WORKING PAPER 60

VIDEO ERGO SCIO

an essay on some things we would like a vision system to know

by

D. MARR* and C. HEWITT

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

November, 1973

Abstract

An approach to vision research is described that combines ideas about low level processing with more abstract notions about the representation of knowledge in intelligent systems. A particular problem, of the representation of knowledge about the three-dimensional world, is discussed: the outline of a solution is given, and an experimental world of simple mechanical assemblies is described, in which the solution may be implemented and tested. A tentative summary is given of the knowledge that is required for operating in this world, and a research project is proposed.

*On leave from the M.R.C. Laboratory of Molecular Biology, Cambridge, England.

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

Working Papers are informal papers intended for internal use.

0 Introduction

This article outlines a research project that we have been designing during the summer. It is centred upon the idea that the principal reason why vision programs at present perform so poorly is that the amount of knowledge they can bring to bear on the seeing process is so limited. For example, Waltz's program (Waltz 1972) is the latest in a line of development called scene analysis, which was originated by Guzman (1968), and pursued by Huffman (1970) and Clowes (1971). The usefulness of this approach is called into question by the difficulty of extracting, from information about intensity, the near perfect line drawings that such programs require; and by the restricted nature of the line drawing representation itself. Nevertheless, within these constraints, Waltz showed that, after a certain minimal amount of information has been included in a program's database, the problem of interpreting a scene becomes easier, rather than more difficult, as more information is added. In the extreme case, he showed that when the database is in a certain sense complete, and the incoming line drawing is perfect, the interpretation of a scene is uniquely determined.

We greatly admire Waltz's program, but we feel that the approach that it embodies is open to several criticisms. The first is that the knowledge that it uses is in a certain sense not explicit enough. Although it contains a great deal of information about the appearance of line drawings, this information is essentially in a compiled form: one reflection of this is that the structure of Waltz's program makes it

inherently unable to use either explicit information about the three-dimensional form of what is being viewed, or the many pieces of special and general knowledge that we surely bring to bear on the process of seeing. There is no way in which pieces of its knowledge can be pulled out and examined while it tries to create an interpretation of, for example, a scene in which several lines are missing. Unless such knowledge, suitably embedded in a hypothetico-deductive system, can play a large part in the operation of a vision program, we see no prospect of such a program being able to interpret the incomplete information that is the diet of daily life.

The basic trouble with the labelling approach of scene analysis is that it is too limiting and stultifying a paradigm for vision, in much the same way that resolution is for deduction. The fundamental principle of resolution, that (not A) and (A or B) together imply B, is occasionally useful. But attempting to make a uniform resolution proof procedure, to mechanise deduction in a way that cannot be very sensitive to hints, hunches, and a wide variety of higher level knowledge about the particular domain in question, is a cul-de-sac. Similarly, the line and vertex labels are local predicates that are occasionally useful, and are of some mathematical interest in their own right; but the problem of creating a uniform procedure to label arbitrary line drawings is not a central one for vision. Hence, we believe that the kind of knowledge contained in Waltz's program is probably relatively unimportant; and that the way in which it is made available there is certainly too restricting.

The proper endeavour of vision research is to decide what knowledge

should be used to help a vision system to see, and to discover methods that make it possible to use such knowledge. How can one pursue this goal more effectively? There are two kinds of answer. The first is to abandon the restrictive format of line drawings, so that programs can use information about visual features that are not coded in this form. To this end, a whole field called picture processing has arisen, that studies simple low-level algorithms for picking out regions from visual scenes. People who study picture processing are however greatly hampered by not knowing what they are processing the picture for: evaluation of the success of a technique is therefore a subjective matter, and is often avoided altogether. (For example, a recent elementary book by Rosenfeld (1969) discusses a number of operations that can be applied to pictures, but does not evaluate them. Some of the 500 or so papers Rosenfeld (1973) surveyed later may contain useful algorithms, but the reader is given no help in trying to find them. Uncritical surveys of this kind are almost useless.)

The second kind of answer, lying at the opposite end of the spectrum, consists of a more abstract approach to how to represent knowledge in intelligent systems, (Fillmore 1968, Abelson 1973, and Minsky 1973). Roughly, the force of these ideas is that knowledge should be organised into quite large chunks, called frames. We believe that these ideas are exciting because they suggest ways in which a system might be given access to much more knowledge than has hitherto been possible: but general theories often skate over the thorny issues of low-level vision, which are probably mainly responsible for holding up

progress in the area.

Our proposal is to combine both approaches. Because this cannot be done in the abstract, we intend to take a particular visual domain, and set out carefully an explicit catalogue of the knowledge that ought to be used by a program trying to see things in it. We have chosen a world in which most of the important issues of three-dimensional vision are raised, yet which is sufficiently simple that implementing an experimental system is not out of the question. The world is a Fischertechnik construction kit (referred to henceforth as FT). This is a well-designed set of parts of various shapes and sizes, made of metal or plastic, from which small mechanical assemblies may be constructed: to be able to operate successfully in this world requires a considerable knowledge of spatial relations, and knowledge of shape and of function.

This article describes briefly how we propose to do this, and sketches the catalogue of knowledge that we hope eventually to obtain, and to prove useful. We subdivide the catalogue into MINI-WORLDS, according to the criterion that relations between items in a MINI-WORLD are much denser than between items in different ones. Because we have a partly procedural model in mind, a MINI-WORLD should be thought of as an active collection of knowledgeable specialists. The list of MINI-WORLDS that we offer here, and the partition of knowledge that it represents, is a tentative one.

