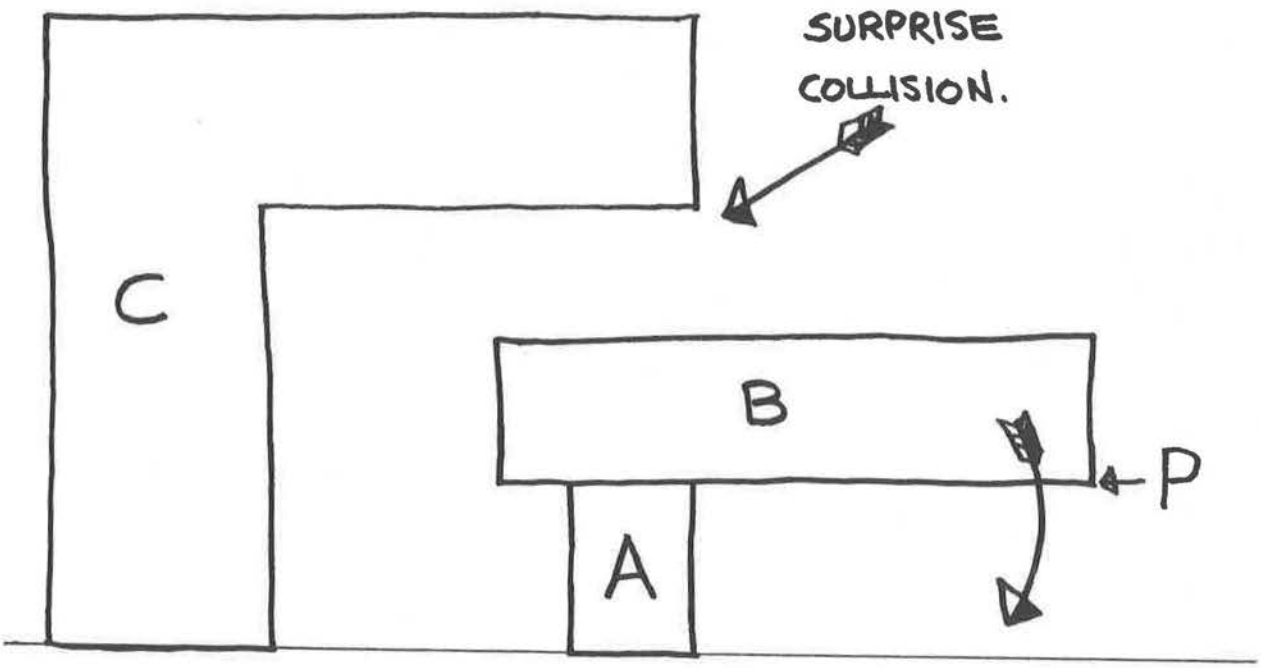


FIGURE II-12

relying on the computation of collisions of only point P on object B, or other strategies of partitioning the class of possible candidate collision objects on the basis of being members of the same structure or being below the current object would more than likely overlook such situations.

II-5 Updating The Diagram To Reflect A Rotation

Once the termination point of an object's rotation is known then the redrawing transformation procedures are called to rotate it in the diagram. Because the retina has only 36 sectors, the angle of rotation at which, through visualization, the first collision is detected is an approximation of the actual angle through which the object must be rotated in the diagram. The transformation procedures are called with a slightly cautious estimation of this angle so that the rotating object will not overshoot its collision point. The result of this first rotation in the chain reaction problem is shown in figure II-13. After this rotation has been made, the eye is moved to the predicted collision point. The spacing between the moving object and the one with which it is to collide is measured with the eye, and this measurement is used to compute the extra twist necessary to close the gap (figure II-14). Although it is not usually necessary, the eye is used to recheck the spacing. The rotation is complete.

II-5.1 Gripe Situations

The one situation in which the gap would still exist is depicted in figure II-15. A facility for handling this situation has not yet been included in WHISPER, but it is a simple matter to see how a complaint message (Fahlman termed these messages 'gripes') could be sent back to the

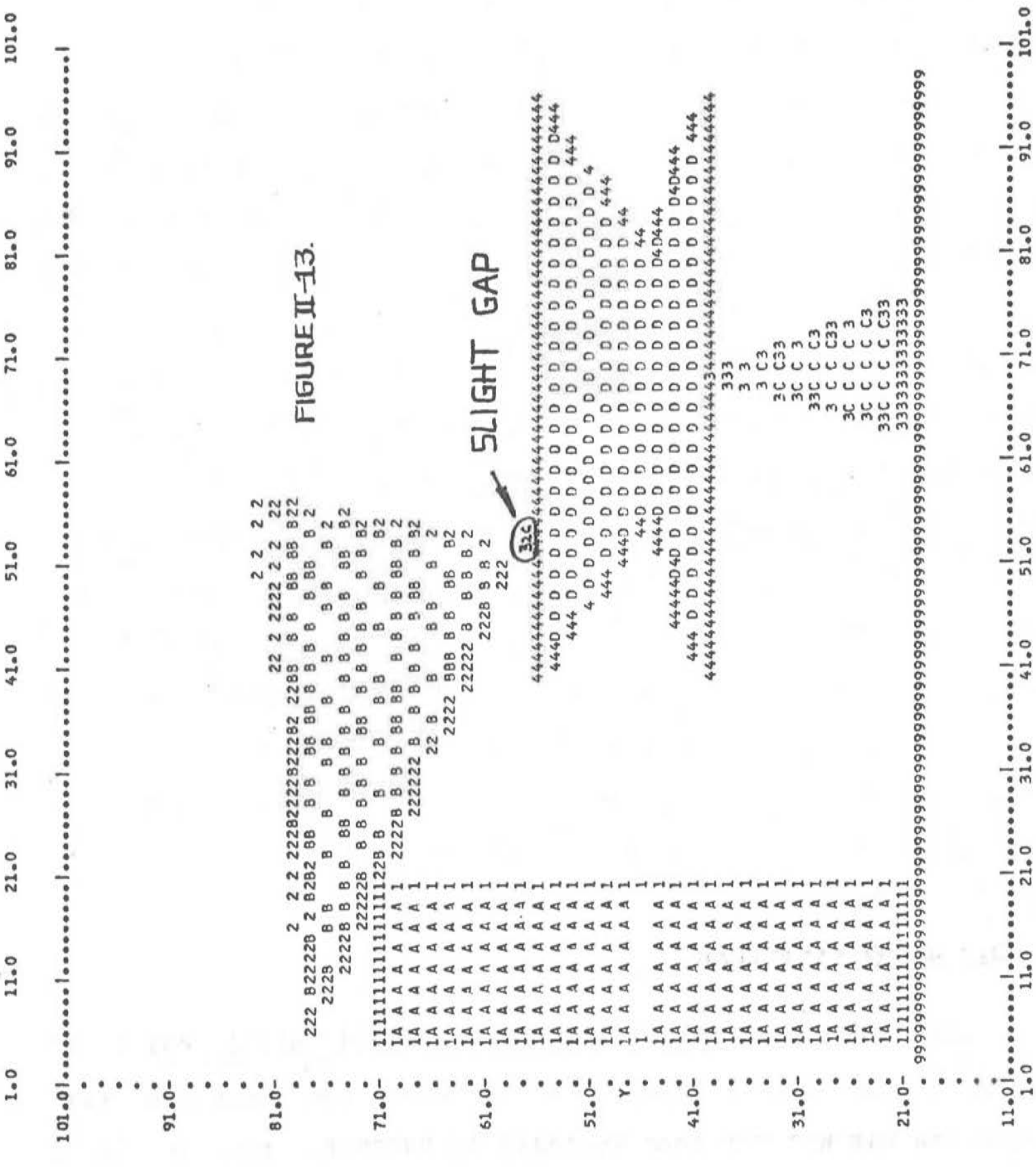


FIGURE II-13.

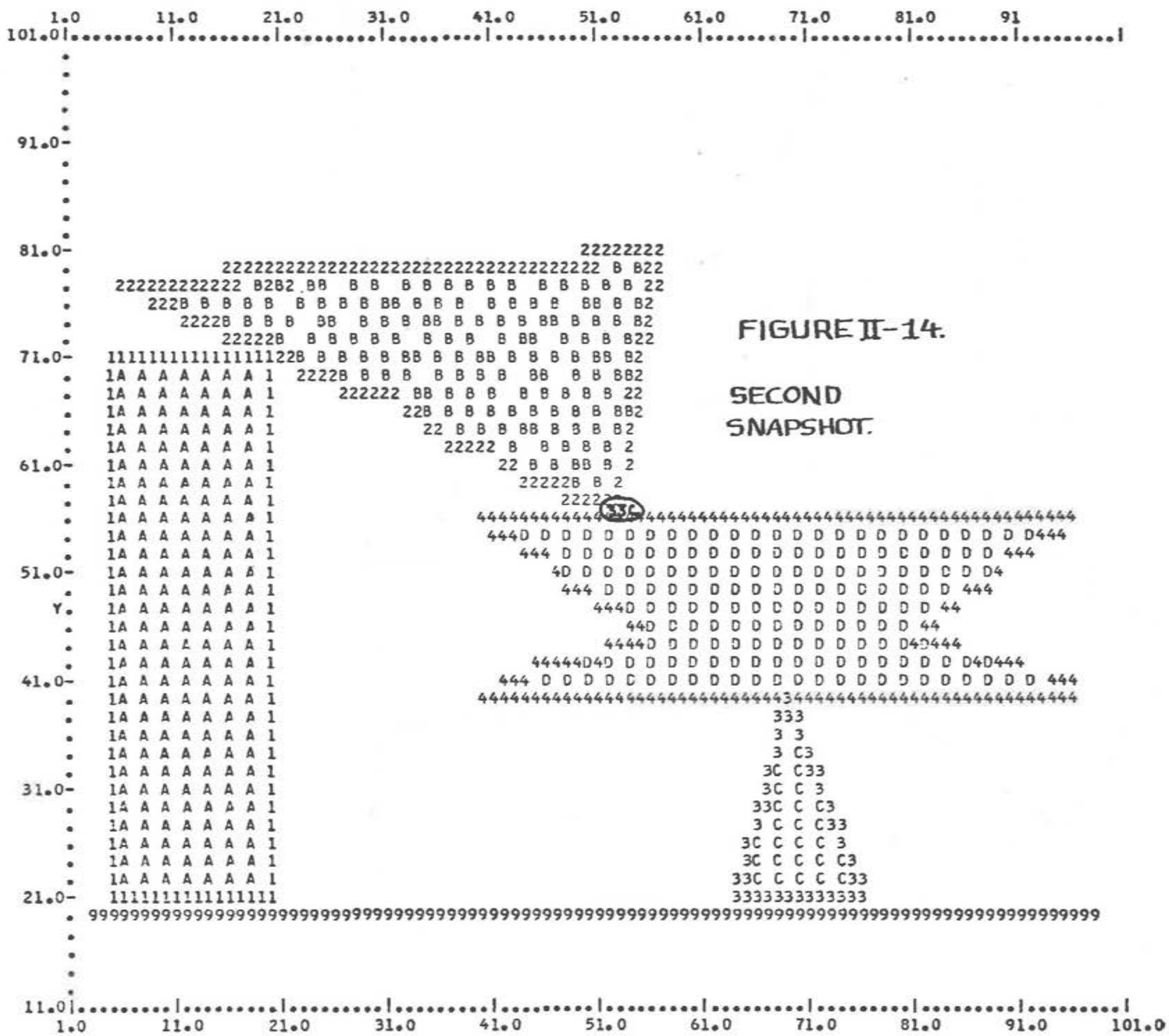


FIGURE II-14.

SECOND SNAPSHOT.

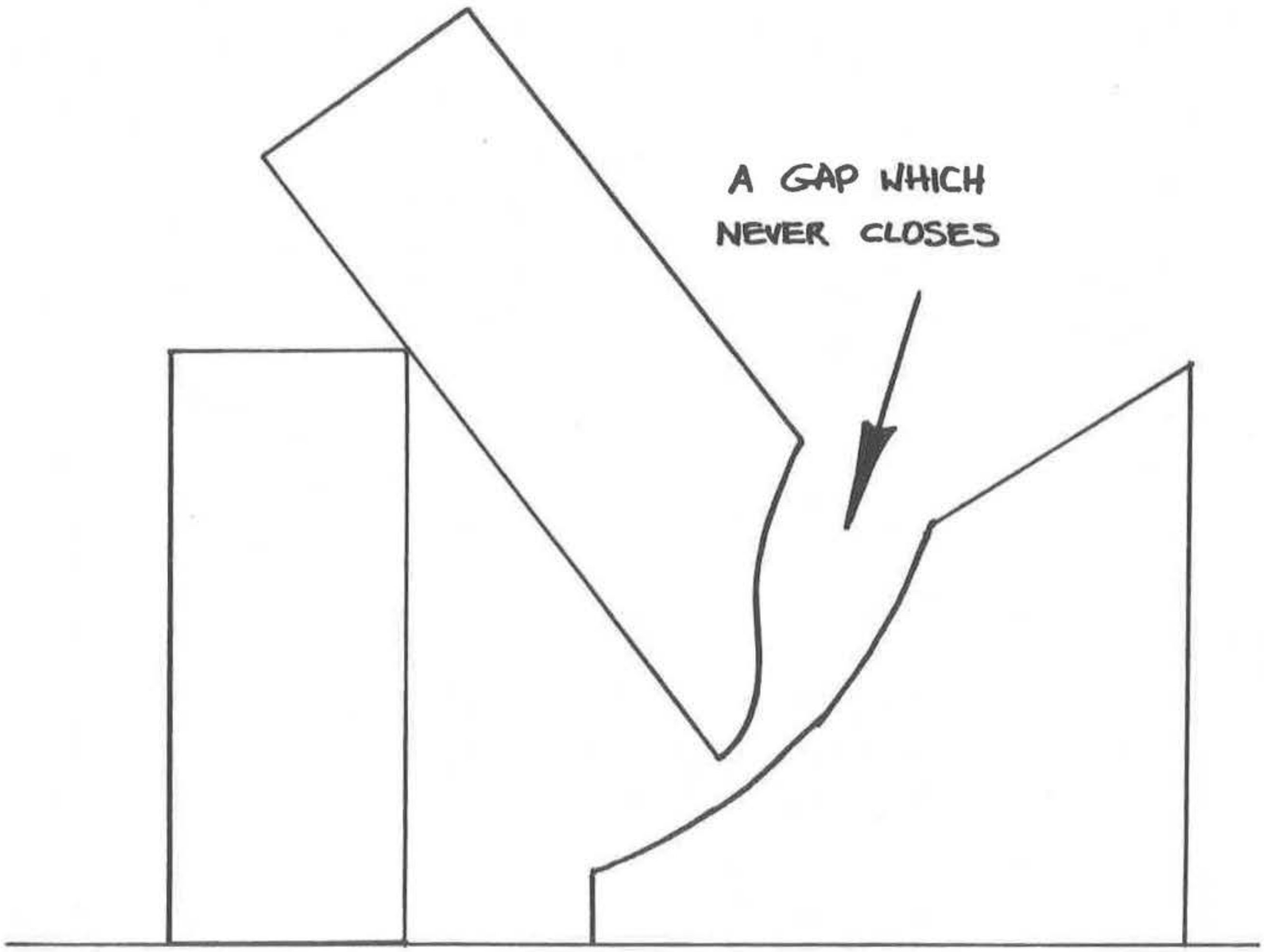


FIGURE II-15.

visualization procedures requesting the generation of the next collision point. This is a qualitatively different type of gripe situation than those arising in Fahlman's BUILD system. The gripe here is not computable in advance _but arises purely out of the experiment with the analogue;_ gripes in BUILD arise because the code was more conveniently written in a form which put off error checking as long as possible. All gripes in BUILD could be eliminated, that is they could be predicted and thus avoided, by executing the same code in different sequence. For example, Fahlman[10] discusses the example of a routine named MOVE, finding an object already in the spot where it is requested to place another object. In this case MOVE generates a gripe indicating that the two objects have collided. The discovery that the spot is occupied is made by executing some of BUILD's code to explicitly test for the presence of the gripe condition. This test could have been executed earlier, before MOVE was called upon to perform an impossible task. In our example the gripe situation is not predictable through the execution of any of WHISPER's code in any order. The gripe arises experimentally in the analogue and is only recognized, not computed, by WHISPER.

II-5.2 Advantages Of The Feedback Method

It is this feedback from WHISPER's experiments with the diagrammatic analogue that provides conceptual simplicity in discovering rotational termination points. Both visualization and gap closure rely on feedback. These two methods in combination represent WHISPER's pragmatic equivalent to the experiment of repeatedly rotating the object in the diagram by small increments until the first collision occurs. Using feedback in this manner has generally been spoken of in terms of a robot immersed in a real world environment. Here it is being obtained not from the real world of falling objects, but from an analogue of the real world situation, namely the combination of a diagram and appropriate transformation procedures. Reliance on this feedback from experiments with the diagrammatic analogue eliminates the necessity for equations of motion and touch tests for arbitrary shapes. WHISPER is not forced to use sophisticated 'number-crunching' techniques in establishing the points at which an object's motion will change. This is appropriate because the hard part of the problem, the part which involves the qualitative physics, is predicting what the motion will be, not when it will terminate.

II-6 The Eye Movement Protocol For The Chain Reaction Problem

As the problem solving process proceeds the retina constantly moves from place to place in the diagram. A trace of the eye movements is given by the circled numbers in figure II-16, figure II-13, and figure II-14. Each circle represents a fixation of the retina at its location in the diagram. The numbers give the order in which the fixations occurred. A number with the letter C attached to it indicates that the central portion of the retina was fixated at the location; a number without a letter indicates that the periphery of the retina was fixated at the location. The structure of the retina is discussed in section III-2. Although moving the two parts of the retina separately would be unnecessary if there actually were many processors operating in parallel, it saves a considerable amount of computation in the pseudo-parallel simulation. A list of the fixations plotted on the diagrams with reasons the qualitative knowledge directed them follows:

(1) Move to center of diagram; return names of all the objects in the scene.

(2-4) Find the center of gravity of A; find supportees of A.

(5-6) Find the center of gravity of B; find supportees and supporters of B.

(7) Move central section of retina; find exact contact point of A and B.

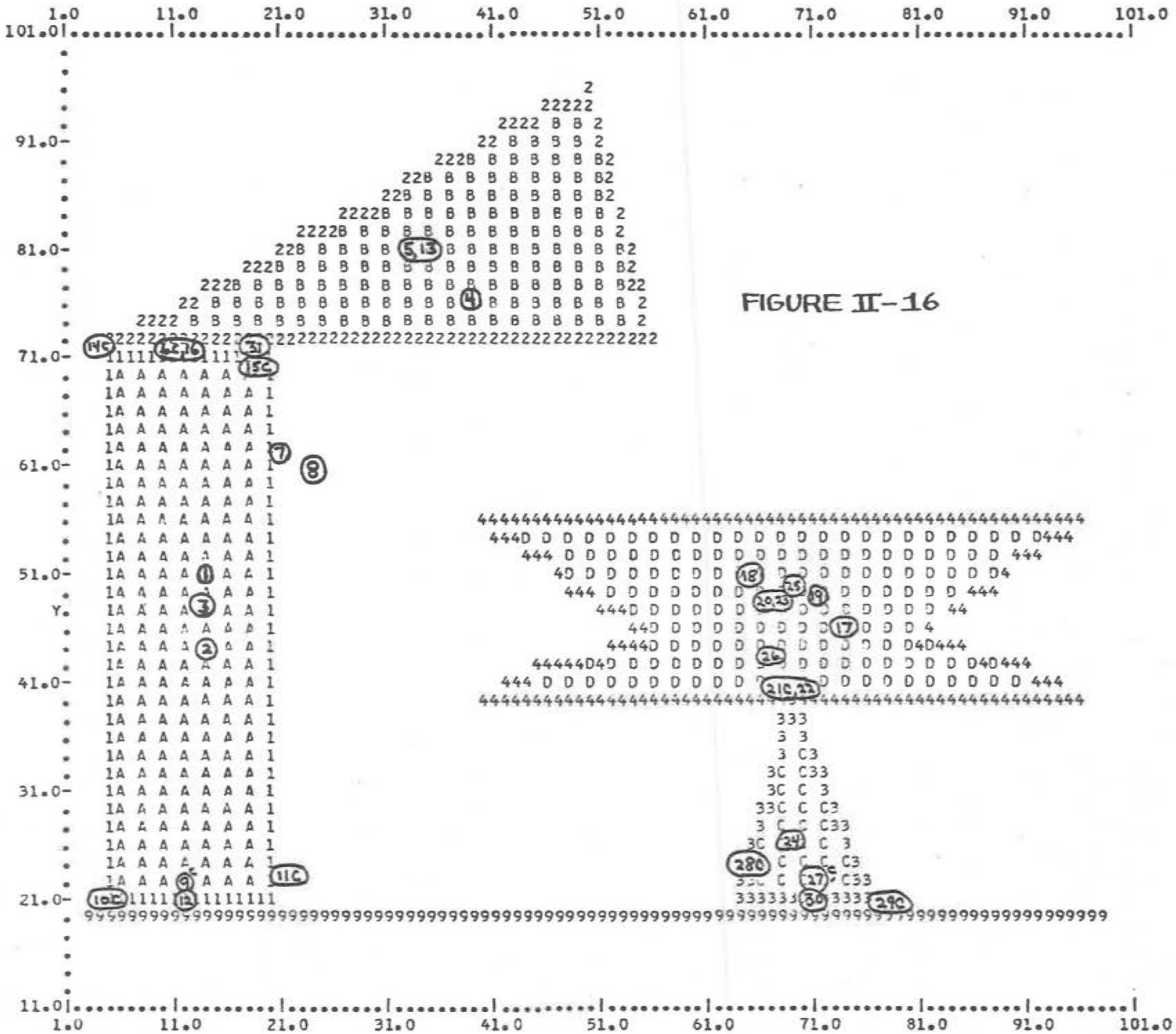


FIGURE II-16

(8-9) Find center of gravity of AB; find supporters of AB.

(10) Move central section; find exact contact point of AB with table.

(11-12) Move central section; find extremities of contact surface.

(13) Find the slope of the contact surface.

(14) Move to center of gravity of B3 look at contact between A and B.

(15-16) Move central section; find extremities of contact surface between A and B; (5 , 72) and (19 , 72) are returned.

(17) Determine the slope of the contact surface.

(18-20) Find center of gravity of D; look for supporters and supportees.

(21-22) Move both the central section and the periphery; find the exact point of contact with C. Discovers that support is a point not a surface indicating possible equilibrium situation.

(23) Move back to center of gravity of D to check for symmetry of D; equilibrium is found to be ok.

(24) Finding center of gravity of C; look for supportees of C.

(25-26) Finding center of gravity of CD; find supporters of CD; finds the table.

(27) Move central section; find exact point of contact of CD with table.

(28-29) Move central section; find extremities of contact surface; returns (64 , 22) and (76 , 21).

(30) Determine the type of contact and its slope.

(31) Move to the pivot point of the rotation of B to visualize the rotation.

****The rotation is then carried out in the diagram, see figure II-13.****

(32) Move central section to estimated point of collision between B and A to see if they touch; the gap is seen; the amount of the next rotation is estimated.

****Another rotation is carried out in the diagram, see figure II-14.****

(33) Move central section to estimated point of collision between B and A; now they are seen to touch.

II-7 Subsequent Snapshots Of The Chain Reaction Problem

In the problem solving process thus far, the state of the world has been considered by examination of the diagrammatic analogue, and it was concluded that object B would rotate. Rotating an object is an action which changes the state of the world; the effects of this change must be reflected in WHISPER's world model if it is to successfully continue with the problem solving process. The action is represented by the application of a rotational transformation to B in the diagrammatic analogue, and the new state of the analogue represents the state of the world resulting from the action.

II-7.1 The Frame Problem

The problem of updating a system's representation of the state of the world to reflect the effects of actions performed in the world is the 'frame' problem. In illustrating this problem Raphael[11] used the example of a situation with a robot at position A, a box B1 at position B, and another box B2 on top of B1. If the robot moves to a new position, C, then the statement describing A as the robot's position has to be replaced by one stating that the robot is at C, while all the other statements must be left unchanged. If the robot pushes box B1 to C, then both the descriptions of the position of the robot and the position of B1 must be changed. In addition, facts derived from the initial situation, such as B2 is at B, may no longer hold. In order to make these transitions from an initial state to the state which results from an action, the causal connections between actions and properties of states must be specified.

One aspect of the 'frame' problem is the necessity to explicitly know which properties remain unaffected by which actions. If this is not known, then it is not possible to infer that these properties still hold after an action by which they are actually unaffected.[12] Another aspect is that there must be some way to state that the problem description exhaustively describes all the causal connections which exist between objects. For example, it is possible that B2 is connected by a wire to the ceiling so that when B1 is moved, B2 actually remains at position B instead of moving with B1. Even if these difficulties are surmounted, the problem remains of effectively organizing an inference mechanism to efficiently reason about and discover the chains of causal connection along which the side effects of actions propagate.

The transition between WHISPER's snapshots is exactly the type of situation in which the 'frame' problem would trouble a system based entirely on a descriptive representation. It involves the representation of action, the effects of action, the issue of exhaustiveness, and chains of causality. Because WHISPER relies on a diagrammatic analogue as a representation of the state of the world instead of a description it is not troubled by the ubiquitous 'frame' problem. <u>The state of the</u>

world is represented by the state of the analogue, and action in the world is represented by corresponding action in the analogue. The corresponding action is the application of the appropriate transformation, and the effects of the action are correctly represented by the resulting state of the analogue.

