USING MULTI-LEVEL SEMANTICS TO UNDERSTAND SKETCHES OF HOUSES AND OTHER POLYHEDRAL OBJECTS

by
Jan A. Mulder
and
Alan K. Mackworth

Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada V6T 1W5

Abstract

HOUSE, a computer program, can interpret sketches of houses and other polyhedral objects. This paper describes the design and current implementation status of HOUSE. The program uses seven levels of representation of the meaning of the sketch. It achieves a consistent interpretation at each level before proceeding to the next level. The interpretations produced on one level are used as cues to invoke models at the next level. The notion of consistency is extended to include both internal and external consistency. Consistent interpretations are arrived at through a uniform network consistency algorithm. The program is presented in the context of the goals of a sketch understanding project. HOUSE is evaluated with respect to its contributions towards satisfying those goals.

1. Motivation

The purpose of this paper is to report on the design of a program, HOUSE, that interprets sketches of polyhedral objects composed of meaningful parts, such as houses. The program, which has recently been implemented, is the latest result of the SEE project, a project set up to explore the interpretation of images designed for person to person communication. The goals of this project are:

i) to develop methods of exploiting the semantics of images designed for communication as typified by sketches,

ii) to explore possible solutions to the chicken and egg problem in perception: sensible segmentation requires interpretation and *vice versa*,

iii) to broaden the scope of vision programs by applying lessons learned in the blocks world to other domains,

iv) to provide an experimental vehicle for studying control structures required to implement schema-based theories of perception,

v) to make available useful interpretation programs for some restricted but important classes of sketches,

vi) to explore the relationship between natural and conventional representations.

2. The cycle of perception and MAPSEE

HOUSE is an offshoot of MAPSEE, a program designed for interpretation of sketches of geographic maps (Mackworth, 1977a). The assumption underlying both programs is that perception is an active process both data-driven and model-driven in character. Mackworth (1977b) has argued that all perceptual processes can be viewed as a cycle consisting of four processes: cue discovery, model invocation, model testing and model elaboration (see Fig. 1). In particular, all vision programs can be usefully characterized by how they embody this cycle. MAPSEE shows that a viable solution to the perceptual chicken and egg problem can be obtained by closing the cycle. In MAPSEE, the cycle is entered in the cue discovery phase, that is, a conservative, tentative segmentation into picture primitives (chains and regions) is done first. A number of picture fragments are identified as cues in this segmentation. These cues give access to a number of domain dependent models.

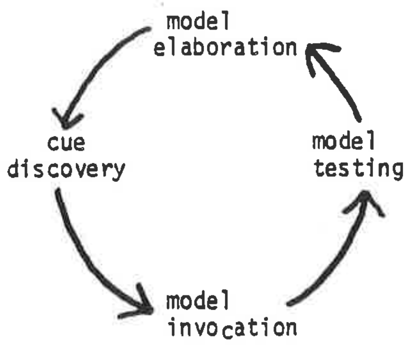


Fig. 1 the cycle of perception.

A model here is an interpretation, a naming, of the parts of the cue. A network consistency algorithm takes these models, together with the primitives in the picture which the models are supposed to interpret, and tests the consistency of the possible interpretations for the different primitives in the picture. The interpretations that survive the consistency tests provide a basis for sensibly refining and extending the segmentation.

3. Why HOUSE?

Clowes (1971) argued that all picture interpretation tasks involve formulating and manipulating descriptions in two distinct domains: the picture domain and the scene domain. Besides simply avoiding the confusion of linguistic category errors (lines and regions exist in the picture domain, edges and surfaces in the scene domain) this approach led to greater precision in the formulation of picture interpretation tasks. In MAPSEE, for example, chains of connected line segments in the cartographic picture domain correspond to rivers, roads, shore lines, coast lines, mountain sides and parts of bridges in the geographic scene domain; regions correspond to seas, lakes and landmasses.

In many tasks, however, the requirements of descriptive adequacy alone dictate that we need more than two distinct domains. Consider, for example, the sketch in Fig. 2a. We need first of all to distinguish the primitive connected points from the straight lines and the regions they appear to define. We need to represent the shape of the edges depicted (convex, concave, crack or occluding), the three-dimensional orientations of the edges and the shapes and orientations of the surfaces depicted (horizontal, vertical,...). Most importantly, we must, in this domain, go beyond three-dimensional geometric structure. We must be able to name surfaces according to their function in this architectural domain (wall, door, window), be able to describe and use their attributes (the walls are vertical) and interrelationships (the window is surrounded by the side wall and coplanar with it), and be able to interpret the whole as a functional entity, a house, as well as a three-dimensional polyhedral object.

