

# WHISPER: A PROBLEM-SOLVING SYSTEM UTILIZING DIAGRAMS

## AND A PARALLEL PROCESSING RETINA

Brian V. Funt

Artificial Intelligence Laboratory"  
Stanford University, Stanford CA

### ABSTRACT

Diagrams are of substantial benefit to WHISPER, a computer problem-solving system. In testing the stability of a "blocks world" structure and predicting the event sequences which occur as that structure collapses. WHISPER'S components include a high level reasoner which knows some qualitative aspects of Physics, a simulated parallel processing "retina" to "look at" its diagrams, and a set of re-drawing procedures for modifying these diagrams. Roughly modelled after the human eye, WHISPER'S retina can fixate at any diagram location, and its resolution decreases away from its center. Diagrams enable WHISPER to work with objects of arbitrary shape, detect collisions and other motion discontinuities, discover coincidental alignments, and easily update its world model after a state change.

### 1. INTRODUCTION

Diagrams are very important tools which we use daily in communication, information storage, planning and problem-solving. Their utility is, however, dependent upon the existence of the human eye and its perceptual abilities. Since human perception involves a very sophisticated information processing system, it can be argued that a diagram's usefulness results from its suitability as an input to this powerful visual system. Alternatively, diagrams can be viewed as containing information similar to that contained in the real visual world, the canonical entity the human visual system was presumably designed through evolution to interpret. From this latter perspective, diagrams are a natural representation of certain types of primarily visual information, and the perceptual system simply provides an appropriate set of database accessing functions. Both these viewpoints underly the work described in this paper.

The role of diagrams is explored in a computer problem-solving program, named WHISPER, which refers to diagrams during its processing. A strong case for computer use of diagrams as models for Geometry has been made by Gelernter[1963], and as general analogical representations by Sloman[1971]. Hayes[1974] discusses the theoretical nature of these representations. WHISPER'S high-level reasoning component (HLR), built along the lines of traditional procedural AI problem-solving programs, has the additional option of requesting observations in a diagram. It does this by asking its "perceptual system" to "look at" the diagram with its parallel processing "retina". The questions that the perceptual system can answer are called perceptual primitives. If necessary, the HLR can also make changes to the current diagram.

A major portion of this research was done at the University of British Columbia and is fully described in the author's Ph.D. thesis[Funt 76].

### 1.1 MECHANISM? PPF PIAQRAM INTERACTION

The retina and perceptual primitives are designed to provide WHISPER with a new set of operations whose execution times are of the same order of magnitude as conventional machine instructions. To achieve this a high degree of parallelism has been incorporated into the system. The retina is a parallel processor, and the perceptual primitives are the algorithms it executes. (Do not be misled by the term "retina"; it refers to a general system of receptors and processors for the early stages of perceptual processing, rather than implying any close resemblance to the human retina.) Each perceptual primitive, when executed by the retina, determines whether some particular feature is present in the diagram. WHISPER'S retina mixes parallel and sequential computation, so the features it can recognize are not subject to the same theoretical limitations that perceptrons [Minsky and Papert 1969] are.

#### 1.1.1 The Retina

WHISPER'S retina is a software simulation of hardware which, given the rapidly advancing state of LSI technology, should soon be possible to build. It consists of a collection of processors, each processor having its own input device called a receptor. As with the human eye, WHISPER'S retina can be shifted to fixate at a new diagram location (also a feature of a program by Dunlavey[1976]), so that each processor's receptor receives a different input from the diagram. This fixation facility is important because the resolution of the retina decreases from its center to its periphery. Without being able to fixate, it would be impossible for WHISPER to examine the whole diagram in detail. Economy of receptors and processors dictates the use of decreasing resolution. (A declining resolution is also a characteristic of the human eye.) Each receptor covers a separate segment of the diagram and transmits a single value denoting the color of that region. The geometrical arrangement of the receptors and the area each covers is shown in Figure 1. The "circles" in the figure are called bubbles, and they are arranged in wedges (rays emanating from the center) and ring? (one of the concentric circles of bubbles). The resolution varies across the retina because a larger portion of the underlying diagram is mapped onto a bubble depicted by a larger circle. Since the complete group of receptors is assumed to sense and transmit all signals in parallel, fixations are fast.