1 Outlines

Our enquiry is basically epistemological, and explores the interesting territory between knowledge that one would like to use, and knowledge that is actually available to be used. The first issue that we wish to raise concerns the nature of low-level processing. Line drawings are easy to use but difficult to obtain. Questions like where are the obvious one-inch blobs in a picture are in principal very easy to answer, but this knowledge is probably considerably more difficult to use. It is also impracticable to procure using current hardware, even though the nature of computation involved is trivial. A serial processor would take as long to obtain such knowledge as it would to extract a line drawing - yet the latter on its own is far more informative. Little wonder therefore that people do not take seriously the possibility of extracting blobs, bars and spots in all kinds of positions and orientations when a clever serial region finder or line extractor will provide much more valuable results in the same time.

The only circumstances in which one might even consider doing something like this is if one had several orders of magnitude more computing power available; and if one had it, it is clear that ones basic approach to the problem of vision might be different. Yet animals probably do have a great deal of special purpose computing power available for early visual processing (see e.g. the brief review by Marr & Pettigrew 1973). Perhaps the most powerful idea that it becomes feasible to contemplate is that of running an analysis simultaneously at

several different image resolutions. One of our objectives is to study how to make use of this kind of knowledge about a visual scene, with the ultimate intention of formulating a prescription for a piece of hardware capable of providing it.

Our second principal interest arises because the world is basically 3-dimensional, and so if the system is to be able to handle a large amount of knowledge about the world, a decision has to be made rather early about whether to represent such knowledge in a 2-D (knowledge solely about appearance) or in a 3-D language. A major theme running through our approach, and one that we shall try to justify, is that the representation of visual objects should be translated into a 3-dimensional language as soon as possible. Thus we distinguish between what we call a VIEW of an object, and an underlying 3-D MODEL of that object. (In this respect we disagree with Minsky 1973).

The VIEW is a peripheral mini-world in which low level visual routines can leave assertions about what they see outside. The VIEW knows about the possible different directions away from the viewer, which directions are near which other directions, and it has a crude knowledge about distance away from the viewer. Closely associated with this world are other mini-worlds that can describe two-dimensional shape, movement, colour, and disparity (between e.g. the images on two cameras) leaving their information bound to the names of directions in the VIEW. The VIEW is thus like a canvas on which these features are painted to be studied and interrogated by more central routines.

The final ingredient of the view is the CLUE. One form of a CLUE is

a particular combination of predicates (e.g. of shape predicates), which can be bound to the VIEW, and which suggests the presence of a particular 3-D structure. Such CLUES are like masks that act as triggers to specific central 3-D representations, and cause them to ask particular questions to try to verify that what they represent is actually visible. There are other, more general kinds of CLUES, and we give a number of illustrations of them later in the paper.

The VIEW communicates with 3-D models of objects in the world. A 3-D model is an abstract description by assertions of objects, and it is this that the system tries to keep consistent with the information in the VIEW. 3-D models are arranged into a number of mini-worlds: the first concern descriptions of primitive 3-D shapes, and later ones, the elaboration of these into representations of the objects with which the system has to deal. Each 3-D model has its set of triggers, (e.g. the CLUES from the VIEW mini-worlds), and a body of code that it will try to execute if it is triggered. For example, a typical model might have a dozen or so triggers - ranging from low resolution visual clues, through high resolution clues that detect a particular detail, to clues arrived at from a guess at the whole object (e.g. that piece of black must be the fourth wheel). Notice that triggers are simply special ways of using an important kind of knowledge - akin to the role that features play in a systemic grammar for natural language (Halliday 1967, 1968). 3-D models are designed to suicide as early as possible, if they are inappropriate: and should be regarded as specific suggestions about how to see the information that caused their activation. Knowledge about the failure of

a 3-D model can act as a trigger to other 3-D models - yet another kind of information that the system must be able to use. It is a basic philosophy of the system that the 3-D representations of the world are what the system "remembers", and what it tries to maintain consistent with the information bound to the VIEW.

Our insistence on using 3-D models for the basic representation of objects does not preclude the use of catalogues of appearances of objects from different viewpoints. Indeed we regard knowledge about appearances as an indispensible kind of CLUE. But three factors predispose us against using such catalogues as the main representation technique. Firstly, there is the sheer number of such appearances that would be required. For a very common object such as a hand, with a standard illumination, the number of appearances that would have to be stored to cover views from all directions at various degrees of clenching, is something like 400. (This estimate is based on a simple calculation made from a set of defocussed photographs of a hand). For something as important as a hand, it is probably worth keeping a catalogue of this size, though it would have to be expanded considerably to allow for a decent range of lighting conditions. But it is unlikely that one could do the same for every object, even though most objects probably have at least one stored, standard view. For those cases where an object must be recognised from a view that is not stored, our system would split it into parts, compile a 3-D description of it, use part or all of that description as a CLUE to access the world of 3-D models, and gradually modify the description until something is derived that satisfies both a 3-D model and the VIEW.

The second consideration concerns updating a representation. If a CLUE is slightly wrong about the appearance of an object, it does not greatly matter: the consequence will be to degrade the effectiveness of that CLUE, but it will not upset the ability of the system to reason about the object, because the 3-D model is used for that. If a system's primary representation is in terms of appearances, however, updating becomes a major chore, since it has to be done on all appearances before the system can reason reliably again.

The third consideration concerns the design of 3-D structures in the FT world. Although it is difficult to be certain of this, we feel that the manipulation of 3-D descriptions directly (e.g."attach part1 at right-angles to the centre of part2") provides a more promising environment for FT architecture than the manipulation of descriptions of appearances of objects - though again one might find some sort of compromise that is useful.