In order to make these distinctions it is necessary to fracture the picture and scene domains into seven distinct domains. Since these domains are at least partially ordered with respect to semantic content or abstraction from the original image we shall call them levels. Contrasting this with MAPSEE where image cues invoke scene models we can see that HOUSE requires a cue/model hierarchy. The interpretation strategy in HOUSE is to achieve a consistent interpretation by following a MAPSEE-like cycle of perception at each level before proceeding onto the next.

4. Description

4.1 Levels of Representation

The seven levels of representation in HOUSE are:

1) Sketch level: the picture is represented as an interconnected set of points.
2) Line/region level: straight line representation and region representation.
3) Vertex level: the lines are interrelated by vertices, the region boundaries and shapes are computed.
4) Edge level: lines are interpreted as edges, relating the surfaces connected by the edge. The edge types possible are: convex (+), concave (-), occlude (>), occlude-concave (>)

and crack (c).

5) Orientation level: the three-dimensional orientations of both surfaces and edges are represented. This classification is very crude. Possible orientations are: vertical, horizontal or slanted.
6) Surface naming level: the surfaces carry meaningful names. For example, a surface can be ground, ground* (a horizontal surface coplanar with the ground such as a path), roof, window or door-handle. A surface is a side-face or top-face if it is part of a cube or a wedge.
7) Object level: the image is represented as an object. The possible objects in HOUSE are a cube, a wedge and a house.

Fig. 2 shows an image, interpreted as a house at the object level, represented at the seven different levels in the hierarchy.

4.2 Input

HOUSE receives a sketch in the form of a procedure for drawing it, created by the routines that track the stylus on a data tablet. The input is a sequence of plotter commands, a command being Move (pen up) to (x,y) or Draw (pen down) to (x,y) from the current position. Each series of pen down commands forms a chain of connected line elements.

4.3 Multi-levels of processing

The interpretation process strives to represent the image at the highest level possible. This is achieved by systematically bootstrapping up through the seven levels described above. A consistent interpretation of the image has to be achieved at each level before the step into the next level can be made. The cycle of perception serves as a metaphor for the description of the process. The cycle can be found at each level of processing, stepping through its four stages: cue discovery, model invocation, model testing and model elaboration. The objectives at each level of processing are always: 1) to construct a consistent representation and 2) to find the cues that allow bootstrapping into the next level.

