78-03



# VISIONS: A Computer System for Interpreting Scenes[1]

Allen R. Hanson
School of Language and Communication
Hampshire College
Amherst, Massachusetts 01002

Edward M. Riseman
Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

## Abstract

The design of a general system for interpreting static, monocular, color images of natural scenes is described. Interpretation of an image involves the construction of an internal model which is a description of the major semantic elements in the scene, as well as their three-dimensional relationships in the physical world. This paper examines the structure of the interpretation system including the representation of knowledge, the processes which form the model, control of these processes, and search through the space of possible models. All components have been designed modularly to provide a tool which will facilitate the local exploration and evolution of a very complex system. Initial results demonstrate selected aspects of the combined segmentation and interpretation processes.

## Table of Contents



## I. Introduction

This paper discusses the design and initial performance of the semantic processes of a computer vision system, called VISIONS (Visual Integration by Semantic Interpretation Of Natural Scenes), for interpreting static monocular scenes [RIS74,HAN75, HAN76a,b,WIL77].[2] This work represents an empirical approach to the design and implementation of an extremely complex system. We have decomposed the system into "low-level" segmentation processes which operate on numeric arrays of visual data, and "high-level" interpretation processes for constructing a description of the world portrayed in the scene. The global structure of the VISIONS system is outlined in Figure 1. The companion paper in this volume [HAN78] describes the segmentation system of VISIONS. In the present paper, we assume that the low-level segmentation and feature extraction processes have provided adequate information and describe how this can be used to interpret the image. Initial results demonstrate

[1]This research has been supported by the National Science Foundation under Grant DCR75-16098.

[2]Many of the ideas in this paper are under continuing development by members of our research group. We will reference those papers that already document these efforts, and mention in footnotes the individuals responsible where documentation is not yet available.


ISBN 0-12-323550-2

selected aspects of the combined segmentation and semantic processes.

For alternative views on complex interpretation processes, in this volume refer to [BAJ78, BAL78, BAR78, BRA78, LEV78, MAC78, MAR78, RED78, SHI78, UHR78, WOO78]; as a sample of previous efforts towards general vision systems refer to [BAR72, FIS73, YAK73, LIE74, TUR74, MIN75, BAR76, TEN76, KAN77, RUB77] and for related work in speech recognition systems refer to [ERM75, LOW76, LES77, WAL77, WOO77].

## I.1 A Strategy for the Evolution of a Very Complex System

The goal of VISIONS is the transformation of patterns of sensory visual input from a two-dimensional (2D) image of a scene into a description which captures the meaning of the scene. Interpretation of an image, then, involves the construction of a model which includes a description of the major conceptual entities present and a volume-surface occupancy description of the three-dimensional (3D) space of the world in the scene. The process by which this description is constructed is called model-building.

Our approach to model-building consists of four major components, as shown on the right-hand side of Figure 1:

1) representation - multiple levels of representation for both the image-specific model and long-term general knowledge,

2) processes - a set of modular knowledge sources for the transformation of data (patterns) between particular levels of representation,

3) control - a hierarchical modular strategy to control the application of the knowledge sources, and

4) search - a tree representing the history of search through the space of possible models.

One of our design goals is to provide maximum flexibility during the evolution of a system which may need to be modified along lines which cannot be anticipated in advance -- for example, representations may have to be added or modified, processes may not prove to be sufficiently reliable, etc. It is naive to believe that all of the complexities in controlling such a wide range of processes can be predicted without empirical investigations. Therefore, the initial design of VISIONS has been modular -- geared towards the development of a tool which allows local investigations of a range of issues and which permits modifications to the system without major disruptions in its functioning as a system. This work was primarily influenced by some of the system design criteria of the HEARSAY-II speech understanding system [ERM75, LES77], and we have further extended the general design philosophy to other aspects of system development, in particular the strategies for controlling the processes.

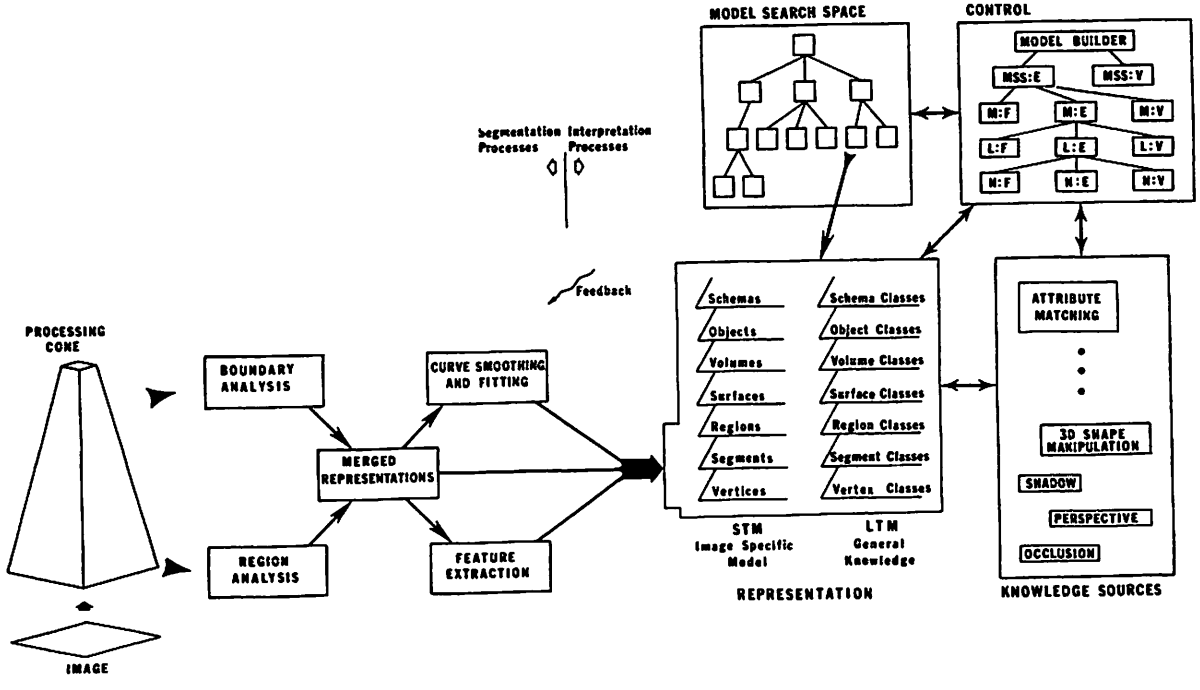



Figure 1. Overview of the VISIONS system. The left hand side represents the segmentation processes which are described in the companion paper in this volume. The right hand side represents the mechanisms for the formation of a descriptive model of a scene. The interpretation components are structurally divided into a representation of knowledge, modular knowledge sources (processes) for the formation of hypotheses, a hierarchical modular strategy to control the application of the knowledge sources, and a tree for representing the history of search through the space of models.