In WHISPER's current problem the qualitative knowledge procedures know that the action of B's rotation is represented by calling the rotation transformation procedure to redraw B at its new location in the diagram. Almost all of the information that it needs to continue its problem solving is correctly represented by the updated diagram. It can proceed just as if the new snapshot were its original input and it were starting a brand new problem. The most important information which has changed in the transition between the states as a result of the rotation is: the position and orientation of object B; the position of its center of area; the contacts it makes with other object; and the shape of the areas of empty space. There are also a multitude of things which have not changed and are correctly left unchanged by the rotational transformation, such as the position of all the other objects, the shape of all objects, the area of all objects, and the contact relationships of other objects not involving B. All of these things work out correctly without the need of any deduction or inference on WHISPER's part. All that it need do is to use its retina to look at the diagrammatic analogue and extract whatever information it needs.

The visualization process works because of the exhaustive nature of the diagrammatic analogue. All the objects which could affect the motion of B are in their proper positions in the diagram. None is missing. We could add some external force such as a strong magnetic field which would interfere with B, but there is no problem in expressing the assumption that such a force does not exist in the current situation.

The discovery of causal chains is also facilitated by the diagrammatic analogue. In particular, what causes termination of B's rotation - its collision with D - is easily found by the visualization process. After the rotational transformation is applied, all its side effects are immediately propagated throughout the diagrammatic analogue. WHISPER is in a good position to apply its qualitative procedural knowledge to determine what effects will follow from this new state. In following the causal chain from the initial input snapshot through the intervening sequence of snapshots to the final snapshot, there is continual interaction between the higher level procedurally represented qualitative knowledge and the more mundane though voluminous information contained in the diagrammatic analogue.

An expanded WHISPER system could not completely avoid the pitfalls of the 'frame' problem because not all of the information about the current state of the world can be represented by the state of the analogue. For example, once an object starts moving it acquires some momentum and this momentum will cause a force greater than that resulting from the force of gravity alone to be applied to the supporters of the moving object in the subsequent snapshot analysis. Velocities are not part of the state of diagrammatic analogues and thus velocities in the world are not represented by them. For the qualitative solutions which WHISPER currently obtains, a consideration of the velocity and momentum of objects is not necessary. Although the 'frame' problem cannot be totally eliminated, WHISPER demonstrates that it can be circumvented to the extent that the system need not be hampered in its search for a qualitative solution by the many messy details involved in propagating the effects of simple causality.

II-7.2 The Third And Final Snapshots

WHISPER begins thinking about the second snapshot as a new problem and proceeds through some of the same considerations as for the first stage. Object D is found to balance on C except that there is an extra force on D from B and this force lies outside the supports of D. Therefore D is noted as rotationally unstable. Object B is at first expected to slide to the right except that this is based on the assumption that its supporters are stable. Supporter D is not stable, however. The rotational instability of D is given precedence over the sliding instability of B, and D is rotated about its contact with C until it hits the table. As with B toppling, D is visualized as rotating with the eye centered on the pivot point, an approximate angle of rotation is obtained, D is rotated in the diagram by this amount (figure II-17), and then the eye is moved to the gap to measure it for the computation of the final twist. The result is the third snapshot, figure II-18.

Again, this third snapshot is taken as a new problem. Object B is found to be rotationally unstable, and is rotated until it hits D again, figure II-19. This is the point at which WHISPER's analysis stopped. Some of its first order approximations to simultaneity and velocity are simply no longer viable.

The essential elements of the action involved in the collapse of the initial structure are portrayed by the four snapshots that WHISPER produced. The initial instability of B is shown to result, as B hits D, in the subsequent toppling of D, and their eventual tumble to the table. The action could be specified somewhat more precisely by including the two snapshots of figure II-20 between the current second and third snapshots. These two extra ones could be determined by rotating D only part way to the termination point of the rotation and then starting the stability testing procedure over again.

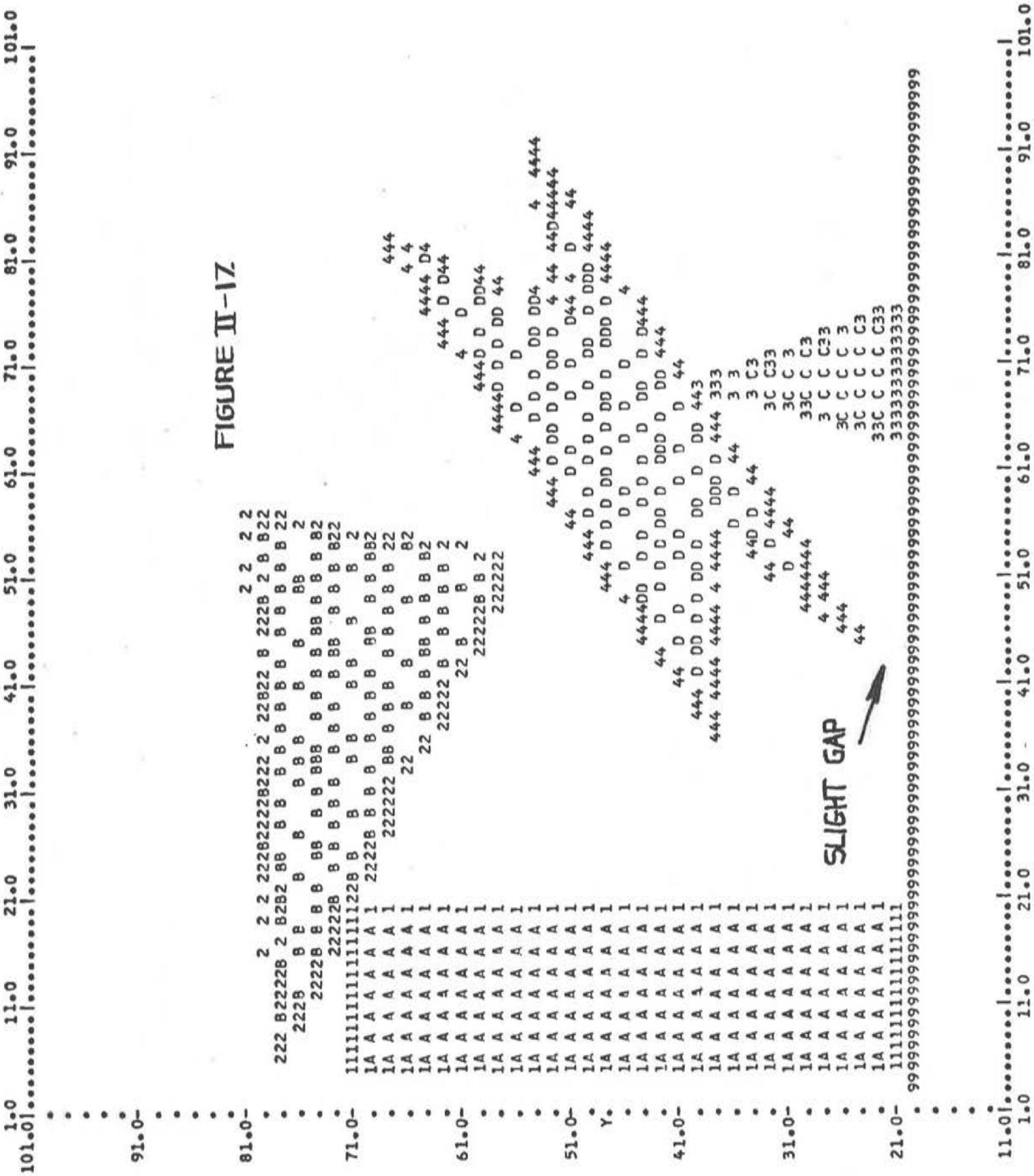
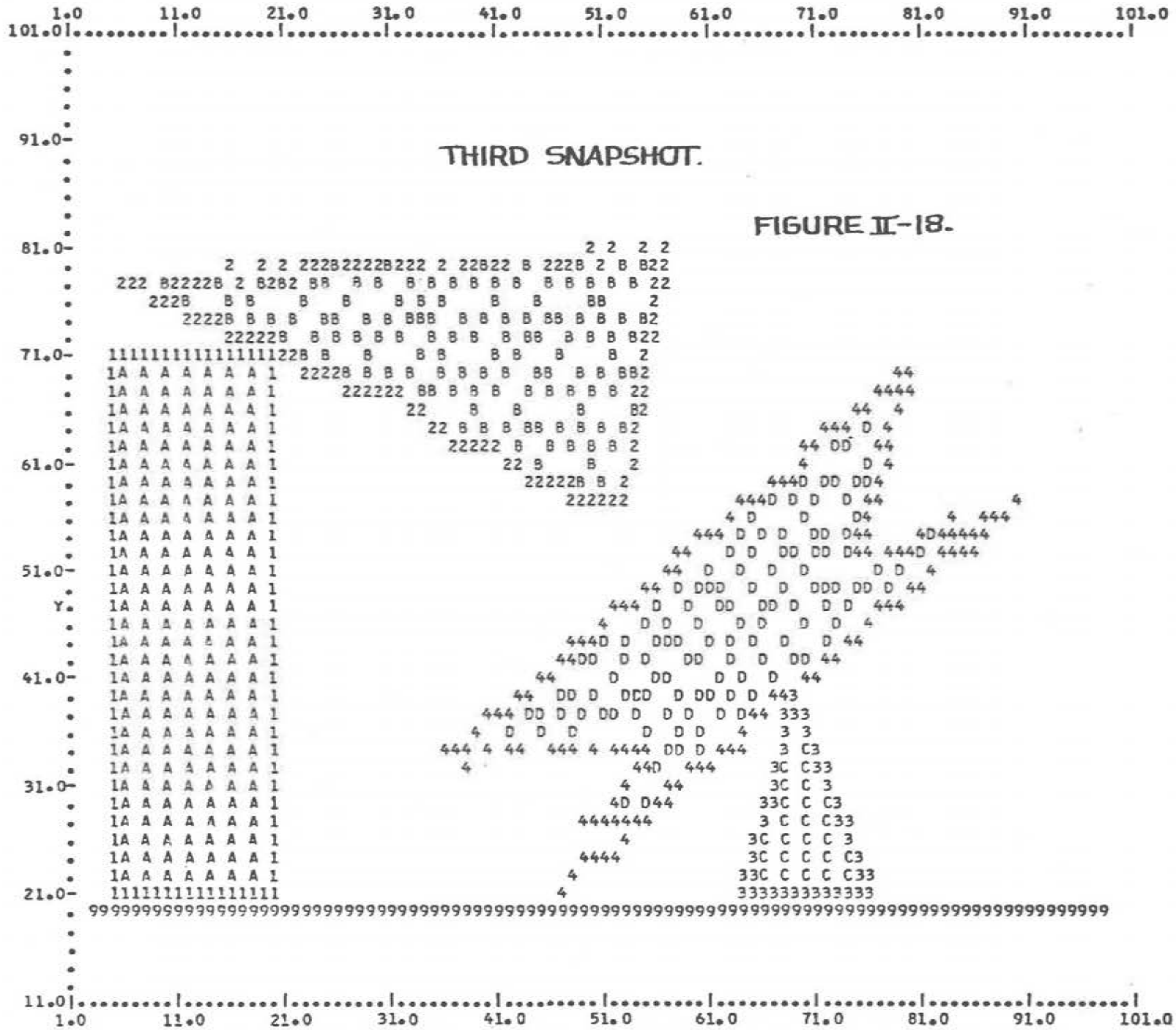
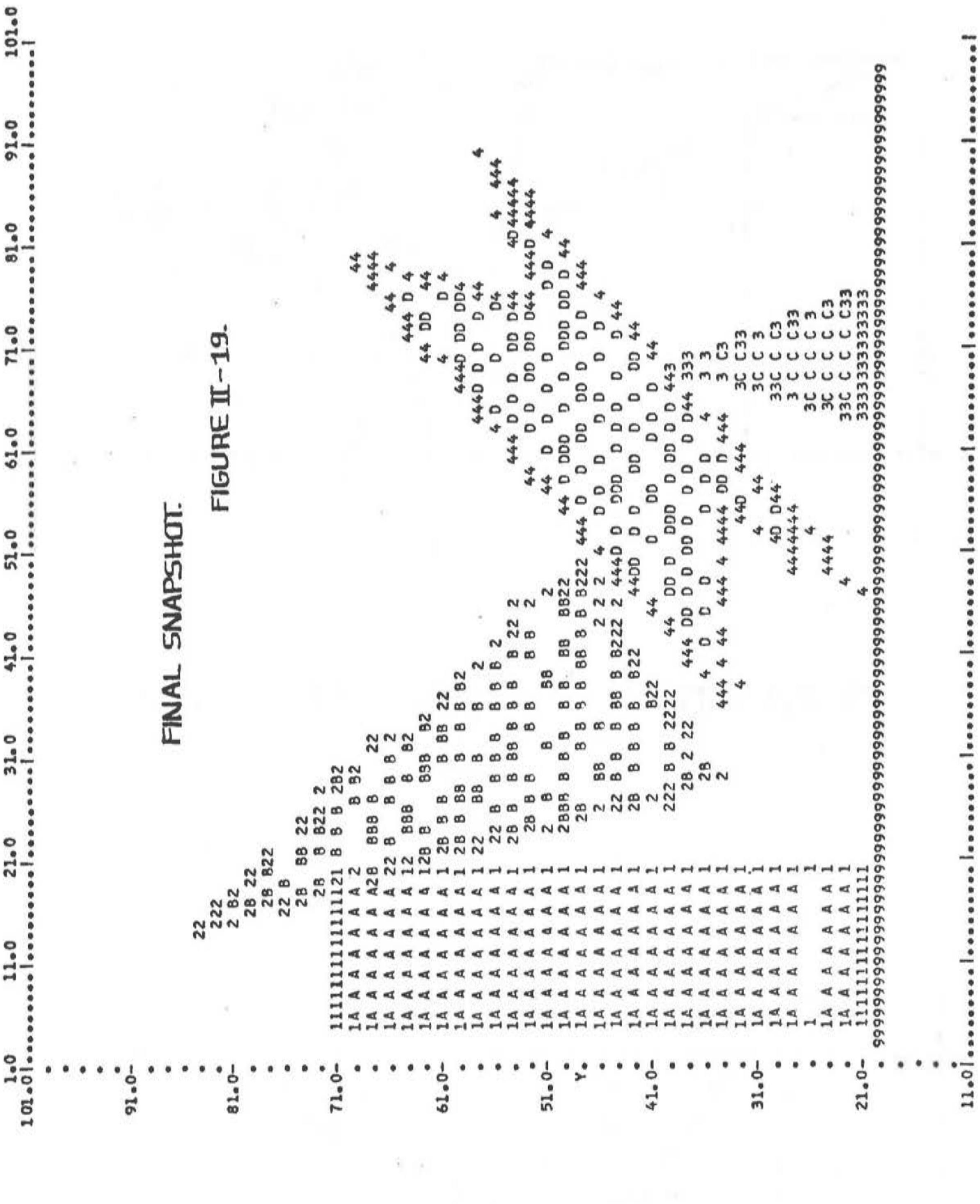


FIGURE II-17



THIRD SNAPSHOT.

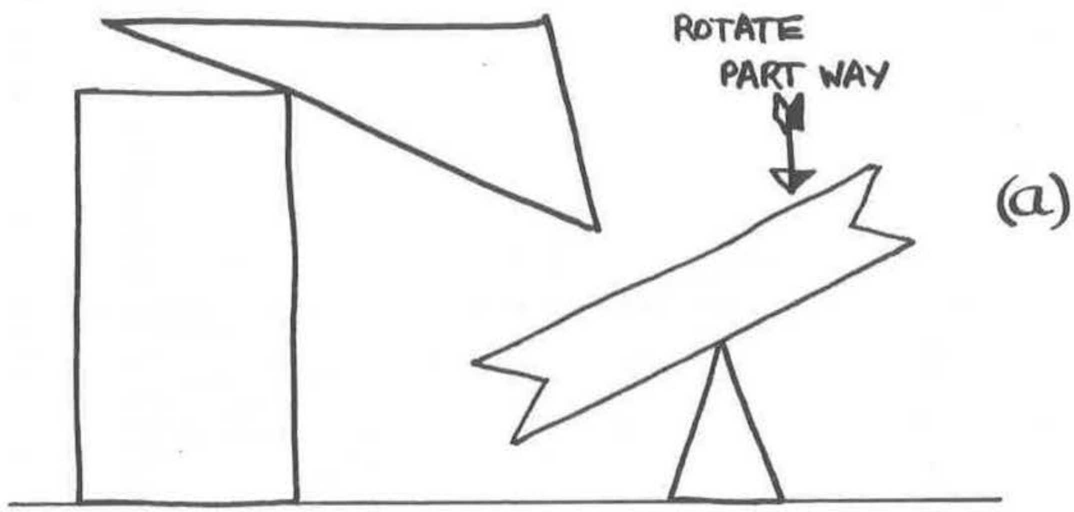
FIGURE II-18.



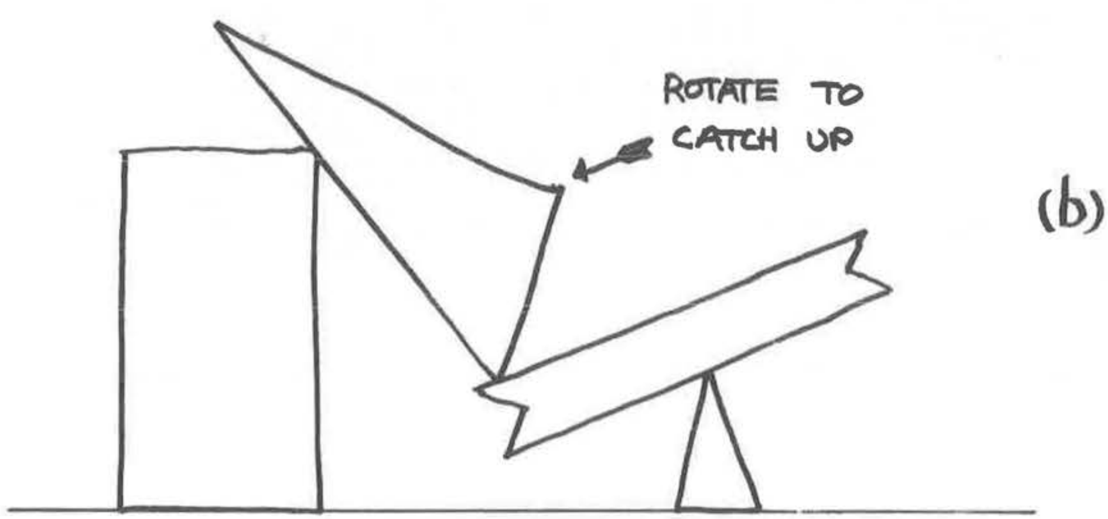
FINAL SNAPSHOT.

FIGURE II-19.

FIGURE II-20



INTERMEDIATE SNAPSHOTS.



II-8 Translational Stability

The chain reaction problem did not involve any translationally unstable objects. The example of figure II-21 will be used to illustrate how WHISPER determines whether an object will slide, and if it slides, how far it will travel. As described earlier (section II-3.1), the stability of a complete structure is tested by subdividing it into conglomerate objects whose stability is tested separately. The translational stability of these conglomerate objects is tested at the same time as their rotational stability.

WHISPER decides on translational stability of an object by examining its contacts. There are three types of contact that are considered: surface-to-surface, surface-to-point, and point-to-surface. The stability criterion for a particular contact is whether or not the tangent to the surface involved in the contact is horizontal at the point of contact. (Tangent finding is another perceptual primitive.) If the tangent is not horizontal, then the direction of downward tilt is taken as the resultant direction of motion of the object. If a conflict in the direction arises, one contact indicating leftward motion and another indicating rightward motion, then WHISPER reports its inability to decide on what the motion will be. This illustrates the need arising in some situations for a quantitative investigation in order to resolve the qualitative ambiguity. (Resolving qualitative ambiguities by quantitative

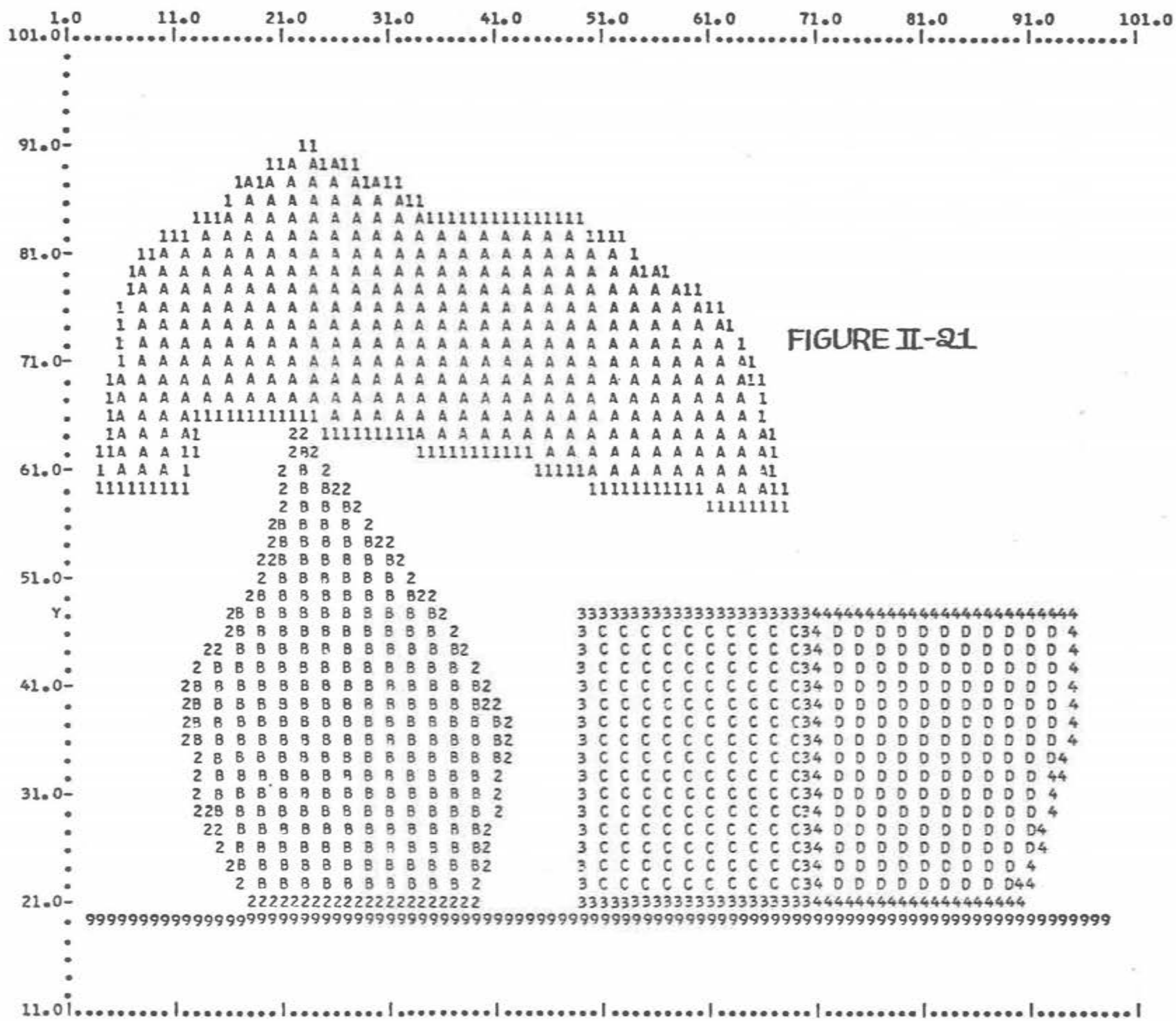

1.0
11.0
21.0
31.0
41.0
51.0
61.0
71.0
81.0
91.0
101.0
101.0
91.0
81.0
71.0
61.0
51.0
Y.
41.0
31.0
21.0
11.0

FIGURE II-21

reasoning is discussed by DeKleer). There is, of course, no conflict between a horizontal contact slope and a non-horizontal contact slope, the former simply does not contribute to the motion. In the example of figure II-21 object A rotates until it hits object C, figure II-22, just as in the previous example. At this point the eye is moved separately to each of the contacts. The surface-to-point contact between A and B is noted as is the rightward tilt of the surface of A at the contact, and the point-to-surface contact between A and C with the horizontal slope of C at the contact. The A-to-B contact is classified as surface-to-point, with the rightward tilt of the surface of A at the contact noted as contributing to a rightward motion for A. Similarly, the A-to-C contact is classified as point-to-surface with no contribution to the motion of A because of the horizontal nature of the slope of C at the contact point. Thus WHISPER concludes that A will slide to the right along the surface of B.