Each retinal processor has direct communication links to its 4 nearest neighbors plus one additional link via a common databus connecting all the processors to a supervisory processor called the retinal supervisor. The communication topology has been restricted in this simple way to ensure a feasible future hardware implementation.

The bubble processors are each small computers with

Independent memory. They all simultaneously execute the same procedure; however, each bubble does not necessarily execute the same instruction at the same time. In the current implementation, a call to the LISP evaluator simulates a processor; and LISP mapping functions simulate the parallel control structure.

Although the bulk of the processing of the perceptual primitives is done in parallel, there is also a small amount of sequential processing which is performed by the retinal supervisor. The retinal supervisor also directs the parallel processing by choosing which procedure the bubbles should execute next and broadcasting it to them.

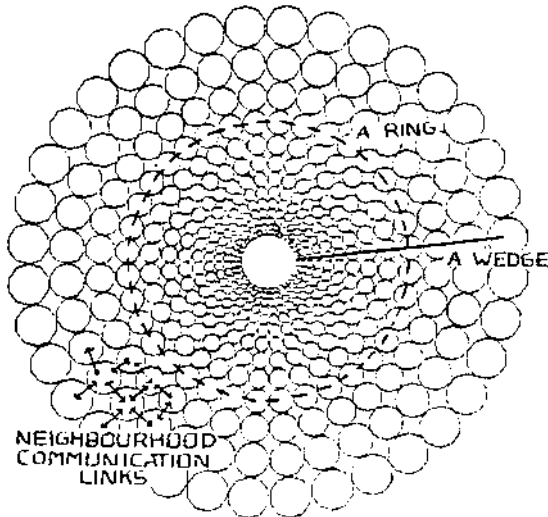


Figure 1

### 11.2 The Perceptual Primitives

Each perceptual primitive detects a problem domain independent diagram feature. The HLR assigns these features interpretations pertinent to the problem it is solving. The current set of implemented perceptual primitives include ones to: find the center of area of a shape; find the points of contact between a shape of one color and a shape of another; examine curves for abrupt slope changes; test a shape for symmetry; test the similarity of shapes; and visualize the rotation of a shape while watching for a collision with another shape.

The center of area perceptual primitive is an illustrative example of the general operation of the perceptual primitives. It computes the center of area of a shape relative to the origin defined by the center of the retina. For each piece,  $AA$ , of the total area we need to compute the  $x$  and  $y$  components of its contribution to the total area. Dividing the vector sum of these contributions by the total area yields the coordinates of the center of area. Since each retinal bubble receives its input from a fixed sized area of the diagram and is at a fixed location relative to the retina's center, each bubble can independently compute the components of its contribution to the total area. The bubbles whose receptors do not lie over any part of the shape simply do not contribute. The retinal supervisor performs the summation and the division by the total area. A separate primitive computes the total area. It simply totals the area of

all the contributing bubbles. If the computed center of area is far from the retina's center then its accuracy can be improved by fixating the retina on the estimated center of area and then recomputing. The decision to iterate is made by the retinal supervisor. The accuracy improves because more of the central, high resolution portion of the retina is used.

The center of area is used for more than simply providing the center of gravity of the objects in WHISPER'S problem domain. Other primitives (symmetry, similarity, and contact finding) fixate on a shape's center of area before beginning their calculations. For example, if a shape is symmetrical its center of area will be on its axis of symmetry.

Another important primitive is visualization. What is "visualized" is the rigid rotation of a shape about the retinal center. While the shape is rotating the collision detection primitive can be called a demon to watch whether the rotation causes the shape to overlap with another stationary shape. This is useful both in "blocks world" environments involving moving objects and in testing whether two shapes are equivalent under rotation. The process is termed visualization because it does not involve modifying the diagram, but instead is totally internal to the retina itself. It simply entails an organized and uniform exchange of information amongst neighboring bubbles.