## I.2 An Overview

Figure 2 portrays the layered graph which is the underlying representation of stored general knowledge about the world; in this representation, an interpretation of a specific image becomes an inter-linked collection of instantiations of the stored concepts. General a priori knowledge can be viewed as long-term memory (LTM). The set of instantiations, viewed as short-term memory (STM), constitutes the system's internal model, or description, of the world in that particular image.

Declarative knowledge about the world has many forms. The system must have access to knowledge of objects, including attributes of color, texture, size, shape, etc., including the functional and spatial relationships between (parts of) objects, as well as information about the way simple 3D volumes and surfaces project as regions, boundaries, and vertices in a 2D image [MIN75]. The SCHEMA level of the knowledge base will be used to describe the prototypical structure of common scenes (road scene, house scene) and objects (house, person, car) in terms of their parts; the importance of each part to the schema and the spatial relationships between the parts will be stored with each schema. This will provide a hierarchical description of the world down to finer levels of detail, but instantiation

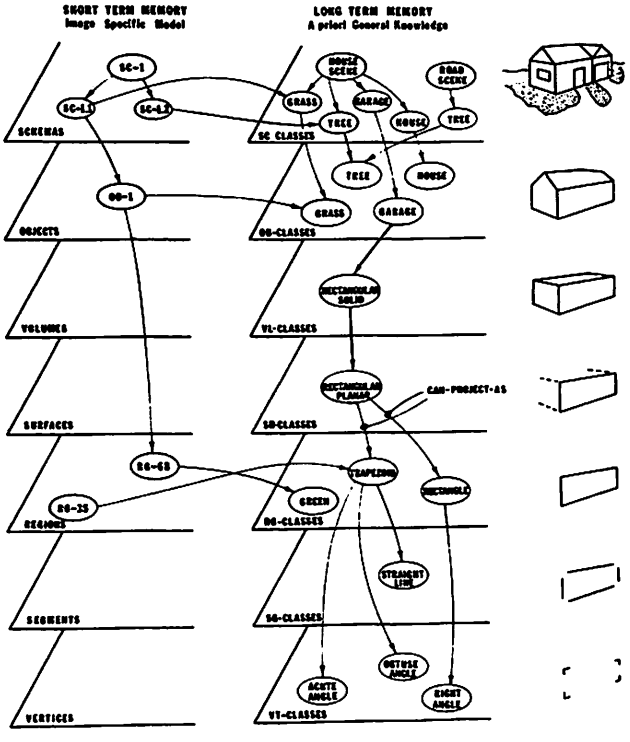



Figure 2. An example sketch of a partial model and the decomposition of declarative knowledge. The knowledge is divided into: 1) a hierarchy of levels of representation defining the key levels of abstraction which are necessary for a general system of visual perception; and 2) short-term memory (STM) representing an interpretation of the specific image and long-term memory (LTM) representing general visual knowledge of the world.

from the image can take place at any level of detail (e.g., instantiation of the house as a whole or instantiation of the parts of house).

For other kinds of visual knowledge, a procedural form is more natural; for example, the laws of perspective which govern the mapping between the 3D world and its projection onto a 2D image. Thus, we have implemented a procedure to compute the expected image size of an object of known dimensions at a given distance viewed through a lens with a given focal length. Each such process is called a "knowledge source" [ERM75] and will be used to form instantiations, also referred to as hypotheses, in the process of building a model of the scene in the image.

There are a number of knowledge sources (KSs) potentially relevant to vision. Many of these are sketched in Figure 3 across the levels of representation at which they operate. Although several KSs have already been implemented or are nearing completion, some of them are not as sophisticated as one might desire. Here our strategy is to incrementally improve a particular KS after a simpler working process has been made available to the system.

The construction of an interpretation is guided by a strategy which we refer to as a control strategy. When the knowledge that is being manipulated is procedural, it is necessary to determine which processes are to be activated, in what order, and how new information is to be integrated with existing information. When the knowledge is declarative, as in our long-term memory, an active mechanism is needed to employ this information. For example, the predictive power of a schema rests in the stored relationships of the schema to its parts. The way in which it is used still remains to be specified since the information itself does not lead to the generation of hypotheses; in this case a useful strategy might involve the examination of the most likely unexplored schema part suggested by an instantiated schema.

The interpretation of an image is achieved by constructing a description in short-term memory and it requires the generation of many hypotheses. As we have just shown, the paths available for hypothesis formation (Figure 3) involve both procedural and declarative knowledge. These paths are often divided [ERM75] into those which are bottom-up (or generative) and those which are top-down (or predictive):

(a) bottom-up: hypotheses are formulated on the basis of characteristics and features stored in STM which were derived from the specific image being analyzed; examples include the analysis of spectral properties (color and texture) of regions for hypothesizing objects; and the fitted shapes of boundaries and regions for hypothesizing surfaces, volumes, and objects;
(b) top-down: hypotheses are formulated by analysis of predictions from stored knowledge in LTM; examples include object prediction from instantiated schemas, or the manipulation of stored 3D shape representations for matching regions.