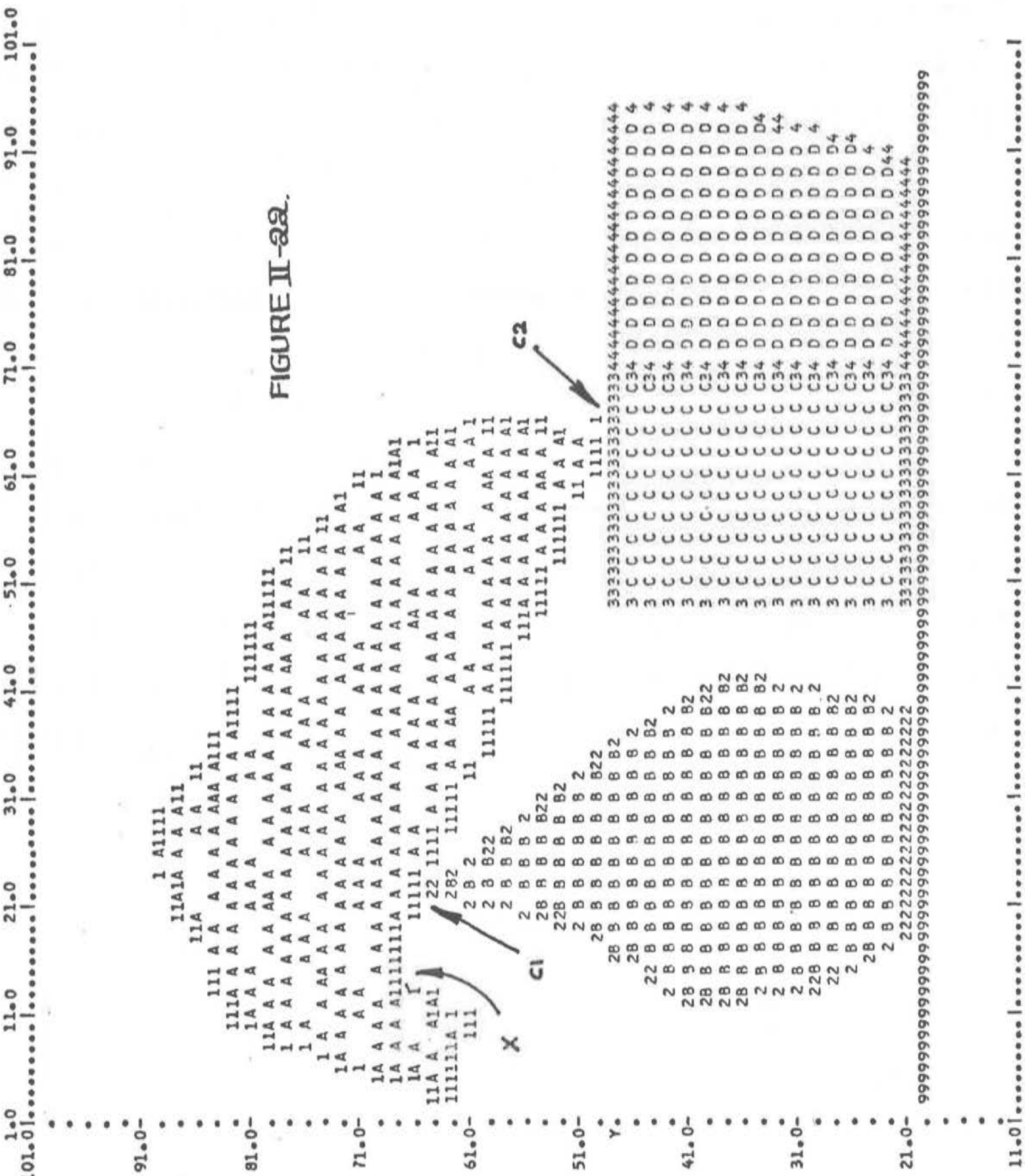


FIGURE II-22.

II-9 Sliding An Object Along An Irregular Surface

An object's sliding motion, unlike a rotational motion, cannot be visualized on the retina, because the object's path is not easily found. WHISPER's method of performing slides is outlined in figure II-23 . Instead of visualizing sliding motions, WHISPER examines the contacting surfaces of the object and its supporters for a circumstance causing the object to stop. The stopping point is called an *interruption point.* To update the diagram the object is translated to the nearest interruption point. After the translation, a rotation generally is required to make the surfaces touch at all their proper locations. To find the correct orientation of the object, its rotation is visualized to see the angle at which the right support relationships between the surfaces occur.

II-9.1 Surface Examination

There are two elements to the examination of the sliding surfaces for termination points. Exactly what portions of which surfaces are to be examined, and what features of the surfaces are important? The basic objective is both to examine those surfaces on the moving object which will slide past a point on a stationary object, and those on a stationary object which will have a point of the moving object ride over them. Thus in the current example (figure II-22) the surface of A will be examined from C1 to the left, and the surface of C and

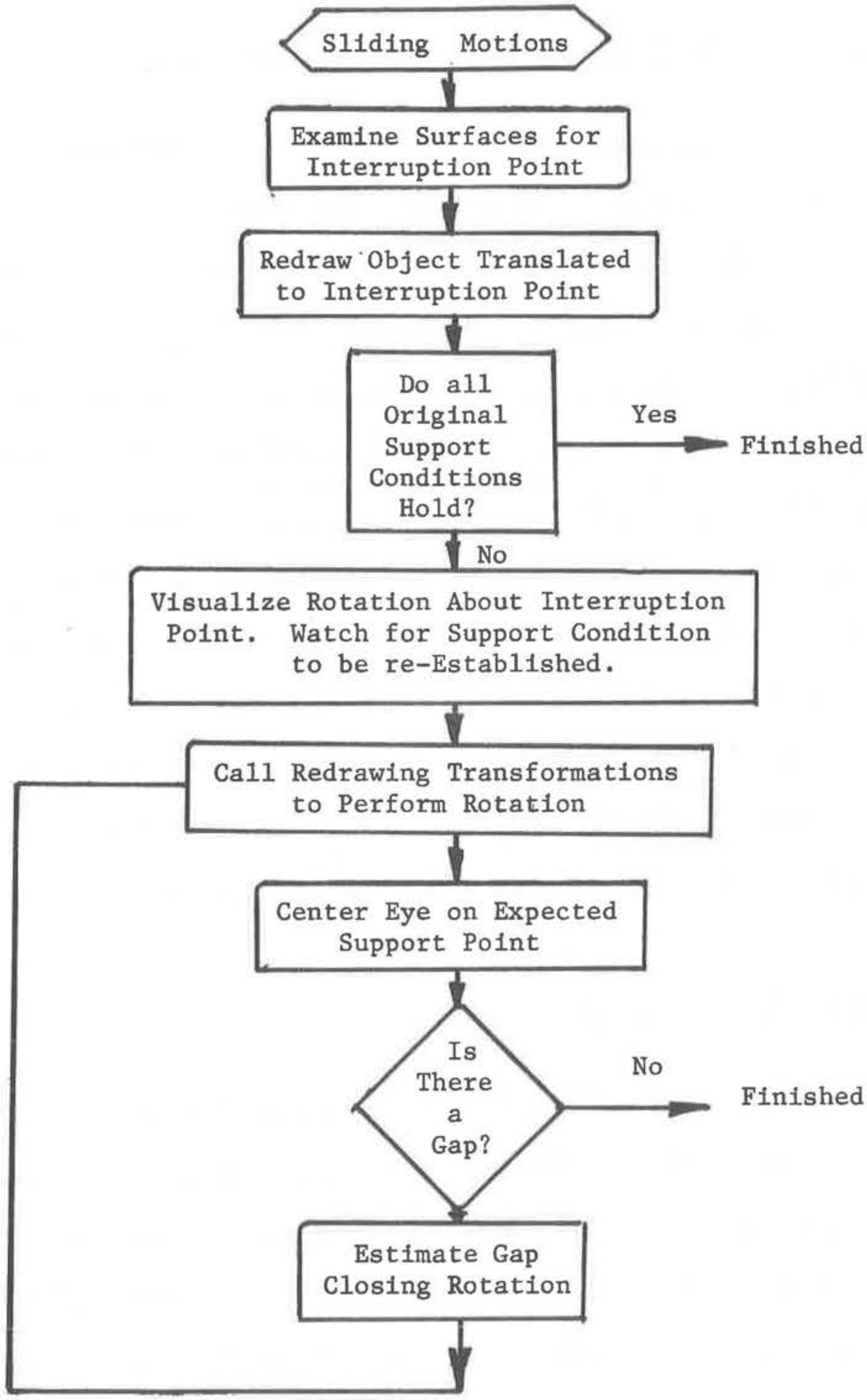

Sliding Motions
Examine Surfaces for Interruption Point
Redraw Object Translated to Interruption Point
Do all Original Support Conditions Hold?
Yes
Finished
No
Visualize Rotation About Interruption Point. Watch for Support Condition to be re-Established.
Call Redrawing Transformations to Perform Rotation
Center Eye on Expected Support Point
Is There a Gap?
No
Finished
Estimate Gap Closing Rotation

FIGURE II-23.

possibly D will be examined from C2 to the right. Surface-to-surface contacts involve the examination of both the contacting surfaces in this manner. It is only the first occurrence of a termination condition that is of interest, since it is the one which will stop the object's sliding motion. Thus WHISPER can constrain its examination of surfaces subsequent to the discovery of one termination condition to only that portion of them which would cause a prior termination condition to arise.

The second element - what is to be looked for while examining a surface - is dependent upon whether the surface under consideration is an upper or a lower surface. For example, it is necessary to look for objects sitting on a lower surface with which the sliding object might collide, whereas this is not necessary for an upper surface. The other conditions which could be relevant to the object's motion are: a sharp bend in a surface; a hill which is higher than the object's intitial position; reaching the end of a surface in the direction of motion; and as mentioned, an object on the surface or close enough to the surface that a collision between it and the moving object would occur. These conditions are illustrated in figure II-24. Currently, WHISPER examines surfaces for any of these conditions in one fixation by centering the eye on the relevant starting contact point. The detectors for these conditions are built in at the level of perceptual primitives (section III-3). However, because of

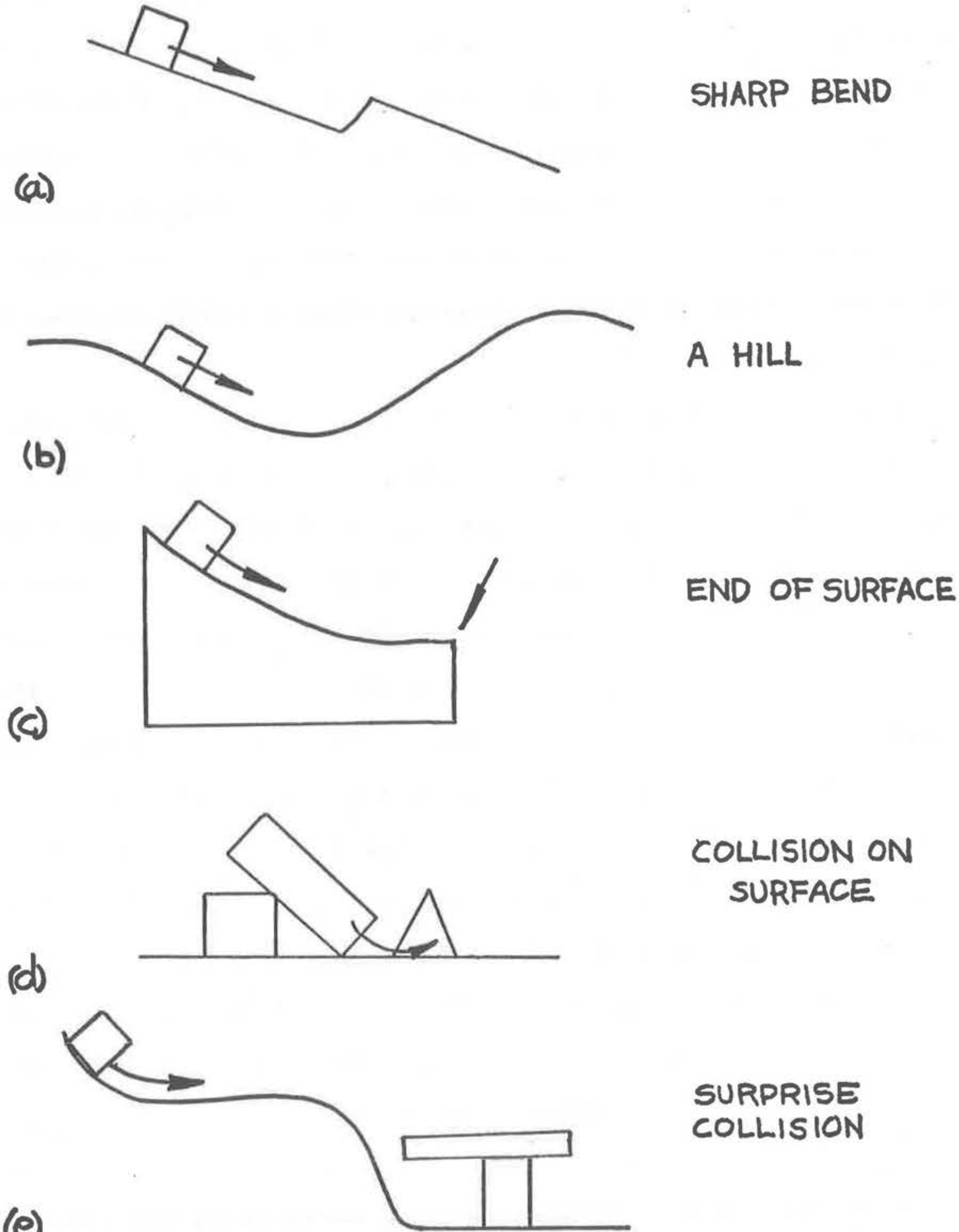

(a)
SHARP BEND
(b)
A HILL
(c)
END OF SURFACE
(d)
COLLISION ON SURFACE
(e)
SURPRISE COLLISION

FIGURE II-24.

their specialized nature for this problem domain, I would hesitate to call any but the sharp bend and collision detectors truly primitive. It would be possible and desirable in terms of accuracy to use the current detectors to propose further fixaticn locations to which the eye could be moved to test more precisely for the fulfillment of a particular condition.

WHISPER must make multiple fixations along the surface if it is to detect the 'surprise' collision of figure II-24(e). Although it does not currently handle this case, it is clear how it easily cculd by fixating the retina at regular intervals along the supporting surface. This is illustrated by figure II-25 in which an x indicates a fixation point, a semi-circle indicates the area of the diagram to be checked by the retina at each fixation (checking a circular region is easy because of the retina's ring structure), and the space between dashed line and the surface indicates a clear 'corridor' for object. The radius of the semi-circle is a function of the object's size and the fixation interval. The same sized corridor can be examined with fewer fixations by using a larger radius. The only disadvantage is that the probability of false alarms increases, because the distance between the dashed line and the circumference of the semi-circles is greater. False alarms can be handled by making more fixations in the region where they occur. This method of detecting collisions is very good for two reasons: (i) because the retina can check large segments of space in a single glance, the number of fixations

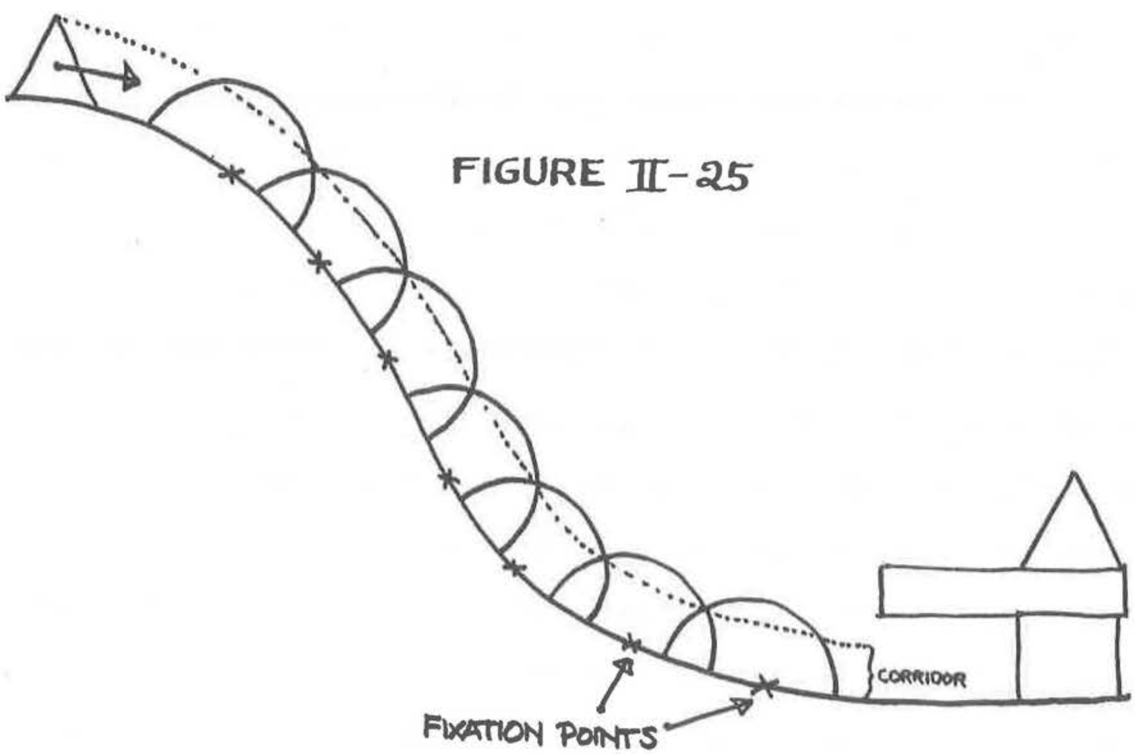


FIGURE II-25
CORRIDOR
FIXATION POINTS

required to examine the space near the surface is relatively small; (ii) a collision will never be missed.

II-9.2 Advantages Of The Analogue In Analyzing Slides

There are several respects in which WHISPER's analysis of the termination conditions for sliding motions is simplified by the analogue. One is that there is no need for the application of numerical methods in finding the curve features. The features are found by inspection of the diagram. Of course, there is computation involved in this process, but it is a comparatively small amount because WHISPER is working with the curve itself rather than an equational description of the curve. Additionally, such a descriptive equation often is not available, and may itself have to be computed through a curve-fitting process.

Another important respect in which WHISPER's slide analysis is simplified is in detecting coincidental alignments of two or more objects whose surfaces form one continuous curve over which an object could slide; such coincidences also result in smooth curves in the diagram. Notice that in figure II-22 object A will slide along the surface of both C and D. The surface examination testing should thus be carried out along only the upper edges of C and D. The fact that C and D together form a smooth surface is a property which emerges from the coincidence that they have the same height and that they are touching. It would be very difficult for a system relying on separate encodings of the shapes of objects to discover this emergent property. To find it would first of all require the built in expectation that it might happen. Then its existence would have to be continually checked. This check would involve establishing all the contact relationships of the object on which the sliding object is initially resting with all the other objects in the universe, already a difficult problem, followed by the amalgamation of the descriptions of the two separate curve descriptions into a new curve description. Establishing the contact relations of the supporting object cannot be simplified by asserting them as part of the initial problem description, because it might have rolled or slid into its current location. In addition, not only must touching objects be considered, but also objects which are almost touching, since a small gap may not inhibit the sliding motion of an object.

It is unnecessary for WHISPER to concern itself with questions of how contours and other properties might have combined to produce a smooth curve. This is another instance of the amalgamation problem discussed in section IV-4.1.3. Since a smooth curve in the real world is modeled by a smooth curve in the analogue, WHISPER requires only a recognizer of smooth curves. The amalgamation of contour descriptions is solved by simply ignoring the distinguishing colouring of all objects except the sliding one. The touching faces of C and D thus become inside points of the combined CD, and hence are not seen as part of the contour. The resulting contour and its shape arise directly in the diagram, not as the net product of a complex chain of deductive or computational inferences about the properties of the independent objects.

II-10 Updating The Diagram To Reflect A Slide

Using feedback from the diagram, WHISPER can avoid having to establish the exact locus of motion of a sliding object. The locus followed by each contact is known to be the same as the surfaces over which it slides which is why those surfaces were examined; nonetheless, the motion of the object itself is a composite of the loci of the surfaces involved. It is of substantial benefit to WHISPER that it can avoid the calculation of this composite. The termination point of the slide has already been found; what remains is to move the object to that point. The first stage is simply to translate the object so that the specified point on it is aligned with the specified point on the contact surface. This is shown in the change from figure II-22 to figure II-26 in which point X is aligned with C1. This translation is accomplished by simple matrix multiplication much the same as for rotations. After this translation, WHISPER checks that the contact relationships which existed before the translation still exist. This examination in the example of figure II-26 reveals that the contact, C2, existing in figure II-22 has changed. A rotation about the termination point will correct the problem. With the eye centered on this pivot point a rotation is visualized until the contact is re-established. The amount of this rotation is used to perform a rotation in the diagram (figure II-27). As with the purely rotational examples, another gap closing

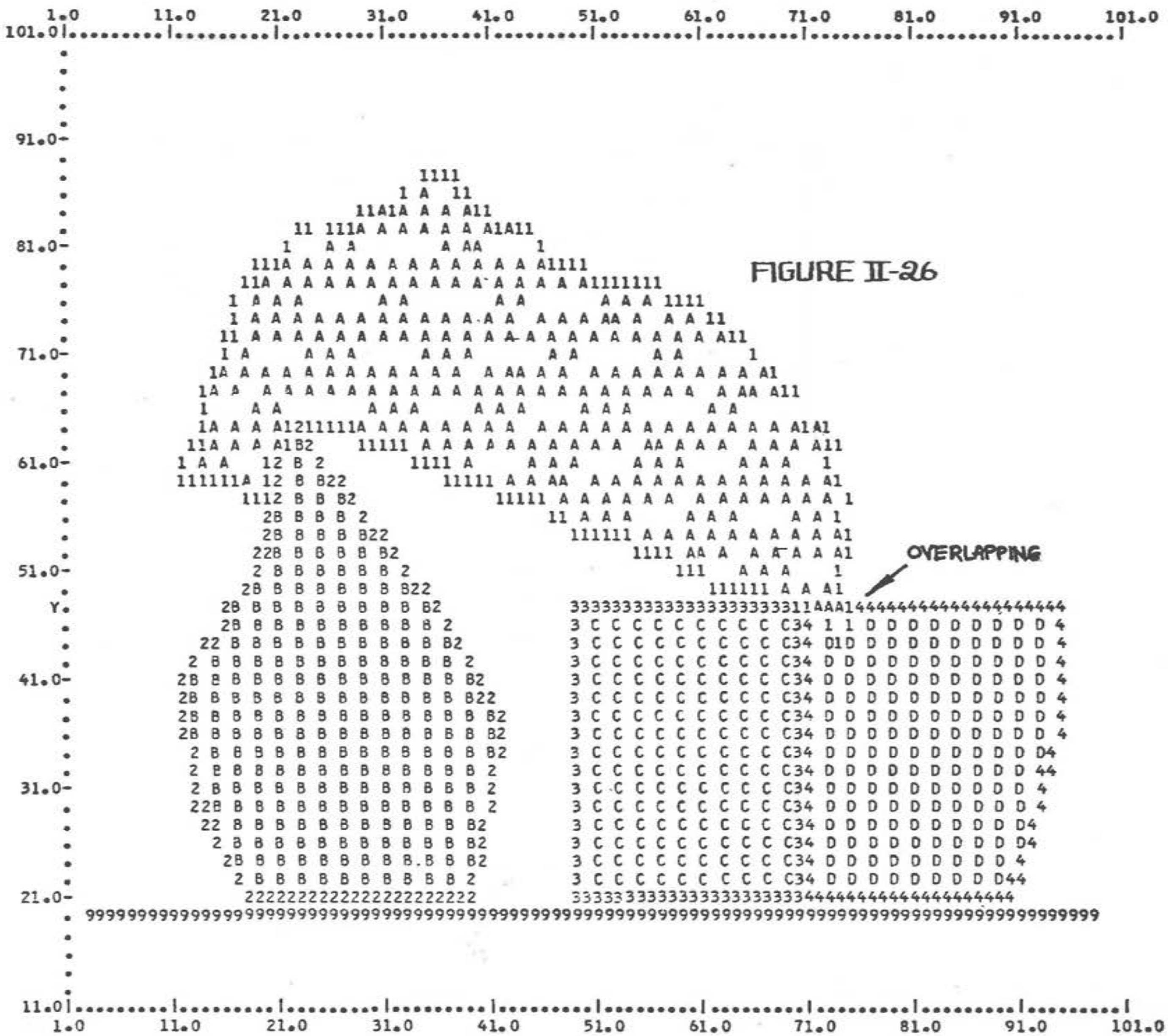


FIGURE II-26

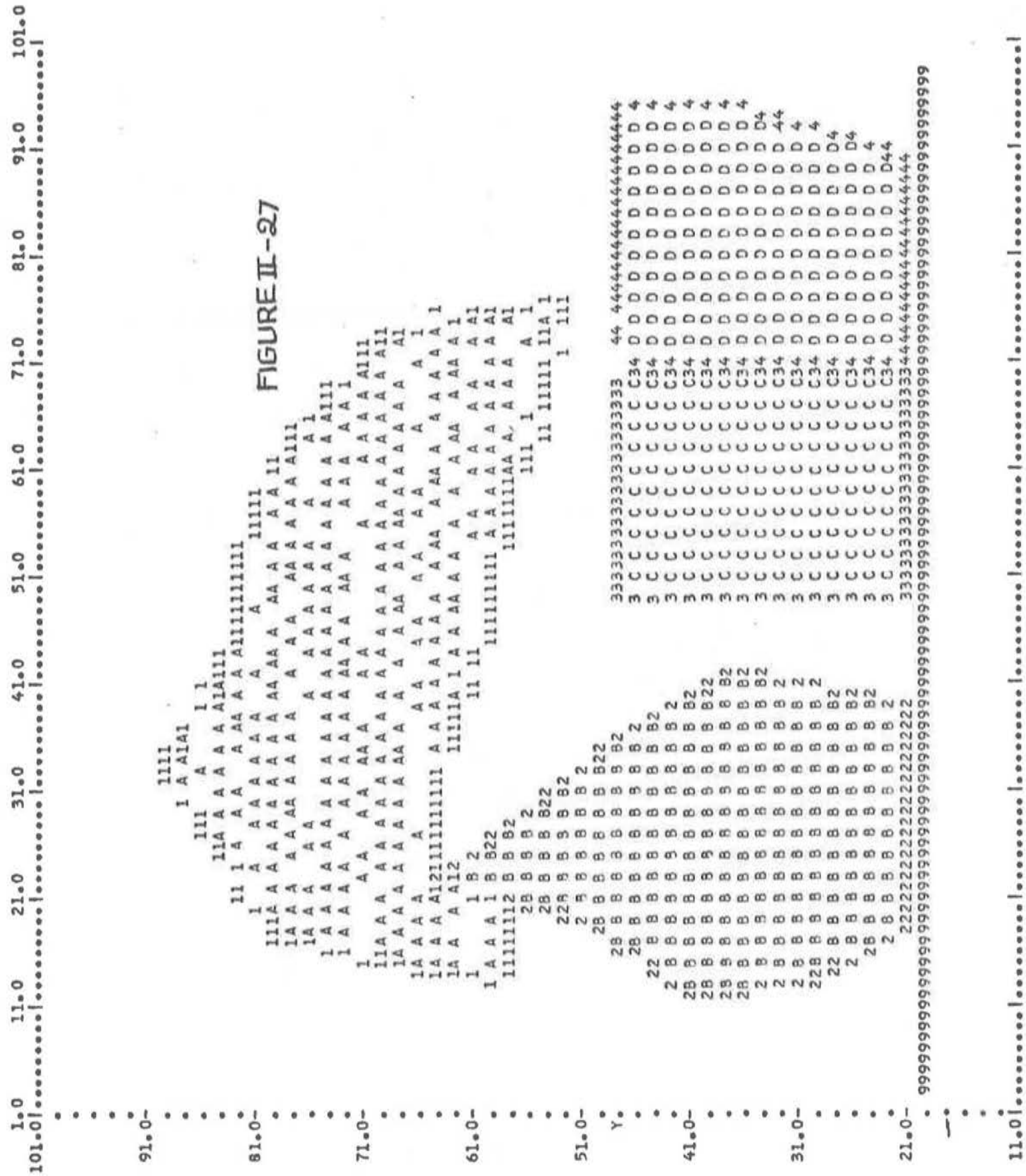


FIGURE II-27

rotation may be required (figure II-28) because of the approximate nature of visualization.