4.3.1 Low Level Segmentation

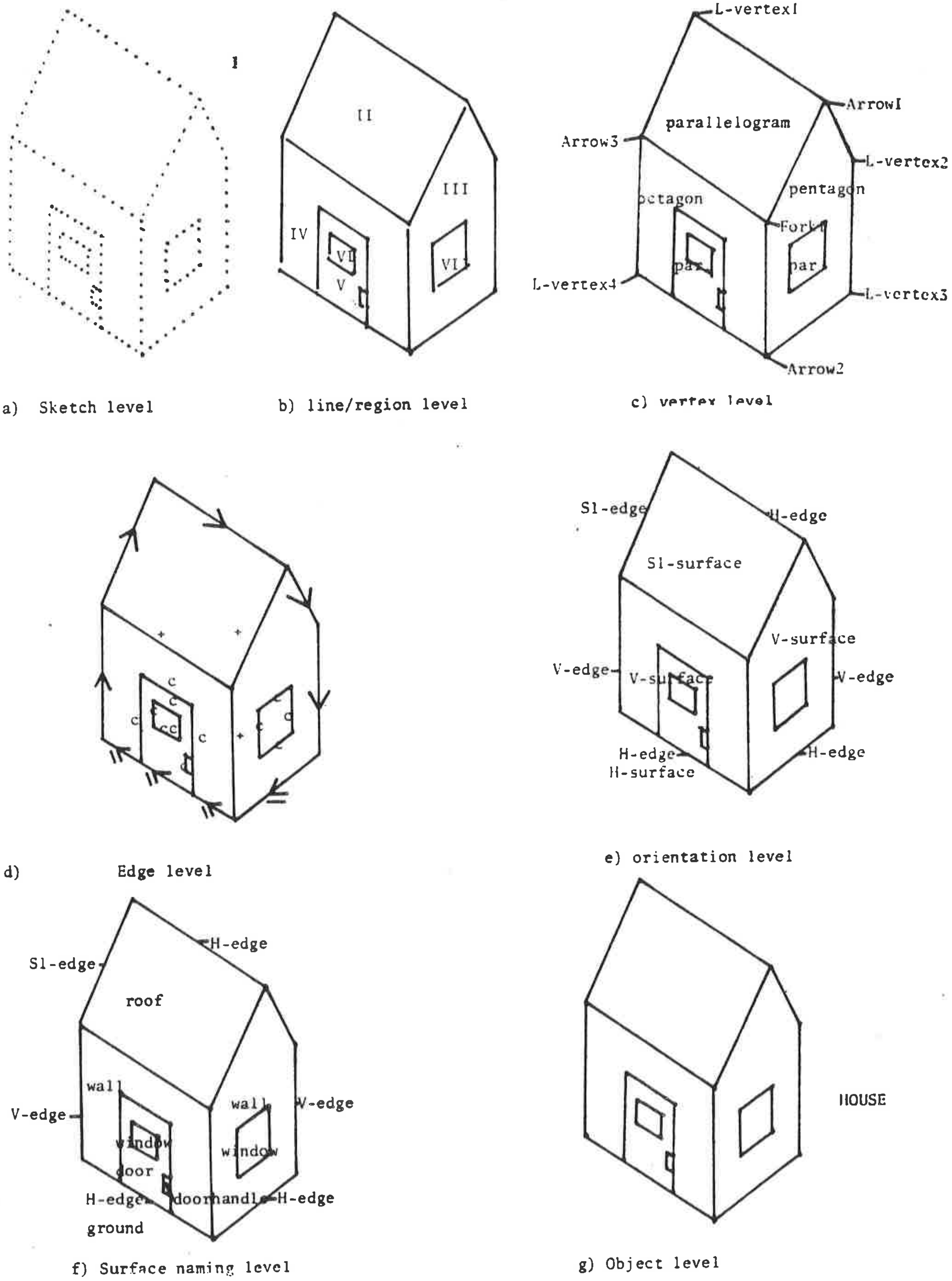
One of the lessons learned from the blocks world is that one needs to maintain a variety of representations each at various levels of detail in order to meet the demands of the interpretation task. These representations are created by means of four different segmentation procedures resulting in point, line, region and vertex representations.

Point formation. The points in the picture are represented in two different ways. First there is a network representation of the set of all points in the picture. Apart from this a coarse array representation is maintained (32x32). Each cell contains the list of points in that area. Quick answers to questions such as "what am I near?" can be given this way.

Line formation. A chain is defined as a set of interconnected points. The coarsest line representation of a chain is the straight line joining its end points. A procedure searches for the point in the chain furthest from that line and uses this point to split the line into two components. The chain is recursively subdivided until there are no free points left.

Vertex formation. The vertices used in HOUSE are: Free-ends, Links, L-vertices, Tees, Arrows and Forks (Fig. 3). Each vertex has its own formation procedure. These procedures are efficient in the sense that they use the line representation of each chain just up to the level of detail they require. The procedures are also conservative. For example, a merge of two Free-ends into an L-vertex or Link will occur iff the distance between the ends is very small. Thus, one prevents vertices from being merged that were not intended to be. Conservative segmentation will often miss genuine cues but, crucially, it will not supply false cues (Mackworth, 1977a).

Region formation. A region segmentation is achieved by subdividing the picture into empty patches, a patch being subdivided only if it is not empty.