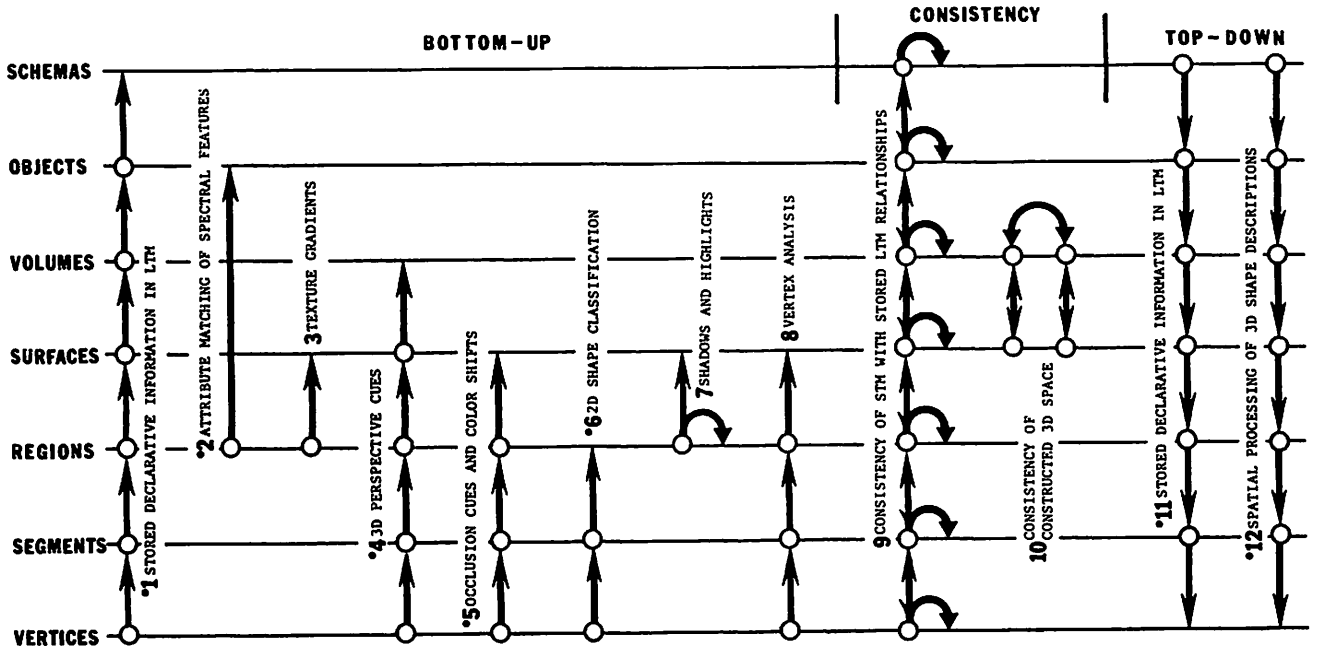



Figure 3. Bottom-Up and Top-Down Paths for Hypotheses. Knowledge sources, using both declarative and procedural forms of knowledge, can generate and verify hypotheses along many paths through the multi-level representation. These are some of the prime cues in static, monocular, 2D color images of 3D scenes. Additional paths are available from motion and stereo data. Those mechanisms for hypothesis formation which are in a more advanced state of development in the system are marked with a • in the diagram. (1) Hypothesis generation via symbolic labels and relationships stored in LTM, including primitive types at each level, subclass/superclass and part/whole relationships, shape, spatial relationships, etc. (Figures 2 and 10). (2) Object identity hypotheses via attribute matching of spectral features of regions in STM with stored objects in LTM (Figure 4). (3) Surface hypotheses via texture gradients. (4) Volume and surface hypotheses via 3D perspective cues and projective geometry; this analysis includes vanishing points and lines, type of polyhedral vertices, comparison of region size in STM with stored object size in LTM, etc. (Figure 6). (5) Hypotheses of relative distances via occlusion cues (Figure 6) and color shifts. (6) Hypotheses from analysis and classification of 2D shapes (Figures 27-29 in companion paper). (7) Shadows and highlights provide cues to surface orientation and can direct merging of regions for truer projection of surfaces (Figure 27 in companion paper). (8) Analysis of polyhedral vertex types for surface and volume hypotheses; extensions to smoothly varying surfaces (Figure 6). (9) Verification of the consistency between STM and stored relationships in LTM. (10) Verification of the consistency of volume/surface occupancy in constructed 3D space. (11) Predictions via symbolic labels and relationships stored in LTM (inversion on arcs in Figures 2 and 10). (12) Predictions via rotation and projection of 3D shape descriptions stored in schemas (Figure 5).

Some type of control is required for manipulating both procedural and declarative information. In order to generate hypotheses, knowledge sources must be activated, and/or information in LTM must be examined. The hierarchical modular control structure (Figure 8) is intended to provide the flexibility necessary to explore a range of strategies for the construction of an interpretation. This structure is decomposed horizontally according to the classic hypothesize-test paradigm (here, focus-expand-verify). The vertical decomposition captures the sequence of events which must occur in order to expand a partially constructed interpretation: selection of one of a collection of competing interpretations, selection of a level in the multi-level structure, and finally selection of a node (or set of nodes) upon which to focus.

Let us illustrate one of very many strategies for applying these KSs in the construction of the model of a rural road scene. For sake of discussion, assume the road has grass and trees to either side, a dotted yellow line down the center, and that there is a red car coming towards the camera. Initially, the regions which are largest, brightest, and most highly color saturated can be examined first. Stored attributes of object classes (which have been extracted from a training set in our data base of images) are compared to the features of each region. The sky region should be easily recognized, but there may be some uncertainty between the grass and tree regions. If a dark (possibly brown) region with parallel vertical sides is found below a green textured region with an irregular boundary below the sky region, this will imply that a tree is present. Note that here the tree schema would be used to recognize the parts of tree.

Projective geometry can be used to determine that the width of the trunk region and the height to the top of the crown region imply a physical object in the world whose dimensions lie within the stored size range of the parts of tree. In order to perform this analysis, hypotheses would be made concerning the levelness of the ground, the vertical orientation of the tree to the ground, and the lowest point in the image where the trunk region touches the ground. Occlusion analysis would be invoked to determine whether the bottom of the trunk appears to be obscured. The hypothesis of grass would be increased when the trunk is instantiated. The converging sides of the road region, the gray color of that region, and the presence of small elongated yellow regions contained in the road region are cues for road; perspective analysis would verify consistency between the sizes of image region and the stored size range of road in a fashion similar to the tree; here the road is assumed (via stored information) to be in the horizontal ground plane. If the road-scene schema is instantiated, the identity and locations of the regions already examined fit the schema. Now the road-scene schema predicts the possibility of car, and the saturated red region (man-made objects are often highly saturated) matches the shape and expected location of car; of course the car might be recognizable without the road-scene schema. The analysis can proceed in this fashion using the stored knowledge and any of the KSs at various places in the image. There are usually many redundant visual cues present and obviously many control strategies for employing these cues.

### I.3 Goal Orientation

The successful construction of the model description is the goal of our current research in this first phase of system development. In most practical applications, however, the goals of a vision system would guide this model-building process. Manipulation of the 3D volume-surface levels in the model would allow a mobile robot to form plans for movement in the environment; here, the goal-orientation of the robot (e.g., to move to a particular place) would require the system to selectively construct a detailed model only in the relevant areas of the environment, that is, in areas which have semantic importance in relation to the current goal(s). In some cases the description being formed will by its very nature attain the desired goals; e.g., if we consider the automatic scanning of tissue samples for malignancies, the naming of the regions at the object level (or the formation of the attributes of these regions/objects) could actually be the results desired by the pathologist.

The current design has not been directed towards any particular goal, but rather is an attempt to design a general system in which a variety of goals can be incorporated. Therefore, in this paper we will usually refer to an interpretation without reference to the particular goal at hand. We expect that the presence of more constrained goals would allow greater efficiency in model construction.

### I.4 Schemas: A Bridge Between General-Purpose and Special-Purpose Vision Systems

Schemas[1] are the highest level knowledge structures currently in our system. They describe objects and scenes in terms of a related set of parts; spatial relationships between parts and the importance of a part to the schema as control for model building are included (refer to Section VI). The house scene schema would encode expected information about the relationships of house, lawn, driveway, garage, etc. The house schema would describe the relationships of windows, doors, walls, roof, etc. These structures are related to Minsky's frames [MIN75], Schank's scripts [SCH77], Piaget's schemas [PIA71], Neisser's schemas [NEI76], and Arbib's slides [ARB72] and schemas [ARB77]. Note that interpretation should not require the use of scene schemas -- there cannot be a schema for every situation encountered, and objects ought to be recognized in unexpected contexts.

