MODEL REPRESENTATIONS AND CONTROL STRUCTURES IN IMAGE UNDERSTANDING

Takeo Kanade

Department of Information Science
Kyoto University Kyoto, Japan

ABSTRACT

This paper overviews and discusses model representations and control structures in image understanding. Hierarchies are observed in the levels of description used in image understanding along a few dimensions: processing unit, detail, composition and scene/view distinction. Emphasis is placed on the importance of explicitly handling the hierarchies both in representing knowledge and in using it. A scheme of "knowledge block" representation which is structured along the processing-unit hierarchy is also presented.

I. INTRODUCTION

Image Understanding System(IUS) constructs a description of the scene being viewed from an array of image sensory data: intensity, color, and sometimes range data. Image understanding is best characterized by description, whereas pattern recognition by classification, and image processing by image output. The level and scope of the goal description depend on the task given to the IUS: whether it is interpretation, object detection, change detection, image matching, etc. It may appear that the discussion in this paper will take usally the flavor of scene interpretation from a monocular intensity image.

Observing that there are hierarchies of levels of description along a few dimensions, this paper overviews and discusses model representations and control structures in image understanding. Emphasis is placed on the importance of explicitly handling the hierarchies both in representing knowledge about scenes and in using it, especially processing-unit hierarchy and scene/view domain distinction.

In the next section, the levels of description are identified. Then section III gives an overview and discussion on object-model representations, together with presentation of our knowledge block representation scheme. Section IV deals with the problems of control structure, and finally the role of low-level processing is discussed in section V.

II. LEVELS OF DESCRIPTION IN IMAGE UNDERSTANDING

Descriptions are not only the goal constructs, but also the media through which various components of an IUS communicate in the course of understanding the image. There are a few orthogonal dimensions.

a) Processing-unit Hierarchy

This is a hierarchy in the levels of units used in processing. Let us identify five levels for the moment. For a region-based IUS, they are pixel (an image point), patch(a group of contiguous pixels having similar pixel properties), region(a meaningful group of patches corresponding to a surface of an object), subimage(a part of an image corresponding to an object or a set of objects), and object(an object as a real entity). For a line-based IUS, the level of patch can be replaced by line segment, region by line, and subimage by a set of lines corresponding to an object, Fig. 1 illustrates these levels for a region-based IUS.

Akin & Reddy(1976) observed that six levels are used when human subjects understand the contents of an image through verbal conversation: scene, cluster, object, region, segment, and intensity. The number of levels is not very significant. These levels as well as those in Fig. 1 depend on the units on which different levels of processing are performed and for whose description different vocabularies are used. Processing in the pixel-to-patch level is often called as low-level processing. The region-to-subimage level is high level in the picture processing domain. It clearly needs to deal with semantics which stem from the highest, object level. The patch-to-region level might be called as intermediate.

b) View Domain / Scene Domain Distinction

The point to be noted here is the clear disparity existing between view-domain and scene-domain descriptions; in Fig. 1, the lower four levels are in the view domain and the upper one in scene domain. The need for this distinction was argued for first and most effectively by Clowes(1971). He used the term "picture domain" in place of "view domain". But the latter is used in this paper to mean the domain of observable facts by viewing the scene in either intensity or range data. The importance of this distinction is readily understood by thinking that, for example, the actual meaning of "adjacency" in the view-domain description is fully understood only after the relation is interpreted in the scene-domain description. Note that the scene-domain descriptions are not necessarily in a metrical 3-D coordinate space; e.g., Waltz's labels of edge is a symbolic system to represent the edge types in the 3-D space, or even a gross subjective space will suffice.

c) Detail Hierarchy and Composition Hierarchy

The detail hierarchy is along preciseness of description. It can exist in both the view and the scene domains. Section 5.2 presents examples in the view domain. An example in the scene domain is the description of overall/detail shape of an object, which is found in section 3.2 b). The composition (or part-of) hierarchy represents part/whole relationships in the scene domain.

The processing-unit hierarchy actually contains somewhat both aspects of the detail and composition hierarchies in the sense that the low-level entities are parts and details of an upper-level entity. Unfortunately this revealed hierarchy does not directly correspond to the hierarchies which naturally exist in the scene domain. This fact makes image understanding difficult, and it is why the models often need to represent the natural hierarchies

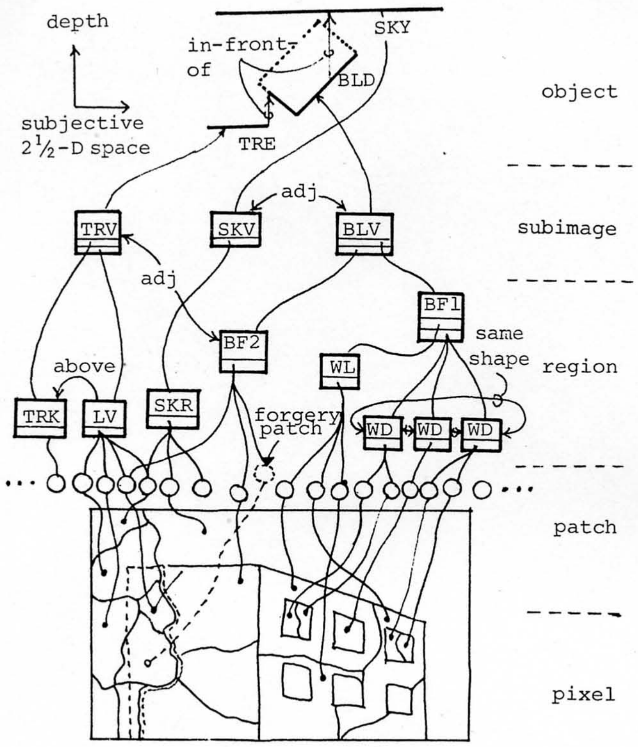

Fig. 1 Illustration of Levels of Description in Processing-Unit Hierarchy

explicitly using processing-unit hierarchy in order to bridge the gap between the view-domain and the scene-domain descriptions.