The main part of this article is concerned with summarising the knowledge needed for the VIEW, and for low level 3-D representation of the FT world. We believe however that an important part of the theory of vision should be to try to determine the extent to which knowledge about very high level properties of a domain makes it progressively easier to see it: in this case, the high level knowledge has to be about how to construct models using the FT components. One of the first pieces of knowledge that one needs is a mini-world that knows about how to make primitive types of joins between components: we already know that even this simple functional knowledge brings immediate rewards in terms of a

greater ability to see the domain accurately. We would like to know, for example, whether advanced architectural ability in the FT world is a substantial, or only a marginal benefit. Later in this article, therefore, we set out some of the issues that arise in the more advanced domain of FT architecture.

The implementation of so large a system as this, even omitting the necessary manipulative skills, will of course be a lengthy task - one not to be undertaken lightly. We feel however that the issues that we describe above are the issues that ought to be faced now. The primary questions are epistemological: what knowledge do you need to see well; and how should it be organised - what pieces of knowledge will need to be able to interact with (to read, debug, run, or be run by) what other pieces? Questions of implementation are of great importance, but should probably be answered second, because epistemological interactions that are found necessary at quite a late stage can affect the implementation details rather early on.

Finally, there seem to be two other great advantages of the FT world as a medium for experimental investigation. The first is that it is expandable; some of the extensions, for example the introduction of gears, introduce major new concepts into the domain. Thus the FT world allows us to study the effect, on a system that is already extremely sophisticated, of introducing entirely new kinds of knowledge. This is an indispensable quality for an experimental system given the present state of the art.

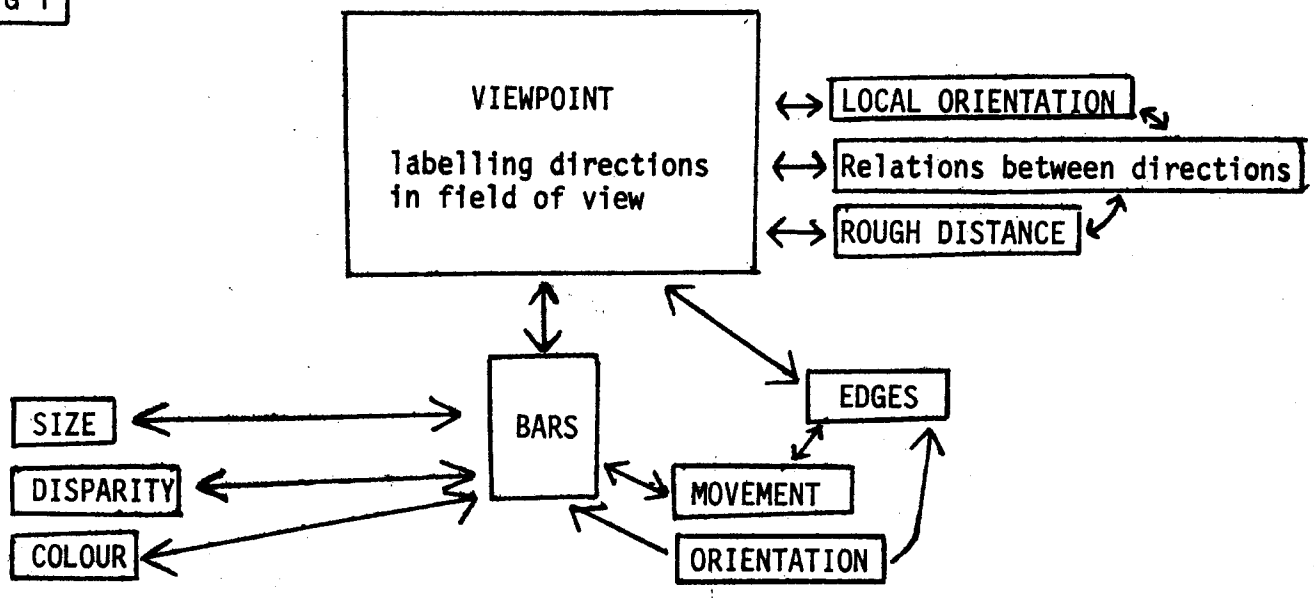
The second advantage is that the FT world is discrete in a sense in

which the real world is continuous. This means that results that are expressed as qualitative assertions about the world will usually be either right or wrong. The final step that is needed in a continuous world, of interpolating between two competing qualitative answers, is unnecessary in the FT world. We feel that this is the right kind of simplifying factor to have available.

2 The VIEWPOINT and associated mini-worlds

The next few chapters present a preliminary catalogue of the system's knowledge, arranged in the system of mini-worlds that was described in the introduction. The information is presented as a combination of lists and diagrams. The first collection consists of the VIEWPOINT, and the worlds that can leave assertions bound to the VIEWPOINT (see figure 1). The term VIEW is used to denote the information that is bound to the VIEWPOINT at any time.

FIG 1



2.1 The VIEWPOINT

The VIEWPOINT provides a naming system, for directions in space away from the viewer, together with knowledge about the relative orientations of those directions. It is essentially a special purpose box of knowledge and techniques that represent important aspects of the structure of 3-space in a coarse, symbolic manner. It is quite close to a representation of the space around a person that could be read by motor routines for moving and placing ones arms and hands. The naming scheme is based on spherical, rather than Cartesian co-ordinates.

2.1.1 A space frame (SF) consists of:

(a) a plane P

(b) a point C in P

(c) a unit vector **f** from C in P.

The plane P is called the frame's horizontal plane: C is called the centre of the frame, and **f** defines the direction forwards in the frame.

Figure 2 gives the names of directions that are assigned in the space frame relative to (P,C,f). Labelling conventions are F=forwards, U=up, D=down, V=vertical, B=backwards, R=right, L=left, Q=quarter, and H=half.