It should be noted that the above two-step method - translation to align the object with its interruption point followed by a correcting rotation - works for curved as well as straight surfaces (figure II-29). This conceptually simple approach, incorporating experimental feedback from the analogue, is a very natural form of qualitative reasoning embodying a first order theory of the motion of sliding objects.

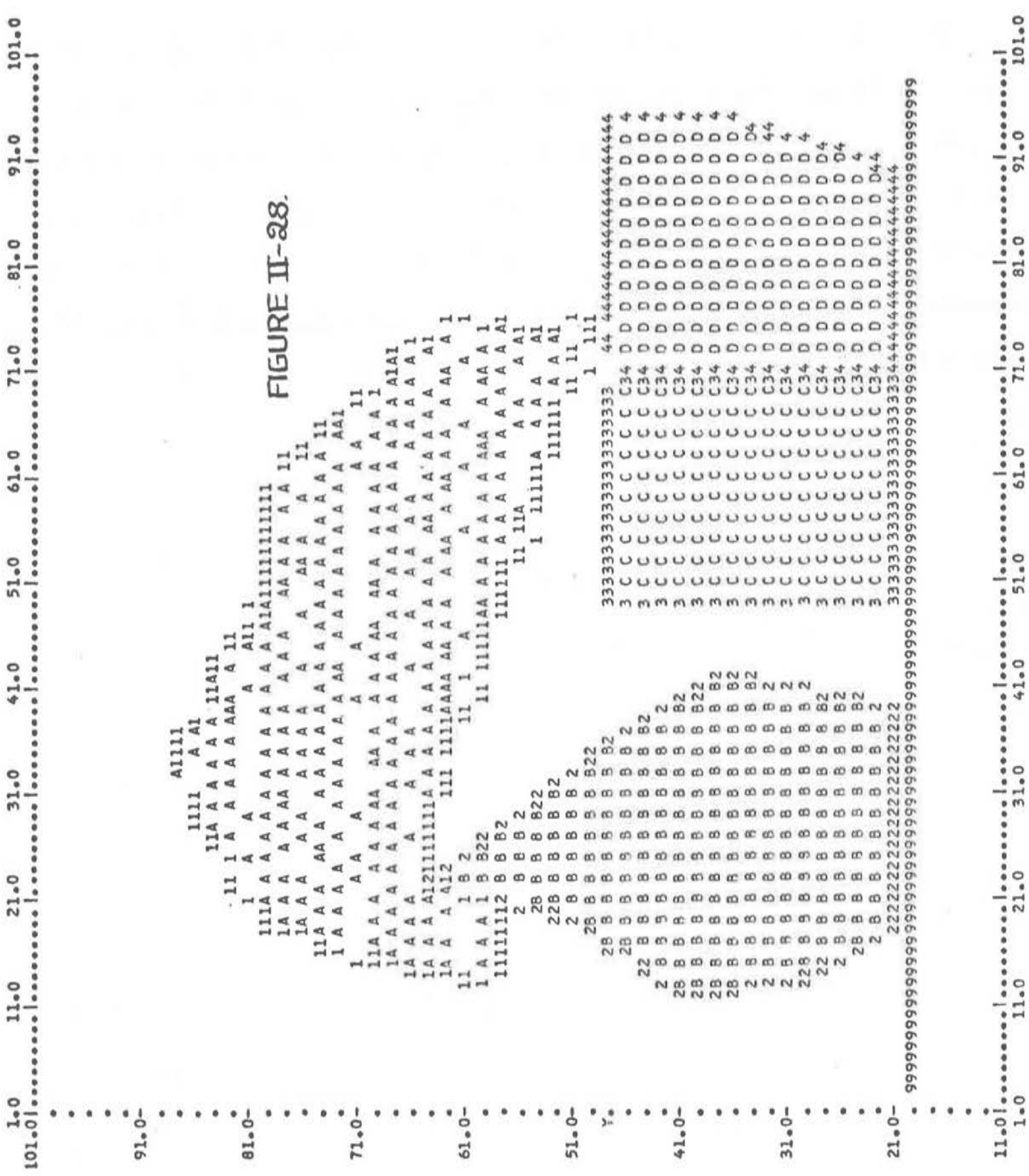


FIGURE II-28.

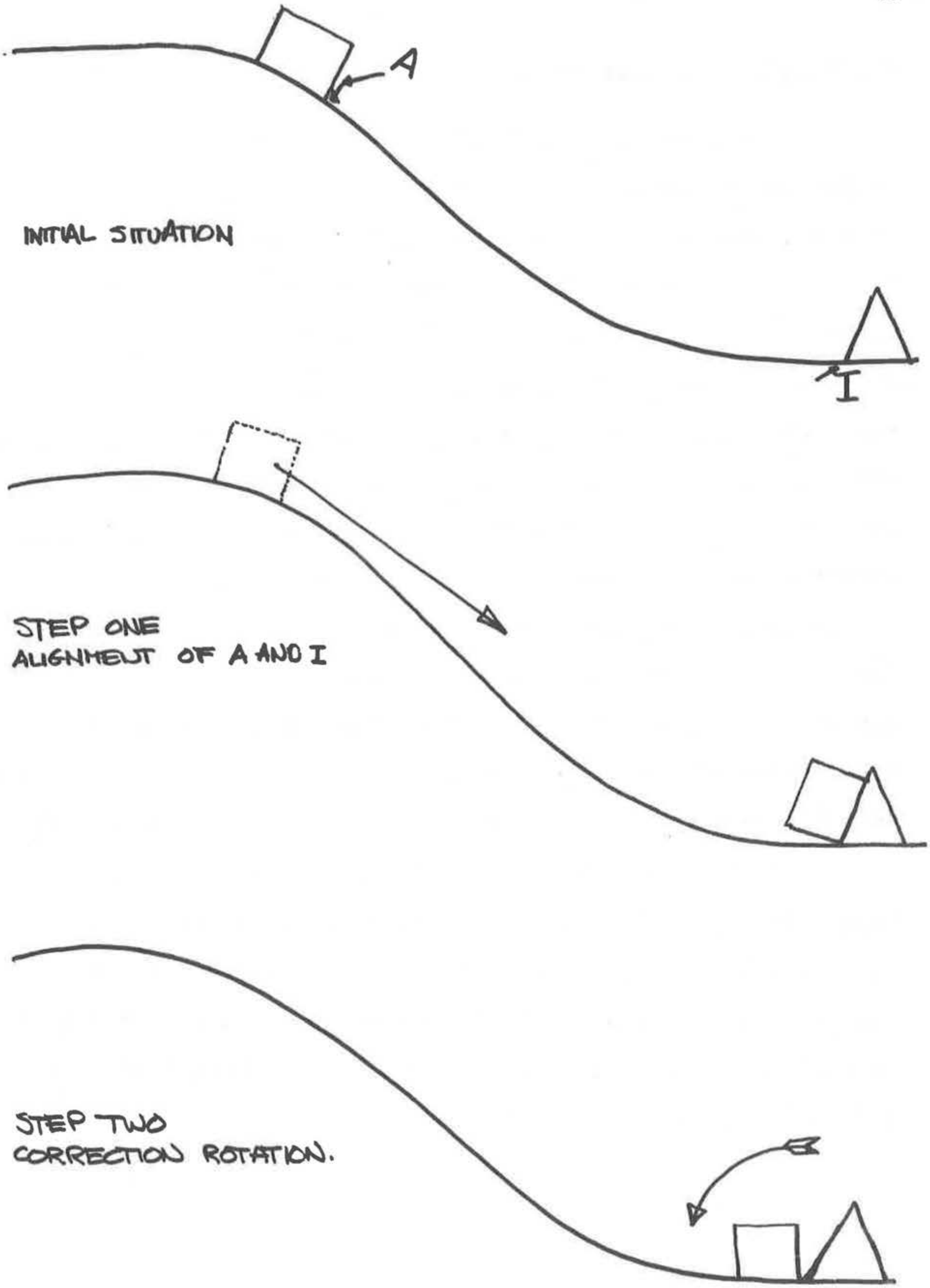

A
INITIAL SITUATION
I
STEP ONE
ALIGNMENT OF A AND I
STEP TWO
CORRECTION ROTATION.

FIGURE II-29.

II-11 Summary Of Qualitative Knowledge

In concluding this section on qualitative reasoning I would like to summarize a few points. Taken together, the diagram, and the set of transformations that are applied to it, is an analogue of the real world objects. Future motions of these objects under the force of gravity are predicted by WHISPER. Connection with the analogue is maintained via continued interaction using the eye and its perceptual primitives, and via the transformation procedures. Problems are solved by interaction of the procedurally encoded qualitative knowledge of blocks world physics with the diagrammatic analogue. This interaction is through experiments directed by the qualitative knowledge, performed in the analogue, and accessed by the simulated retina. When an experiment, such as the rotation of an object, is complete, WHISPER only needs to 'look' at the resulting diagram to determine the new state of the world. _It is because WHISPER need only be adept at directing experiments and interpreting their results rather than at predicting their outcomes that, instead of becoming bogged down in combinatorially exploding computations, it is able to obtain conceptually simple solutions to simple, though non-trivial, problems._

Chapter III: The Retina And Its Primitive Percepts

III-1 Introduction

The eye is the primary connection between WHISPER and its diagrammatic analogues. The perceptual primitives provide answers to the basic questions that WHISPER can ask of the eye. These questions concern topological features of configurations in 2-space, and are independent of any interpretation that the diagram might have as an analogue. Interpretations are assigned to the topological features perceived by WHISPER in accordance with the analogy that exists between the diagram and the particular domain of interest. The perceptual primitives thus constitute a fixed set of operators, applicable to a variety of different domains.

The significance of the eye and its perceptual primitives for the utility of analogues is that they provide a new set of conceptual primitives. The world is divisible, rather arbitrarily, into many different conceptual categories; the choice of a conceptual segmentation is primarily a function of the available conceptual primitives. High level programming languages provide an example of the influence of different sets of conceptual primitives. Some may quarrel, but it seems reasonable to suggest that there have been programs written in, say CONNIVER or PLANNER, which would never have been written in, say, FORTRAN, (let implementations of CONNIVER in FORTRAN be ruled out) even though there is no theoretical reason why they could not have been. The conceptual complexity of a program is related to the interweaving of the conceptual primitives in its construction. If the conceptual primitives are less powerful, that is to say less suited to the problem at hand, then the resulting conceptual complexity of the solution will be greater. In providing WHISPER with an eye and associated perceptual primitives I have endeavoured to expand the available set of conceptual primitives to include some which are tailored to spatial problems. A new set of machine instructions or a new language is established.

There is an impressive physical and computational structure imposed by the senses on human conceptual primitives. Our visual perceptual understandings would be very different if we were endowed with x-ray vision, for example, or if we had a third eye in the back of our heads, or any of the other multitudinous possibilities. Expressing such examples of expanded conceptualization in terms of our familiar percepts would probably be as difficult as expressing the concept of colour to a blind man. Communicating shapes instead of colours would not be so difficult if, rather than stating facts about the geometry or topology of the shapes, they are simply manipulated and understood by touch. The argument that the structure imposed by our senses influences our thinking about the world is not new. I raise it here merely to highlight the necessity for an enlarged set of conceptual primitives in AI systems and to justify the simulation of some of the computational and structural properties of the human eye as a sensible approach tc this expansion.

III-2 The Retinal Simulation

WHISPER's eye is based on the simulation of the structural and computational aspects of the human retina's most pronounced features, modified and guided by pragmatic considerations of computational expense on the available hardware, and ease of implementation both of the simulation itself and of the perceptual primitives dependent on it. It is not to be interpreted as a model of the operation of the human retina or the early perceptual processing stages. In order to establish the computational feasibility of utilizing diagrammatic analogues, a computationally feasible implementation of the perceptual primitives which examine them must be provided. A simulation of some aspects of the human eye, especially its parallel operation, provides the framework in which these primitives can be implemented.

III-2.1 Retinal Geometry - The Periphery

The overall structure of the periphery of the retina is shown in figure III-1. Each 'circle' represents one 'receptor' processor, and is called a bubble. Since the diameter of the bubbles increases with increasing distance from the center of the retina, the acuity is a decreasing function of this distance. The size of the bubble is dependent on the function which maps the array values of the diagram onto the retina. This function is equivalent to the process of overlaying the

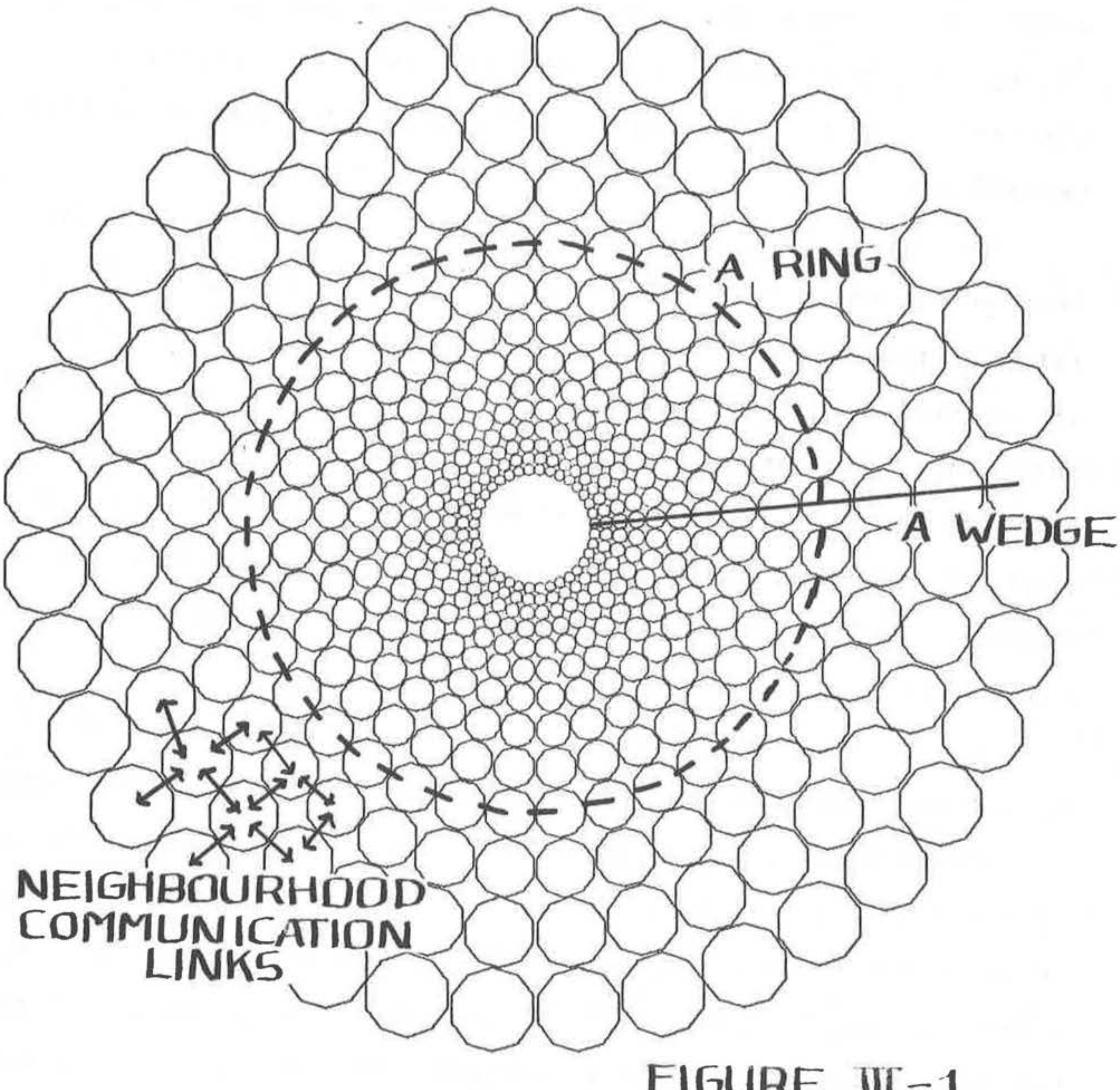


FIGURE III-1

bubbles of figure III-1 on the diagram which the eye is to look at, and shading in each bubble with the colour under it as in figure III-2. A bubble can be assigned multiple values if it covers more than one colour. (Only single values would be required if there were more bubbles than can currently be afforded due to the expense of the pseudo-parallel mode of operation.)

The retina is a circular array of bubbles, each bubble consisting of a processor, some memory, and addressable in terms of its wedge and ring coordinates. A wedge is one line of bubbles radiating outwards from the center, as shown by the solid line in figure III-1; a ring is one circle of bubbles equidistant from the center, as shown by the dashed line. The rigid alignment of the bubbles was chosen so the location of a bubble center relative to the retinal center could be calculated from its wedge and ring coordinates.

It is important to realize that the retina is not confined to a single position over the diagram, but is free to move and be refilled with a fresh view of the diagram under commands from a higher level process. This refilling process is assumed to be accomplished in parallel, all the bubbles receiving new values simultaneously. All the receptors on the human retina obviously receive new inputs in unison after a saccade to a new fixation point.

In addition to the addressability of the bubbles as an array, each bubble has direct pointers to each of its four

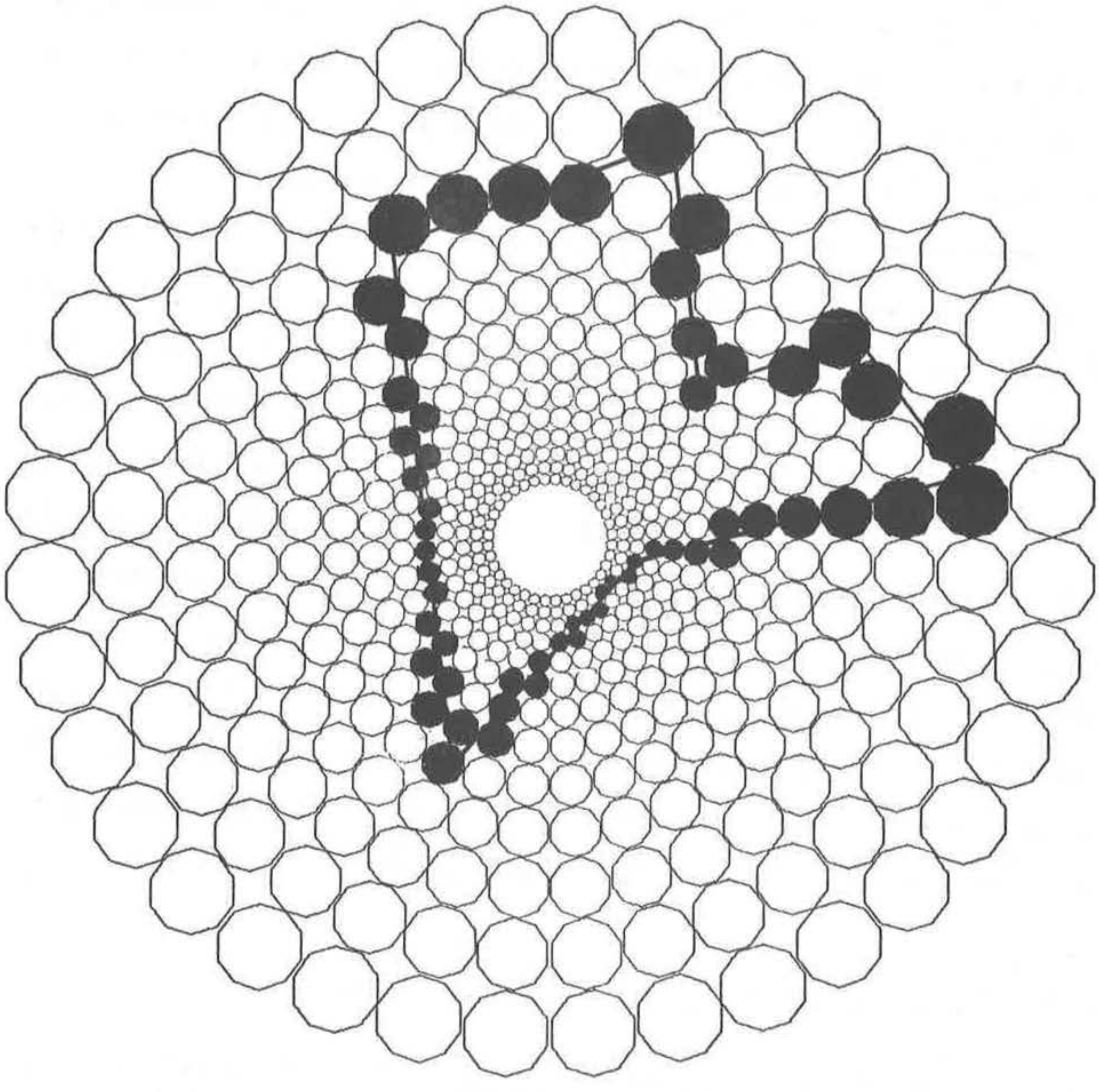


FIGURE III-2

nearest neighbours, the two in the same ring and the two in the same wedge, as depicted by the arrows in figure III-1. These pointers are software equivalents of what would be direct communication links between processors in a hardware implementation. In other words the retina is a circular array consisting of bubbles, each bubble a list composed of a two level stack of current and previous values, pointers to its four nearest neighbours, and its own coordinates.

Each wedge and ring is addressable as a list of bubbles as a result of the linkages from each bubble to its nearest neighbours. These lists facilitate the use of the LISP mapping functions which apply a single LISP function uniformly to each list element. The uniform application of a single function to all the bubbles is a form of pseudo-parallel processing. As long as the applied function has no side effects, then there is no time or order dependence between its invocations. Thus if multiple processors are available then all the separate invocations can be executed simultaneously. It is in this sense that a processor is considered to be associated with every bubble, and that all these processors are identical.

III-2.2 Retinal Geometry - The Retina's Center

In the central area of the retina the diameter of the bubbles becomes less than the width of the squares of the array grid in which the diagram is encoded. Thus, paradoxically, in contrast to the human eye where the central section, the fovea, is very important, the central area of the simulated retina becomes in many respects less interesting than the periphery because its resolution begins tc exceed the resolution of the diagram it is viewing. This problem stems from the poor resolution of the diagram rather than the construction of the retina, and could be solved simply by increasing the size of the diagram grid. It would not be bothersome except that the current simulation operates only in pseudo-parallel mode so the extra bubbles in the central area must be paid for in terms of increased total computation time. In the interest of economy the central region was separated from the periphery allowing the two sections to be moved and refilled individually. It was possible to implement many of the perceptual primitives using the processors from only one of the two retinal areas, reducing the amount of computation required to simulate the retina's parallel processing.