III. MODEL REPRESENTATION

There are many kinds of knowledge for an IUS. The task world of an IUS is defined first of all by what objects (or class of objects) are concerned and how they behave and interact to form a scene. Therefore let us confine ourselves to the object-model representations. Corresponding to the levels of description it can take several forms.

3.1 View-Domain Models

This is an approach in which properties and relations in the view-domain descriptions are stored as the model of an object. It is interesting to observe that most of the region-based scene-interpretation programs have taken this approach. The region analysis of image was first used by Brice & Fennema(1970).

a) Graph Matching at Region Level

The first group of the region-based view-model approach includes Barrow & Popplestone(1971) and Preparata & Ray(1972). They store properties of and the relations between regions in the form of a graph; the nodes correspond to the regions in the image which correspond to surfaces of an object or part of an object, while arcs correspond to relations between regions. Interpretation consists of graph matching or subgraph matching; i.e., finding the "best" assignments of part or object names to nodes of the graph obtained from the image, so that specifications stored in the model graph are maximally satisfied.

It is apparent that the approach can not cope with occlusions or rotations of objects that change the graph structure drastically. But a more serious limitation is that it is assumed that the (almost) perfect, meaningful partitioning of the image into regions(i.e., region-level descriptions of the image) is obtained independently, which is now known to be very difficult.

b) Semantic Grouping of Patches into Region

Yakimovsky & Feldman(1973) and Tenenbaum & Barrow(1976) are the next group. They tried to overcome that limitation. The image is first partitioned into many small patches of uniform color. To this patch-level description of the image, a merging operation is repeatedly applied, with which the operation of giving an interpretation to the patch is combined. That is, besides intensity and color data, semantic constraints are introduced in deciding a possible merge, by means of a set of probabilities(Yakimovsky, et.al.,1973) or a constraint table(Tenenbaum, et.al.,1976) about combinations of object-name labels and region-level properties and relations such as adjacent, above, etc. Some uniform, domain-independent procedures are used to find a "best" segmentation; in the former, it is (sub) optimization of combinational probability that regions have correct labels, and in the latter, it is the use of filtering procedure together with a relaxation method which repeatedly eliminates and suppresses inconsistent labels from a set of possible labels for each patch.

It was an advance that they succeeded in introducing some semantics into image segmentation. But because all the patches and regions are uniformly treated and knowledge is scattered in the constraint representation of one level(patch level), the program does not know what objects it is dealing with at each moment. Therefore, neither explicit processing of the shape of object nor object-dependent processing on part of image is easy. The reason, in the context of Fig. 1, is that the uniform procedure does not construct a subimage-level or even region-level description to control where and what to look at.

The limitations mentioned above are not inherent to region-based interpretation schemes. They stem mainly from failures in handling the processing-unit hierarchy explicitly and in constructing each level of descriptions in the interpretation process. If this is solved properly, then regions can be powerful description primitives for natural scenes, as lines are for polyhedral scenes.

3.2 Three-Dimensional Shape Model

a) Computer Graphics Metaphor

The most straightforward object model stores 3-D shape data as in computer graphics. In the pioneering work of Roberts(1965), the object model was very straightforward; it was a set of 3-D coordinates of the vertices of polyhedra.

The input image is processed and transformed into a line drawing. The matching process consists of the following processes; 1) junctions in the line drawing are selected that may constitute a simple polyhedron (wedge or parallelepiped), 2) the proposed polyhedron in the 3-D model is generalized-transformed (rotated, translated, scaled, and projected) to match the selected junctions, and 3) the matched piece is then removed and the remainder is considered. Here no view-domain model is employed. Everything is considered in the 3-D space. The interaction of objects is treated as addition and subtraction of volumes in the 3-D space. The upward flow from view to scene is based on simple heuristics relating a view to the object model; e.g., a set of junctions that form three parallelograms are possibly from a parallelepiped. Falk(1971) is in the same spirit; in order to deal with imperfect line drawings, an aggregate of local cues such as Y, L or ARROW junctions are used to hypothesize a plausible object model.

More recent extremes of this line of pursuit, perhaps with increased manipulation power, are the geometric modeling by Baumgart(1974) and a program that uses Braid's scheme(Braid, 1974) for computer vision (Popplestone, Brown, Ambler & Crawford, 1975). The system by Baumgart produces polyhedral approximations to 3-D shape from multiple views of an object. The system by Popplestone, et. al constructs 3-D body models by using the technique of projecting light stripes tc know the surface shape. These might be described as the inverse of computer graphics problem.

b) Generalized-Cylinder Representation

A generalized cylinder is formed by moving a two-dimensional cross section along an axis. The cross section need not remain constant and the axis need not be straight; an ordinary cylinder is a circle moved along a line through its center. If the circle shrinks linearly, then it is a cone. An object is represented by decomposing it into parts each of which is a generalized cylinder. Agin & Binford(1973) and Nevatia & Binford(1973) used range data by means of a laser range finder to obtain the generalized-cylinder representation of objects such as a doll and a horse, and then matched the obtained description against the models.