Forwards View

UL	UHL	UQL	U	UQR	UHR	UR
HUL	HUHL	HUQL	HU	HUQR	HUHR	HUR
QUL	QUHL	QUQL	QU	QUQR	QUHR	QUR
L	HL	QL	F	QR	HR	R
QDL	QDHL	QDQL	QD	QDQR	QDHR	QDR
HDL	HDHL	HDQL	HD	HDQR	HQHR	HDR
DL	DHL	DQL	D	DQR	DHR	DR

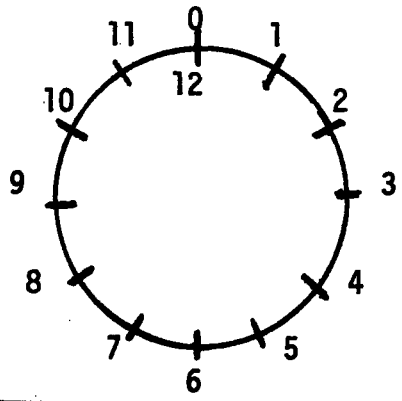
2.1.2 The VIEWPOINT. Let P be the true horizontal, C be the position of an observer, and f the direction that is forwards for the observer. Then the VIEWPOINT is the space frame associated with these (P,C,f). Let g denote the direction of gaze: then in a human, from knowledge of posture, and of eye, and head position, the direction g can be expressed as a label in the VIEWPOINT. This is one of the many pieces of knowledge that we shall not need to include in our system.

Implicit in figure 2 is much knowledge, about the relative positions of the direction labels, which must be made explicit before it can be used. That knowledge is set out in the next few sections.

2.1.3 Local orientation names. Let g be a direction in the viewpoint.

Then we denote by g-1,...,g-12 the twelve orientations in the plane perpendicular to g, arranged, at a separation of 30degs with g-12 vertical, like the numerals on a clock face (see figure 3).

FIG 3



2.1.4 Angular separation of directions. The system needs to know the angular distance between labels in the VIEWPOINT, and also the vertical and horizontal components of the angular distance from F of any label in the VIEWPOINT. This knowledge could be represented by three functions, SEPANG, HORANG, and VERTANG, whose values are expressed in terms of a small collection of names of angles.

2.1.5 Orientations in the VIEWPOINT. Knowledge about orientation in the VIEWPOINT needs to be present in two forms. Firstly, given any two labels that are not too far apart, the orientation of the line joining them needs to be available. e.g. (OR F QUHR) = g-2. Secondly, given a label in the VIEWPOINT, a local orientation, and an angular separation, the system needs to be able to estimate the nearest label in the VIEWPOINT that has this orientation and separation from the given one (i.e. the inverse of the above functions).

2.1.6 Rough distance measures. Although the VIEWPOINT is mainly a two-dimensional viewing frame, it needs a crude idea of distance away from the viewer. Notice that this is somewhat different from the idea of local distances in and between the objects that it sees. The accuracy with which this distance is needed is not great: about five distance names may suffice - call them D1,...,D5. These distance names need to be related to the direction labels in the following crude way: given two pairs (a, Di) of direction labels and distance away in that direction, the system needs to know, again rather roughly, the distance between the two points thus represented.

All the functions described above can be provided easily, either by simple computation, or by using a number of look-up tables.

2.2 Shape primitives

The second mini-world is concerned with the description of two-dimensional shapes. It consists of bar-shaped region descriptors of various sizes, orientations and positions, including descriptors of very thin, long cracks, and of edges. It also contains some multiple bar, or high spatial frequency, detectors that are useful for suggesting the presence of serrated edges, gears, and cogs. A first guess at the necessary predicates follows: programs that can detect these predicates in the FT world are presently being written.

The following set will provide a preliminary vocabulary of bar shapes and sizes:

(a) bar widths of 16, 8, 4, 2, 1, 0.5, 0.25 degs.

(b) bar lengths equal to 1*width, 3*width, and open ended.

(c) 12 orientations, g1,...,g12, as in 2.1 above.

(d) specialist region detectors designed to pick up the prominent holes that occur in FT pieces: there are about six kinds needed for this.

(e) very thin crack specialists, designed to see the junction between two joined pieces.

(f) multiple (3 or 4) bars, of width 2, 1, and 0.5deg.

2.3 Colour

The colours that occur in the FT world are very limited: they are black, grey, red, silver, and a name for holes, dark.

2.4 Disparity

Stereo disparity information can be very useful for separating the nearby world into objects, so we would like the ability to use this information in the experimental system. Disparity measurements will be associated with every bar predicate, and will take values between about 4 degrees of convergence to about 1 degree of divergence.

2.5 Movement

Only a very primitive notion of motion is required in the FT world. To each bar is attached a flag indicating one of the following values: STILL, SLOW (up to 1deg/sec), and FAST (more than 1 deg/sec). The system needs a primitive idea that movement changes the direction in the view: it is sufficient however that the knowledge described in 2.1.5 be

available for computing the expected position of a moving object. Tracking ability, and the complex allowances that must be made during active tracking, are not necessary for our FT world, since we do not plan to monitor actively the assembly process.

2.6 Visual properties of FT parts

FT parts have various advantages and disadvantages from a purely visual point of view. Their surfaces are not appreciably textured- and the kinds of curved surfaces that can be constructed are limited to wheels and gears of various sizes. The surfaces have a moderate reflectance, so that reflexions other than of a light source are practically absent, and the specularities that are present are not too intense to be used. The red components will cause trouble with the vidissector, but not with the vidicon, so that no special treatment of any of the surfaces will be necessary. We may wish to undertake a superficial exploration of the value of our techniques for curved surfaces, but these can be made using other objects, specially selected for the purpose.