Let us consider the relationship between a schema and a system tuned to a particular application. A special purpose system (e.g., a system to analyze a chest x-ray image [WEC75,BAL78]) incorporates many constraints from the domain being viewed. It can take advantage of a fixed context, use features that are particularly useful in achieving the specific goal, and choose a strategy which may be very bad in general but which is effective in the given application. To some degree the use of schemas reduces the complexity of general vision systems towards that of special purpose systems.

Consider the problem of intermediate difficulty -- a general system which is being used to recognize known objects and scenes [BUL78,NEV78]. Once the correct schema is selected, the system could use recognition strategies which vary the visual features used in segmentation, the order in which parts are searched for, the location in which matching processes are attempted, the type of matching, the confidence which is acceptable

---

[1] In our earlier work the term "frame" [MIN75] was used for the representation of these stereotyped situations. We have chosen to change our terminology in order to avoid confusion with the common usage in TV and motion pictures. There is also confusion with the "frame problem" in AI literature [McC69] which concerns the determination of what remains constant as the environment changes. The term "schema" has a history of usage with a general sense that fits our purpose. From this point on we will use "schema" in place of frame, while acknowledging the general influence of frame theory. With respect to the choice of "schemata" or "schemas" as the correct English plural of "schema," we quote Arbib [ARB77]

> "Most English dictionaries offer the Greek plural schemata for the word schema, but we prefer the English plural schemas. Fowler's 'Modern English Usage' states: 'Of most words in fairly common use that have a Latin as well as or instead of an English plural the correct Latin form is given in the word's alphabetical place. ... There is a tendency to abandon the Latin plurals, and when one is really in doubt which to use the English form should be given the preference.' (The cited comments refer to Latinized-Greek as well.) This tendency to abandon classical plurals is part of the pattern of historical change of English. For example, the 19th century had already seen dogmas achieve parity with dogmata, and in current usage only the English form appears unaffected. Amongst authors who share our preference for schemas, we may cite P.H. Lindsay and D.A. Norman: 'Human Information Processing', 2nd Edition, Academic Press (1977)."

for a match, etc. The front view of a specific known house should be no more difficult (and probably far easier) than an unknown chest x-ray image. Now the problem is that for a general vision system to be functional, it will often not have schemas for particular objects/scenes, but a more general description of a class of objects/scenes. In addition, at many stages of the processing one cannot be sure whether the correct schema is being used or which one to use next.

Our approach is to separate the problem of recognizing a house when a house is assumed to be present and actually is present, from the problem of first hypothesizing that a house is present or recovering from the erroneous hypothesis that it is a different object. Consequently the development must proceed along two lines, one dealing with the specific, and the other with general levels of processing. The development of an individual schema and the verification that it is applicable may be as tractable as the development of a particular strategy in a special purpose system. The general strategy by which schemas are instantiated, verified, and rejected is a separate research question whose answer will bring us a long way towards the development of more general vision systems. Our argument is that there are many steps by which the advantages of a special purpose system can be extracted so that there is a continuum in the development of general vision systems from special purpose systems [NEV78]. We believe this formulation of the problem will allow the incremental development of a general system, which can take advantage of constrained contexts. This is a major characteristic of the design in [BAL78].

## II. Representation of Declarative Knowledge and Models

### II.1 Multiple Levels of Representation

From our point of view the process of image understanding is a series of transformations from numeric entities representing the sensory information to a symbolic structure capturing those concepts relevant to the goals of the system. The input is a very large array of integer values representing brightness and color at local points of the image, while the information that we expect our system to derive involves the names of classes of objects and the important relationships between these classes. In order to manipulate data across such a range of representations, we have found it necessary to form a hierarchy of representations (or abstractions) [ERM73] which capture many of the natural characteristics of our visual and physical world. These levels of abstraction divide into two types of representation: a) 3D concepts concerned with the physical world; and b) 2D concepts associated with projections of the physical world onto an image plane.

Figure 2 depicts the levels that currently are being employed in our system. The primary levels of representation include schemas of stereotypical situations, the objects taking part in those situations, the volumes and surfaces which delimit those objects, and the regions, boundary segments and vertices defining the visible portions of those surfaces from particular points of view [WIL77]. The lower region, segment, and vertex (RSV) levels provide the means for describing information in a photograph; RSV is the data structure which receives segmentation output in symbolic form and is a major interface between the low-level and high-level systems (see companion paper [HAN78], this volume). It also is the representation for describing the way physical objects appear from particular views. The surface, volume, object, and schema levels permit description of the physical world. The 3D world and the 2D projections interface at the surface and region levels [MAR76b], where the laws of projective geometry govern the relationships between them.

We believe that if any of these levels are removed, the system will be deficient in its ability to interpret images in a general manner. For example, removal of surface descriptions will leave the system deficient in its understanding of three-dimensional space, while removal of schemas will leave the system ignorant of situations such as road scenes in which there are many constraints on the relationships between members of a particular set of objects. Neither of these deficiencies can be easily overcome in a vision system which is to exhibit some of the characteristics of general visual perception.

Of course all of these levels of representation are not necessary if the environment or goals are restricted to some special-purpose application. Analysis of blood cells [BRE74] would not require a general representation of surfaces, while an assembly line system designed to inspect a single industrial part [BAI75] would not need a general schema level since the system has a fixed context in which it operates.

### II.2 A Visual Model in Terms of Stored General Classes: STM and LTM

The hierarchical set of representations provides rich descriptive capabilities, but we must also distinguish between the representation of a specific instance of a visual entity and the general class to which it belongs. For example a particular object labelled TREE appearing in the image is a member of the general TREE-CLASS at the object level, and it will inherit the basic properties of TREE from its class association. However, the instance of TREE in the image will have specific values for these properties and will in general differ from its prototype. These values should be stored with each particular instance of tree, since there may be several in the image.

This suggests that, in addition to dividing visual knowledge into hierarchically related levels of abstraction, each level should also be subdivided into short-term memory (STM) and long-term memory (LTM); these are also depicted in Figure 2. Consequently, a model of a specific image can be viewed as a set of instantiations of stored general concepts which are relevant to understanding the image. The information that is specific to a particular image will be stored as a set of instantiations in STM, while the general knowledge about the world (which the system has been given prior to interpreting a given image) will be stored as classes in LTM.