The geometrical arrangement of the bubble receptors is what facilitates the visualization of rotations. From Figure 1 it can be seen that aligning the bubble centers along wedges results in a constant angular separation between bubbles of the same ring when they are from neighboring wedges, and that this constant is independent of the ring chosen. Thus, to rotate a shape clockwise each bubble marked by the shape simply sends a message to its clockwise ring neighbor asking it to mark itself. The sender then erases its own mark. A collision is detected if a bubble receives a message to mark when it is already marked by a shape other than the rotating one. Although the shape is rotated in sequential steps, the time required is still short because (i) there are, as a maximum, only as many steps to be made as there are wedges on the retina (currently 36); and (ii) all the message passing and collision checking occurs in parallel during each step.

### 11.3 The Underlying Diagram

We began with the view that the retina is a special purpose parallel processor designed to detect diagrammatic features without saying anything about the precise nature of the diagrams themselves. With the retinal processor in hand, we can now see that the representation of the diagrams is unimportant as long as each bubble receives its correct input. This is analogous to a program which issues a READ command without caring whether the input is coming from a card reader, a file, or a terminal. The method of mapping from the diagram to the retinal bubbles' input must be fast, however, because the retina is re-filled everytime it is fixated at a new diagram location.

There are at least two different types of representing media for the underlying diagram. The first is the conventional medium of visible marks on a two-dimensional surface, usually paper. The map from diagram to human retina is accomplished by the lens of the eye focusing the incoming light. Since there is simultaneous stimulation of the receptors, it is a very fast process.

The second possible type of diagram representation is similar to that used in generating computer graphics. The diagram is specified as a list of primitive elements (in graphics applications, usually line segment equations), in a similar vein, Kosslyn [1975] proposes that human visual imagery is in some ways analogous to the storage and display of graphics images. The parallel processing capacity of WHISPER'S retina can be used to quickly map each primitive element into the proper bubble inputs. To mark all bubbles lying on line segment, S, the retinal supervisor directs every bubble to determine independently if it is on S, and if so, to mark itself. Since this simple test -- do a circle and a line segment intersect? -- is performed by all bubbles simultaneously, the time required is independent of the length of S. The same method can mark all bubbles within any simple shape such as a circle, square or triangle in time independent of its area. Regardless of the type of primitive element, the time taken to "draw" the diagram on the retina is, however, proportional to the number of primitives in its description. They must be processed sequentially. With the fast parallel processing of each primitive this should only pose a problem for very detailed diagrams.

Due to the lack of true parallel processing, neither of the above two types of diagram representations is used in WHISPER. Instead, the diagram is implemented as a square array. Each array cell denotes a point on a real world, pencil and paper diagram.

#### 11.4 The Re-drawing Transformations

The re-drawing transformations are the procedures the HLR can call to change the underlying diagram. In WHISPER there are transformations for adding and removing lines, and for rigidly translating and rotating shapes. Other non-linear transformations could be added if required. These re-drawing procedures are of course dependent upon the representation of the diagram they modify, and the ease and efficiency with which they can be implemented could affect the choice of diagram representation.

### III. WHISPER IN OPERATION

With the basic mechanisms for interaction with the diagram now understood, it is appropriate to see how they are used in the course of solving a problem. We will consider problems of the type: predict the sequence of events occurring during the collapse of a "blocks world" structure. The structure will be a piled group of arbitrarily shaped objects of uniform density and thickness. If the structure is stable, there are no events to describe; if it is unstable, then the events involve rotations, slides, falls, and collisions. WHISPER accepts a diagram of the initial problem state, and produces a sequence of diagrams, called snapshots, as its qualitative solution. A quantitative solution specifying precise locations, velocities, and times is not found; however, deriving one from a qualitative solution should not be too difficult [deKleer 1976].