Since the retina's central area operates independently, it does not continually slow down the simulation. Thus for uniformity, the retina's center has the same circular array structure as the periphery even though its resolution then exceeds the diagram's. The blank area in the middle of figure III-1 is covered with more bubbles in the same pattern as those on the periphery. The remaining small blank area in the absolute center of the retina is covered with a separate central bubble. It is likely that a different organization of the central bubbles would be required if the resolution of the array diagram (currently 101 by 101 units) were improved.

III-2.3 The Retina As A Data Structure

The simulated retina distinguishes itself as both a data structure and a computational structure. Consider first its properties as a data structure. The retinal array in conjunction with the diagram-to-retina mapping which fills the bubble value slots, possesses the varying acuity property of the human eye. This property is important in providing a focus of attention and a varying degree of concentration on detail. The current resolution of the eye, poor but workable, is provided by a total of 540 bubbles (15 rings by 36 wedges) on the periphery and another 540 on the central region. The complexity of the human eye is of a different order of magnitude with an approximate 116.5-131.5 million receptors (6.5 million cones and 110-125 million rods) on each retina, although the number of bubbles might more reasonably be compared with the one million fibers in the optic nerve.[13]

The mapping from the diagram to the retina is in effect a re-representation of the diagram in which some detail is blurred. The topological structure of the diagram is preserved in transferring to the retina. In particular, within the constraints of the resolution of the simulated retina, the bubble-fill mapping ensures that no object depicted in the diagram is missed on the retina. The choice of circular bubbles results in an equal degree of blurring in both the radial and circumferential directions. Other retinal designs were considered such as that of figure III-3 in which the radius of the rings increases as the tangent of their distance from the center, but the unequal spread in the two directions resulted in such a severe distortion of the diagram that it became almost unrecognizable.

III-2.4 The Retina's Computational Structure

Parallelism is the overriding characteristic of the retina when viewed as a computational structure. This is a characteristic shared with at least the initial processing stages of the human perceptual system. In the current implementation, each of the processors has been given the full power of the LISP evaluator. All of the bubbles are filled in parallel from the diagram and then the individual processors each simultaneously execute a common program. A supervisory sequential process, called the retinal supervisor, constructs the common program and initiates the parallel execution.

An important characteristic of the retinal parallelism is that the number of processors is fixed. There is no need for the introduction of new processes and the creation or hook-up of new processors as the computation proceeds, as Fahlman's system requires. Although it is possible to envision the

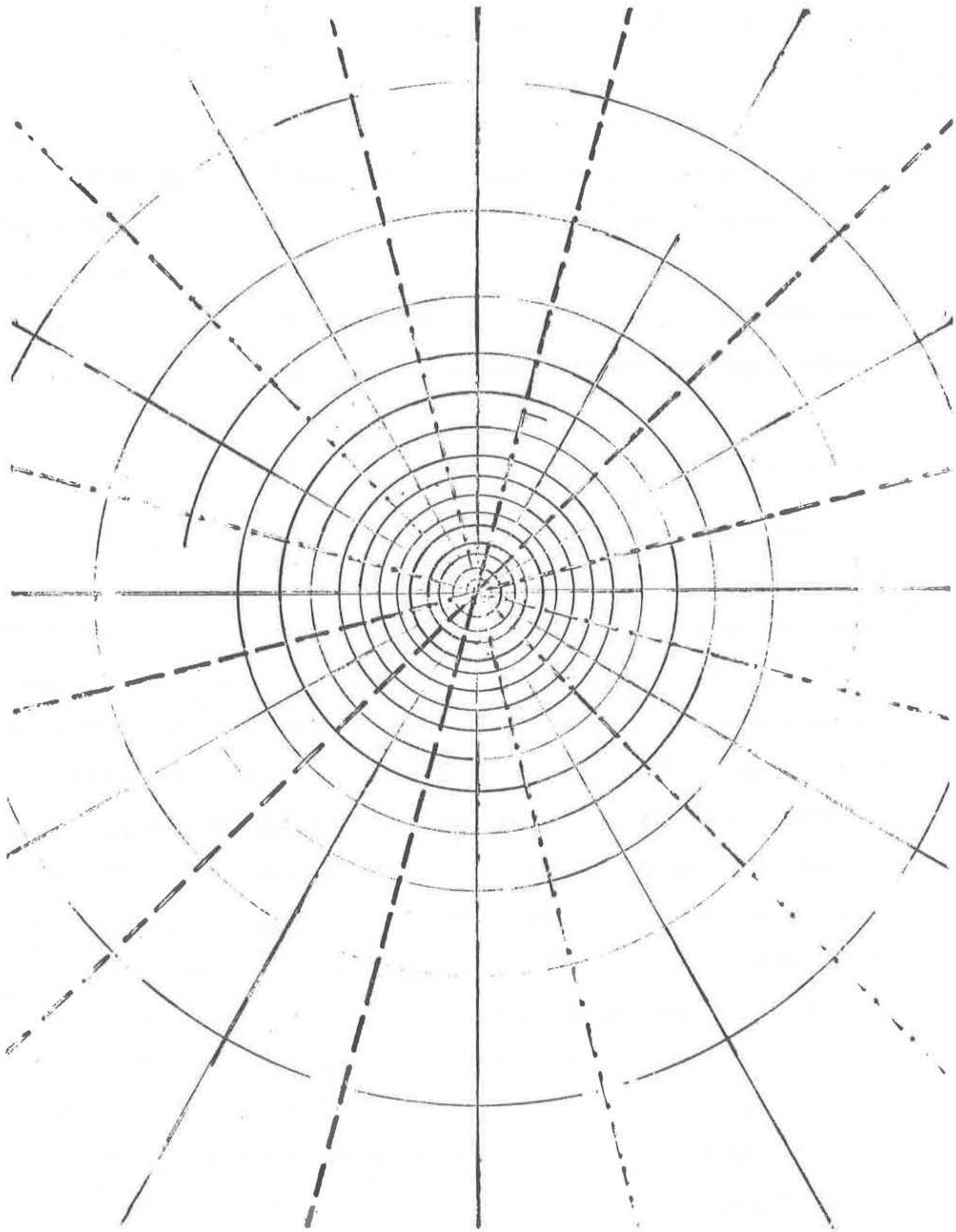


FIGURE III-3

growth of new processors, from a hardware standpoint it would be much less complicated to be able to construct a fixed number of processors in a predefined and fixed configuration.

The computational structure of the retina is also affected by the decision (again with a view to a feasible hardware implementation) that communication between processors be restricted to their nearest neighbours. The only exception is a link from each processor via a common data bus to the retinal supervisor. Because of the local spatial nature of the perceptually primitive operations (e.g. contact point finding), neighbourhood communication is all that is generally required.

The computational assumption of parallelism with neighbourhood communication is justified by the feasibility of implementing perceptual primitives. Without parallelism their computation would be grossly inefficient. The question of efficiency for the purposes of Artificial Intelligence is to be decided not on the total amount of computation involved, but on the total amount of elapsed time required. The reduction in total elapsed time which can be effected by using parallelism is proportional to the number of simultaneous processes. In the current situation this reduction is significant even though in many problem solving systems, where the computational requirements grow exponentially, it would not be. It is significant here because perceptual primitives can be incorporated in a new programming language as primitive operations with execution times of the same order of magnitude as for other more conventional language constructs. The number of processors required is fixed (in the current implementation there are 1080) but large. Perhaps as few as one million would yield the resolution of the human eye. Here, saving such a large constant factor is important.

III-2.5 Comparison With Perceptrons

The retinal structure and its use of parallelism is not the same as that of a perceptron (Minsky and Papert[14]); and the theoretical limitations of perceptrons do not directly apply. Minsky and Papert impose various restrictions on the devices they study to eliminate any aspects of sequential computation in order that a non-trivial theory of purely parallel computation and its limitations can be established. This is very different from the intent here which is to intermix parallel computation with sequential computation to as great an extent as possible in an attempt to increase the computational feasibility of efficiently computing the perceptual primitives. The main differences between retinal and perceptron computation are:

(a) The retinal supervisor, which plays an analogous role to that of the perceptron's linear threshold function, can perform arbitrary computations, not merely weighted summations.

(b) The bubbles can talk to each other.

(c) The retina can sequentially fixate at numerous locations in the diagram during a single computation.

III-3 The Perceptual Primitives

The implementation of the perceptual primitives is reasonably straightforward. Their implementation generally adheres to the computational restrictions imposed by the retinal structure. Although the current set of primitives is adequate for WHISPER in its problem domain, it could certainly be extended. The ultimate goal would be to expand the current set of primitives to include all the perceptual operations performed by the human perceptual system, although as yet we do not know the constituents of this set.

Each of the perceptual primitives will now be discussed in turn.

III-3.1 Center Of Area

Calculating the center of area of a closed figure is a particularly simple parallel computation. Each bubble first checks to see if its value is the 'colour' of either the interior or contour of the object whose center of area is to be found; if so, it returns to the retinal supervisor the pair of values representing the x and y components of the contribution of its area to the center of area, i.e. (xA,yA) where A is the area of the diagram mapped onto the bubble. The supervisor sums in parallel all the component pairs it receives and divides by the total area of the object. The total area of the object is obtained by summing the areas of all the correctly coloured bubbles. This is the usual notion of the center of gravity of a body. The final result is the coordinate of the center of area relative to the current center of the retina. If the retina is centered far from the object then this result, because of the blurring of the peripheral bubbles, will be only a rough approximation to the actual center of area. It is improved by centering the retina at this estimated location of the center of area and computing the center of area again. The current implementation generally converges to an acceptable result in three iterations.

An object's diagrammatic center of area provides a canonical point which is used as a focal point for many of the other primitives. It helps establish the presence of symmetries, since if an object is symmetrical, the center of area must lie on the axis of symmetry, thereby providing a clue as to where to look for symmetries. The similarity test primitive uses it to align two objects for comparison. Another feature of the center of area is that except for objects with holes or large concavities, it lies within the boundaries of the object at a relatively central location. It is thus a good point on which to focus the eye when looking for contact points.

The center of area primitive is not simply an ad hoc addition made in response to a particular requirement unique to WHISPER's problem domain. Its use in computing symmetry, similarity, etc. provides independent justification of its inclusion as a perceptual primitive apart from its utility in the current domain.

III-3.2 Contact Finding

To find the points at which an object touches other objects the retina is first fixated on the center of area of the object and then the retinal supervisor directs each retinal bubble to execute the following steps:

(1) If the bubble value is not the colour of the object then stop.

(2) For each of its 8 neighbouring bubbles do step (3).

(3) If neighbour's value is the colour of a different object send a collision message to the retinal supervisor.

(4) stop.

The difference amongst contacts involving the support of another object, the support of the current object, or simple touching without support is determined by a comparison of the coordinates of the bubbles as to their relative vertical position with respect to 'up', as defined by the diagram. This involves an interaction between the qualitative knowledge and the primitive contact finding since the assignment of a vertical is a function of the problem domain and the correspondence between it and the diagrams being used.

Once the individual contacting bubbles have been found they must be grouped together. Even though there will be many separate contacting bubbles, there will only be a few distinct areas of contact between the objects. To form the groups requires sequentially following the neighbourhood links from one contact bubble to another. As the chain of neighbouring contact bubbles is followed, each bubble in the chain is recorded as being a member of the same group. If no neighbouring bubble is a contact bubble, then the chain is broken. The length of the chain is used in classifying a contact as either a surface of contact or a point of contact. The coordinates of the bubbles at the ends of the chain provide the extremities of a contact surface. Averaging the coordinates of all the bubbles in the group yields a representative coordinate for the whole group to which the eye can be moved for more detailed analysis. When the eye has been moved there, the central portion of the retina is examined for contacting bubbles, the coordinates of which will be the precise points of contact between the objects. Although the less accurate peripheral determination of the contact points is sufficient for establishing support relationships, exact contact finding is necessary when a contact is the pivot point for a rotation, or when the center of gravity of an object is near the balancing point. It is also used in the feedback method of rotation to check whether a gap has been closed and contact established.

III-3.3 Finding Nearest And Farthest Bubbles

It is frequently necessary to find the location nearest or farthest relative to the retinal center which satisfies an arbitrary condition. For example, in estimating the final amount of twist in rotations, the size of the gap between the objects must be determined. To do so it is necessary to focus the eye on the gap and then find the nearest bubbles whose values are those of the objects involved. The organization of the retina into rings, each an increasing distance from the center, facilitates the search for the required nearest and farthest bubbles. For example, to find the nearest bubble to the center of the retina satisfying condition C, the retinal supervisor executes the following algorithm:

(1) Direct each bubble to test C and save the result (either 'true' or 'false').

(2) For n = 1 to the number of rings on the retina do steps (3) and (4).

(3) Direct each bubble to report its wedge and ring coordinates as a message to the retinal supervisor if the following hold: (a) it belongs to ring n, (b) its saved value is 'true'.

(4) If there is a message pending for the retinal supervisor from step (3), return the coordinates specified in that message to the calling procedure.

This algorithm is a good example of the difference between efficiency in sequential and parallel computation. Since testing C could be an arbitrarily long computation, it is more efficient in terms of elapsed time to simultaneously test C on all bubbles as in step (1), than to test it for only those bubbles in the scanned rings of step (3). On a sequential processor it would be best to test C as few times as possible; whereas, on a parallel processor the total number of times C is tested is irrelevant (if we assume that the time to compute C on failing bubbles is never longer than the time to compute C on successful bubbles). It is the number of times C is tested sequentially which is important.

III-3.4 Visualization

The retinal visualization of rotations is also an exercise in neighbour communication. I have used the term visualization because <u>the process is occurring within the retinal structure, not directly on the diagram.</u> This process is faster, but because of the large size of the peripheral bubbles, less precise than rotation in the diagram. An object can be rotated uniformly on the retina around the retinal center using neighbour communication because the angular shift between bubble centers in neighbouring wedges is the same for all rings. Generally, as the rotation is taking place, there is a predicate P, for example a test for collision, which each bubble is to test. If the predicate succeeds then the bubble interrupts the retinal supervisor with a message indicating that a collision has been detected at the bubble's location. During the visualization process each bubble executes the following steps:

(1) Save a copy of its current value on its two-level stack.

(2) If its current value is the same as the 'colour' of the rotating object, send it to its ring neighbour in the clockwise or counterclockwise direction depending on the direction of rotation.

(3) Set the current value to the value expressed in the incoming message; NIL if there is no message.

(4) Pass the pair of current and saved values to P as arguments.

(5) If P succeeds, report this to the retinal supervisor; if it fails, repeat from (2).

The process repeats until either the predicate succeeds or until it has been executed as many times as there are wedges. The latter occurs when the object has rotated through a full circle.

Because of the coarseness of the retinal resolution, the visualization process is much faster than the alternative of rotating the object by small increments directly on the diagram. This speed is gained at the expense of the possibility of false alarms generated by the predicate succeeding during the visualization when it would not succeed for the actual conditions in the diagram. In particular this is true for collision detection. Although false alarms may arise in which the collision predicted succeeds for the retina when it would not for the diagram, it will, however, never be the case that it succeeds for the diagram but fails for the retina. This is a result of the slight expansion in the size of objects which occurs in the mapping from the diagram to the retina, because points in the diagram are blurred into larger areas of the retina. False alarms cannot be detected or handled on the retina alone. The visualized rotation must first be carried out in the diagram. It is then examined by moving the retina over it to determine whether or not the situation is as predicted. If it is not as anticipated then a false alarm was generated during the visualization. Visualization is a quick, highly parallel, method of anticipating the ramifications of rotating an object through a segment of space.

III-3.5 Symmetry

The symmetry primitive tests for symmetry about a designated vertical axis by comparing the values of symmetrically positioned bubbles. An object is symmetrical (WHISPER tests for vertical and horizontal reflective symmetry, but other types could easily be included) about a given axis if each bubble having its 'colour' as value has a symmetrically located bubble having the same value. In the test of the vertical reflective symmetry of a blue object, for example, if, say, the bubble in the third wedge clockwise from the vertical axis and in the fourth ring from the center has the value 'blue', then the value of the bubble in the third wedge counterclockwise from the vertical axis and in the fourth ring must be checked to see if it is also 'blue'. If it is not then possibly the discrepancy can be ruled out as insignificant, or else the object is asymmetrical.

In addition to the comparison of symmetrically located bubbles, there is an 'excuse' mechanism whereby a non-matching pair of bubbles can query their neighbours' values in an attempt to resolve a conflict. The 'excuses' which can be generated help to eliminate failures of the symmetry test on objects which are actually symmetrical but whose shape on the retina lacks total symmetry because of round-off error arising in representing the diagram as an array. The type of asymmetry which results is usually due to a single bubble covered by the object not having a corresponding symmetrically placed matching bubble, but having all of its neighbcur bubbles successfully match. This is the only type of asymmetry which WHISPER currently knows how to excuse. To determine if a mismatch is excusable the neighbours of the troublesome bubble are checked. All having the same value must have successfully matched, and at least one of them must be a neighbour in the same wedge. The 'excuses' have been designed to deal with the anomalous situations which arose for WHISPER's retina; different excusable asymmetries would arise for different retinal geometries.

The 'excuse' mechanism is easily implementable within the bounds of neighbourhood communication; however, the symmetry comparison itself would be more easily implementable if communication links also existed between the symmetrically located bubbles. Only one such set of extra links would have to exist in order to handle both vertical and horizontal reflective symmetries (or any other axis for that matter), since the visualization process could be used to rotate the retinal projection of the object into a testable orientation. Such extra links are not essential for the organization of symmetry comparisons. A workable technique would be to use the neighbour links to cause whole wedges to shift in a manner perhaps best described as analogous to the closing of a Japanese hand fan. As two wedges come together the bubbles in corresponding rings are compared.

The symmetry test must be supplied a proposed axis of symmetry. As mentioned earlier the center of area offers partial information on determining this axis. The center of area must lie on any axis of symmetry of an object. This does not, however, provide the orientation of the axis. Although the simplest solution may be to test the object in all of the wedge orientations by using the rotational visualization, if one more point on the axis of symmetry could be found the axis would be uniquely determined. Such a point is the center of the circumscribing circle of the object. The only problem is that thus far I have not managed to devise a quick parallel algorithm for finding this center. Although in some cases they could be coincident, in general I expect the center of area and the center of the circumscribing circle to be distinct for objects with only a single axis of symmetry.

III-3.6 Similarity Testing

The similarity test seems to be an important primitive necessary for domains such as geometry, the findspace problem, space planning, jigsaw puzzles, and other Physics problems, although so far WHISPER has not had an opportunity to use it. The purpose of the similarity test is to determine whether two objects, A and B, are similar under any combination of translation, rotation and scaling, and if so to return the angle of rotation, direction and distance of translation, and scale factor.

To test similarity one object is translated, scaled, and rotated to match with the other. Since the center of area of an object is unique, the centers of area of the two objects must be made to coincide. The parameters of translation are simply those required to align them. To test for scaling and rotation the translation is first 'visualized' by having every bubble covered by A save its value while the eye is focused on the center of area of A, then refocusing the eye on the center of area of B. After the translation is 'visualized' the 'image' of object A on the retina is scaled (see next section) by a factor equal to the square root of the ratio of the areas of the two objects (i.e. scalefactor = sqrt(area(A)/area(B)). (The total area of an object can be easily obtained as a by-product of the calculation of the center of area.) Clearly, if the objects are similar this will yield the correct scaling.

After translation and scaling, retinal visualization can be applied to find the angle of rotation and thereby finish the similarity test. Some clues to the most likely angle of rotation could be utilized, although WHISPER currently tries all the possible wedge orientations until one yields an acceptable match. An excuse mechanism similar to that used in the symmetry tests can again be employed here (although WHISPER does not) to handle the cases of objects which are similar but which do not appear precisely similar on the retina.

The similarity test, like the symmetry test, only provides results which are within the resolution of the retina. The parameters of the translation, rotation and scaling are approximate; the similarity test specifies only that a match is probable after the designated transformation. Although WHISPER does not do so, further testing of the similarity of the objects could be carried out by moving the eye to other locations on the contours of the objects and making further comparisons.

III-3.7 Retinal Scaling

An unexpected and interesting property of WHISPER's retinal geometry leads to a simple solution, employing neighbourhood communication, to the problem of scaling the retinal 'image' of an object. An object is scaled correctly (i.e. without distorting its shape) if each bubble having its value, sends this value to a bubble in the same wedge, but a fixed number of rings away. As long as each value is moved the same number of rings either inwards or outwards from the bubble which originally holds it, the size of the 'image' of the object is changed but its shape is preserved (figure III-4). This is the case because the constraint of aligning the bubbles into wedges such that each bubble touches all of its immediate neighbours is satisfied by increasing the bubble diameters by a constant factor from ring to ring. A proof of the scaling property is given in figure III-5. Scaling an object by neighbourhood communication is implemented by having each bubble simultaneously send its value as a message to its neighbour in the same wedge in either the appropriate inwards or outwards direction, and repeating this message passing process sequentially as many times as necessary to bring about the required scaling.

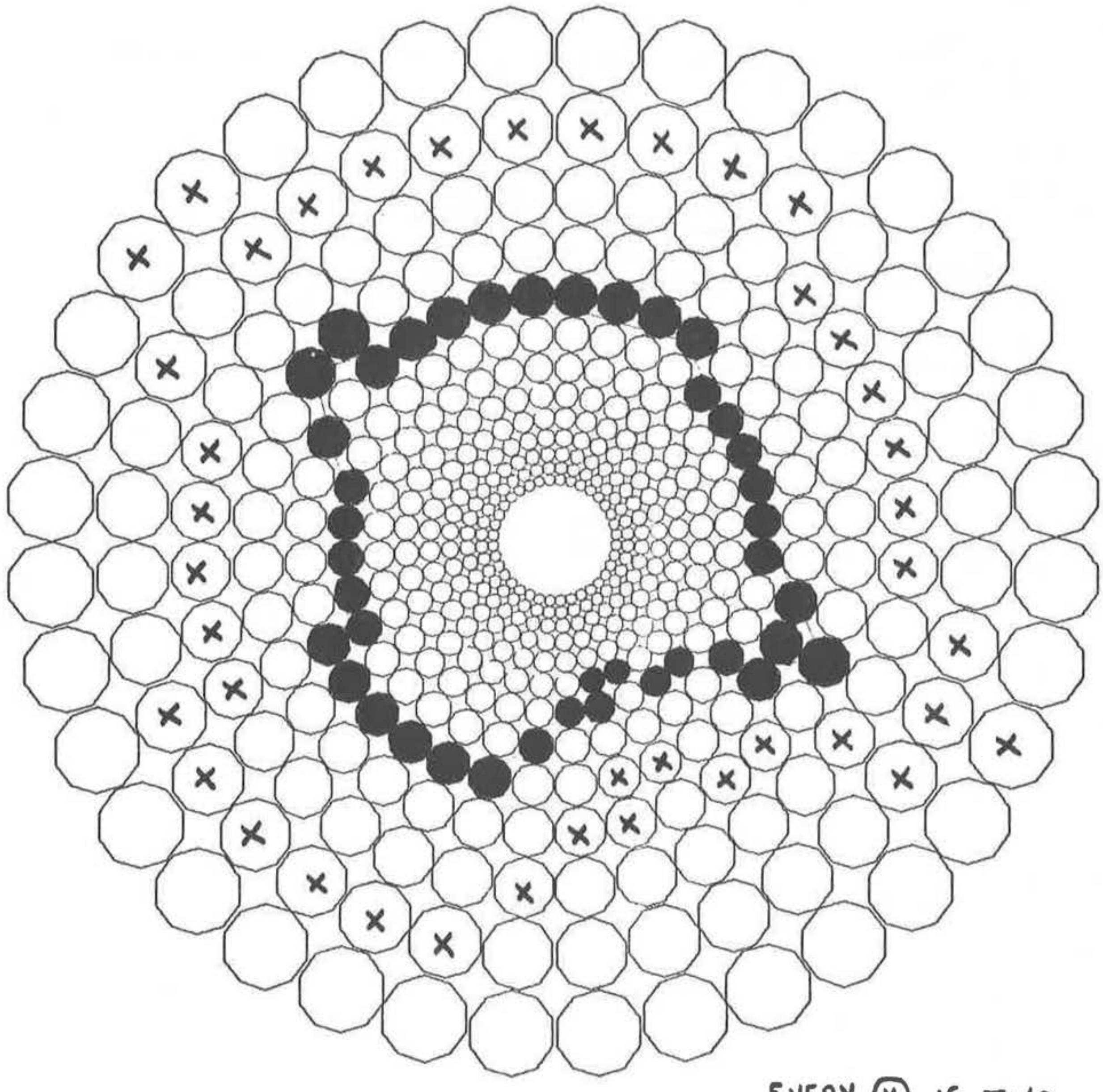


FIGURE III-4

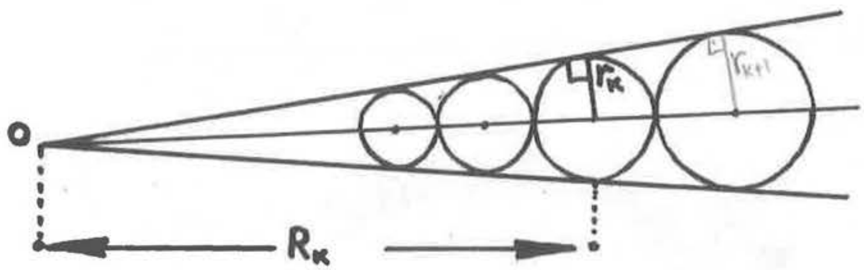
PROOF OF THE RETINAL SCALING PROPERTY

Let:

r_k be the radius of the k^{th} circle.