It is important to distinguish between what is known about cars -- the fact that they have four wheels, for example -- and the appearance of a particular car in which only two of these wheels are visible in the image. Each of the two regions would be instantiated as a member of the class of wheels and general knowledge from the car class in LTM would indicate that two more wheels are probably present. There is further information in the image (occlusion cues) and in LTM (the 3D spatial relations between the parts of car) which can be used to verify that the absence of two wheels in the image is consistent with our stored knowledge of cars.

One point of potential confusion in the discussion to follow is the distinction between a physical object (surface, volume,...) in the world and the representation of that object (surface, volume,...) in short- and long-term memory. In most cases, the context of the discussion should make the intended meaning clear. Wherever necessary, we will distinguish between the physical object (or scene, volume, surface), the representation of that particular object, and the representation of that object class.

Finally, it should be understood that both the range of knowledge in LTM and the low-level system's ability to extract from an image the attributes stored in LTM will delimit the range of the interpretation process. If color attributes do not appear in LTM or if they cannot be extracted from the sensory data, then color cannot be used in the interpretation process. If LTM does not contain the decomposition of car into its parts, then the interpretation process will only be able to use the attributes of car as a whole.

## II.3 The Implementation of Declarative Knowledge

We represent declarative knowledge as a directed graph in the classic style of semantic networks [QUI68,SIM73,FAH75,HEN75,RUM75,WOO75]. Nodes represent primitive entities (objects, concepts, situations, etc.) and labelled arcs represent relationships between them. Although each arc is directed, this does not imply that it can only be traversed in a single direction. The directionality of edges is for increased semantic power and is not meant to imply a strict one-way access in the pointer structure. This directed graph is partitioned into the different levels of abstraction, so that it is actually a collection of directed graphs, each residing on a distinct labelled plane, with labelled directed arcs between planes. Each level of abstraction is further subdivided into two planes, one for image-specific entities in STM and the other for general class knowledge in LTM. This structure has been implemented in GRASPER [LOW78], a graph processing language (implemented in LISP), and is an extension of the work of [FRI69,PRA71].

Arcs between levels relate concepts at a given level to the concepts on planes in neighboring higher and lower levels; they are organized in terms of AND/OR relationships. Thus, arcs leading up from an object node can be followed to nodes at the schema level, representing those schemas in which the object participates, and arcs leading down can be followed to volume and surface nodes indicating the spatial occupancy or 3D shape of the object; the same is true for nodes on the object class plane where prototypical information about a class of objects is stored.

Arcs between the STM and LTM planes at the same level are instantiations of general classes for image-specific elements. For example, an instantiation of a green car could be represented by an image-specific node, say OB-38, with arcs to the car class node and to the green class node in LTM. In addition, each node at any level can have a list of properties and values associated with it. These will be very useful for matching image features to class descriptions, as in the case of matching region attributes with object class attributes. Further discussion of this representation appears in [WIL77].[1]

Finally arcs and nodes in long-term memory carry information about the coarse likelihoods of concepts and relationships [YAK73,DUD76]. Each node will have an a priori estimate of its likelihood and each arc will have a conditional probability attached to it.[2] This will provide useful control information for ordering alternative hypotheses, and allows belief in hypotheses to be represented. For example a cylinder CAN-PROJECT-AS a rectangle with a very low probability because there is a very restricted set of views from which this can take place; a rectangular solid will have rectangular planar surfaces which CAN-PROJECT-AS regions whose 2D shape is rectangular with some probability, or trapezoidal with higher probability. It is obvious that these estimates will be quite heuristic, but they do provide relative weights on paths upward and downward where we feel there is justification to prefer some paths.

## II.4 An Example of a Partial Model

Figure 2 is an example of a partially developed model. Region RG-68 has been recognized as having the color GREEN, while region RG-35 has been recognized as an instance of the shape TRAPEZOID. On the STM side RG-68 has been identified as an object OB-1 which is an instance of the object class GRASS, and plays the role of a schema part in schema SC-1 which is an instance of the class of schemas called HOUSE-SCENE. The remainder of this example should be obvious.

Note that Figure 2 is just a rough sketch of the network. The actual representation is developed more carefully, since there would be an exclusive-OR on the CAN-PROJECT-AS relationship in this example. Also there are various part-whole and subclass-superclass relationships on the arcs [WIL77]; for example a rectangular planar surface is just one part bounding the rectangular solid and 5 other surfaces must be defined. And finally the system must know that GREEN is a type of COLOR;

[1] The development of the knowledge structure is part of the ongoing work of John Lowrance of the COINS Department, University of Massachusetts

[2] Note that if necessary a procedure for analyzing a given situation could be associated with an arc or node if a number is not sufficient to represent the likelihoods of the alternatives.

this can be represented either with an intra-planar LTM arc between COLOR and GREEN, or else it could be attached implicitly as (COLOR GREEN) on the property-value list associated with GRASS. However, there are still serious difficulties in recognizing a distribution of the input data as the color "green", which is a rather fuzzily defined term. We bypass the symbolic naming of color and texture by storing distributions of spectral features for objects and comparing them with the distributions of regions in the image. There is not room to discuss further the details of all these problems. However, the part-whole relationships at the schema and object levels will be expanded on in Section VI.

This highly structured form of declarative knowledge forces the incomplete development of a specific model to be readily apparent. A node which does not have expected arcs to levels above and below may be a portion of the model which is not fully developed. Examples of incomplete portions of a model include a region not explained as a surface (i.e., a region node of STM without at least one arc from a surface node in STM), a surface not explained as (part of) an object, and a surface not explained as a region. Of course, most human perception seems to be similarly incomplete in the sense that not all areas of an image can be, or need be, interpreted; only those that are necessary to achieve the current goals of the perceptual system need be interpreted.

## III. Knowledge as Processes

Model-building is viewed as the process by which image specific entities in STM, at all levels of abstraction, are constructed and related to general entities on the LTM side of the hierarchy. At any instant during model-building, the current set of hypotheses about the image is represented in STM as a collection of nodes and arcs dispersed through the different levels of abstraction. This set of hypotheses is a partial model and includes not only inter-planar and intra-planar arcs in STM, but intra-level arcs from STM to LTM as well. Partial models are expanded incrementally by adding hypotheses (nodes and arcs) to the current set. These hypotheses result in the creation of arcs between two or more levels of STM, and/or between the STM and LTM planes at the same level. This section discusses some of the processes which form these hypotheses, while the next section discusses ways in which they can be applied as part of a comprehensive model-building process.

### III.1 Independence and Interaction of Modular Knowledge Sources

The levels of abstraction and the decision to construct models incrementally (see next section) provides a structure which leads directly to the decomposition of the active processes. To the degree that processes -- knowledge sources -- can be defined to operate upon information at one level and produce hypotheses at another level, the levels of representation selected provide the input-output relationships required of potential model-building processes.