Figure 2 is a typical example of WHISPER'S input diagrams. They all depict a side view of the structure. Each object is shaded a different "color" (alphanumeric value) so it can be easily distinguished and identified. Objects' boundaries are also distinctly colored. The diagram depicts a problem, called the "chain-reaction problem", which is particularly interesting because the causal connection between objects B and D must be discovered.

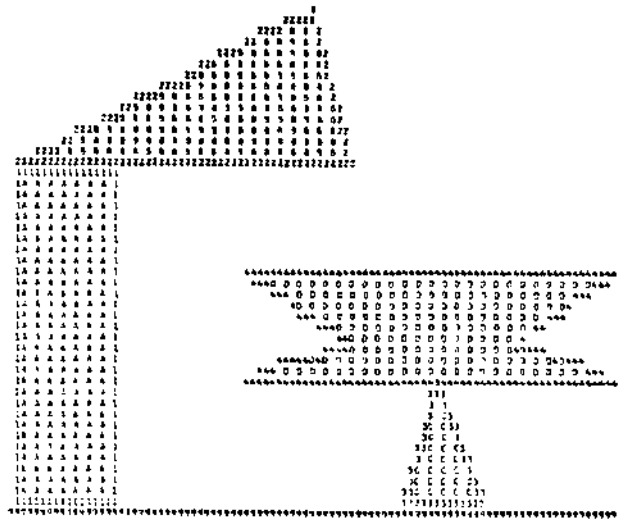


Figure 2

#### III.1 The Qualitative HLR

The HLR is the top level of the WHISPER system. It is solely responsible for solving each problem; the diagram and retina are simply tools at its disposal. It consists of procedural specialists which know about stability, about the outcome of different varieties of instability, how to interpret each perceptual primitive, and how to call the transformation procedures to produce the solution snapshots. There are two types of instabilities - rotational and sliding. For clarity, sliding instabilities will not be discussed for the present. Operation of the system follows the steps:

- (1) determine all instabilities
- (2) pick the dominant instability
- (3) find pivot point for rotation of unstable object
- (4) find termination condition of rotation using retinal visualization
- (5) call transformation procedure to modify diagram as determined in step (4)
- (6) output modified diagram as a solution snapshot
- (7) use snapshot from step (6) as input and restart from step (1).

In what follows we elaborate on each of these steps.

The diagram and retina are an invaluable aid to the HLR in discovering what stops an object's motion, and in accomplishing the necessary state change. The chain reaction problem demonstrates this. The stability specialist directs the retina to fixate at numerous locations while perusing the diagram, and from an analysis (discussed below) of the visible support relationships determines that B is the only unstable object. B will pivot about the support point closest to its center of gravity. The retina is fixated there (the top right corner of A), so B's rotation can be visualized. As the object rotates, two events are possible. It may

collide with another object, or it may begin to fall freely. The conditions under which either of these occur are monitored during the visualization. From this simulation of B's rotation, its collision with D is discovered, and its angle of rotation and location of first contact with D are found. Because of the coarseness of the retinal resolution, this angle of rotation is only approximate. This approximate value is used in conjunction with feedback from the diagram to refine the angle of rotation as follows. First the re-drawing transformations are called to produce a new diagram in which B is rotated by slightly less than the estimated value. The rotation is made on the short side so that B will not overshoot. The retina is then fixated on the anticipated point of collision so that the gap between the two objects can be examined. If there is none, then the update is complete; however, if there is, then B is rotated again until the gap is closed. The resulting diagram is output as WHISPER'S first snapshot of the solution sequence.

### III.2 Motion Discontinuities and Experimental Feedback

There are several important observations to be made at this point. One is that discovering the reason for the interruption of an object's motion, accomplished so simply here for B through visualization, is generally found to be a very difficult problem. Physics provides equations for object motions, but these equations describe a condition which theoretically lasts indefinitely. They do not indicate when new boundary conditions should take effect. Certainly it is possible to design a set of special heuristics specifying when and where collisions are most likely to occur (e.g. below the rotating object). However, it is quite probable that the collision occurring in Figure 3 would be overlooked, whereas WHISPER'S visualization process would definitely detect it.