R_k be the distance from 0 to the center of the k^{th} circle.

$C = \frac{r_{k+1}}{r_k}$ which is a constant from the construction of the retina.



The Scaling Property Hypothesis is: $\frac{R_{k+n}}{R_k}$ constant for all k.

By similar triangles $\frac{R_{k+n}}{r_{k+n}} = \frac{R_k}{r_k}$

$$r_{k+1} = Cr_k$$

hence $r_{k+n} = C^n r_k$

and $\frac{R_{k+n}}{R_k} = \frac{r_{k+n}}{r_k} = C^n$

which is independent of k.

FIGURE III-5.

III-3.8 Curve Features

In order to establish the features of a curve it is first necessary to determine which bubbles are part of the curve. Given one bubble on the curve, the others can be found by following the chain of bubbles each having the same value. One of the conditions imposed upon WHISPER's diagrams was that the contours of the objects were 'coloured' a different shade from their interiors. This prevents the curve following process from getting lost tracing chains of bubbles which are part of an object's interior. It is not necessary to distinctly code the contours of the objects, since it is possible to determine contour points by the type of neighbours surrounding them, but coding is cheaper and easier.

Once the set of bubbles on the curve is found, each bubble in the set can individually test in parallel for the occurrence of a particular feature. Sharp bends in a curve are detectable as an imbalance in the number of bubble neighbours on opposite sides of the curve which are themselves not members of the curve. This is illustrated by figure III-6 in which bubble A has three neighbours on each side of the curve indicating that the curve is smooth at that point in contrast to bubble B which has six neighbours on one side and only one on the other. A bubble can test for bends by simply asking its neighbours to respond whether or not they are also members of the curve, and counting the responses from the two sides of the curve. For a

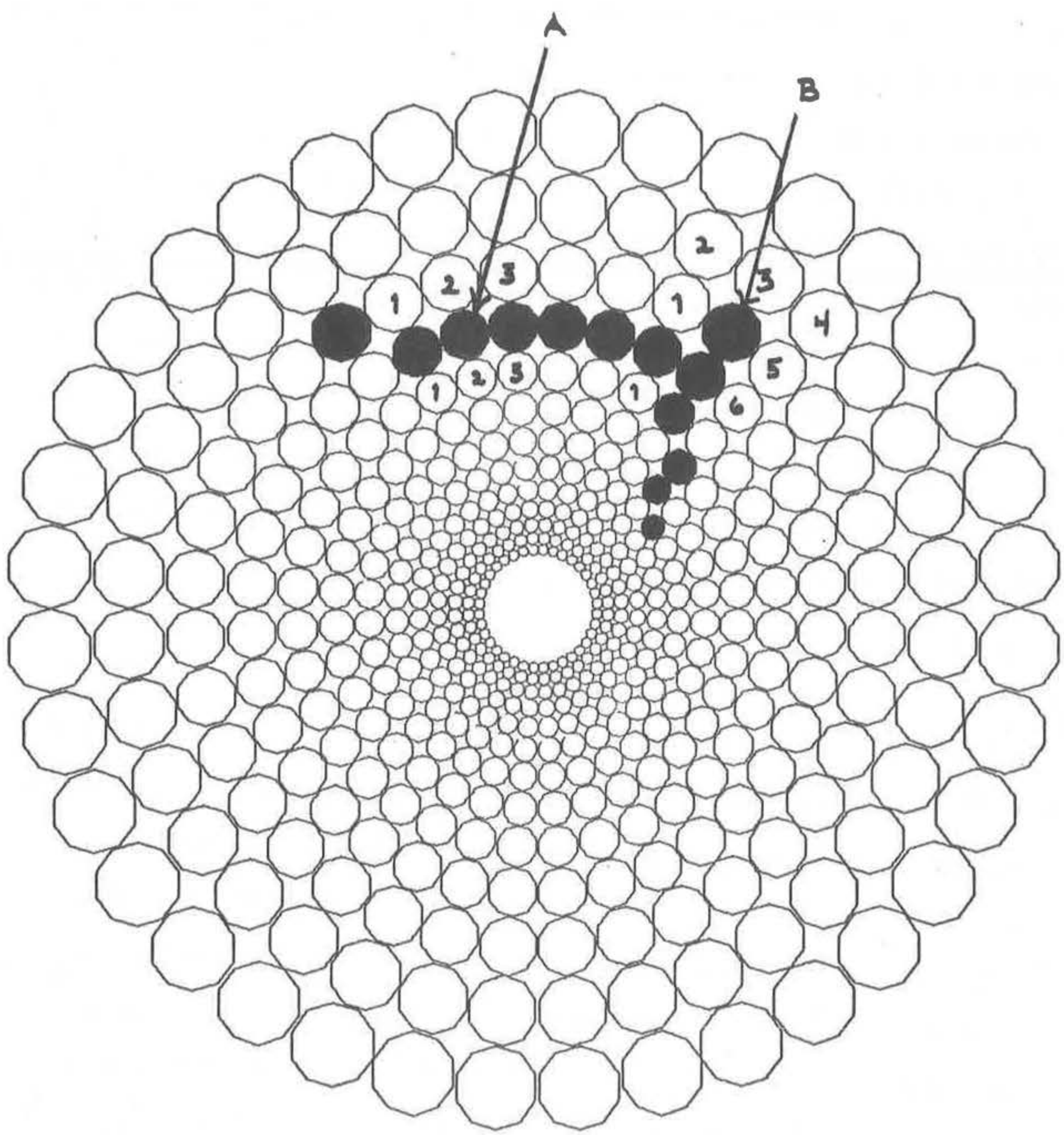


FIGURE III-6

simple closed curve, if it is known which of the bubbles are interior and which are exterior, then the bend can additionally be classified as convex or concave.

The slope of a curve at any curve bubble is determined as the perpendicular to the bisector of the angle between the centers of its neighbouring bubbles on the curve. This yields a rough approximation to the actual slope, but it is sufficient for testing drastic changes in the slope of the curve over its whole length. A more accurate determination of the slope at a particular point on a curve is obtained by re-centering the eye on that point and then utilizing the higher resolution central portion of the retina. The perpendicular to the bisector of the angle between the wedges most densely covered with points from the curve is the tangent to the curve at that point. The angle between wedges can be used because they emanate directly from the center of the retina, just as the curve must when the eye is centered on it. This is more accurate than measuring the angle between neighbouring bubbles because there are more wedges than neighbours. This more accurate slope determination is used to measure the slope of surfaces at contact points to decide whether or not they are horizontal.

A sliding object cannot slide over a hill which is higher than its initial location. The high spots of a curve are found by each bubble on the curve comparing its vertical coordinate with the height of the initial location.

Finally, WHISPER must know how to check whether there are objects resting on a surface with which an object sliding along the surface might collide. Objects on a surface are found by having each curve bubble test that none of its neighbours is empty space, the interior of the object of which the curve is the contour, or part of the curve.

III-4 Summary Of The Retina And Its Perceptual Primitives

WHISPER uses the retina and its perceptual primitives to extract information from a diagram. The perceptual primitives provide a domain independent set of diagrammatic features which are interpreted by a higher level reasoner relative to the current problem domain. The retina is a parallel processor with restricted communication between processors. The perceptual primitives are the algorithms that it executes. It is moveable, and when it is focused on a point in the diagram each of its processors receives an input simultaneously. The function mapping points in the diagram onto processor inputs defines the topology of the retina. Although there could be many other retinal topologies, the current one has useful properties. Rotations and scalings can be visualized by a neighbourhood communication process, and its resolution decreases with increasing distance from the retinal center.

III-5 Implementation Details

III-5.1 Languages

The languages used in WHISPER's implementation are: LISP/MTS, a subset of CONNIVER, and FORTRAN. The qualitative physical knowledge and the perceptual primitives are primarily written in LISP with some calls to CONNIVER's pattern matcher. CONNIVER's[15] associative database is used to store assertions pertaining to features extracted from the diagram. Each specialist requiring information from the diagram first checks the database for a relevant assertion made by an earlier specialist before it calls the retinal supervisor to look at the diagram. The array diagram is stored in a FORTRAN array, and the diagram-to-retina mapping is a LISP callable FORTRAN subroutine. The redrawing transformations are also written in FORTRAN. FORTRAN was chosen for these tasks because they require extensive numerical calculations, and because it is LISP/MTS callable.

III-5.2 Timings

The timings which can be given are at best very approximate, and highly dependent on machine speed and language implementation. Running LISP/MTS[16] interpretively under the Michigan Terminal System on an IBM 370/168, WHISPER took approximately 2 minutes of processor time for each snapshot. The amount of memory required including space for the LISP interpreter was approximately 250 K words (32 bits per word, 2 words per CONS cell). The time to fill the retina at each fixation is 1.4 seconds for the periphery and 0.6 seconds for the central section. These times would be inconsequential in the total problem solving process if there were true parallelism. Since the central section and the periphery each have 540 bubbles, each complete fixation would require (1.4 + 0.6)/540 = 0.0037 seconds. In producing the first snapshot to the chain reaction problem WHISPER made 33 fixations (10 central section, 23 periphery) requiring 10 x 0.6 + 23 x 1.4 = 38.2 seconds. Virtually all of this time (38.2/540 = 0.07) could be factored out from the 125 seconds required for the first snapshot. I would estimate from isolating portions of a trace of WHISPER's behaviour during the first snapshot that approximately 40 plus or minus 15 seconds were spent in the qualitative physics specialists. The rest of the time is spent fixating and computing the perceptual primitives, and would be substantially reduced by parallelism. The time spent in the qualitative physics specialists would be significantly reduced by compilation (LISP/MTS does not support a compiler).

Chapter IV: Human And Machine Use Of Analogues

IV-1 Introduction

The use of analogues as aids in human problem solving is very common. It is a central aim of this paper to demonstrate how computer programs can derive some of these same benefits from the use of analogues in problem solving as do people. The importance of the role of analogues in human problem solving has been underrated. This is particularly true with respect to diagrams, which are generally regarded as simply an addition to the memory capacity of the brain. When viewed from this perspective there is no sense in using diagrams in machine reasoning because there are more convenient ways of extending computer memory. Currently AI systems are not severely constrained by memory capacity limitations, but rather by the lack of good methods of organizing the information which can already be stored. It is the purpose of this chapter to explore some of those aspects of analogues which are more important than their possible use as memory extensions.

A basic premise underlying my examination of the benefits of analogues in problem solving is that there is no essential difference in the way people utilize them and the way machines should be able to utilize them. This introduces a prior and unsolved problem concerning the interaction of a computer with its environment; WHISPER serves as a demonstration that this problem can be overcome if it is accepted that a hardware incarnation of its software simulation is realizable. This premise also raises the question discussed in section I-2, namely, what is to be considered the computational structure of a machine. To insist that a computer not be allowed to utilize diagrammatic aids in problem solving would be comparable to requiring a student to write a Physics exam without allowing him to draw any diagrams. Why would this be a handicap? A possible reply - that it is a result of human short term memory limitation - is a partial, though insufficient answer to the question. The advantages that WHISPER derives from using a diagrammatic analogue (e.g. amalgamation, motion discontinuities) and the further issues raised in this chapter are evidence that diagrams may play a more fundamental role in the student's reasoning.

I would like to point out a few issues that the use of analogues does not concern. One is that the ability to discover analogies is not a prerequisite to the use of analogues; the use of analogues is not concerned with their discovery. There are many analogies which have been discovered and transmitted from one generation to the next. Secondly, analogues have no special connection with the property of continuity. Certainly, a characteristic of a particular analogue may be that it is continuous, but this is hardly a prerequisite condition. Similarily, although the emphasis in this paper is on analogues which have as one of their properties two-dimensionality, this is not a necessary characteristic. Finally, reasoning with analogues is not intended as a substitute for other kinds of more abstract reasoning, nor do I claim that through the use of analogues one can enlarge the class of computable functions as defined by Church's Thesis.

IV-2 Analogy And Analogues

When there is an analogy between two entities then they are each analogues of the other. Polya defines analogy: "*Analogy* is a sort of similarity. ...similar objects agree with each other in some aspect. If you intend to reduce the aspect in which they agree to definite concepts, you regard those similar objects as *analogous.* ...two *systems* are analogous, if they *agree in clearly definable relations of their respective parts.*" (original emphasis).[17] Although almost all things are analogous on a trivial level, some things are strongly analogous in that there are numerous relationships involving total or large subsets of their respective parts which agree. The ingredients of Polya's definition are present in Sloman's discussion of analogical representations: "if R is an analogical representation of T, then (a) there must be parts of R representing parts of T, as dots and squiggles on a map represent towns and rivers in a country... and (b) it must be possible to specify some sort of correspondence, possibly context-dependent, between properties or relations of parts of R and properties or relations of parts of T..."[18]

There are many examples of analogues: a map is an analogue of the geography of an area; water waves are an analogue of light waves; the circuit simulation in the SOPHIE system of Brown, Burton and Bell[19] is an analogue of a real electrical circuit; a two-dimensional array is an analogue of a piece of paper; the diagrams used by WHISPER are an analogue of a blocks world. In the case of water versus light waves and in the case of circuit simulation, the correspondences constituting the analogy are not between static aspects of the entities but rather between aspects of their behaviour. Thus the 'parts' referred to in the above definitions, particularly in Polya's, must be interpreted very broadly to encompass a notion of a 'part' of behaviour. For example, diffraction is a part of the behaviour of light for which there is a clearly definable correspondence to the diffraction of water waves.

A lot of the murkiness surrounding questions of the use of analogues, especially their efficiency, derives from confusing the issues of the construction of the analogue with the issues of its use. Once an analogue exists it can be utilized independently of its origin. A central question then is whether it is possible to obtain more results from the analogue than are <u>explicitly described</u> in the construction of that system.

The answer to this question is yes on two grounds. The first is that the analogue need not be strictly internal to a machine, i.e. it need not take the form of a simulation. This is the case for analogues such as maps, diagrams, and scale models. For example, if we wish to determine the stability of a pile of blocks in a blocks world situation on the surface of the moon, then we could construct a similar pile of blocks on earth and determine the result by experiment. The experimental result is acquired 'for free' in that there is no need to describe the behaviour of the blocks on earth in order for them to behave correctly. However, since the behaviour of the two situations is not identical, the experimental results must be interpreted on the basis of the analogy. Interpreting the results (e.g. converting accelerations) requires establishing and executing a method to handle discrepancies. The basic point is that we don't need a theory of blocks behaviour on earth or the moon; we need only know the analogy (i.e. the mapping) between them.

The second reason for answering yes is demonstrated by WHISPER. The two-dimensional structure of a piece of paper is partially simulated by a two-dimensional array. Thus it is necessary to describe some of the aspects of two-dimensional space and to simulate these aspects computationally. All that this simulation consists of is the usual function, mapping coordinate pairs into positions in the storage vector. It is not necessary to describe any of the other topological properties of 2-space (e.g. the triangle inequality, since its validity derives from the properties of the array). In addition, there are procedures for rotational and translational transformations of points in the array. The resulting system consisting of the array and the redrawing transformations is analogous (when objects are represented by sets of points) to the blocks world with respect to the behaviour of objects in terms of their possible motions. Thus we do not get motions for free since it is both necessary to describe them and to compute them. However, when an object is moved there are many side effects of that action which propagate in the analogue without their being explicitly described in advance. When WHISPER moves an object none of the other objects move; the shape of the object is preserved; the total amount of empty space is conserved; the areas of empty space are updated properly; its contacts with other objects are updated; and the shapes formed by groups of objects change. These properties are accessible as a quick fetch by the appropriate perceptual primitives. <u>These aspects of the behaviour of the analogue were not explicitly described and built in during its implementation.</u>

IV-3 Interaction With An Analogue

Using an analogue in reasoning involves interacting with it through experimentation. To find the outcome of a given change in situation S, an experiment is conducted in an analogous situation A (a corresponding change is made to A), and the result of this experiment is interpreted in terms of the original situation S. The class of meaningful experiments and their interpretations is determined by the correspondences defining the analogy between S and A. Since the analogy provides a means of interpreting states and events of A as indicating the occurrence of particular states and events of S, A represents S. The analogy defines the semantics of the analogue, A, as a representation of S. In Pylyshyn's[20] terms, it provides us an SIF (semantic interpretation function).

In addition to a static state of the analogue A denoting a state of S, A can also be used to represent the dynamic behaviour of S. This is a result of the two modes of experimentation: measurement of aspects of the current state, and measurement of aspects of the subsequent state or sequence of states arising from some change in the initial situation. These will be called <u>configurational</u> and <u>behavioural</u> modes of experimentation. An experiment, therefore, either determines the current physical configuration or it determines the subsequent behaviour deriving from a change in the current configuration. It is because an analogy can specify correspondences between situations in terms of their static configurations, as well as in terms of their behaviour, that these two modes of experimentation arise, and that an analogue can represent both the static state and the dynamic behaviour of a situation.

An example of a configurational experiment is measuring the distance between two points on a map. The interpretation of this measurement is made on the basis of the analogy between the map and the geography it represents. An example of an experiment in the behavioural mode is increasing the velocity of the wind in a wind tunnel containing a model airplane. Changing the velocity is a change in the state of the situation from which many other effects of the behaviour of the wind-model system follow, such as an increase in lift of the wings. These resultant effects are interpreted as indicative of similar effects which would occur as the result of increasing the velocity of a full sized airplane in flight. The interpretation may, however, not involve simple linear scaling as in the case of the map, but a more complex conversion.

Although analogues can be used as representations this does not imply that they immediately serve as representations manipulable internally by a computer. A representation is useful, nonetheless, if it is more manipulable than that which it represents. Thus a pile of blocks on earth, representing a pile of blocks on the moon, is a useful representation since determinations about the blocks on the moon can be made without visiting the moon to observe them. Such representations are useful to people at least, and I see no reason why they should not be considered to be useful to a machine.

The advantages of analogues to WHISPER arise primarily during problem solving rather than in the long term storage of information. A problem solving program need not be able to invent the appropriate analogue. The method of constructing an analogue from a descriptive representation can be defined in advance for the program. Thus a program does not need to store a large inventory of 'pictures'.

This is similar to the graphics metaphor which Kosslyn[21] has advanced as a theory for the storage and construction of visual images which humans subjectively experience. His analogy is that of a computer graphics terminal displaying pictorial information presented to it in terms which are very non-pictorial in nature (usually in terms of the coordinates of two points between which a line segment is to be drawn). If some sort of equivalent exists which is the 'screen' for visual imagery, then it is not necessary to literally store a picture in order to have the subjective experience of visual imagery. Similarly, it is not necessary to store pictorial diagrams in order to utilize diagrammatic analogues. This issue will be discussed at greater length in Chapter V where I propose an alternative to the array encoding of diagrams.

The behavioural and configurational modes of experimentation and the representation of the states and events of one situation by the analogous states and events of another situation result in two advantages (behavioural advantages and configurational advantages) which a system can derive from interaction with an analogue. These advantages will now be discussed.

IV-4 Advantages Of Analogues

I want now to classify the benefits derivable from analogues in terms of the twc modes - behavioural and configurational - of experimentation discussed earlier as the basis of the interaction between a system and analogue. The WHISPER system relies upon and demonstrates the usefulness of many of these benefits. There are, however, further aspects which have not been touched upon by WHISPER, and also further examples of problems whose solutions are simplified by appealing to analogues. Different aspects of analogues are, of course, applicable to different problem domains; this is one reason they are not all demonstrated by WHISPER. To take further advantage of analogues, as well as tc apply them to other domains, would require extensions and modificaticns to the current system. On the basis of the current implementation it should be clear that only reasonable extensions to the perceptual primitives would be required in order to handle the examples discussed below. The qualitative knowledge is the domain dependent part of the system, and as a result would require almost complete replacement with the knowledge relevant to any new domain. The feasibility of using analogues in problem solving has been demonstrated, I believe, by the WHISPER system; the generality of using analogues, especially diagrammatic analogues, follows from the examples below.

IV-4.1 Behavioural Advantages

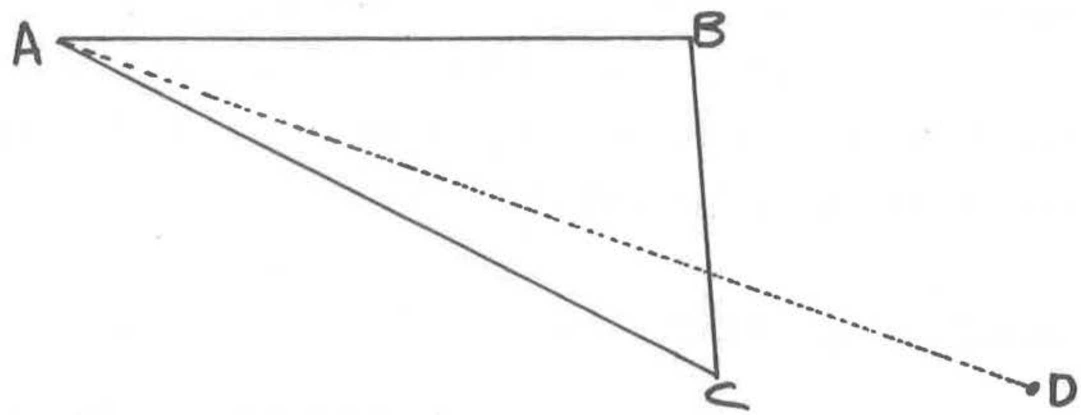
One of the main advantages of analogues is that it is possible to represent action by analogous action. There is a difference between the descriptive encapsulation of action in laws of behaviour (e.g. as equations of motions), and the representation of action by dynamic behaviour itself. It is computationally easier to reason about action by observation than to reason with denotative descriptions of action. In a given situation the effects of an action propagate until all the ramifications of the action have been effected. If such a situation is used as an analogue of another situation, and if the analogy is strong enough, then the effects which are propagated in the analogue will correspond to effects which follow from an analogous action performed in the represented situation. The 'frame' problem is concerned precisely with this propagation of the side effects of actions, which is why analogues overcome (for those situations in which an analogue can be found) the 'frame' problem.

WHISPER has already demonstrated some of the behavioural advantages which result from the use of analogues. When the action of moving an object is performed in the diagrammatic analogue the side effects of the changing contact and support relationships, the changing shape of surfaces over which objects might slide, the changing shape of empty space, and the unchanging shapes, areas, and positions of the unmoved objects, are all propagated correctly and in an analogous manner to the way in which these relationships change or remain unchanged in the physical world the diagram represents. A discussion of further behavioural advantages follows.

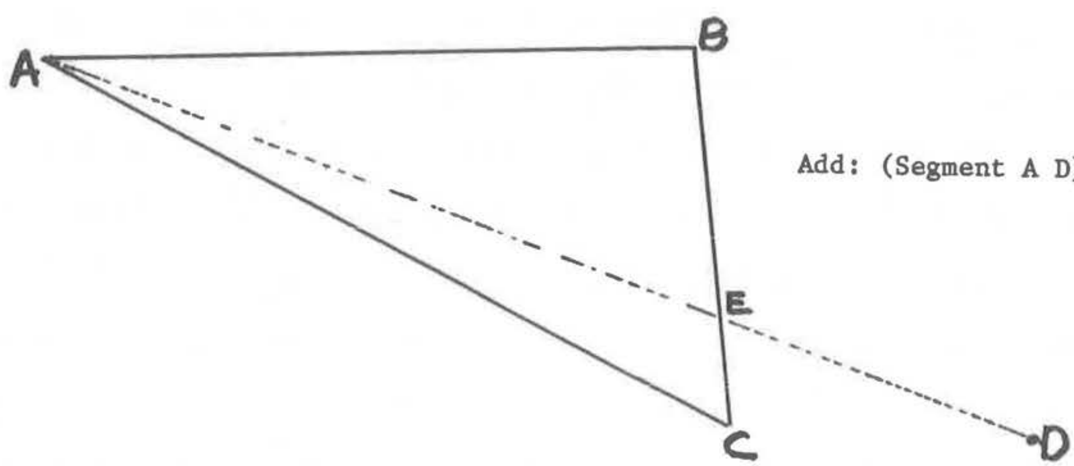
IV-4.1.1 Implicit Derivation

Figure IV-1 depicts a geometry example. Two representations of the triangle ABC are given, one diagrammatic, the other in terms of its vertex coordinates. An advantage to using the diagrammatic representation arises if one makes a geometric construction, e.g. joining points A and D. Assuming that an intersection point recognizer is available as a perceptual primitive, the new point, E, created by intersecting AD and BC, is automatically present in the diagram. In the coordinate representation the construction would be represented by the simple addition of the assertion (SEGMENT A D), but this does not produce the new point.