a) Sketch level

b) line/region level

c) vertex level

d) Edge level

e) orientation level

f) Surface naming level

g) Object level

Figure 2 Levels of Representation

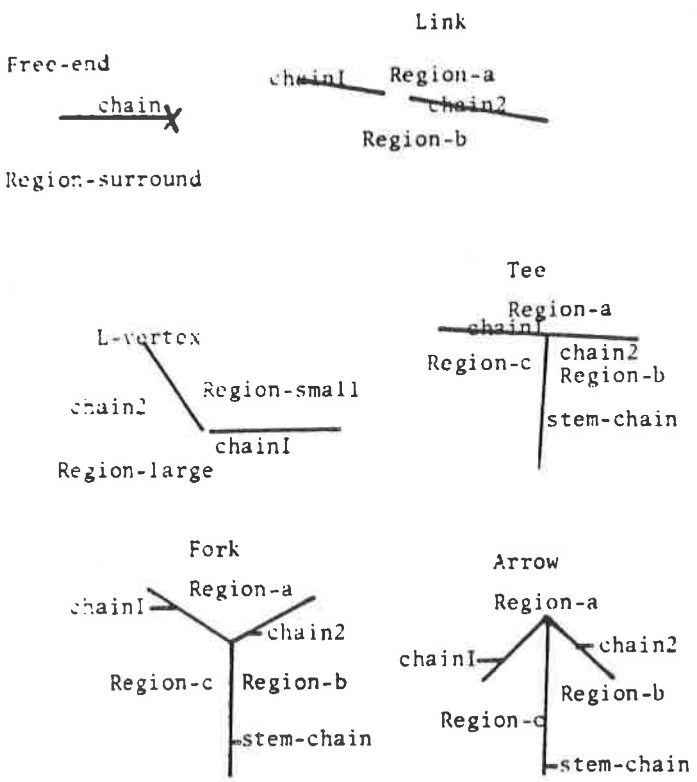


Figure 3. the vertices

Again, for conservative reasons, this process stops at a relatively large patch size.

The low level segmentation process continues through levels 1 to 3. This process cumulates in the formation of the low level cues (the vertices in Fig. 3) that allow bootstrapping into the interpretation cycle of the edge level, level 4. Chains and regions are the primitives in HOUSE. They are constrained by the vertices. Each vertex has a procedure at its disposal by means of which it can find out which regions it constrains. This procedure crawls along the bisector of each vertex line pair. This process is conservatively biased in the same way as the region formation procedure was. It will travel over a distance less than the size of the smallest patch along the bisector. If no region is found a region-ghost is created (Mackworth,1977a). Such a region-ghost stands for the region which has that relationship to the vertex but cannot yet be identified.

4.3.2 Cue interpretation tables

For each level of processing beyond level 3 there exists a set of cues which have procedures attached to them that will allow one or more interpretations for the primitives at that level. Fig. 4 shows the primitives at each level. A few examples of the constraints imposed at each level might be useful.

At the edge level we have used traditional Huffman (1971), Clowes (1971) and Waltz (1972) junction interpretations to interpret the edges.

At the orientation level, we have used extremely crude characterizations of the orientations of surfaces and edges (similar to but much cruder than those suggested by Waltz, 1972) into horizontal, vertical and slanted. The cues here are the edge types. A typical inference is that two surfaces separated by a crack must have the same orientation. A vertical line lying in a surface with a vertical orientation must be a vertical edge. These constraints are essentially compiled versions of the gradient space constraints exploited by POLY (Mackworth, 1973).

At the surface naming level, we use inferences such as, "the ground is horizontal, walls are vertical, roofs can be slanted or horizontal". Relational information such as, "windows share a crack edge with walls or doors and are surrounded by them" is also exploited here.

At the object level, certain cues must be present before the object can be called a house, cube or wedge. Some parts of a house (a putative wall containing a door or a window and connected via a convex edge to a putative roof) must be there before it can be a house. Other parts (e.g. door-handles) are optional, as in Winston's (1975) architectural models.

Edge level

cue type	domain(s)		
	chn		
free-end	convex,concave,occl-conc,occlude,crack		
	chn1	chn2	
l-vertex	occlude	occlude	
	occl-conc	occlude	
	occlude	occl-conc	
	crack	crack	
link	occlude	occlude	
	convex	convex	
	concave	concave	
	occl-conc	occl-conc	
	crack	crack	
	stem-chn	chn1	chn2
arrow	convex	occl-conc	occl-conc
	convex	occlude	occlude
	concave	occl/crack	occlude
	concave	occlude	crack
fork	convex	occlude	occlude
	concave	concave	concave
	convex	convex	convex
	concave	occl-conc	occl-conc
tee	convex	occlude	occlude
	crack	crack	crack
	crack	occl-conc	occl-conc