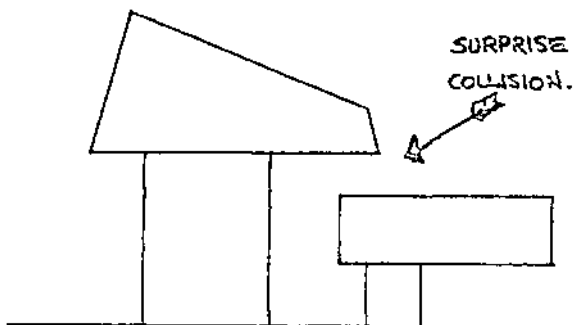


Figure 3

WHISPER relies on experimental feedback to successfully update its diagram in its method of visualization followed by gap closure. This method is basically a pragmatic equivalent to the unfeasible experiment of rotating the object in the diagram by very small increments until a collision occurs. Usually feedback is thought of in terms of a robot immersed in a real world environment. In WHISPER'S case, however, the feedback is from a situation analogous to that in the real world - the diagram and diagram transformations - rather than from observation of actual falling objects [Funt 1976]. Using this feedback WHISPER is able to find when and where discontinuous changes in an object's motion occur without being forced to use sophisticated, time-consuming, "number-crunching" algorithms for touch tests [Fahlman 1073] (for arbitrary shapes), and avoids considering every object in the

universe as a possible candidate for a collision.

Once WHISPER has produced the first snapshot, it is ready to compute the next one. All the information the HLR needs for this is contained in the first snapshot diagram. Thus to produce the next snapshot, the HLR takes its last output snapshot as input, and begins processing exactly as if it were working on a fresh problem. Although some results derived while working on the previous snapshot remain valid (e.g. some contact relationships will still hold), many will be inapplicable to the new problem. It is easier to disregard this old information than to sort it out and update it, since the retina provides a fast and efficient method of fetching it from the new diagram.

### HL3 Subsequent Snapshots of the Chain Reaction Problem

The analysis producing the second and third snapshots is very similar to that for the first. In Figure 4, B's weight on D causes D to be unstable. Its rotation is visualized with the retina fixated at the peak of C leading to the discovery of its collision with the table. The diagram is updated to produce the second snapshot, Figure 5, which is again input for further analysis. B now lacks sufficient support, and topples to hit D again as shown in Figure 6. The complexity of the problem rises sharply at this point, and WHISPER'S analysis ends, as, I believe, would most peoples'.

B and D could be shown to fall simultaneously (WHISPER currently does not) by rotating D only part of the way to the table before allowing B to catch up, and then iterating this process a few times until D reaches the table.