3 Primitive three-dimensional knowledge

The second group of mini-worlds is concerned with the representation of basic three-dimensional shapes, with simple position relations, elementary notions of length, width and diameter, and with the basic

elements of the FT world. Associated with this knowledge is information about what shape cues from those set out in section 2 are appropriate CLUES for these 3-D structures, and a certain amount of additional information that can be used to verify or to reject a CLUE.

It is helpful to introduce a tentative basic unit of organization of knowledge, called a PLAN. A 3-D model is a particular type of PLAN. A PLAN consists of code that serves the following three purposes:

(i) A TRIGGER, that defines the circumstances in which the PLAN is run. The trigger defines the CLUES for the PLAN, and a given CLUE may depend on more than one circumstance. Furthermore, CLUES may be armed by particular circumstances, so that information that in one context fails to elicit a particular PLAN, in more suggestive circumstances will succeed.

(ii) A BODY of code that runs when the plan is triggered. The body can apply further tests to see if the plan is really applicable and otherwise record its impressions of the situation in which it is being executed. It can contain other kinds of information, like what to do if the PLAN fails, and how to acquire more information about something that the PLAN ran successfully on.

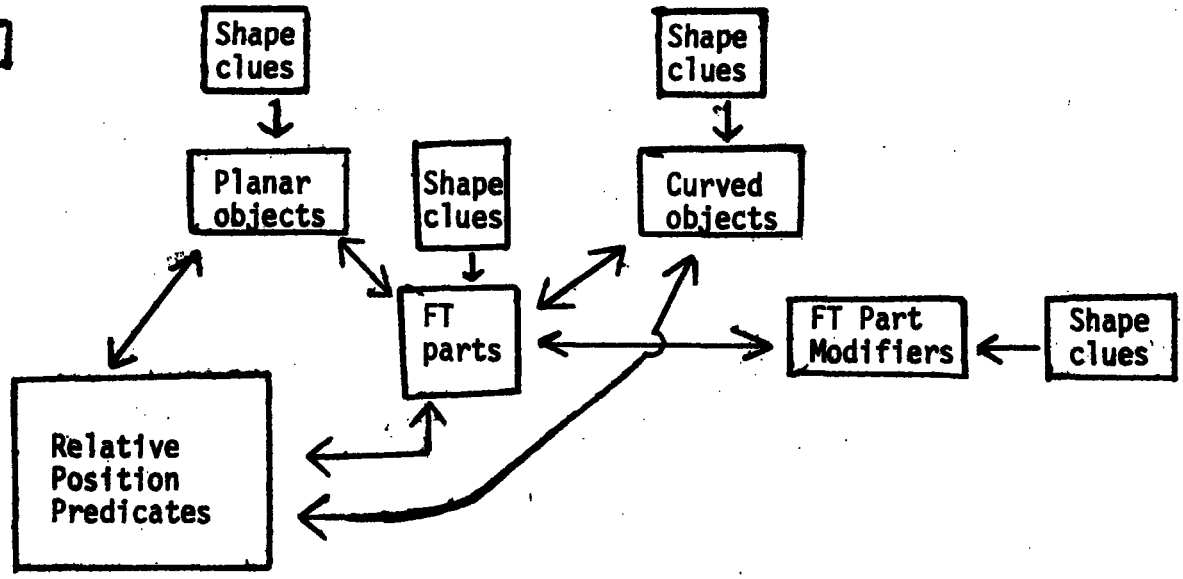
PLANS are derived from a mixture of sources: from the "state vectors" of McCarthy (1964), from PLANNER-69, from an unpublished character recognition program that used PLANS to represent the characters (Blomfield, Marr & Mollison 1972), and from ideas that come under the general heading of frame theory (Minsky 1975 and works cited earlier).

This section describes knowledge about some simple objects.

Although there are no cubes or pure rectangular blocks in the FT world, knowledge about such items is useful because much of the thinking about the results of putting FT pieces together can often be done using these simpler objects. Thus they can act in a sense as abstractions from the real pieces that make the formulation of seeing hypotheses, or of designs for construction, much easier than they would have to be if the original pieces were considered. The step from a rough design or hypothesis to the correct analysis is then achieved by some simple debugging. This illustrates an important principal, namely, keep the thinking as simple as possible for as long as possible. Make sure that the program is not going to have to worry about peripheral issues until it has made up its mind about how to handle the central issues. Then make the old peripheral issues the central issues for another period of thinking. It is also appropriate to draw attention to what seems to be another fact of life: namely that low resolution visual images of blocks and of FT bricks are very similar. Thus low resolution triggers for a block could be activated by piece FT1. The fact that such a piece can be thought of as if it were a block for many purposes is a phenomenon that is not restricted to the FT world.

The knowledge described in this section is summarised in figure 4.

FIG 4



3.1 Primitive 3-D plan r objects

This MINI-WORLD contains the basic notion of a small-object as something that can be moved around, associated with a position in space, and can be used to fill space up. It contains the basic concepts related to 3-D properties of a block - of a face, edge, and corner; the associated notions of length, width, and diameter (as being the size of a gap into which the object will fit); a simple idea of number (one, two, three, many); and a primitive idea of what parallel means. Associated with this is the knowledge that opposite faces and edges of a block are parallel, the idea that three sides and three edges meet at a corner, that an edge joins two corners, and that the same two edges only meet at one corner.

3.2 Shape knowledge for 3.1

Attendant on 3.1 is knowledge about what predicates in the VIEW should act as CLUES for the 3-D assertions of 3.1. This includes a coarse catalogue of appearances of primitive 3-D objects at different viewing angles and at different levels of resolution. Many MINI-WORLDS generate their own satellite collections of shape knowledge.