Orientation level

cue type	domain(s)		
	chn	region-a	region-b
occl-conc	h-edge	v-surface sl-surface	h-surface
	h-edge	v-surface sl-surface	h-surface
concave	h-edge	h-surface	v-surface sl-surface
	v-edge	v-surface	v-surface
	v-edge	v-surface	v-surface
convex	h-edge	sl-surface v-surface	sl-surface h-surface
	h-edge	sl-surface h-surface	sl-surface v-surface
	sl-edge	v-surface	sl-surface
	sl-edge	sl-surface	v-surface
	h/sl-edge v-edge	v-surface	v-surface
crack	h/sl-edge	sl-surface	sl-surface
	h-edge	h-surface	h-surface

Surface naming level

cue type	domain(s)		
	chn	region-a	region-b
occl-conc	h-edge	wall/door roof side-face top-face	ground
concave	h-edge	wall/door roof side-face top-face	ground*
convex	v-edge	side-face	side-face
	h/sl-edge	wall/roof	roof
	h/sl-edge	roof	wall/roof
	v-edge	wall	wall
	h-edge	ground	ground*
crack	h/v-edge sl-edge	window	wall
	region-a	region-b	
inside	window	wall/door	
surrounds	wall/door	window	
common	wall/roof side-face top-face	ground	
	wall/door-handle	door	
	door	wall/door-handle	
	region		
v-surface	wall/door/window/door-handle side-face		
sl-surface	roof/top-face		
h-surface	ground/ground*/roof/top-face		

Object level

cue type	domain
	object
common-wall-door-* convex-wall-roof-h-edge	house
convex-wall-roof-h-edge inside-window-wall-*	house
convex-side-face-top-face-h-edge convex-parallelogram-parallelogram-h-edge convex-side-face-side-face-v-edge convex-parallelogram-parallelogram-v-edge	cube
convex-side-face-top-face-h-edge convex-parallelogram-triangle-h-edge convex-side-face-side-face-v-edge convex-parallelogram-parallelogram-v-edge	wedge
convex-side-face-top-face-sl-edge convex-triangle-parallelogram-sl-edge convex-top-face-top-face-h-edge convex-parallelogram-parallelogram-h-edge	wedge

Figure 4 Cue interpretation tables

4.3.3 Network Consistency

At each level of interpretation, certain configurations of primitives, known as cues, invoke models which specify the allowable interpretations for the primitives. To ensure a globally consistent solution we must find an interpretation for each primitive such that each cue has at least one satisfied model. Two different types of consistency are required for each interpretation. An interpretation should be consistent with the internal description of the primitive (internal consistency) and it should be consistent with at least one of the interpretations for related primitives (external consistency).

External consistency is achieved by a network consistency algorithm, NC (Mackworth, 1977a). Input for this algorithm is a list (actually, a queue) of variable/relation pairs. The variables are the primitive chains and regions, the relations are the cue instances constraining the primitives they are paired with. The domain of each primitive is formed by its set of possible *a priori* interpretations. NC takes the first pair (X,R) from the queue and checks for each value a in the domain of X to see if the other variables also constrained by R have at least one value in their domains that is directly constrained by R. If such a value cannot be found then a is deleted from X. An empty domain for X implies that there is no consistency in interpretations possible. If this is not the case then the queue is replaced by the union of the queue and the set of pairs obtained from all relations other than R that directly constrain X. These two steps are repeated until the queue is empty. If two or more primitives have more than one value left in their domains after this operation, the domain of one of the primitives is split in half and NC is applied recursively. An extensive discussion and elaboration of network consistency algorithms is given in (Mackworth, 1977c). The network consistency algorithm used in HOUSE is the same at all levels of processing beyond level 3.

Internal consistency is provided in the form of filters both in the cue interpretation tables and in the network consistency algorithm. These filters can prevent a model from being validated. For example, a slanted surface cannot have a vertical edge as part of its boundary.

Internal and external consistency are the equivalents of model testing and model elaboration, respectively, in the cycle of perception. The completion of model elaboration either starts the cue discovery at the next higher level or it leads to a resegmentation of the picture. Note that all the interpretations obtained at the present level and lower levels are potential cues for the next higher level.

4.3.4 Resegmentation

Resegmentation can be initiated for many reasons. The door-handle of the house in Fig. 2 is a good example. Because this region is too small, the conservative segmentation will initially overlook it. A region-ghost is created in its place because the region finding procedures crawling up along the bisectors of the angles will all fail to find a region. In the interpretation process, however, a region-ghost is treated as all other primitives. Their domains are provided with a set of initial interpretations, whose consistency is tested by the network consistency algorithm. A region-ghost with an interpretation left in its domain initiates resegmentation, that is, the region segmentation is further refined until the region-ghosts can be located and labelled as regions. Although region resegmentation is used in MAPSEE, it has not been implemented in HOUSE at the time of writing.