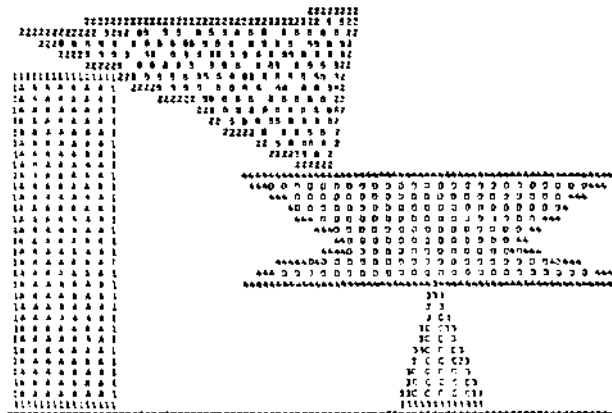


Figure 4

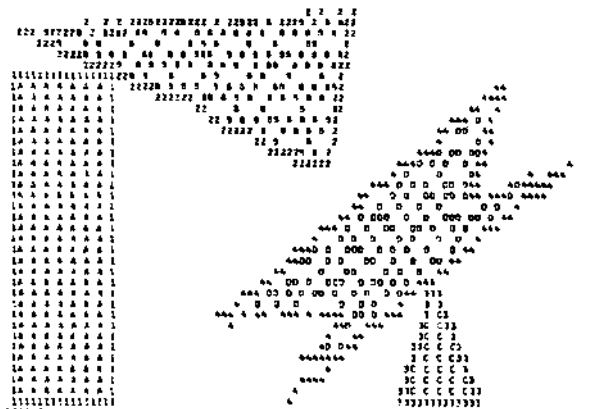


Figure 5

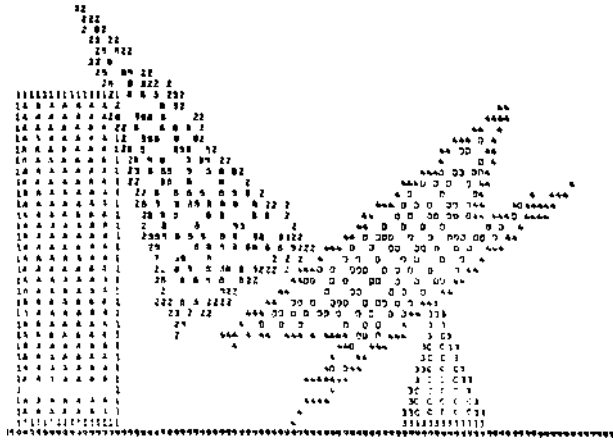


Figure 6

#### III.4 Slide Problems

Unstable objects may also slide. Although the basic outline of the solution process -- test stability, find termination point of motion, update diagram, output snapshot, restart with the output as Input -- is the same, there are some essential differences in handling sliding objects. The most important arises because it is not possible to visualize the slide of an object down an arbitrary curve. What WHISPER'S does instead is examine the curve itself with its retina.

A variety of conditions can terminate an object's slide. For example, there may be a sharp rise (a bump), a sharp fall (a cliff), or a hill which is higher than the starting point. Also the object may slide into another object resting on or near the surface. WHISPER tests for these conditions by fixating at regular intervals along the curve. At each point it invokes the perceptual primitives relevant to each of the above conditions. There are some cases, for example when a plank slides off a box, when WHISPER examines the surface of the underside of the sliding object.

When it finds the endpoint of a slide, WHISPER updates the diagram using the re-drawing transformations in a two step process. First, it translates the object to the new point, and then, if necessary, rotates it. A rotation is usually required, because if the curve is not a simple straight incline, the orientation of the object changes as it moves. WHISPER determines the magnitude of this corrective rotation by re-establishing, via rotation visualization, the contact relationships which existed between the object and the surface prior to the start of the slide.

#### III.5 Benefit of the Diagram During SHDe Analysis

In the curve examination and diagram update process, the diagram is very useful to the HLR in the course of curve following, as well as providing feedback as it did in the case of rotations. The main pitfall in curve following is the possibility that two objects will coincidentally align so that a smooth curve is formed across them both. An object could then begin its slide on one object and continue sliding along the other as shown in Figure 7. This emergent property of the curve must be noticed, and the two curve segments appended. In a system relying on an independent description of each object, this would pose a significant problem because one would require: (i) that it have a built in expectation that the situation might arise; (ii) that it

continually check for the situation; (iii) that its check involve testing whether the first object touches any other object in the universe; and (iv) that it know how to amalgamate the two separate curve segment descriptions into a new curve description. For WHISPER there is no problem because two aligned surfaces of neighboring objects form a continuous curve in the diagram; WHISPER only has to look at this curve, rather than, in a sense, discover or construct it.