Adding the assertion triggers a demon which computes any possible intersection that AD might make with all other lines (curved or straight). This has several disadvantages. The possibility that a new point will be created by the construction of a line segment must be anticipated in order that the demon be written. The demon must be reasonably complex in that it must know how to calculate the intersection of a line segment *with any possible curve which can occur.* If



(Coordinates A 0 0)
(Coordinates B 5 1)
(Coordinates C 6 -4)
(Segment A B)
(Segment A C)
(Segment B C)
(Coordinates D 9 -3)



Add: (Segment A D)

FIGURE IV-1

a new curve type is introduced, the demon must be expanded to handle it. When a line segment is constructed the demon procedure may be called totally unnecessarily if the segment does not intersect any other curves. More important, however, is that the possible intersection of the new segment with each of the other curves must be considered separately.

In the diagram the new point is derived implicitly through the two-dimensional structure of the diagram, and the explicit representation of all the points constituting the curves. The new point must of course be recognized as such, but we may say it exists, provided that a recognition primitive is available to fetch it. It is not necessary that the point be recognized when it first appears and continually re-established every time a change is made to the diagram. It is possible that a construction could have some other effect which would in the proof of some theorem lead to other constructions. Only then would the first intersection point become significant. The point can be ignored until such time as the examination of the diagram for the set of all points is motivated by a requirement of the proof process. The point will still be correctly recognized, if needed, even though no computation has been expended to assert its existence.

IV-4.1.2 The Amalgamation Problem

A common problem is the description of a situation which results from the combination of the descriptions of two or more other situations. The original descriptions must be amalgamated in some way to form a new one. WHISPER's amalgamation of object and curve descriptions was discussed in sections II-3.2 and II-9.2.

Another example of the amalgamation problem arises in the space planning problems described by Eastman[22] in which furniture or machinery is to be placed in a room so as to satisfy a given set of constraints. The original specification of these constraints has the form: 'the sofa must be against the wall' and 'it must be possible to see out the window from the sofa'. To find the set of all positions satisfying these constraints it is necessary to combine the constraints in some way which is more meaningful than their simple conjunction. Eastman suggests a technique which he did not implement called _projective location generation_, which is based on the use of a diagrammatic analogue of the room, objects, and constraints.

His suggestion is to display a schematic of the room at a graphics terminal, and then to shade that space in which the next object to be located is constrained to lie. The shaded area represents the constraint on the object's possible positions, and the conjunction of one or more constraints is represented by the greater screen intensity resulting in the areas where the shading overlaps. The constraint space is thus represented by space on the screen, and the way in which constraints interact is represented by the analogous way in which multiple shading of the screen results in increased intensity. Amalgamating the constraints on the objects to be placed in the room is therefore very easy.

Eastman's proposal facilitates human interaction with the problem solving program. The representation of the constraint space, by space on the graphics terminal screen would be very useful to a human problem solver; there is no reason why it should not be as useful to a problem solving program if array space were used instead of graphics screen space. Greater screen intensity could be represented by array elements of greater magnitude, and the array could be examined by WHISPER's eye.

It is easy for WHISPER to amalgamate the descriptions of its objects. It refers to the objects by the array values which compose them, so combining two objects into a new object only requires the construction of a name which is the union of their values. The combined set of points of both objects is the complete description of the new object. All the properties which are needed in deciding its stability are derivable from this description. The amalgamation of the descriptions of the contour of objects is essentially a by-product of the amalgamation of object descriptions.

IV-4.2 Configurational Advantages

Configurational advantages derive from the representation of relations by analogous relations, in contrast to the behavioural advantages which derive from the representation of action by analogous action. Static states or configurations of an analogue, A, of a situation, S, correspond in a simple way to configurations of S. Parts of A correspond to parts of S, and relations between parts of A correspond to relations between parts of S. As Sloman[23] has pointed out this means that relationships of S are represented by A without being explicitly _named_ in A.

An important consequence of not having to name the relationships is that an analogue can explicitly represent many more of them than would otherwise be feasible. For example, the distance relationships between all pairs of geographical points is explicitly represented by the distance relationships between corresponding pairs of points on a map. It would not be possible to store this potentially infinite set of relationships if each had to be explicitly stated. Some examples in which the representation of such relationships is important are discussed below.

IV-4.2.1 First Approximation Diagrams

An heuristic that Polya emphasizes about solving problems is to draw a figure illustrating the data. One problem he discusses to which I will give a related but somewhat different approach is: "We are given three points A,B, and C. Draw a line through A which passes between B and C and is at equal distances from B and C."[24] The solution to construction problems of this sort is not really the final diagram but an algorithm for its construction. A detailed account of a computer program which solves geometrical construction problems is given by Funt[25] A diagram of the problem would act as a model and could be used in the same fashion as Gelernter's Geometry Machine[26] did for the proof of theorems. Gelernter's system is provided with the diagrams it uses as models in the form of a coordinate specification of their points; however, creating the diagrams from the hypotheses of a theorem or the statement of a problem is generally a non-trivial task.

Visual feedback from the diagrams as they are being constructed can be used to bootstrap from a diagram only partially fulfilling the conditions of the problem to one which completely satisfies them. A partial diagram for Polya's example would be just the three points as in figure IV-2. A line can then be drawn through A in any direction and then rotated (figure IV-3) about A until the distances between it and the points B and C are equal. Measurement is the basis of

FIGURE IV-2

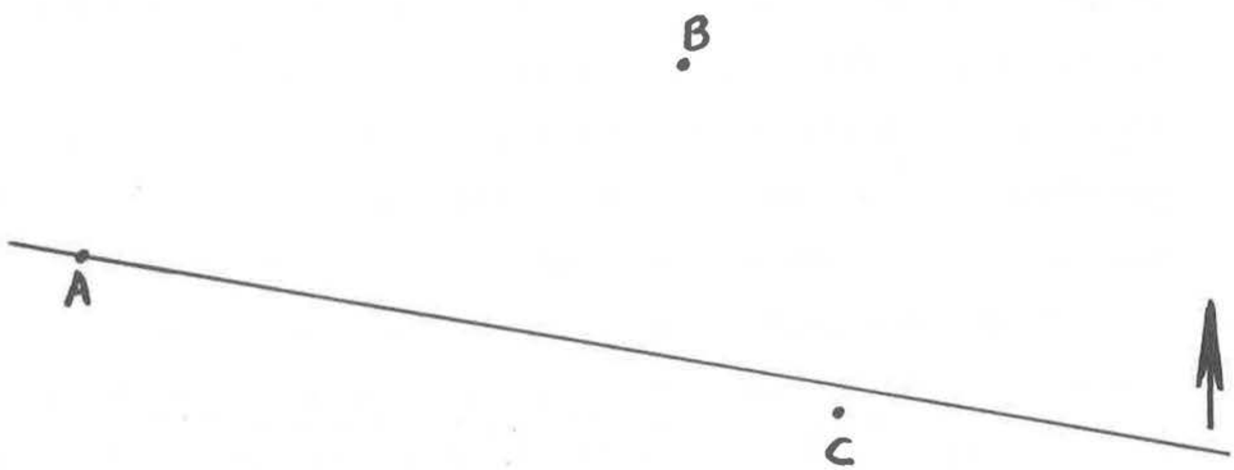


FIGURE IV-3

this test. The rotation could be accomplished by WHISPER's visualization process or it could be done by the systematic generation of new lines. The main point is that feedback from the diagram is used both to test for the condition and to suggest a better orientation for the line; constructing a diagram which looks correct in this way is easier than solving the original problem.

In the final diagram, figure IV-4, all the relationships expressed in the problem statement are explicitly represented. It is an analogue of the more general entity described by the conditions of the problem. All the relationships between the generalized points A,B,and C, and the generalized line passing through A hold in the diagram between the specific points marked by the dots on the paper. Thus, in this case at least, an analogue is a model of the problem. It is necessary to distinguish two analogues in this example. These are: the analogical representation of the three points and a random line used to help in the construction of the final model, and the final model itself, also an analogue.

After the approximate orientation of the line is determined by feedback the problem is easily solved using the analogical properties of the diagram. Connecting points B and C results in the intersection point D. The high level reasoner now uses the retina to inspect the diagram for interesting features (e.g. symmetries, similarities, equal angles or lengths) and finds that BD and CD are of equal length

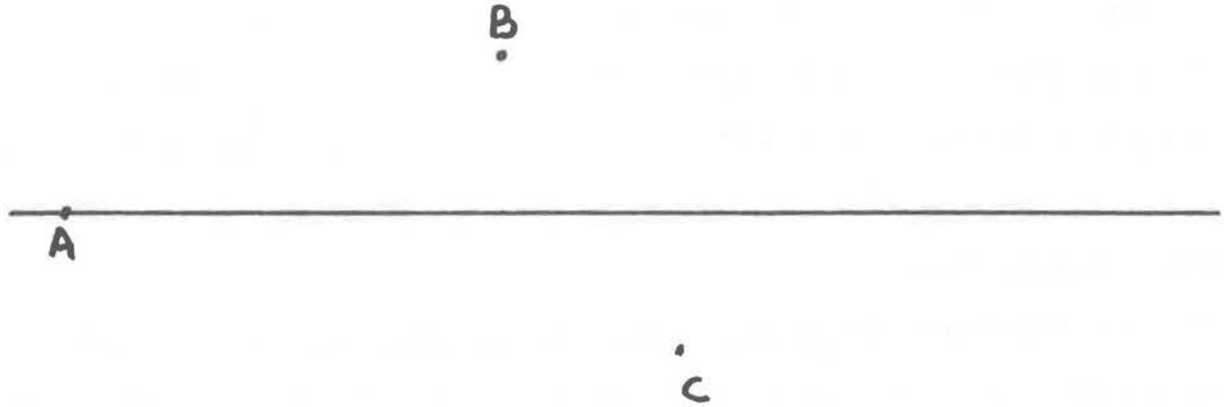


FIGURE IV-4

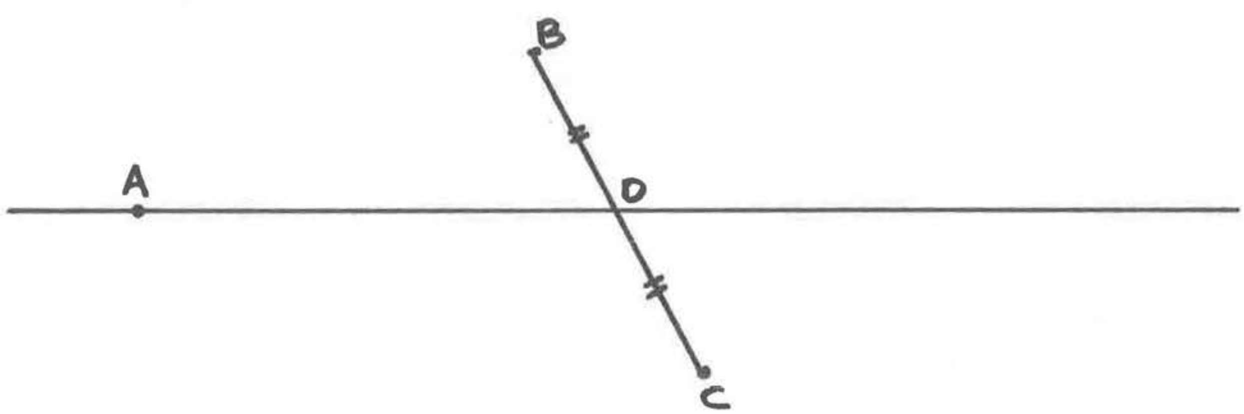


FIGURE IV-5

(figure IV-5); this is the key to the problem. The diagram is essential to its discovery.

Strong clues to the derivation of the final solution are provided by approximation diagrams, since relationships between parts in the specialized situations they depict have strong analogical correspondences with the general relationships in the abstract geometrical situation. These correspondences provide some justification for turning a specific relationship in the diagram into a general hypothesis about the general validity of the relationship. Proving this hypothesis becomes a subgoal of the original problem. In the current example, establishing BD=CD as a worthwhile hypothesis is the most formidable part of the problem; proving this hypothesis is not difficult.

IV-4.2.2 Changing Level Of Detail

We are all familiar with the experience of being so near to a picture that it is unintelligible, while standing back from it makes it suddenly understandable. This is particularly common with digitally produced pictures on a line printer. Fine detail seen close to the picture, e.g. the individual characters in the printout, or patterns or words they might form, is irrelevant to the interpretation of the picture. The picture must be examined at a grossed level of detail as some vision systems (Kelly[27] and Shirai[28]) have done. The extraneous detail cannot just be ignored, it must be smoothed or blurred in such a way that the important qualities of the next level of detail are not simultaneously eliminated. The amazing thing is that the process of moving away from the picture creates exactly the right kind of transition from one level of detail to another. This smoothing is a function of the optics of the situation and the fixed resolving power of the eye. The detail eliminated and that retained is a consequence of the physics of the situation, not of the execution of any filtering algorithm. The physics of the situation dictates that proximity is the basis of the blurring. Thus, neighbouring elements meld.

Part of the effect of blurring is to turn digital data into a form which has more of the characteristics of continuous data. Smoothing reduces the number of discrete facts which have to be dealt with. Formal representations have a digital or discrete quality in that they are composed of a large number of distinct axioms or assertions. The problem is that there is no easy way to blur an axiom or a group of axioms. The problem of blurring axioms is Minsky's[29] nearness problem: If we know Near(A,B) and Near(B,C) then under certain circumstances it would be correct to deduce Near(A,C); but this cannot continue indefinitely for Near(C,D) and so on, because at some point it will not be the case that A is near X where X is related to A by a sufficiently long chain of nearness links. The proximity of points in a diagram is not explicitly characterized by statements of this kind, but is a function of the diagram's topological structure. The result is a smooth transition of the relationship between points as the context and level of detail in viewing them changes.

Blurring which is based on proximity may or may not be a useful transformation of detail, depending on what is represented by the diagram's proximity structure. In many cases physical proximity in a diagram does seem to indicate a kind of organizational proximity in the domain represented. Also grouping on the basis of proximity is frequently a beneficial reorganization of the situation represented. For example, clusters of boxes on a flowchart or components in a circuit diagram are often indicative of a clustering on the functional level. Elements related by their physical proximity function together closely as part of a larger global module. The colour spectrum produced by a prism has proximity of frequency represented by spatial proximity. In normal musical notation proximity in the vertical corresponds to proximity in frequency while proximity in the horizontal corresponds to proximity in time. Although it is not the case in all of these examples that appropriate clustering will automatically occur simply by moving away from the diagram, the explicit representation of a proximity relationship in the situation of interest by analogy to the spatial proximity of marks in a diagram frequently means that changing the level of detail at which the diagram is viewed results in an appropriate and corresponding change in the level of detail at which the represented situation is considered. If the proximity information of a situation is not contained in a representation, then it is very difficult to vary the level of detail at which the situation is to be considered.

Since the resolution of the eye decreases with the distance from the center of the retina, fixating the eye at a different location while maintaining it at a fixed distance from the diagram also results in a restructuring of detail. The blurring effect is similar to that already discussed except in this case the resolution is varying while the distance remains fixed, whereas in the previous case the distance was varied while the resolution remained fixed. Changing the fixation point changes the relative attention paid to the detail in different parts of the diagram. The coarser detail of the surrounding area provides a context for the finer detail at the fixation point. In a sense the diagram is being viewed simultaneously from varying distances. The structure which remains after blurring on the periphery establishes a context for interpreting the structure of the detailed unblurred information on the central portion of the retina.

IV-4.2.3 Planning

An obvious example of planning is discovering a path between two points along which a given object can be moved. A straight line joining the two points is a first order plan. If this plan is represented analogically as a line in a diagram, then the plan can be examined to find the bugs in it. It is easy to anticipate collisions by simply looking at the amount of empty space near the path. If at some point the distance between the path and another object is less than the 'radius' of the object to be moved, then a collision will result. The collision points can be dealt with in the order they arise on the path and be eliminated by introducing detours. It is important that a detailed analysis of possible collisions is required at only a few points, and not at all (or even a finite approximation to all) points along the path. Additionally, it is generally straightforward to determine a detour which will circumvent the difficulty without much trouble, because it can be seen whether or not the collision is a result of objects on only one side of the proposed path. Without the explicit representation of empty space and the proximity relationship of the path to objects in the environment, these two advantageous factors would be lost.

Clearly, the three-dimensional version of this problem is more difficult than the two-dimensional case just discussed. There are two solutions: use a three-dimensional analogue such as a scale model, or decomposa the three-dimensional case into a sequence of two-dimensional problems. I think that most of the problems which people are able to solve easily can be handled by the decomposition method. If a problem is difficult, then often a 3-D scale model is built or the problem is directly present in the world, so the world serves as its own representation. The type of planning which can be accomplished is very similar to the two-dimensional case, whichever method is used.

IV-4.3 Parallelism

Although so far I have only emphasized parallelism as facilitating a feasible implementation of the perceptual primitives, this is just one side of the issue. An important advantage of diagrammatic analogues, at least, is that they provide an application for a powerful, but seldom used tool - parallel processing. The parallelism and organization of WHISPER's retina make it a particular kind of computational structure which is applicable to only certain kinds of underlying data structures. The prime example of these being WHISPER's diagrams.

Parallelism is applicable to diagrammatic analogues because of the amount and uniformity of the information they hold. They contain the partial extension of a set of axioms rather than the axioms themselves. For example, WHISPER's diagrams contain the points on a line segment AB rather than just the two points A, B, and an axiom expressing which other points are on the line. If the extension is first generated, than as WHISPER demonstrates, parallelism is applicable.

IV-5 Why Analogues Work: Some Speculations

There is a spectrum of analogues with respect to the complexity of their underlying descriptions. In section IV-2 I discussed how an analogue could have components of both physical and computational structure. The computation is based on the underlying descriptive model. A decrease in the structural similarity between the physical aspects of the analogue and the situation to which it corresponds leads to an increase in the complexity of the requisite description.

At one end of the spectrum is the world itself. The world simulates itself and does not operate by following an underlying description of its own behaviour. The next point on the spectrum corresponds to scale models, like maps or model airplanes. In these there is a strong similarity between the physical structure of the analogue and the physical structure of the reality they represent. All that is required in descriptive terms are relatively simple rules for discrepancies due to scaling. Simulations such as SOPHIE, which are equation based, lie at the far end of the spectrum. They are totally dependent upon an underlying descriptive model. It is necessary to be able to completely describe the domain before simulating it computationally. Also this final point on the spectrum corresponds to analogues which rely on deduction and axiomatization. A theorem prover will simulate any domain described by the axiomatization supplied to it. Again, it is necessary to completely describe the domain of interest. Formulating a sufficient axiomatization is often a very difficult task.

In general, formal descriptions of situations are incomplete, and more than one situation will fit the description. (The axioms have more than one model.) It is difficult to design a description such that a unique situation will fit it. Often we are interested in a single situation, so it would be advantageous to construct descriptions which are exhaustive in the sense that they do constrain the set of possible situations to just that one of interest.

When a description admits more than one model the conclusions drawn are of greater generality. Any conclusion is valid for all the situations which fit the description not just the situation of interest. It seems self-evident that generality costs. A reasonable goal would be then to reason in no more generality than necessary.

Consider again the spectrum of analogues. Although it is theoretically possible to provide adequate descriptions for the denotative model, it is seldom the case that they are provided. At the end of the spectrum requiring the least formal description, the requisite information is contained implicitly in the physical medium or data structure. The medium acts as a complete description of itself. Further along in the spectrum, the properties of the medium have to be described as well, and it is often precisely some of these properties which are overlooked or taken for granted when formalizing a domain. Thus invoking analogues at the world end of the spectrum is a way of utilizing information which is present in the representing medium, but which is not yet sufficiently understood to characterize descriptively.

Admitting information into the problem sclving process in this way avoids two problems. The first involves reasoning in greater generality and hence at greater cost. The second involves solving subproblems which require deducing results about the medium which are harder to obtain than a solution to the original problem appears to be.

In avoiding this second problem it is often necessary to give such high level tailor-made descriptions that the solution is virtually contained in the description. This is the case, for example, in the 'monkey and bananas' problem where usually it is given that if the monkey stands on the box, then it will be able to grasp the bananas (This is not to say that the monkey does not require some high level qualitative knowledge and a high level planner, just as WHISPER does). If this is not given, how is the monkey to deduce that standing on the box will solve its problem? A real monkey in a real situation could simply try the experiment of standing on the box; an abstract monkey thinking about the abstract situation would have to be given further information about its height, the length of its arm, the height of the box, and the height of the bananas. If any of this information is omitted, it will not be able to solve the problem. In addition, the subproblems of determining the distance from the top of the box to the bananas and the length of the monkey's reach are now becoming more difficult problems than the main problem of hypothesizing the box as the possible key to the solution.

The advantage of using analogues is, therefore, that they provide a way of bringing information present as properties of the medium into the problem solving process without explicitly knowing and describing what these properties are. Furthermore, the presence of this information is important in both (a) reducing the generality of the reasoning by eliminating the unanticipated matching of one situation's description by other situations, and (b) eliminating the need to deduce results about the medium itself from a description of its properties. This is true whether the medium is a physical one existing outside the conventional confines of a machine, or existing within through simulation. A medium can be simulated, as in the case of the simulation of two-dimensional media by two-dimensional arrays, without first describing its properties. Thus even in the case of a simulated medium, when it is used as the basis for an analogue, its properties still become part of the problem solving process without the need of prior description or formalization.

Chapter V: Visual Imagery And WHISPER's Retina

V-1 Introduction

WHISPER's use of array diagrams parallels human use of diagrams. Nevertheless it can also use analogues without an array diagram by creating 'images' on its retina paralleling human visual imagery. The retina can 'look at' and process information represented in a non-pictorial form.

Analogues very similar to those provided by the array diagrams can be created and used directly on WHISPER's retina, because it also has two-dimensional structure. The retina is basically a two-dimensional array of processors, although, in contrast to the diagram array, it is circular, not square. The neighbourhood linking of each processor and the way that processor values are set is responsible for the two-dimensional character of the retina. Of course, the processors themselves need not be aligned in a plane as they are depicted in the picture of WHISPER's retinal bubbles, figure IV-21.

The diagram can be eliminated and the analogical use of diagrams retained, if there is a method of filling the retina from an underlying representation distinct from a diagram. Since all of WHISPER's interaction with the diagram is through the retina, it is dependent only on the retina and not on the diagram itself. As long as there is a mechanism whereby the retina is filled correctly, WHISPER will be unaware that a diagram is missing. The method of filling the retina from the underlying representation must be fast, however, because the retina is refilled every time it fixates at a new location.

V-2 Filling The Retina

The retina's parallelism can be exploited in mapping from an underlying representation to the retina. Assume that the underlying representation is in the form of a set of straight line equations together with the coordinates of the endpoints of the line segments. For each line segment equation, the retinal supervisor would broadcast it and its endpoint coordinates to all the bubbles, asking them each to execute the following simple algorithm:

(1) Let D equal the perpendicular distance from the center of the bubble to the line.

(2) If D is less than the radius of the bubble, then set the bubble value 'on'.

(3) Otherwise, do nothing.

The time taken to map from the underlying line segment description to the retina would be directly proportional to the number of line segments.

The same method is applicable to mapping descriptions based upon other types of primitives (other curves, surfaces, or volumes) for which there is a simple way each bubble can determine if it is within the section of the retina affected by the presence of a primitive element. Irregularly shaped areas could be described by their decomposition into triangularly shaped pieces (figure V-1). In filling the retina, each bubble would only have to determine its own presence within the area

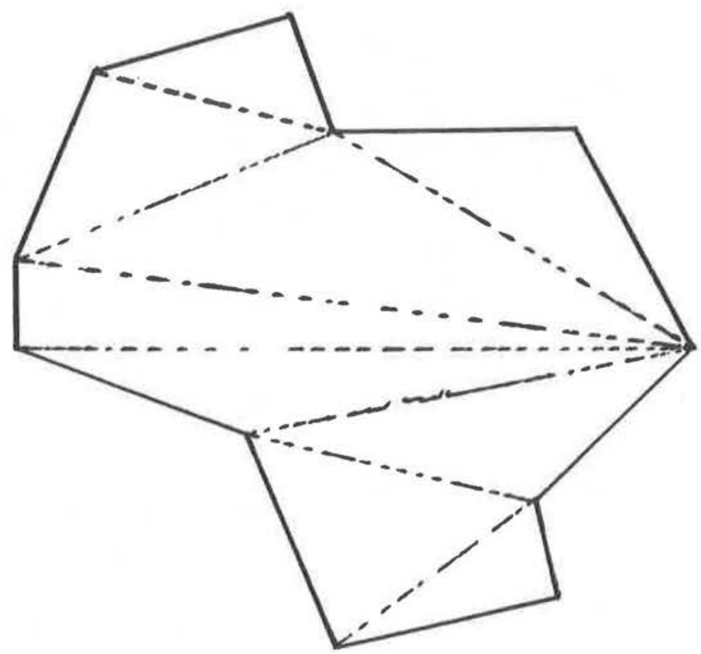


FIGURE V-1

defined by a triangle. The time required would be proportional to the number of triangles in the decomposition. In the case of a 3-D primitive, each bubble would be required to determine whether it is in that section of the retina lying under its 2-D projection.

V-3 Advantages

The above scheme has several advantages over the array representation of diagrams:

(i) There is a significant saving in storage space. Describing surfaces in terms of primitive segments of area is much more compact than saving all the points which are on the surface.