Marr & Nishihara(1976) showed a hierarchical representation of 3-D shapes using cones. A human figure is first approximated by a cone. It is then detailed by joining component cones(HEAD and LIMBs) to the principal cone(TORSO). Each further component(e.g. ARM) can be redetailed in turn, and so on. This scheme can answer overall questions about the object like the gross shape, the direction in which it points, etc. and also one can go into as appropriate detail as wanted. Note that the hierarchies represented here are the detail and composition hierarchies intrinsic to the object.

As for the recognition problem from a monocular image, Marr & Nishihara(1976) write that work has been done to obtain the projected axes of the component cones from the image of a shape and also that the task can be done nearly independently of other higher-level tasks. From experiences with line finding in simple polyhedral scenes and with skeletonization in 2-D figures, however, it is felt that it is very difficult to obtain the axes reliably and to construct the description *which* the model of that object expects, without some feedback of hypothesis-and-test (except the case of obiects basically composed of sticks). More remains to be done in this respect, but it is true that the representation of overall/detail hierarchy will provide a sequence of cues that can be used in hypothesizing plausible shape models for understanding the input image, if it is explicitly related to the processing-unit hierarchy. In fact, it is shown that the contour, an overall description of the processing unit, may be closely related to the generalized-cylinder representation(Marr,1976).

3.3 Relating Scene Constraints to View Constraints

In several cases, image-forming models are successfully formulated and provide methods of working in the view domain to obtain the scene-domain description.

a) Line-Drawing Interpretation

Waltz's program(Waltz,1972) for line-drawing interpretation can be counted as a most typical success. Though seemingly similar, it has a very different spirit from Guzman(1968) which decomposes a line drawing into bodies. Guzman classified junctions of lines in the image, and considered them to give heuristically some local evidence concerning the possible association of regions in forming bodies. For example, an ARROW junction provides evidence for some association between the two regions on either side of the central line. Guzman's program worked fairly well, but its limitation came from sticking to the view domain.

To the contrary, following Huffman(1971) and Clowes(1971), Waltz(1972) classified edges according to their scene-domain, physical meaning: concave, convex, crack, shadow, etc. Possible junctions are enumerated from the possible views of physically possible vertices that arbitrary trihedral solids will generate. Then they are indexed to give a catalog of legal line combinations for each type of junctions. Having the catalog prepared, the interpretation of a line drawing reduces to searching for a set of line labels that provide a legal configuration at each junction in the image. The procedure systematically eliminates incompatible labels from a line when the junctions at both ends of the line is considered together. Astonishingly enough, it was found that this iteration scheme, called "filtering", rapidly converges to a unique interpretation or to a small number of interpretations for most cases.

This success achieved by shifting from Guzman to Waltz is that Waltz begins with the deep structure(scene-domain meaning) and relates it to the constraints computable in the view domain. A similar approach for line drawings of curved objects is taken by Turner(1974), although with more complexity.

b) Shape from Shading

This addresses how the intensity of the observed image can be used to reconstruct the 3-D nature of the corresponding surface of the object. Horn(1977) gives an elegant formulation by use of the gradient space approach(Huffman,1971; Mackworth,1974). If $z=f(x,y)$ defines the surface of the object, the local surface orientation can be expressed using $p=\frac{\partial f}{\partial x}$, and $q=\frac{\partial f}{\partial y}$. The gradient space is the space defined by (p,q). For a particular choice of viewer-object geometry, light source, and a kind of object surface, it can be calculated how the intensity data(I) depends on p and q; i.e., $(p,q) \rightarrow I(p,q)$. For example, in the simplest case where a light source is at the viewer and the object surface reflectivity is proportional to cosine of the incident angle of light, then

$$I(p,q)= 1/\sqrt{1+p^2+q^2}$$

This means that the points with a given intensity have such orientations determined by a circle in the gradient space. Other constraints in the gradient space can be used in conjunction with this constraint to determine the orientation uniquely for each point. For example, the three surfaces of a trihedral vertex correspond to three points in the gradient space and the lines connecting these points must be perpendicular to the three lines joining at the corresponding junction in the image.

A practical example of combining picture processing with simple camera geometry is found in Yoda, Motoike & Ejiri(1975). It provides an insensitive-to-noise algorithm that obtains the normalized top view of a box-like object on a table from its oblique image. The histogram of edge-segment directions, when modified in accordance with camera angle, gives reliably the orientation of the box on the table.

All the representations and procedures mentioned above can be thought of as compiled knowledge that relates scene-domain constraints to view-domain constraints(Winston, 1973).

To sum up the three preceding subsections:

1) Scene-domain knowledge is crucial.
2) The processing-unit hierarchy should be explicitly represented and handled. Mingling them results in the misuse of the descriptive vocabulary for shape and relations.
3) 3-D shape models embody generative nature to manage various cases, but in general they are weak in hypothesizing appropriate candidates for object models in recognition. Hierarchical overall/detail representations may be helpful, if properly related to the processing units.
5) Compiling the knowledge of relating scene constraints to view constraints is a powerful technique, but it works in the controlled world whose constraints have been "compiled" into the method.