If no process can be defined to transform data from the representation at one level to the representation at the next level, there are several possible implications. One possibility is that an intermediate level of data description is absent; if it were present, several different processes could cooperate in performing the task, some forming hypotheses at the new level while others operate on that new level to form hypotheses at other levels. A second possibility is that there is not sufficient information for any process to be constructed which operates on a given level alone and that it must be a function of several levels simultaneously; this can be handled in our structure and some KSs will use input from several levels. A third possibility is that the levels of representation are organized along insufficient or incorrect dimensions.

During our initial definition of the model-construction processes in VISIONS, the sufficiency of our levels of pattern description for processing by independent knowledge sources is being tested. In most cases it appears that the patterns of information at one level do bear clear relationships to the patterns at other levels. Often a KS can be defined to operate independently, using its own set of cues and analyses of the available data in order to focus upon, generate, and/or verify hypotheses. A region may represent (a part of) a surface and cues derived from shading or from 2D shape, when they are present, provide important clues about the type and orientation of the surface. The boundary between a pair of regions can provide spatial information about the occlusion of surfaces and can be used to determine which surface is nearer the viewer. The size of a volume can provide hypotheses about the identity of an object. The identities of objects and the relationships between them, such as a car on a horizontal surface with two parallel bars down the middle of the surface, provide hypotheses about the type of scene present, in this case a road scene.

Before more complex approaches to KS development will be attempted, the efficacy of the modular decomposition of processes operating on a small subset of levels will be explored. In order to achieve the desired reduction in complexity, it is quite important that the choice of levels decompose the model-building processes into subprocesses. To the degree that each KS can be implemented more or less independently, they can be developed and evaluated separately. If the assumptions made concerning the structure of the problem space are correct, the approach will help to clarify the component parts of the model-building process. The implementation of a KS will be unrestricted except that it should execute independently of all other KSs, although it certainly will operate upon the output of some KSs and provide the source of input to other KSs.

### III.2 Bottom-Up and Top-Down Formation of Hypotheses

The overview in Section I.2 described bottom-up and top-down processing, but this dichotomy is

not quite as clear as one might expect. Bottom-up processing is data-directed based upon the analysis of the particular image, while top-down processing is considered to be knowledge-directed (or goal-directed) based upon prior knowledge stored in LTM. It is clear that the fitting of shapes to regions, or the examination of region adjacency for occlusion cues, are both bottom-up (see Figure 3). The prediction of a garage on the basis of an instantiation of the house-scene schema is an example of top-down analysis. However, the distinction blurs to some extent when region attributes extracted from the image are compared to stored attributes of object classes in LTM, since both types of information are involved; nevertheless, attribute matching of regions and object classes is usually thought of as a bottom-up process.

The confusion is most evident when the declarative knowledge paths of symbolic labels in LTM are considered. Figure 2 showed examples of relationships between levels via symbolically labelled nodes. Each node at one level has an arc from (to) a node at a higher level if the nodes have a part/whole relationship (e.g., straight line and rectangle, or rectangular solid and house), subclass/superclass relationship (e.g., house and building), or special relationships such as shape relationships (e.g., rectangular surface CAN-PROJECT-AS trapezoid). Once a region node in STM is instantiated, say with a symbolic label to the class of shapes, then a set of surfaces/volumes/objects are accessible via these paths of declarative information. In order to avoid this confusion, paths will be considered to be bottom-up or top-down depending on whether the hypothesis is formed from information stored at lower or higher levels relative to the hypothesis, recognizing that hybrid processes are also possible.

It is also worth emphasizing that some of the knowledge source mechanisms of Figure 3 are based on active processes which perform non-trivial computation while forming hypotheses. However, paths based on symbolic labels are basically declarative in nature and a relatively simple process is required to access alternative hypotheses and order them for consideration. A node instantiated in STM by whatever means is then available to be operated on by any of the defined mechanisms.

Clearly, the range of processing described in Figure 3 can be quite complex. We are not in a position to discuss results obtained by bringing all of these KSs together. They are in various stages of implementation at various levels of sophistication. If they were not being developed independently, the problems would appear insurmountable. However, each KS is being developed separately, tested in the system, and then refined. As it is added into the system, the clearly defined discrete levels of representation will allow the full system to come into being incrementally. The quality of the KSs and the degree to which the KSs are redundant will determine the effectiveness of the system.

Space does not permit a complete discussion of each of the KSs. Therefore, in some of the following sections of the paper, we will provide only a brief description of several, and then sketch a simple example of the type of cooperation between processes that is being incorporated.

### III.3 Matching of Attributes: Object Hypothesis and Verification

The variability in the spectral features of color and texture is too great to provide a high degree of reliability in the labelling of objects. The lighting, distance, perspective viewpoint, distortions in the photographic and digitization processes, and the inherent variability in the physical characteristics of a class of objects affect our ability to form prototypes for each object in some feature space. Given that such problems are unavoidable, an approach that has computational advantages for directing analysis in productive directions is to divide the problem into two stages: object hypothesization and object verification. The goal of forming object hypotheses is to determine plausible alternative object identities for a given region such that the probability that the correct identity is included in the set is high. The goal of object verification is to examine a small set of alternative hypotheses, while weighting more heavily those features which provide the best discrimination between the hypotheses [TEN73,HAN76c]. It should also be possible for the verification process to be invoked with hypotheses formed from other stages of the model-building system.

The comparison of color and texture features requires knowledge of the typical patterns exhibited by objects. In order to form an initial knowledge base that is representative of some of the general classes of objects that the system must recognize, we have formed a data base of approximately 25 outdoor scenes.[1] Approximately 60 features have been extracted for 77 samples of trees, shrubs, sky, grass, roads, etc., across the images; the number of samples must be greatly expanded in the future. These features are identical to the features which are extracted from the segmented regions of an unknown image under analysis. This is a rather massive amount of information which must be analyzed in many different ways. It has turned out to be extremely useful to use the relational data base facility [KON77] implemented in ALISP [KON75], the UMass version of LISP. This is an effective tool for adding, deleting, and restructuring the different features which are labelled with image name and the region from which they were extracted. Information can be indexed by image, region, object class, and/or feature subset to facilitate testing of the utility of the features.

The feature set includes raw RGB (red, green, and blue) data, YIQ (color TV standard) [KEN77], intensity, edge/unit area [ROS71], number of extremum/unit area [CAR77], moments around orientation-dependent adjacency matrices [HAR73], spatial

[1] This data base consists of about 8 outdoor scenes kindly supplied by R. Reddy at Carnegie-Mellon University, while the remainder are new images selected by our group and digitized at the University of Southern California.

features such as centroid and the coordinates of the enclosing rectangle in X and Y dimensions, etc. Many of these features are initially extracted as a histogram of feature values, but typically we are using scalar values of the mean ($\mu$) and the variance ($\sigma^2$) as representative parameters of these distributions. Thus, a scalar feature $f_i$ may be a $\mu_i$ or a $\sigma_i$, but to avoid confusion in what follows it will be referred to as $f_i$. The problem, now, is that not only is there variation, for example, between samples of tree within one image, but even greater variation across different images. These variations are rather unpredictable and we do not believe it is reasonable to utilize a theoretical statistical model of this information which requires any assumptions about the data until a very large image data base is formed.