(ii) The resolution is less limited. The retina could zoom in on any section of the represented entity to whatever extent it wished. In the case of the array, it could not obtain better resolution than that defined by the size of the array. Using the proposed scheme, resolution is limited only by the number of primitives which are used in describing the entity. This number can be increased indefinitely when the description is constructed. If suitably organized, only a subset of these need be projected onto the retina at any one time; more can be projected if greater resolution is required for some purpose.

(iii) There is the possibility of utilizing 3-D entities. Each fixation would only provide a 2-D projection, but the fixations could be made from different perspectives.

The primary difficulty which arises in replacing the array diagram with a different description is in applying transformations to it. Non-linear transformations would cause the greatest difficulty because the shape of primitive elements would be changed. For linear transformations, the transformation would be applied identically to each of the primitive elements. In line segment descriptions, for example, the recomputation of the endpoints of each segment according to the transformation is all that is necessary. If the primitive elements are solids, described in terms such as Fahlman's AT arrays which utilize a homogeneous coordinate representation, then transformation can again be applied uniformly to each element.

V-4 Relationship To Visual Imagery

Employing a WHISPER-like retina filled from a non-pictorial underlying representation agrees with Kosslyn's theory of human visual imagery.

"A computer graphics metaphor: A visual image is considered to bear the same basic relationship to its underlying structure as a pictorial display on a cathode ray tube (CRT) does to the computer program that generates it. The underlying 'deep' structure is abstract and not experienced directly, whereas the image itself seems pictorial in nature. We are not claiming, however, that the psychological analogue to the CRT displays pictures as such; rather, this structure is characterized as supporting internal representations (whatever they may be like) similar to those that engender the experience of perceiving a picture when a person is viewing one."[30]

The value cells of the bubbles of WHISPER's retina serve the role of the display screen, and the function of the electron beam is replaced by parallel execution of the retinal processors in filling this 'screen'. The retinal structure does support an internal representation similar to that produced by viewing one of WHISPER's array diagrams (the closest thing to viewing a picture). Kosslyn also suggests that "subroutines for displaying lines, arcs, and a set of basic patterns might serve as primitives" (original emphasis) in a "hierarchical representation" from which the display would be generated.[31] Again this is similar to the above proposal for replacing the array diagram with a line drawing composed of line segment primitives.

One hypothesis Kosslyn is lead to by the graphics metaphor is that there would be capacity limitations. There might be limitations on how detailed the generating description could be, and limitations on the amount of information that the display itself could hold at one time. This latter limitation, he suggests, might be like the flicker occurring in CRT displays when refreshed with insufficient frequency as happens when there is a large number of lines to be drawn on the screen.

WHISPER's retina suggests a different source of 'flicker' than the simple fading suggested by analogy with the CRT's phosphor. Because the primitives (e.g. line segments) must be displayed serially, as their number increases, so does the time taken to 'draw' the image on the retina. Since WHISPER is dependent on fixating the retina at many different locations, an extreme increase in fixation time would overload the system with the overhead of moving the retina, leading to a limit on the practical amount of information which can be displayed.

Chapter VI: Conclusion

VI-1 Limitations And Future Directions

VI-1.1 Physical Knowledge

WHISPER's knowledge of Physics is far from comprehensive. The 'snapshot' by its very nature portrays all objects as stationary, whereas some may be moving. To take velocities into account requires the addition of a quantitative reasoning component to its current qualitative knowledge. WHISPER currently does not integrate knowledge of velocity, acceleration, momentum or moments of inertia. These would have to be represented in terms of equations which could be applied after WHISPER makes its current predictions.

WHISPER approximates simultaneity by moving objects one after another. This process works for the problems discussed in Chapter II; however, cases exist for which this approximation is insufficient. In figure VI-1, for example, if B is moved after A is moved, then they will not collide; however, if they are moved simultaneously they will collide. The diagrammatic analogue can be used to overcome this problem by shading the space swept out by each object as it moves. The shaded areas of the diagram could be examined for overlap,

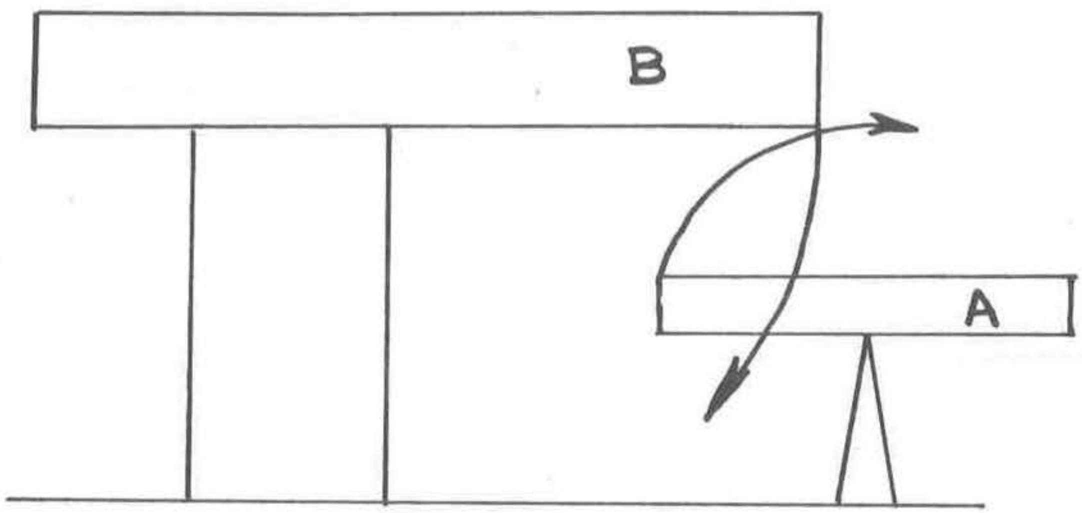


FIGURE VI-1

thereby indicating that a collision might occur. If they do overlap, further quantitative analysis of the angular velocities of both objects is necessary.

VI-1.2 The Retina And Perceptual Primitives

The restricted resolution of the retina is its principal limitation. The resolution could be increased by adding more bubbles, but this makes no theoretical difference to the types of perceptual primitives it could compute.

There are two main directions for further research relating to the retina:

(i) Replace the software simulation with parallel processing hardware.

(ii) Experiment with other retinal geometries and communication links. The rotational visualization and the scaling property are artifacts of the current geometry. Other geometries or extra communication links might facilitate the visualization of translations or rotations of 3-D objects. A different geometry might also affect the rate at which resolution decreases with increasing distance from the retinal center. A retina whose resolution decreases at the same rate as the acuity of the human eye decreases might be a good choice.

The current set of perceptual primitives is adequate for WHISPER's two-dimensional blocks domain. A significant extension to the perceptual primitives would involve the addition of primitives that extract features from two-dimensional projections of three-dimensional scenes.

VI-1.3 Limitations Of Analogues

I am not claiming that analogues are advantageous in all situations; they do have limitations. In section IV-5 I discussed how analogues help reduce the level of generality. It is not always best to reason about a specific rather than a general situation. Sloman gives an excellent example: "If I start in room A and then move back and forth between room A and room B, which room will I be in after exactly 377 moves?"[32] We do not want to perform an experiment in an analogue of this situation to find the answer, although some experimentation with an analogue could help formulate or substantiate a general hypothesis about the location after an odd number of moves.

In section IV-4.2.1 and IV-4.2.4 examples were given of how analogues can establish a reason or evidence for an hypothesis. This does not, however, guarantee that the hypothesis is valid; a good deal of effort could be expended in vain attempts to prove it. Thus relationships in the analogue may be misleading. For example, if two angles of a triangle appeared equal in a geometry diagram the hypothesis that it is isosceles is substantiated, but unfortunately so, if the equality of the angles is coincidental. Some features of the analogue may not be generalizable at all, and if the ones which do not generalize are not known, the analogue can be misleading. For example, WHISPER knows that an object which is symmetrical about a vertical axis through its support point will balance (section II-3.3.1). However, if the objects were not of uniform density the symmetry of an object could be taken as evidence that it would balance, but it would be purely coincidental if this hypothesis were valid.

VI-1.4 Psychological Correlation

Since WHISPER's retina fixates at a sequence of points on a diagram understandable by a human, one can directly compare human and machine problem solving behaviour. One of WHISPER's eye movement protocols was given in section II-6. The eye movement protocol of a human subject could be recorded by presenting him with one of WHISPER's diagrams. This has not been done, but could be an interesting exploratory area. Using WHISPER as a model for interpreting the results of human eye movement might reveal something of the human subject's knowledge of Physics, his problem solving strategy, and his set of perceptual primitives.

VI-2 Key Ideas

A brief summary of the important ideas presented in this paper follows.

VI-2.1 The Analogue

WHISPER reasons by experimenting with an analogue of blocks world situations. The analogue consists of a diagram and transformation procedures which modify it. The static configuration of marks in the diagram is analogous in terms of spatial relations to the configuration of objects in the real world, and the behaviour of objects in the diagram under the linear transformations is analogous to the behaviour of physical objects. WHISPER's interaction with the analogue is shown in figure VI-2.

VI-2.2 Benefits Of The Analogue To WHISPER

(a) Arbitrary Object Shapes: There are no object descriptions except for the object itself as it appears in the diagram. As a result WHISPER handles objects of arbitrary two-dimensional shape.

(b) No Frame Problem: The diagram contains all the information concerning the shapes and positions of objects which WHISPER needs to continue its reasoning after a change in the situation. Only the positions and support relationships of

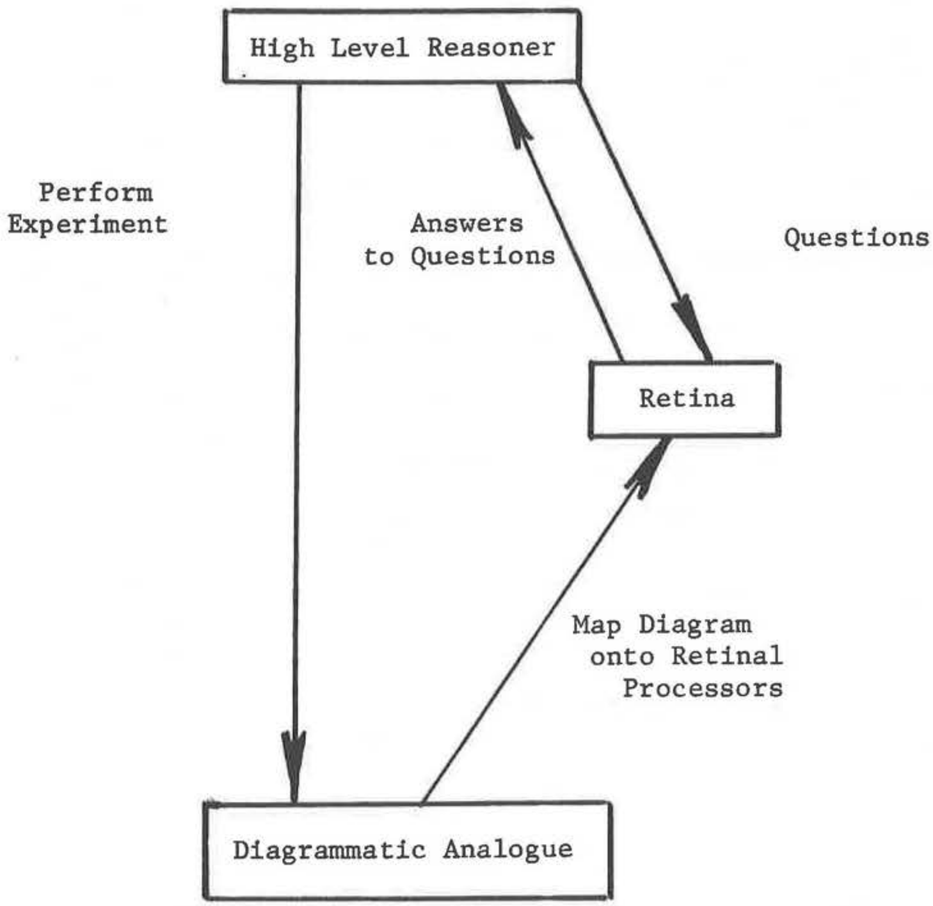

High Level Reasoner
Perform
Experiment
Answers
to Questions
Questions
Retina
Map Diagram
onto Retinal
Processors
Diagrammatic Analogue

FIGURE VI-2.

objects which should be affected by a change are affected.

(c) Amalgamation: Because the only description of the shape of an object is the object itself, amalgamating the shapes of two objects is simply a matter of ignoring their 'colour' difference.

(d) Emergent Properties: These relate to the amalgamation problem. When two objects are combined, coincidental alignments may cause the combined object to have an interesting or useful property. For WHISPER one such useful property is that two or more objects may align so that their surfaces provide a continuous smooth curve for a sliding object.

(e) Description Of Empty Space: Space is represented by space in the diagram. There is no difficulty describing the areas of empty space. This is important when finding a clear path for an object. There is no need to prove that a particular point in space is unoccupied.

(f) Detecting Motion Discontinuities: Retinal visualization can be used for detecting collisions during rotations. Discontinuities in sliding motions are found by examining the sliding surfaces for bumps, hills, cliffs, and objects in the way.

VI-2.3 The Retina

(a) Parallel Processing: The retina consists of a large but fixed number of processors operating in pseudo-parallel. There is no need to dynamically spawn new processors.

(b) Retinal Supervisor: A single sequential processor which directs the parallel processors.

(c) Neighbourhood Message Passing: Each processor exchanges messages with its immediate neighbours and with the retinal supervisor. The restriction to neighbourhood communication is important in facilitating a future hardware implementation of the retina.

(d) Retinal Topology: The current retinal layout has several useful properties: (i) The resolution decreases from the retinal center. (ii) Objects can be scaled by message passing along wedge links. (iii) Objects can be rotated about the retinal center by message passing along ring links.

(e) Domain Independent Perceptual Primitives: The perceptual primitives extract features from diagrams. These features are assigned different interpretations depending upon the problem domain. The current set of percepts includes: similarity, center of area, symmetry, contact points, visualization of rotations to discover collisions, nearest and farthest locations satisfying an arbitrary predicate, curve tangents, convexities and concavities.

Footnotes

1. Marvin Minsky and Seymour Papert, Artificial Intelligence Progress Report, Cambridge, Mass.: Massachusetts Institute of Technology, AI Memo No. 252, 1972.

2. G. Polya, Mathematical Discovery: On Understanding, Learning, and Teaching Problem Solving, New York: John Wiley & Sons, vol. II, p.8.

3. Loc. Cit.

4. Scott E. Fahlman, A Planning System for Robot Construction Tasks, Cambridge, Mass.: Massachusetts Institute of Technology, AI Technical Report No. 283, 1973.

5. Johan deKleer, Qualitative and Quantitative Knowledge in Classical Mechanics, Cambridge, Mass.: Massachusetts Institute of Technology, M.Sc. Thesis, 1975.

6. WHISPER's problem domain was suggested by Alan Mackworth in discusssion with Raymond Reiter.

7. Terry Winograd, Procedures as a Representation for Data in a Computer Program for Understanding Natural Language, Cambridge, Mass.: Massachusetts Institute of Technology, PhD. Thesis, MAC Technical Report No. 84.

8. Scott E. Fahlman, op. cit., p.134-135.

9. Charles M. Eastman, "Automated Space Planning," Artificial Intelligence, vol. 4(1973), pp.41-64.

10. Scott E. Fahlman, op. cit., p. 100.

11. B. Raphael, "The Frame Problem in Problem-Solving Systems," Artificial Intelligence and Heuristic Programming, ed. N.V. Findler and B. Meltzer, Edinburgh: Edinburgh University Press, 1971, pp.159-169.

12. P. Hayes, "A Logic of Actions," Machine Intelligence 6, ed. B. Meltzer and D. Michie, New York: American Elsevier Publishing, 1971, pp.495-520.

13. M.H. Pirenne, Vision and the Eye, London: Chapman & Hall, 1967, p. 32.

14. Marvin Minsky and Seymour Papert, Perceptrons: An Introduction to Computational Geometry, Cambridge, Mass.: M.I.T. Press, 1969.

15. D.V. McDermott, CONNIVER Reference Manual, Cambridge, Mass.: Massachusetts Institute of Technology, AI Memo No. 259, 1972.

16. B.Wilcox and G.J.Sussman, LISP/MTS User's Guide, Ann aRbor, Michigan: Mental Health Research Institute, 1973.

17. G. Polya, Induction and Analogy in Mathematics: Volume I of Mathematics and Plausible Reasoning, Princeton, N.J.: Princeton University Press, 1954, p.13.

18. Aaron Sloman, "Afterthoughts on Analogical Representations," Theoretical Issues in Natural Language Processing, ed. R. Schank and B.L. Nash-Webber, Conference Proceedings Cambridge, Mass., June 10-13, 1975,

19. John Seely Brown, Richard R. Burton, and Allan G. Bell, SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An Example of AI in CAI) Cambridge, Mass.: Bolt Beranek and Newman, BBN Report No. 2790, AI Report No. 12, 1974.

20. Zenon W. Pylyshyn, "Representation of Knowledge: Non-Linguistic Forms - Do We Need Images and Analogues?" Theoretical Issues in Natural Language Processing, ed. R. Schank and B.L. Nash-Webber, Conference Proceedings Cambridge, Mass., June 10-13, 1975, p. 174.

21. Stephen M. Kosslyn, "Information Representation in Visual Images," Cognitive Psychology, vol.7(1975), pp.341-370.

22. Charles M. Eastman, *op. cit.*, p. 47.

23. Aaron Sloman, *op. cit.*, p. 179.

24. G. Polya, *How to Solve It: A New Aspect of Mathematical Method*, Princeton, N.J.: Princeton University Press, 1945(2nd ed. 1957), p. 73.

25. Brian V. Funt, *A Procedural Approach to Constructions in Euclidean Geometry*, Vancouver, B.C.: University of British Columbia, M.Sc. Thesis, 1973.

26. H. Gelernter, "Realization of a Geometry-Theorem Proving Machine," *Computers and Thought*, ed. E.A. Feigenbaum and J. Feldman, New York: McGraw-Hill, 1963, pp. 134-152.

27. M.D.Kelly, "Edge Detection in Pictures by Computer Using Planning," *Machine Intelligence 6*, ed. B.Meltzer and D.Michie, New York: American Elsevier Publishing, 1971, pp.397-409.

28. Y.Shirai, "A Context-Sensitive Line Finder for Recognition of Polyhedra," *Artificial Intelligence*, vol.4(1973), pp.95-199.

29. Marvin Minsky, *A Framework for Representing Knowledge*, Cambridge, Mass.: Massachusetts Institute of Technology, AI Memo No. 306, 1974, p.76.

30. Stephen M. Kosslyn, "On Retrieving Information from Visual Images," *Theoretical Issues in Natural Language Processing*, ed. R. Schank and B.L. Nash-Webber, Conference Proceedings Cambridge, Mass., June 10-13, 1975, p. 160.

31. *Ibid.*, p. 160.

32. Aaron Sloman, *op. cit.*, p.180.

Bibliography

1. Arnheim, Rudolf. *Visual Thinking.* Berkeley, Calif.: University of California Press, 1969.

2. Baker, Richard. "A Spatially-Oriented Information Processor which Simulates the Motions of Rigid Objects." *Artificial Intelligence.* Vol. 4 (1973), pp.29-40.

3. Brown, John Seely, Burton, Richard R., and Bell, Alan. *SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An Example of AI in CAI)* Cambridge, Mass.: Bolt Beranek & Newman, 1974, BBN Report No. 2790, AI Report No. 12.

4. Chafe, Wallace L. "Creativity in Verbalization as Evidence for Analogic Knowledge." *Theoretical Issues in Natural Language Processing.* Ed. R. Schank and B.L. Nash-Webber, Conference Proceedings Cambridge, Mass., June 10-13, 1975, pp.158-159.

5. Codd, E.F. *Cellular Automata.* New York: Academic Press, 1968.

6. deKleer, Johan. *Qualitative and Quantitative Knowledge in Classical Mechanics.* Cambridge, Mass.: Massachusetts Institute of Technology, M.Sc. Thesis, 1975.

7. Dreyfus, Hubert L. *What Computers Can't Do: A Critique of Artificial Reason.* New York: Harper & Row, 1972.

8. Eastman, Charles M. "Automated Space Planning." *Artificial Intelligence.* Vol. 4 (1973), pp.41-64.

9. Eastman, Charles M. "Representations for Space Planning." *Communications of the ACM.* Vol. 13, no. 4 (April 1970), pp.242- 250.

10. Fahlman, Scott E. *A Planning System for Robot Contruction Tasks.* Cambridge, Mass.: Massachusetts Institute of

Technology, AI Technical Report No. 283, 1973.

11. Funt, Brian V. A Procedural Apporach to Constructions in Euclidean Geometry. Vancouver, B.C.: University of British Columbia, M.Sc. Thesis, 1973.

12. Gelernter, H. "Realization of a Geometry-Theorem Proving Machine." Computers and Thought. Ed. E.A. Feigenbaum and J. Feldman. New York: McGraw-Hill, 1963, pp.134-152.

13. Gregory, R.I. Eye and Brain: The Psychclogy of Seeing. New York: McGraw-Hill, 1966(2nd ed. 1973).

14. Hayes, P. "A Logic of Actions." Machine Intelligence 6. Ed. B. Meltzer and D. Michie. New York: American Elsevier Publishing Co., 1971, pp.495-520.

15. Hayes, P. "Some Problems and Non-Problems in Representation Theory." AISB Summer Conference Proceedings, July 1974, pp.63-79.

16. Kelly,M.D. "Edge Detection in Pictures by Computer Using Planning." Machine Intelligence 6. Ed. B.Meltzer and D.Michie. New York: American Elsevier Publishing Co., 1971, pp.397-409.

17. Kosslyn, Stephen M. "Information Representation in Visual Images." Cognitive Psychology. Vol. 7 (1975), pp.341-370.

18. Kosslyn, Stephen M. "On Retrieving Information from Visual Images." Theoretical Issues in Natural Language Processing. Ed. R. Schank and B.I. Nash-Webber. Conference Proceedings Cambridge, Mass., June 10-13, 1975, pp.160-164.

19. McDermott, D.V. And Sussman, G.J. CONNIVER Reference Manual. Cambridge, Mass.: Massachusetts Institute of Technolcgy, AI Memo No.306, 1974.

20. Minsky, Marvin. A Framework for Representing Knowledge. Cambridge, Mass.: Massachusetts Institute of Technology, AI Memo No. 306, 1974.

21. Minsky, Marvin and Papert, Seymour. Artificial Intelligence Progress Report. Cambridge, Mass.: Massachusetts Institute of Technology, AI Memo No. 252, 1972.

22. Minsky, Marvin and Papert, Seymour. Perceptrons: An Introduction to Computational Geometry. Cambridge, Mass.: M.I.T. Press, 1969.

23. Norman, Donald A. Memory, Knowledge, and the Answering of Questions. San Diego, Calif.: University of California, Technical Report Chip 25, 1972.

24. Pirenne, M.H. Vision and the Eye. London: Chapman & Hall, 1967.

25. Polya, G. How To Solve It: A New Aspect of Mathematical Method. Princeton, N.J.: Princeton University Press, 1945(2nd ed. 1957).

26. Polya, G. Induction and Analogy in Mathematics: Volume I of Mathematics and Plausible Reasoning. Princeton, N.J.: Princeton University Press, 1954.

27. Polya, G. Mathematical Discovery: On Understanding, Learning, and Teaching Problem Solving. New York: John Wiley and Sons, vol. 1: 1962, vol. 2: 1965.

28. Polya, G. Patterns of Plausible Inference: Volume II of Mathematics and Plausible Reasoning. Princeton, N.J.: Princeton University Press, 1954.

29. Pylyshyn, Zenon W. "Representation of Knowledge: Non-Linguistic Forms - Do We Need Images and Analogues?" Theoretical Issues in Natural Language Processing. Ed. R. Schank and B.L. Nash-Webber. Conference Proceedings Cambridge, Mass., June 10-13,1975, pp.174-177.

30. Pylyshyn, Zenon W. "What the Mind's Eye Tells the Mind's Brain: A Critique of Mental Imagery." Psychological Bulletin. Vol. 80 (1973), pp.1-24.

31. Raphael, B. "The Frame Problem in Problem-Solving Systems." _Artificial Intelligence and Heuristic Programming._ Ed. N.V. Findler and B. Meltzer. Edinburgh: Edinburgh University Press, 1971, pp.159-169.

32. Shirai, Y. "A Context-Sensitive Line Finder for Recognition of Polyhedra." _Artificial Intelligence._ Vo.4(1973), pp.95-199.

33. Slcman, Aaron. "Afterthoughts on Analcgical Representations." _Theoretical Issues in Natural Language Processing._ Ed. R. Schank and B.L. Nash-Webber. Conference Proceedings Cambridge, Mass., June 10-13,1975, pp.178-182.

34. Sloman, Aaron. "Interactions Between Philosophy and Artificial Intelligence: The Role of Intuition and Non-Logical Reasoning in Intelligence." _Artificial Intelligence._ Vcl. 2 (1971), pp.209-225.

35. Wilcox, B. And Hafner, C. _LISP/MTS User's Guide._ Ann Arbor, Michigan: Mental Health Research Institute, 1973.

36. Winograd, Terry. _Procedures as a Representaticn for Data in a Computer Program for Understanding Natural Language._ Cambridge, Mass.: Massachusetts Institute cf Technclogy, PhD Thesis, MAC Technical Report No. 84, 1971.