3.4 Multiple-View Model

Minsky(1975) proposed the theory of frame systems as a unified theory of knowledge representation. As Minsky describes them, when applied to vision most straightforwardly, frame systems are collections of related frames linked together; different frames describe the stereotyped view (not necessarily of a single object, but of a scene with multiple objects like a room) from different viewpoints, and the transformation from one frame to another represents the effect of moving from place to place. Though very attractive as a psychological model, it seems that this multiple-view frame systems, as they are, have gone too far in neglecting direct, metrical processing of 3-D information. Anyway it is not clear yet how well such representations work for real images, since no implementation exists yet.

The frame theory itself presents a foundation of representational schemes. Among others, it advocates that matching against a stored set of expected prototypes and instantiating them are the central recognition process. Each prototype should contain a chunk of data and procedures which are used in applying it. These ideas can be developed in both the view domain and the scene domain.

3.5 Structuring Multiple-Level Descriptions

The importance of explicitly handling the hierarchy of image descriptions has been stressed. One thing we can do about the object model representation is to store knowledge applicable for each level of descriptions together with information about how it is related to the upper- and lower-level descriptions. Sakai, Kanade & Ohta(1976) used symbolic "knowledge block" representations for outdoor-scene interpretation. Here we extend it a little to represent processing-unit hierarchy explicitly in it. Our system is region-based and, an image is first partitioned into a collection of patches by the recursive thresholding technique(Ohlander,1975); see 5.1. Thus we can assume that the image has been completely described in that level; all the properties of and relations between patches are known or computable.

As an example, the representation of a typical building would appear as in Fig. 2. Each block having a name starting with * is called a knowledge block(KB). A KB can stand for not only a processing unit concerning an object, but also material (e.g., *CONCRETE), property(e.g., *RECTANGULAR), or relation(e.g., *INHERIT-PROP). It is like a conceptual object in the KRL(Bobrow & Winograd,1976).

*BUILDING in the object level specifies its gross shape by *RECBLOCK. The parameters are its size, location and orientation. Properties and relations valid in this level can be included. The object model is related to a few of the qualitatively different views depending on the value of parameters. The connecting parameter values are given when working downward, and inferred when working upward. Note that, especially in outdoor scenes, the size, location and orientation are relative matters. They need not be very precise for most cases as long as no contradictions occur. Note also that the distant objects such as sky, or objects with fuzzy shape such as trees can be thought of as flat patterns perpendicular to the viewer. It might be said in this sense that the scene domain is a subjective $2\frac{1}{2}$-D space.

The view-domain units represent the hierarchy; how the subimage corresponding to a particular view is composed of regions, and how the region is composed of patches. The subimage KB contains a procedure to infer the scene-domain parameters. During the interpretation process, each KB for subimage or region-level unit generates its instance by linking region or patch instances as its part descriptions. The pool of those instances together with initial patch-level descriptions constitutes the data base for image description. The boxes in Fig.1 can be regarded as those instances. In the course of interpretation their links might be partially completed. Usually the number of patches to be linked to a region instance is not known beforehand; this makes the actual interpretation difficult.

Generally speaking, each KB contains three types of rules: ASK, SELECT, and CHECK. These rules are represented either by a list of fuzzy predicates, or as a procedural attachment(Bobrow & Winograd, 1976). The ASK rules are used to choose entities as candidates for the component of the instantiation; for example, a patch with the properties of many-holes, many-lines, inheriting-properties-of *CONCRETE, and adjacent-to *SKYREGION with linear boundary may well become a component patch of *WALL. The ASK rules function as bottom-up triggers of proposing an instance of the KB.

The SELECT rules are used in trying to extend the instantiation as far as possible under the KB's own control; for example, *BLDFACE tries to identify the holes in the region of *WALL as *WINDOWs, and checks if they all satisfy some relations expected of walls and windows. The SELECT rules correspond to a goal-directed analysis on a part of image, taking advantage of the facts known under object-dependent control.

*RECBLK	type	3-D shape
	parameter	size(a,b,c), location, orientation
	procedure	to generate a block in the space,given the parameters
*BUILDING	type	processing unit(object)
	shape	(*RECBLK parameters)
	relation	(in-front-of *SKY) ...
	view	(*BLDVIEW1 parameter-range) (*BLDVIEW2 parameter-range)
*BLDVIEW1	type	processing unit(subimage)
	view-of	(*BUILDING parameters) procedure to infer parameters
	part	(*BLDFACE(1) *BLDFACE(2))
	CHECK	check the relation between the parts
	SELECT	if one of the part is instantiated, search for the other
	ASK	region(*BLDFACE)
*BLDFACE	type	processing unit(region)
	part	(*WALL *WINDOW(N))
	CHECK	check the shape, regularity of *WINDOWs ,etc
	SELECT	try to identify holes in *WALL as *WINDOWs
	ASK	region(*WALL)
*WALL	type	processing unit(region)
	part	(patch(N))
	CHECK	check color, shape, etc.
	SELECT	try to expand the region keeping the properties,to a shape(*RECTANGULAR parameter)
	ASK	patch(x); ((many-holes x) (many-lines x)(*INHERIT-PROP *CONCRETE x)(linear (boundary x *SKYREGION)))
*WINDOW	type	processing unit(region subordinate)
	CHECK	check shape, color, etc.

Fig. 2 Part of Knowledge Block Representation of BUILDING

The CHECK rules verify and evaluate the rating of how much the present state of the instantiation satisfies the KB. This rating is used by a control structure in selecting proposals made by the KBs. The possible control structures for these representations will be discussed in 4.2.