3.3 Primitive 3-D curved objects

This MINIWORLD has information about small-objects that do not have edges and corners like those on 3.1: e.g. a sphere, a cylinder, and a disc. It contains knowledge of the difference between a flat and curved surface, the idea of radius, and of thickness, and very coarse knowledge that flat surfaces used as support are stable, whereas curved ones can move.

3.4 Shape knowledge for 3.3

The world of 3.3 induces another that contains CLUES for curved objects from shape predicates. This takes the form of a catalogue of appearances for items in 3.3

3.5 Primitive position predicates

Next, we need the idea that two small-objects can be in various positions relative to one another, and a primitive idea that support is necessary. We use notions of beside, between, behind, in front of, above, below, on top of, resting on, etc., and the corresponding triggers

from information in the VIEW, like position and disparity, that provide CLUES for the diagnosis of these position schema. Also needed is the notion of sloping planes, and a very coarse description of how planes can slope.

3.6 Basic FT parts

Now we come to the description of the basic elements of a core FT kit: BRICK, BLOCK, PANEL, CONNECTOR, WEDGE, BASEPLATE. This world contains low resolution knowledge of the nature of these parts, using knowledge in 3.1 and 3.3.

3.7 Shape knowledge for 3.6

This is a basic catalogue of appearances of items in 3.1, to be used as CLUES.

3.8 Basic FT part modifiers

Here, we have more detailed knowledge of the structure of FT parts: knowledge of a groove, a lug, and a slot; their positions on the various parts, their size, 3-D shape and colour: the use of part modifiers to diagnose FT parts - e.g. if it has a lug, it cannot be a baseplate so try a brick: if it has a slot, it is probably a brick.

3.9 Shape knowledge for 3.8

This knowledge consists of CLUES for 3.8 based on appearance, and leads to CLUES for 3.7 based on the appearance of details on the parts

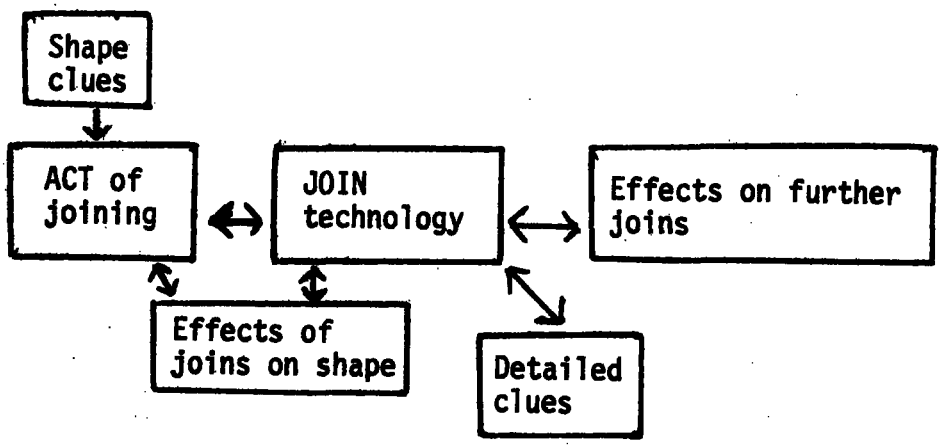
(e.g. if you see something that might be a lug, immediately try a BRICK plan).

An important aspect of the organization of this knowledge is that there should be no restrictions on the directions in which hints can flow. Thus parts can suggest wholes, and wholes can suggest parts: the only criterion is that the CLUE should be a useful one.

4 Basic Joins

The next group of mini-worlds contains knowledge about the means and effects of simple joins. This includes information about how joins are accomplished, what parts can be joined to what other parts, and about the effect on 3-D structure of making simple joins. In parallel with this is information about the appearance of the structures at each stage, and about other simple kinds of CLUE (see figure 5).

FIG 5



4.1 Basic join engineering

The first mini-world contains elementary knowledge about join compatibility: lugs can mate with slots or with grooves; the use of connectors; panel insertion; methods of connexion to the baseplate; and the male-female property.

4.2 Basic effects of joins

Joins turn two small-objects into one, and this mutt be understood at several levels. Here we keep a low resolution 3-D description of the effects of joining two components: joins between bricks (T, L, and end-to-end joins); joins using connectors, panel insertion, and joins to the baseplate.

4.3 Appearance of 4.2

This is a catalogue of CLUES for 4.2 based on low resolution shape predicates for basic joined pairs. It can make tentative assertions from appearance about 3-D structure.

4.4 The act of joining

Next comes simple knowledge about how joining may be accomplished: this includes the notions of entering a lug in a slot, of sliding down a groove and across the baseplate, and of sliding in a panel; and elementary knowledge that space must be available for joining to take place (e.g. panel access must not be boxed in before the panel is slid in). More advanced knowledge, including the common bugs that make joining impossible, is held in more advanced mini-worlds, but expressed in terms of the basic concepts held here.

4.5 CLUES for 4.4 from appearance

Certain CLUES from shape predicates suggest that the act of joinjng will be impossible: e.g. seeing bar shapes arranged in a closed square pattern precludes the insertion of a panel. This is an appearance CLUE, because the analysis does not pass through the representation in the 3-D mini-worlds.

4.6 Effects of joins on further joins

Only one join may be made to a given lug: panels prohibit other joins at all boundaries. Connexion is transitive, and the male-female property is contagious. This world also contains high resolution 3-D knowledge about the structure of joins: about the use of wedges to achieve direction changes; and the knowledge that this induces about rounded corners.