In light of these problems, a strategy which is flexible with respect to the expected variability has been adopted [WIL78]. For each feature $f_i$ of each object $O_j$, there is a range of possible feature values $X_{ij}$. We have formed a template which summarizes the information in the distribution of the means of the training set of samples of $O_j$. Note that each sample that is used has a $\mu$ which has been computed across the set of pixels from the image sample of an object. The template shown in Figure 4 has the minimum and maximum $\mu$ among the samples for an object, and then a $\mu_\mu$ and a $\sigma_\mu$ across the samples; i.e., the mean of the means and the variance of the means. This template will serve in place of the likelihood $P(f_i|O_j)P(O_j)$ derived from Bayesian decision theory, and will not be as prone to error when the number of samples in the training set for $O_j$ is small. Our motivation is to move to the most global level of feature information while retaining information on the image-to-image variation.

The heuristic process of matching unknown region $R_k$ with known object $O_j$ will use the template of each feature $f_i$ to determine the contribution $C_{ij}$ of that feature to a linear decision function $\sum_i C_{ij}$. The contribution of a feature will be largest (a value of 1) if the mean of $R_k$ is within distance $\sigma_\mu$ of $\mu_\mu$ of $O_j$; otherwise the response is scaled down linearly out to zero at $MIN_\mu$ and $MAX_\mu$; there is a negative contribution of -1 for the feature if it falls outside the min-max range because no sample of $O_j$ has been observed outside that range. It is not difficult to extend the template and matching process to particular pairs of good features by capturing information about the way they covary, but we have not yet needed to do so.

The next important aspect in developing this approach is to weight the contribution of each feature in matching object $O_j$ on the basis of its effectiveness in discriminating $O_j$ from the rest of the object classes. This will give a weighted linear discriminant function $\sum_i W_{ij}C_{ij}$ for each $O_j$. If a particular feature returns a similar value for many object classes, it is not useful for object hypothesis. For some feature if the distribution of the means of the samples for $O_j$ and $O_k$ overlap, then this feature will not be useful in discriminating between $O_j$ and $O_k$; e.g., the color of a region will not be effective in separating trees from

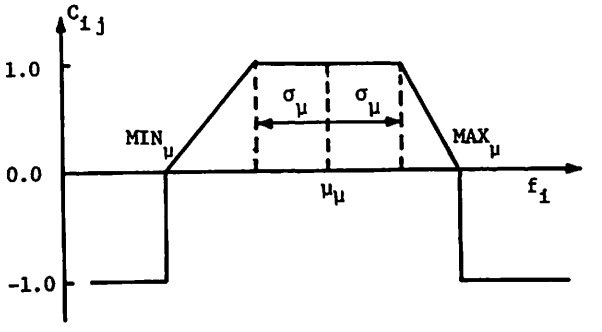



Figure 4. Matching of region attributes with feature templates stored with each object. The value $X_i$ of feature $f_i$ will be used to compute the contribution $C_{ij}(X_i)$ to the discriminant function for the hypothesis of object $O_j$. For each object $O_j$ the information stored about $f_i$ will be extracted from the distribution of the means $\mu$ from the various samples in the training set of $O_j$. This includes the mean of the sample means ($\mu_\mu$), the variance of the sample means ($\sigma^2_\mu$), and the minimum and maximum of the sample means. $C_{ij}$ will be the contribution of feature $f_i$ in matching $\mu$ of the region under consideration. A set of $W_{ij}$ will be computed as a measure of the ability of $f_i$ to discriminate $O_j$ from other objects, forming a linear decision function $\sum_j W_{ij}C_{ij}$ for each $O_j$.

shrubs (although it could be effective for discriminating the foliage superclass, if present, from other classes). This means that the importance of features will change in ways that cannot be predicted as the system moves from hypothesis to verification. The weights for hypotheses among the whole set of objects can be precomputed. However, the proper weight for a feature will change when the set of hypotheses under consideration changes; the system must have the ability to change the weighting of features dynamically, particularly in the object verification stage when a small number of hypotheses is present. Thus, the amount of computation involved must not be exorbitant.

Our first attempt at forming a weight $W_{ij}$ for each $f_i$ of $O_j$ involves the ratio of the number of other object samples ($O_t \neq O_j$) which fall inside to the number that fall outside the range $[\mu_\mu-\sigma_\mu, \mu_\mu+\sigma_\mu]$. Another choice is to intersect the interval at one standard deviation around the means of object classes. Those features of $O_j$ whose range overlaps other classes little will be weighted more for discrimination.

A final point that is noteworthy (and somewhat distressing) concerns the limitations of spectral data. While we have good information for identifying sky, trees, shrubs, grass, roads, sidewalks, telephone poles, and people's skin, we cannot rely on color and texture for shirts, cars, houses, windows, doors, and in general most man-made objects. Some of these objects have distinctive shapes while others really are identifiable only in context. The full impact of these problems and the addition of heuristic knowledge to handle particular cases is a subject currently under examination.

III.4 Representation of 3D Surfaces and Volumes

In order to understand the three-dimensional world we will need to store information about the 3D space filled by objects and the 3D relationships between (parts of) objects in scenes. All such information will be stored in LTM at the schema level for reasons discussed in Section VI. We will briefly describe several representations, and refer the reader to [YOR78a] for more detail.

Most objects in the world are not symmetric in all physical dimensions, and consequently there are definable axes by which they can be oriented. The lack of unique natural axes for objects such as a sphere or cube are unusual in this respect. Therefore, we will choose an axis-based description of objects, which thereby allows the relative orientation of parts to be compactly specified and manipulated [MAR76a]. Given an axis, simple volumes can be described by sweeping a cross-section down an axis to form a "generalized cylinder" [BIN71, AGI72,NEV74,MAR76a]. The axis can be specified to be any curve in 3D space, the planar cross section can be of any shape, and the cross section can be defined to vary down the length of the axis. However, the representation is often restricted to an unvarying (often circular) cross section with a straight line for an axis as shown in Figure 5(a).

One of the problems with this representation is that it does not permit surface descriptions to be accessed with the flexibility available for accessing volume descriptions. Note that surfaces and volumes have a dual relationship analogous to regions and boundaries. The orientation, shape, and spatial relationships of surfaces play a key role in recognizing the projection of volumes and objects as regions. Another problem with this representation is that it is difficult to describe local variations (distortions) of a surface because a description of the variations must be represented as a function across the length of the axis.