IV. CONTROL STRUCTURE

Control structures are the strategy of using the knowledge to efficiently construct the goal descriptions. For one thing, it is strongly dependent on the representations employed. Waltz's program for line-drawing interpretation required no sophisticated control structure. However, this exceptional simplicity was obtained not only because of the careful choice of description schemes but also because of the assumption of (almost) perfect line drawings as input. When one deals with real intensity images, one has to cope with more uncertainty. It should be noted here again that this does not justify a straightforward use of probabilistic or optimization techniques which mix up everything in one level. Let us first have a brief look at the spectrum of the control structures embodied in vision programs which treat actual image data. Then the control structure for the knowledge block representations will be discussed.

4.1 From Pass-Oriented to Heterarchical Control Structure

The pass-oriented structure, or linear (bottom up) sequencing of transformations, is the most straightforward control structure; it builds up higher-level descriptions step by step. The typical sequence is: 1) noise removal, 2) edge-segment finding, 3) grouping of edge segments, 4) line drawing, and 5) interpretation. The lowest level is usually a universal technique. The higher the level is, the more the process is domain-dependent. Though it is simple and modular, such a control structure is not always reliable. Errors in the earlier stages seriously damage the later stages, and it is very difficult or even impossible to make the earlier stages error-free without using knowledge of the later stages. It is noted, however, that recently several people including Marr(1975) raise a reconsideration of this point; Marr claims that it is necessary to clarify how much can be done in each stage independently, before going to rich and complex interactions between stages.

A hierarchical top-down gross-to-detail control, directed by a model, is an efficient way to detect a particular pattern in an image (Harlow,1973; Ballard & Sklansky,1974). The recognition process takes the form of a decision ladder or graph, whose subsequent lower nodes correspond to decisions to be made concerning more detail in a smaller area.

A feedback analysis procedure was described in

Nagao(1972), and exemplified in the face recognition programs(Sakai, Nagao & Kanade,1972; Kanade,1974). The program consists of many routines, each of which corresponds to a component of the face such as nose, eyes and mouth, and is programmed to detect the component in a given small area. The program works basically in a hierarchical top-down manner, in the sense that the ordinary order of calling routines into action is predetermined on the global-to-local basis. But its feature is that when something goes wrong (failure or inconsistent detection) within a routine, then the control goes back to former ones and retries them to correct or refine the parameters that might have caused the error.

A more complex control structure is mixture of bottom-up and top-down (and further middle-out) (Turner,1974; Popplestone, Brown, Ambler & Crawford, 1975). The ultimate style would be heterarchy, in which a number of modules work together like a community of experts with no strict central executive control(Winston,1973). However, it is very difficult to include the low-level processing in such a style of cooperation. Though not heterarchy, the Shirai's program(1973) to obtain line drawings of polyhedra has embodied the most rich interactions between high-level and low-level routines. It is based upon the strategy of constructing a line drawing step by step. At each time the most probable line is hypothesized and verified by making use of previous results. Recognition starts from 1) contour lines with a black background and goes into 2) other boundary lines between two bodies and 3) internal lines of bodies.

4.2 Control Structure for Knowledge Block Representations

A control structure of the system that employs the knowledge block representations will be discussed; it is again based on the control structure employed in Sakai, Kanade & Ohta(1976).

a) Each KB Proposes Its Existence in Its Own Way

It appears that the most crucial point is to know where in the image one should apply which object model. It is very natural to start with a part of the image where strong features exist; the result will help in understanding other parts. But who knows what strong features are important? Each object has its own cues at its own level of description, strong or weak. For example, in a city-outdoor scene, a light bluish patch in the upper part of the image suggests sky rather strongly. On the other hand, a brown patch suggests a wall of the brick building only very weakly. The evidence becomes stronger, if it has a linear boundary with the sky. It becomes much stronger if the patch together with neighboring patches forms a rectangular region with a regular substructure. What cues are used, what relations can support the evidence, and up to what level they are to be grouped? All these depend on each object. Therefore rather than using a centralized hypothesizer, a distributed hypothesis-making is desirable.

The concept of the control structure is depicted in Fig.3. As was mentioned in 3.5, the pool of plausible instances of processing-unit KBs and the initial patch-level descriptions of image constitute the data base that represents the present state of image descriptions. Each knowledge block looks at the data base. If it finds the cues (by ASK rules), it begins its own processing and continues as far as it can (by SELECT rules). If it checks that enough evidence has been found (by CHECK rules), it proposes to add its instance(partial image description at that level) to the data base. Thus the KBs work in parallel and communicate through the common data base. The blackboard model of control structure (Erman & Lesser,1975) used in speech understanding provides a good metaphor at this point.

Primitive functions for accessing the data base are prepared. The existential fetch;

```
EX-FETCH[ x, t,<specification of evaluation>]
```

is such a function. It selects from x, a list of entities, one that satisfies the specified conditions more than t, and returns a pair of the entity and its evaluation value. For example,

```
EX-FETCH[ ALLP, 0.7,
   '(LAMDA (s) (F-AND (below s *RGN) (dark s))) ].
```

The evaluation is the value of fuzzy AND of the two fuzzy predicates. ALLP is the reserved variable for all patches. *RGN is the unit itself that the KB using this EX-FETCH represents.

b) Proposal Selection Is Necessary

It is apparent that if each KB is allowed to add its proposal to the common data base freely, the data base will soon explode. Some rating should be attached to the proposal. The simplest proposal-selection mechanism selects an instance description with the highest rating at a time, and adds it to the common data base. The addition has two effects: 1) It may change the rating of instances of other KBs. Each KB has a list of KBs which refer to it. Thus what KBs have to recompute is known. The effect of recomputation may further propagate, repeatedly. 2) It may trigger the next-level KBs to propose the existence of their instances.