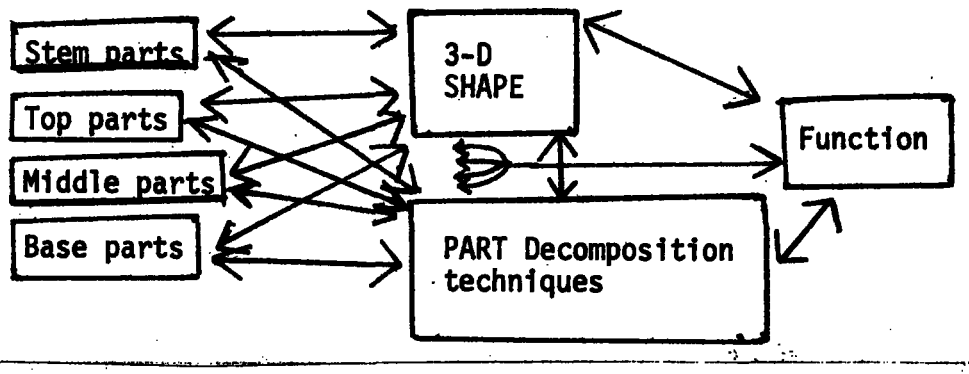
4.7 Detailed appearance CLUES for joins

Associated with detailed information about the nature of joins are high resolution visual information and visual strategies for the recognition of joins: e.g. how to detect a connector join from appearance (not by deduction); how to detect slide connexions to the base-plate, etc.

5 Intermediate 3-D knowledge: useful parts of things

The fourth collection of mini-worlds contains information that is intermediate between the very basic concepts that have been listed above, and the higher level descriptions of real, useful 3-D objects. It provides a repertoire of useful parts for other mini-worlds dealing with descriptions of whole objects, with function, with design, with stability, and with seeing (see figure 6).

FIG 6



5.1 Stem parts

Stem parts are long, thin parts. Knowledge is kept about their

likely composition, and about useful ways of connecting them. The other important concept related to a stem part is that of the distance it induces between the parts that it connects.

5.2 Base parts

Next, there are the various ways of forming a solid base for a structure: the likely places for a base-plate; the idea of a base square and of a base cross; and the concept of legs.

5.3 Middle parts

Above a base part comes some kind of middle part - perhaps more than one. A stem part is a kind of middle part, but there are a number of kinds of fat, middle parts that are quite common.

5.4 Top parts

The last of the simple types of part is the notion of the top of a structure: e.g. a table-like top, a sloping top, a small decorative top, a roof, etc.

5.5 Part decomposition from appearance

Associated with 5.1 to 5.4 are useful criteria for splitting an object into parts: these include sharp changes in cross-section, the end of parallel grooves, a change in general colour, a point of inflexion in an outline, etc. This MINI-WORLD is concerned with medium and low resolution CLUES for the likely parts of a structure.

5.5 The function of parts in a structure

Finally, there is a MINI-WORLD that contains functional knowledge about the role that different parts can play in the composition of a whole object. Such knowledge is intrinsic - like "join the top part to the bottom part," rather than extrinsic - "the object must be able to lift things up from here and put them over there".

6 High level 3-D structures

We come now to some higher level knowledge of whole objects and their appearance, and to a certain appreciation of function. The latter aspect will necessarily remain primitive until the idea of a wheel is introduced.

6.1 3-D repertoire

The basic high-level MINI-WORLD contains descriptions of objects in terms of their parts, and of kinds of joins between those parts. This world consists of 3-D models of a crane, table, bench, chair, box, bridge, car, desk, lectern, stall, garage (with sloping roof), lighthouse, and so forth.

6.2 Appearance CLUES for 6.1

As we mentioned in section 2, appearance clues for whole objects

from all possible viewing angles would occupy a great deal of storage, and it is probably unusual to find objects that are important enough to warrant so large a vocabulary in real life. Neverthless, for many objects, there are sets of viewing angles that are important - like the views of a table from standing height - and in cases like this, whole object CLUES will be necessary. We emphasize that such masks are quite distinct from underlying 3-D representations of the objects in question, though low STATUS assertions about the presence of such objects may often follow quite uncritically from the firing of a CLUE. In real life, and in the FT world, there are many special features that can be used as CLUES, in addition to those provided by low and medium resolution analysis of shape: and conversely, it is often necessary to identify an object before a small feature detail on it can be recognised.

6.3 Extrinsic function specialists

This mini-world will expand greatly as soon as wheels are introduced: for the present, it contains knowledge about support, and an elementary theory of balance and stability: (two legs do not balance, supports must be underneath and near the centre); and some idea of the notion of a container.

A very important group of mini-worlds is the one concerned with debugging aspects of the other mini-worlds. Although some knowledge about debugging will be needed for all groups of mini-worlds, much will be specific to a particular one because the knowledge that it deals with is specialised. We expect this to be one of the areas of knowledge that will eventually differ the most from the preliminary sketch that follows.

7.1 Error realisation

This MINI-WORLD keeps an account of the errors that arise during the seeing process, and tries to decide whether this particular error has occurred before by using descriptions of the error that are created elsewhere. Other routines read the errors from here, and try to cure them by altering plans in the appropriate part of the system.

7.2 Error localisation

The first thing to do is to try to classify the error by deciding who is responsible for it. This world knows about the common bugs that arise in the various worlds, and is able to recognise some symptoms of them.

7.3 Error correction

This library is concerned with suggesting specific cures. It is assumed that the kind of complaint, and the place that it arises, is known, and is concerned with developing a repertoire of remedies.