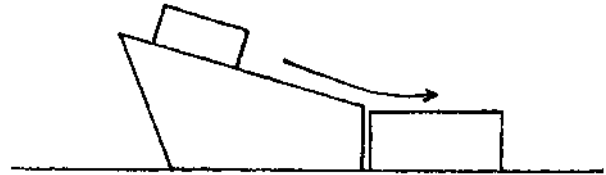


Figure 7

#### III.6 The Stability Test

A full discussion of WHISPER'S stability testing routines cannot be given in the limited space of this paper; however, we shall give an impressionistic view. Rather than solve the problem with a sophisticated general method, as Fahlman [1973] did in his BUILD system, WHISPER seeks qualitative solutions using rules corresponding to those a person untrained in physics might apply. The HLR has specialists which express rules like: "If an object hangs over too far, it will topple"; and "If an object and one of its supporters make contact along a surface (rather than at a single point) and if this surface is not horizontal, then the object will slide." A frictionless environment is assumed.

Overall organization of the stability test is based on the observation that a complete structure is stable if each of its independent subparts is stable whenever their supporters are stable. An independent subpart is any connected group of objects which is not supporting anything. For example, in Figure 2 of the chain reaction problem the independent subparts are objects B, D, AB, and DC, but not A and C. Thus the initial structure is broken down into smaller substructures whose stability as individual units is easier to test than the stability of the structure as a whole. When their stability is tested, B and AB are found to be rotationally unstable, while D and DC are found to be stable. Since B is a subpart of AB, it is the dominantly unstable object, and therefore it is the one rotated. If AB were unstable, but B stable, then AB would be rotated (i.e. objects A and B would be rotated together as if they were a single object).

Incidentally, treating two objects like A and B as a single object AB is another example of a situation in which two descriptions must be amalgamated, it is a trivial task for WHISPER to amalgamate two object descriptions, since all it need do is interpret their two color codings as the same color.

The stability test uses the diagram and retina to find contact relationships, centers of area, and slopes of contacting surfaces. The symmetry perceptual primitive is used to accurately establish that D in the chain reaction problem will balance.

#### IV. CONCLUSION

WHISPER demonstrates the advantages and feasibility of using diagrams as an aid in problem-solving. A fundamental component of the system is a retina which blends sequential and parallel processing while limiting the quantity of processors and processor interconnections to a fixed, not too large number. By asking questions of the retina, the HLR is able to obtain the type of experimental feedback from the diagram which is normally expected only as a result of interaction with the real world.

#### ACKNOWLEDGMENT

I have benefitted from the Inspiration of Raymond Relter, Alan K. Mackworth, Richard S. Rosenberg, and Gordon McCalla.

This research was supported by the National Research Council of Canada under grant A7642, and through the author's NRC Postdoctora\* Fellowship.

#### REFERENCES

1. deKleer, Johan. Qualitative and Quantitative Knowledge in Classical Mechanics. Cambridge, Mass.: MIT, M.Sc. Thesis, (1976).
2. Dunlavey, Michael. "An Hypothesis-Driven Vision System". Advance Papers of the Fourth International Conference on Artificial Intelligence. Tiblisi, Georgia, USSR, September 3-8, 1976, pp.618-619.
3. Fahlman, S. A Planning System For Robot Construction Tasks. AI TR 283.M.I.T. (May 1973).
4. Funt, B. WHISPER: A Computer Implementation Using Analogues In Reasoning. Tech. Report 76-9, Department of Computer Science, Univ. of British Columbia., (1976).
6. Gelernter, H. "Realization of a Geometry-Theorem Proving Machine." Computers and Thought. Ed. E.A. Felgenbaum and J. Feldman. New York: McGraw-Hill, 1963, pp.134-162.
6. Hayes, P. "Some Problems and Non-Problems In Representation Theory." AISB Summer Conference Proceedings. July 1974, pp.63-79.
7. Kosslyn, Stephen M. "Information Representation In Visual Images." Cognitive Psychology. Vol. 7 (1976), pp.341-370.
8. Minsky, Marvin and Papert, Seymour. Perceptrons: Aft Introduction to Computational Geometry. Cambridge, Mass.: M.I.T. Press, 1969.
9. Sloman, A. Interactions Between Philosophy and Artificial Intelligence: The Role of Intuition and Non-Logical Reasoning In Intelligence. Artificial Intelligence2 (1971).209-226.