When a subimage instance is generated, the corresponding object description is inferred as a component of the scene description. Now, following the model downward, the view of the object can be verified in the view description so far developed. The links left incomplete in the various levels of instances, which may correspond to details of the

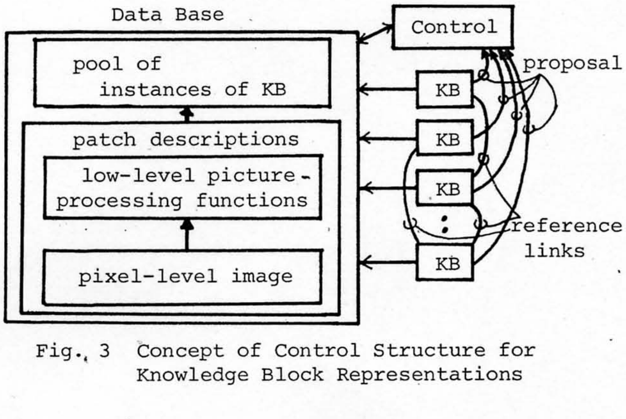

Fig. 3 Concept of Control Structure for Knowledge Block Representations

object, are completed, and the incorrect links which may have existed near the boundary are cut. Instances that substantially overlap in the image with the verified one can be eliminated.

Therefore, usually those objects which reveal their existence strongly, or which are not occluded by others, are processed first. If the image involves only weak occlusion and each object in it shows up clearly, all this will suffice. This is actually what happens in Shirai(1975) for multiple-object recognition in the desk scene which includes a telephone, lamp, and bookstand.

Much more sophisticated optimization mechanism would be possible and sometimes desirable(Barrow & Tenenbaum,1976). However, the most important thing is to build up descriptions by basically data-driven analysis, so that the goal-directed, object-dependent analysis can be applied to an appropriate part of the image.

c) An Example of Occlusion Inference

When an object is recognized and verified, its boundaries are checked. Let us take a simple example. The boundaries may be divided into three parts:

1) The boundary which shows the object's own shape or properties: For example, part of a linear boundary for a building, or zig-zag boundary for a tree. It suggests the occluding boundary.
2) Contrarily, the boundary which shows the properties of the neighboring objects: It suggests the occluded boundary
3) Not clear.

The first and second evidences give some information about relative positions between parts of objects. In-front-of and behind-of pointers are added to the object-level description in order to give a partial ordering in depth.

In order to exploit this finding, the information is passed and added to the description of the corresponding patches; i.e., what part of their boundary is occluding or occluded. A region instance, which has been suspended because part of the necessary area is not yet detected, can use this information as an excuse for that defect, and may raise its rating. The goal-directed process in the SELECT rules can expand the region across the occluded boundary as if the expected conditions were satisfied. A simple method to transfer this effect to the next higher level description is to generate special "*forgery*" patches(see Fig.1) that have the desired properties, and link them to the region instance.

d) Competing Instances

Although the instances kept in the common data base are the most plausible ones, they are not always correct in all details. The instance made in a later stage by the goal-directed procedure of the SELECT rules may want to have a link to an entity to which another instance already has a link These are competing instances.

The problem of conflicting hypotheses is discussed in multi knowledge-source systems(Erman & Lesser, 1975; Barrow & Tenenbaum, 1976). The multi-context control is necessary. However, in image we can further take advantage of the local independence of parts of the image; i.e., the fact that the influence of a decision at some local point often does not extend too far. Unless the overlapping area is large, the competing instances can be treated as if both are correct at the same time. Minor inconsistency of the context is not so serious. The number of contexts to be treated separatedly becomes smaller.

In the verification process, after one object is found, the part of the image in competition can be interpreted through negotiations among the competing units which have links to that part. This time the problem has become classification or comparison between them. Usually it is easier and the results are more reliable.

V. ROLE OF LOW-LEVEL PICTURE PROCESSING

One of the classical and somtimes misleading views about low-level processing is that it is information reduction process. This view leads to attempts to transform the input image directly into very compact, minimally sufficient form such as a line drawing. Rather the low-level process should be viewed as information structuring process of raw image data, so that as much information as possible may become accessable from other knowledge sources. This alternative view leads us to regard the pixel-level image data plus various low-level picture-processing functions as the structured data set that can answer questions from upper levels. This data set is the most basic part of the data base used in the image understanding process(see Fig.3).

5.1 Adequate Descriptive Vocabulary

In order that the structured data set works, adequate descriptive vocabulary is necessary to describe the image in terms of low-level entities. One basic attitude is to *describe* the cases which have been conventionally *detected* as a single YES/NO event(such as "an edge exists"), and to have the next higher-level unit interpret the resultant description on a bit more global basis. Another point is to recognize that a large amount of computation is necessary to obtain adequate descriptions of the image with which to start the bottom-up, data-driven analysis; stinginess in description or computation is not to be pursued.

The primal sketch(Marr,1975) is a rich symbolic description computed from the image, so that it can be the input of the next level. A typical feature description about edge at a point would be like;

```
( EXTENDED-EDGE  (POSITION  (100 125))
                 (CONTRAST  12)
                 (ORIENTATION  63.5)
                 (FUZZINESS  3) ).
```

Creating a primal sketch requires application of local filters of various type, size and direction for each image point.