7.4 Analogy specialists

This mini-world is a specialist at using information in plans that will not quite run in the real world. For example, a normal plan may pass most of its criteria, but fail on a condition that MUST-BE satisfied. If there is no better plan in the database, the failed plan may contain the only useful information in the system, and so it should be used to help cope with the new object or circumstance. The analogy specialist can run the failed plan in a protected environment to see what it suggests, and can annotate its success or failure elsewhere. This is a common method of germinating new plans - they start as comments on existing, slightly inappropriate ones.

7.5 Plan differentiator

The plan differentiator contains expert knowledge about when, and how, a given plan should be reorganised into separate subplans. There are several basic ways of doing it: leaving a plan alone, but creating a new plan for a particular sub-class of items that the old one accepts (or vice versa); splitting a plan into two co-equal neighbours; setting up the new plan as a commentator on the original, etc.

7.6 Plan-writer and historian

This world contains a standard generator that is capable of creating a tentative but callable 3-D model, complete with body and triggers, from an example in the recent history of the system. It can also ask that this plan be called soon so that it gets exposed to the test-and-debug

interaction.

7.7 Trigger specialist

Finally, we need a watchful specialist that is an expert at adding CLUES to an existing 3-D model. It lists interesting circumstances that precede a plan's being called, (especially if that call was as a result of, or resulted in, a bug); formulates new triggers, and writes them into plans to test their usefulness.

8 More advanced areas of knowledge

There is clearly going to arise a need for the system to keep a coarse model of itself so that simple management strategies may be tried and debugged like any other plan. Specialists in ideas like "importance", "goal", "resource allocation", "interesting coincidence", should be able to affect the general flow of control, while leaving the details to specialists. Communication between these and the executive mini-worlds should take place in terms of assertions about how hard to try for things, how much tolerance to apply when testing for a certain thing, and when to call off an approach that is probably unprofitable.

It is also tempting to consider the problems that arise in assembling a plan for a new structure in the FT world. In order to be able to do this, a number of kinds of knowledge about design, construction and debugging techniques have to be available. Eventually,

we hope to be able to specify a function, and have the system produce, debug, and order the detailed construction of a design of its own. Initially, however, we would expect to give the system a top-level description in terms of parts that it understands, and require it to construct the object that fits that description. The interest from the visual point of view would be in the extra knowledge that would be required to oversee the construction, and recognise errors in intermediate stages.

We imagine being able to give the system a description of the form "like two chairs glued together at the arms"; or "like a table with no legs at one end". Sketch architects would be called to suggest the parts of the new object, and join specialists would think about how to join those parts together. Surrounding these two would be other specialists that are capable of taking a sketch plan, and of filling it out with detailed suggestions of exactly which parts to try to use to achieve the intended structure. Attendant on these would be a cluster of specialists watching for bugs that arise as particular attempts at instantiating the sketch plans are found to fail, and higher order debugging specialists watching the course of construction for evidence of systematic errors. These problems may be considered rather distant from the principal interest of this article, but the utilitarian nature of vision makes it important to study it in the context of doing something else; and many of the issues raised here probably also arise in learning to see.

9 Discussion

The account given above is intentionally sketchy. A few of the mini-worlds that we summarized have been studied in some detail, and we are at present compiling more substantial accounts of the knowledge that each contains. A major problem for later study is that of finding a good implementation: the old issues of the interfacing of special and general knowledge, of naming strategies, ways of moving fluently among different 3-D models for the same part of the VIEW (representing different kinds of knowledge about that part), are all raised here. But we feel that the implementation problem is best attacked _after_ we have a very clear and detailed account of the knowledge that has to be expressed. There seems to be no _a priori_ reason why a universal, high-level formalism should exist that is exactly suited to handling so many different types of knowledge, although there will probably be a few methods that are commonly useful.

Finally, at the risk of offending purists, one of us would like to mention that scattered over the clinical and neurophysiological literature are hints that the mechanisms of masks, clues, and underlying 3-D representations that we organised into a small theory of recognition, may have fairly closely corresponding analogs in primate and in human visual systems. When we know in more detail the kinds of assertion that are useful for 3-D representations, it may be possible to formulate a succinct and concrete hypothesis about the kind of single unit response that one would expect from cells in the relevant occipito-parietal

regions. Hypotheses of this kind are extremely difficult, but not quite impossible, to formulate: they are however comparatively easy to test.

Acknowledgement. We thank Ira Goldstein for his criticisms.

References

Abelson, R.P. (1973). The structure of belief systems. Chapter in: Computer simulation of thought and language, ed. K.Colby & R.Schank. W.H.Freeman & Co.

Fillmore, C. (1968). The case for case. In: Universals in linguistic theory, pp1-88. New York: Holt, Rinehart & Winston.

Halliday, M.A.K. (1967 & 1968). Notes on transitivity and theme in English. J. Linguistics 3, 37-81, and 4, 179-215.

Hewitt, C. (1969). PLANNER: a language for proving theorems and manipulating models in a robot. Proc. Int. Joint Conf. Artif. Intell. pp295-301. Bedford, Mass: Mitre Corp.

McCarthy, J. (1964). A formal description of a subset of Algol. Proc. Conf. on Formal Language Description Languages, Vienna.

Marr, D. & Pettigrew, J.D. (1973). Quantitative aspects of the computation performed by visual cortex in the cat, with a note on a function of lateral inhibition. A.I.La. Working paper 1.

Minsky, M. (1973). Frames: a theory of representation of knowledge. (Draft).

Rosenfeld, A. (1969). Picture processing by computer. New York: Academic Press, 196 pages.

Rosenfeld, A. (1973). Progress in picture processing: 1969-71. ACM Computing Surveys, 5, 81-104.