4.4 Output

The output of the program consists of a network of interconnected interpretations. An interpretation at each level consists of a list of all the primitives at that level; each primitive is paired with a valid interpretation for the primitive. Each interpretation at a given level is linked to the interpretation at the next lowest level that spawned it and the interpretations at the next highest level that it, in turn, has spawned. The number of such interpretations as returned by the network consistency algorithm varies from level to level. The general trend is that the number of interpretations at each level increases up to the orientation level and decreases beyond this level. For example, a wedge has 7 interpretations at the edge level, 30 at the orientation level, 11 at the surface naming level and 3 at the object level, two of these representing a wedge, the third one representing its concave interpretation.

5. Discussion

It is not feasible to discuss all the implications of HOUSE in this short paper. For such a discussion the reader is referred to Mulder (1978). HOUSE's treatment of the cycle of perception should speak for itself by now. It is shown in Fig. 5.

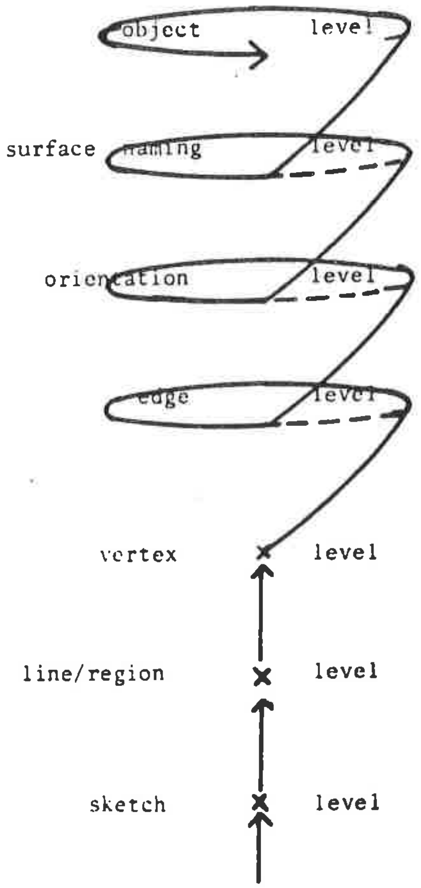


Figure 5. HOUSE's treatment of the cycle

We will limit ourselves here to a short discussion of the contributions of HOUSE to the goals of the project.

i) The exploitation of the semantics of an image in levels buys both modularity (and thus generality) and efficiency. For example, merging the orientation level and the surface naming level would have led to the same end results. However, the total number of combined interpretations to be tested by the network consistency algorithm would be much larger in the merged situation. Ruling out certain combinations at the orientation level prevents similar combinations from being made at the surface naming level.

ii) The chicken and egg problem "segmentation requires interpretation" remains a problem. Both MAPSEE and HOUSE show that a conservative segmentation of the picture, eventually corrected later by means of interpretation, can work. However, it is not difficult to think of examples in which even a conservative segmentation would be incorrect. We are working on solutions of this problem, but none of these has been implemented.

iii) The need for and implementation of multiple representations and levels of details, one of the lessons learned from the blocks world, was mentioned before. For example, the vertex finding algorithms make sophisticated use of multiple representations.

iv) The utility of stratifying the interpretation process does not require one to use a totally bottom-up approach as HOUSE currently does. Within the framework we have presented it should be easy to see that one can start processing at a higher level before finishing at a lower level. With the cycle running concurrently on many levels at once the constraints embodied in the cue interpretation table could propagate vertical consistency (Zucker, 1977) down the levels as well as up.
Schema-based theories of perception require a control structure different from HOUSE's present control structure. Schemata are active processing elements which can be activated in either a

top-down way or a bottom-up way (Bobrow & Norman, 1975). The models in MAPSEE and HOUSE are predicates treated as data structures by the network consistency algorithm. Schemata on the other hand take over control themselves, controlling both internal and external consistency. A next stage of the project may be to implement models that can take over control in order to test internal consistency. Having schemata that control external consistency as well would mean that we have to alter the network consistency algorithm. At this point, the project will probably diverge, one side focussing on the development of a general control structure for the interpretation of sketches, the other focussing on a schemata controlled interpretation. A substantial contribution to schemata-controlled interpretation has been made by Havens (1978a). Havens has designed and implemented a programming language called MAYA (Havens, 1978b) which is a multi-processing dialect of LISP that provides structures for representing schemata, and control structures for coordinating and integrating bottom-up and top-down schema instantiation.