A simple working region-oriented technique is recursive thresholding technique by use of multiple histograms in order to partition the image into patches. This was first used by Tomita, Yachida & Tsuji(1973) in segmenting a textured scene. Ohlander (1975) used it in segmenting natural color scenes.

The method is very simple. First, calculate histograms of each of available local features of the image point. The features are such as intensity, component of color (red, green, hue, etc.), gradient, or occurrence of a local pattern. Then select a feature whose histogram has two (or more) separated peaks; it means that not all the image points have similar feature values, but that they form two (or more) groups. Finally threshold the image at the values which separate the peaks, and partition the image into several connected parts. The sequence of these operations are applied recursively to each of the resultant parts, until each one has monopeak histograms for all the features, which shows that it is almost a homogeneous patch.

The output product of this region-oriented low-level processing is not again simply the segmented picture. It consists of a list of patches, a list of boundaries, and a list of vertices, each of which includes their attributes and links to represent their relations. This does not necessarily mean that all the values have been computed beforehand; it is too inefficient. Facilities such as Memo functions(Michie,1968) are valuable.

5.2 Detail Hierarchy

The product of the low-level processing can show the detail hierarchies at that level along a few dimensions. It has to be structured so that higher-level components can exploit them to work from overall to detail or from strong to weak features.

1) Spacial dimension-- A pyramidal image-data structure is often used by dividing the image into n x n neighborhood and mapping(typically averaging) each neighborhood into one pixel of the next level image. Descriptions constructed corresponding to this hierarchy can give the detail hierarchy along the spacial dimension(Nagin, Hanson & Riseman,1977).
2) Feature dimension-- When the picture segmentation is done sequentially using the most distinguishing feature at a time, as in the recursive thresholding method, the total result gives a tree structure of segmentation. It is along the feature dimension.
3) Reliability dimension-- For a particular feature detection, a linear hierarchy is obtained along reliability, or complementarily, along fuzziness of the detection.

5.3 Associative Retrieval

Once adequate symbolic representational vocabulary is defined to describe the result of the low-level picture processing, it becomes desirable for higher-level functions to retrieve information from the data set in a unified, logical way, perhaps by associative retrieval(Yakimovsky & Cunningham;1976). For instance,

$\mathrm{Area} \otimes \mathrm{Patch} \equiv$ number of pixels
$\mathrm{Boundary} \otimes \mathrm{region} \equiv$ (a set of lines)

In fact, the lack of such an ability of smooth interface between low-level and high-level functions has been an obstacle to developing a vision system which is largely based on symbolic models.

One can think of information retrieval with some inference and/or data manipulation capability, in addition to the simple associative retrieval and the FETCH-type functions mentioned in 4.2 a); for example, "Find a set of patches such that *the region they form* has such and such relations with PATCH3". It suggests research on data set manipulation language for image understanding. The relational data model could supply a basis.

5.4 Controllability

This property is related to the top-down aspect of control. The top-down analysis requests a low-level picture processing program to verify or detect specific conditions. This means that the functions need to be programmed with controllability so that they can do as exactly much as the given guidance requests. A typical example is found in Shirai's Line Finder(Shirai,1973). The circular search procedure is for searching for lines starting at a given point when the direction of the line is not known. This problem is decomposed into successive applications of the line-segment detection in possible directions. The detection procedure is supposed to search (confirm or deny) a line segment with a given direction in a given search area.

This example suggests that a low-level vision program, for instance an edge detector, need not be "general" to deal with a broad class of edges. Interestingly enough, the opposite effort has been made as an image processing technique; i.e., attempts to devise an edge detector that always works for "any" type of edges. Low-level programs become more useful when they are parametralized to enable them to be specialized according to the given specification; for exmple, direction, type, and size of the window for edge detection.

VI. SUMMARY

The problems of model representations and control structures of IUS have been discussed, mainly from the viewpoint of interpretation-oriented tasks. It is very important to handle explicitly the hierarchy in the levels of image descriptions, especially to reflect the natural hierarchies into the processing-unit hierarchy.

After a few types of representations were reviewed with discussion, the knowledge block representations were described that store knowledge for each level of description together with how it is related to the upper- and lower-levels.

The core part of an IUS is basically a symbolic process. The low-level picture processing must have a smooth interface with the symbolic process. Considerations were given about adequate descriptive vocabulary, detail hierarchy, associative retrieval, and controllability in the low-level processing.

ACKNOWLEDGMENT

I would like to thank Professor Toshiyuki Sakai, and Mr. Yuichi Ohta for productive discussion.

REFERENCES

Agin, G.J. & Binford, T.O.(1973): "Computer Description of Curved Objects", Proc. IJCAI-III, 629-640.

Akin, O. & Reddy, R.(1976): "Knowledge Acquisition for Image Understanding Research", Department of Computer Science, Carnegie-Mellon University.

Ballard, D.H. & Sklansky, J.(1974): "Hierarchic Recognition of Tumors in Chest Radiographs", Proc. IJCPR-II, 258-263.

Barrow, H.G. & Popplestone, R.J.(1971): "Relational Description in Picture Processing", Machine Intelligence 6 , Meltzer, B. and Michie, D. (eds.), 377-396, Edinburgh University Press.

Barrow, H.G. & Tenenbaum, J.M.(1976): "MSYS: A System for Reasoning about Scenes", SRI Tech. Note 121.

Baumgart, B.G.(1974): "Geometric Modeling for Computer Vision", Stanford AI Memo, AIM-249.