v) HOUSE is a first step towards realistic architectural sketch understanding.

vi) One might expect that interpretations in the block world are more constrained by euclidian than by conventional representations as contrasted with the domain of geographic maps where conventional signs play a strong role. Data collected in an experiment done by Mulder and Mackworth (1978) however show, that slant estimates of cube surfaces by human subjects are controlled by three different schemata. Only one of these is purely geometrical; the other two display an intriguing mixture of geometry and convention.

6. Related Work

We have already discussed the commonality of the ideas behind the two programs and the influence of previous image understanding research. Levels of processing have also been proposed by Zucker (1977) for processes in a low level vision system. Network consistency algorithms and their development from binary arc consistency algorithms (Waltz, 1972) are described in (Mackworth, 1977c). Related algorithms have been proposed by Gaschnig (1974), Barrow and Tenenbaum (1976), Freuder (1976) and Rosenfeld, Hummel and Zucker (1976).

7. Acknowledgments

This research is supported by grants from the National Research Council of Canada to the second author and by a scholarship from the Cultural Exchange Section of the Canada Council to the first author. We are grateful to Bill Havens and Randy Goebel for discussion and comments on an earlier draft of this paper.

8. References

Barrow H.G. and Tenenbaum J.M. (1976), "MSYS : a system for reasoning about scenes", Tech. Note 121, A.I. Center, Stanford Res. Inst., Menlo Park, CA.

Bobrow D.G. & Norman D.A. (1975), "Some principles of memory schemata", in Bobrow, D.G. & Collins, A, (Eds.) Representation and Understanding, Academic Press 1975.

Clowes M.B. (1971), "On seeing things", Artificial Intelligence 2, 1, 79-112.

Freuder E.C. (1976), "Synthesizing constraint expressions", A.I. Memo 370, M.I.T., Cambridge, Mass.

Gaschnig J. (1974), "A constraint satisfaction method for inference making", Proc. 12th Ann. Allerton Conf. on Circuit Theory, U. of Ill., Urbana-Champaign, Ill., pp. 866-874.

Havens W.S. (1978a), "Recognition as a model for machine perception", Proc. of the 2nd Nat. Conf. of the Can. Soc. for Computational Studies of Intelligence (this volume).

Havens W.S. (1978b), "A procedural model for machine perception", Ph.D. Thesis, TR78-3, U. of British Columbia, February 1978.

Huffman D.A. (1971), "Impossible objects as nonsense sentences", Machine Intelligence 6, B. Meltzer and D. Michie, (Eds.), Edin. Univ. Press, Edinburgh, pp. 295-323.

Mackworth A.K. (1973), "Interpreting pictures of polyhedral scenes", Artificial Intelligence 4, 2, 121-137.

Mackworth A.K. (1977a), "On reading sketch maps", Proc. Of IJCAI-77, M.I.T. ,Cambridge, MA., pp. 598-606.

Mackworth A.K. (1977b), "Vision research strategy": black magic,metaphors,miniworlds and maps", in Computer Vision Systems, E. Riseman and A. Hansen (Eds.), Academic Press (in press).

Mackworth A.K. (1977c), "Consistency in networks of relations", Artificial Intelligence 8, 1, 99-118

Mulder J.A. (1978), M.Sc. thesis, Univ. of British Columbia (in preparation).

Mulder J.A. & Mackworth A.K. (1978), in preparation.

Rosenfeld A., Hummel R.A. and Zucker S.W. (1976), "Scene labelling by relaxation operations", IEEE Trans. On Systems, Man and Cybernetics, SMC-6, 420-433.

Waltz D. (1972), "Generating semantic descriptions from drawings of scenes with shadows", MAC AI-TR-271, M.I.T., Cambridge, MA.

Winston P.H. (1975), "Learning structural descriptions from examples", in Winston, P.H. (Ed.), The Psychology of Computer Vision, McGraw Hill, 1975, pp. 157-209.

Zucker S.W. (1977) "Towards consistent descriptions in vision systems", Proc. Of IJCAI-77, M.I.T., Cambridge, MA., p. 709.