Bobrow, D.G. & Winograd, T.(1976): "An Overview of KRL, a Knowledge Representation Language", Xerox Palo Alto Research Center.

Braid, I.C.(1974): "Designing with Volumes", Cantab Press, Cambridge, U.K..

Brice, C.R. & Fennema, C. L.(1970): "Scene Analysis Using Regions", Artificial Intelligence, 1, 3, 205-226.

Clowes, M.B.(1971): "On Seeing Things", Artificial Intelligence, 2, 1, 79-116.

Erman, L.D. & Lesser V.R.(1975):"A Multi-level Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge", Proc. IJCAI-IIII, 483-490.

Falk, G.(1971): "Scene Analysis Based on Imperfect Edge Data", Proc. IJCAI-II.

Guzman, A.(1968): "Computer Recognition of Three Dimensional Objects in a Visual Scene", Ph.D. Thesis, MIT.

Harlow, C.A.(1973): "Image Analysis and Graphs", Computer Graphics and Image Processing, 2, 60-82.

Horn, B.K.P.(1977): "Understanding Image Intensity", Artificial Intelligence, 8, 2, 201-231.

Huffman, D.A.(1971): "Impossible Objects as Nonsense Sentences", Machine Intelligence 6, 295-323, Meltzer, B & Michie, D. (eds.), Edinburgh University Press.

Kanade, T.(1974): "Picture Processing System by Computer Complex and Recognition of Human Faces", Ph.D. Thesis, Department of Information Science, Kyoto University.

Marr, D.(1975): "Early Processing of Visual Information", MIT AI Lab., AI Memo 340.

Marr, D.(1976): "Analysis of Occluding Contour", MIT AI Lab., AI Memo 372.

Marr, D. & Nishihara, H.K.(1976): "Representation and Recognition of the Spatial Organization of Three Dimensional Shapes", MIT AI Lab., AI Memo 377.

Mackworth, A.K.(1973): "Interpreting Pictures of Polyhedral Scenes", Artificial Intelligence, 4, 2, 121-137.

Michie, D.(1968): "Memo Functions and Machine Learning", Nature, 218, 19-22.

Minsky, M.(1975): "A Framework for Representing Knowledge", The Psychology of Computer Vision, Winston, P.H.(ed), 211-277, McGraw Hill.

Nagao, M.(1972): "Picture Recognition and Data Structure", Graphic Languages, Nake, F. & Rosenfeld, A.(eds.), 48-69, North Holland.

Nagin, A., Hanson, A.R. & Riseman, E.M.(1977): Region Extraction and Description Through Planning", COIS Tech. Report 77-8, University of Massachusetts at Amherst.

Nevatia, R. & Binford, T.O.(1973): "Structured Description of Curved Objects", Proc. IJCAI-III, 641-647.

Ohlander, R.(1975): "Analysis of Natural Scenes", Ph.D.Thesis, Computer Science Department, Carnegie-Mellon University, Pittsburgh.

Popplestone, R.J., Brown, C.M., Ambler, A.P. & Crawford G.F.(1975): "Forming Models of Plane-and-Cylinder Faced Bodies", Proc. IJCAI-IIII, 664-668.

Preparata, F.P. & Ray, S.R.(1972): "An Approach to Artificial Non Symbolic Cognition", Information Science, 4, 65-86.

Roberts, L.G.(1965): "Machine Perception of Three-Dimensional Solids", Optical and Electro-Optical Information Processing, Tippett, J.T. et. al(eds.), 159-197, MIT Press.

Sakai, T., Kanade, T. & Ohta, Y.(1976): "Model Based Interpretation of Outdoor Scene", Proc. IJCPR-III, 581-585.

Sakai, T., Nagao, M. & Kanade, T.(1972): "Computer Analysis and Classification of Photographs of Human Faces", Proc. First USA-JAPAN Computer Conf., 55-62.

Shirai, Y.(1973): "A Context Sensitive Line Finder for Recognition of Polyhedra", Artificial Intelligence, 4, 2, 95-119.

Shirai, Y.(1975): "Edge Finding, Segmentation and Recognition of Complex Objects", Proc. IJCAI-IIII, 674-681.

Tenenbaum, J.M. & Barrow, H.G.(1976): "Experiments in Interpretation-Guided Segmentation", SRI, Tech. Note 123.

Tomita, F., Yachida, M. & Tsuji, S.(1973): "Detection of Homogeneous Regions by Structural Analysis", Proc. IJCAI-III, 564-571.

Turner, K.J.(1974): "Computer Perception of Curved Objects Using a Television Camera", Ph. D. Thesis, School of Artificial Intelligence, Edinburgh University.

Waltz, D.(1972): "Generating Semantic Descriptions from Drawings of Scenes with Shadows", Ph. D. Thesis, MIT.

Winston, P.H.(1973): "The MIT Robot", Machine Intelligence 7, Meltzer, B. & Michie, D.(eds.), 431-463, Edinburgh University Press, Edinburgh.

Yakimovsky, Y. & Feldman, J.A.(1973); "A Semantic Based Decision Theory Region Analyzer", Proc. IJCAI-III,580-588.

Yakimovsky, Y. & Cunningham, R.(1976): "DABI-A Data Base for Image Analysis with Nondeterministic Inference Capability", Tech. Memorandum 33-773, Jet Propulsion Laboratory, Pasadena.

Yoda, H., Motoike, J. & Ejiri, M.(1975): "Direction Coding Method and Its Application to Scene Analysis", Proc. IJCAI-IIII, 620-627.