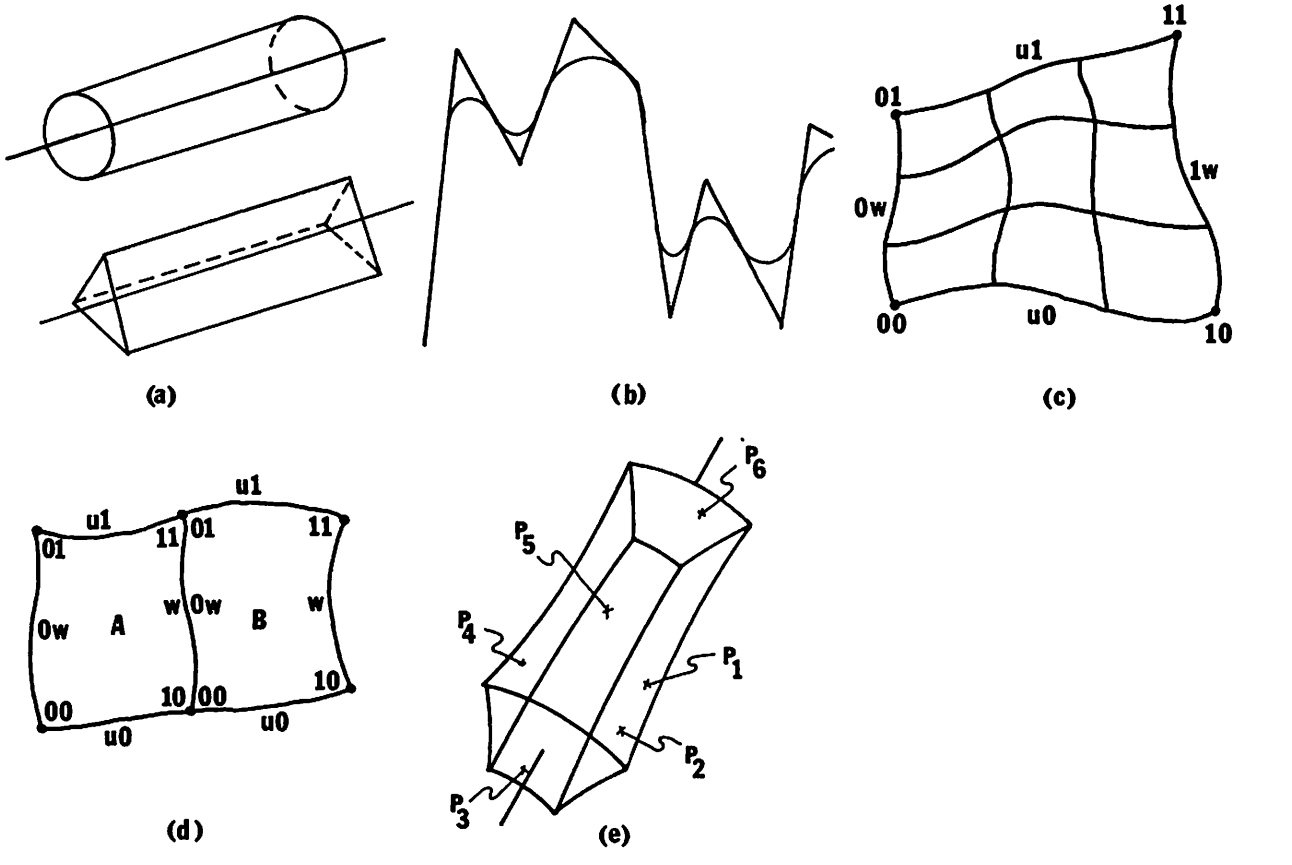



Figure 5. The Three-Dimensional Representation of Shape. (a) The generalized cylinder representation. (b) Cubic B-splines can smoothly fit a polynomial through a set of points in 2-space or 3-space. (c) A Coons' surface patch P(u,w), where u and w are parameterized on the interval [0,1], employs four B-splines P(0,w), P(1,w), P(u,0), P(u,1) to delimit the surface patch boundary; blending functions which are also B-splines interpolate between opposite sides of the surface patch. (d) Two adjacent surface patches A and B can be smoothly joined at a common boundary if the blending functions are constrained properly. (e) Six surface patches can define the shape of a volume around an axis which is used to relate the spatial orientations of such volumes.

Consequently, we are employing techniques from functional approximation and computer-aided design called B-splines [COO74,GOR74] and Coons' surface patches [COO67,74] to form compact descriptions of the surfaces and volumes of irregular objects. Briefly, cubic splines allow a set of points (in 3-space) to be smoothly fitted with a unique curve which is a cubic polynomial on each interval (Figure 5(b)). Proper use of a special type of spline, namely B-splines, allows changes in any interval between a pair of these points to be isolated to that interval and not affect the neighboring intervals. As shown in Figure 5(c), we will define a surface patch P(u,w), where u and w are parameters on the interval [0,1], by using four B-splines P(0,w), P(1,w), P(u,0), P(u,1) to delimit the surface patch boundary. Blending functions (which can also be B-splines) can be used to interpolate between opposite sides of the surface patch. Two adjacent surface patches A and B can then be defined with a common boundary as in Figure 5(d). These surface patches can then be smoothly joined if the first and second derivatives of the blending function at the common boundary are constrained to be equal. Each surface patch can be locally deformed by appropriate choice of the blending functions without distortion of the adjacent patch, and without disturbing the smooth join of the patches.

Let us summarize the surface and volume representations that are now available:

1) an axis with simple pre-defined shapes as cross-sections sweep out volumes;
2) an axis with closed B-splines defining the cross-sections sweep out volumes; and
3) six surface patches spatially related to an axis define volumes as in Figure 5(e); each patch can be described in its own local coordinate system (a point chosen as an origin) and cubic B-splines can be used as the surface boundary and blending functions.

All of these representations are axis-based descriptions where each primitive component is stored in a local coordinate frame of reference. In general these representations require a relatively small number of points and therefore allow compact storage; for example the description in Figure 5(e) would only require about 64 points, if the blending functions are assumed to be stored in a common library which is used by a spatial processor. Thus, rotation of these figures via standard computer graphics techniques is relatively inexpensive.

The third representation allows access to both surface and volume representations. We have not yet had sufficient experience in using the different representations, and therefore will not comment on details for matching of regions to particular views. However, the goals of such processing are to use 2D region shapes as shown in Figure 2 (or some of the other KSs) to access possible object identities for a region. This would be facilitated by storage of standard 2D views corresponding to projections of the 3D representation. The mechanisms for manipulating the 3D description of an object to match 2D projections against 2D regions is the subject of ongoing research.[1]

### III.5 An Example of KS Interaction

In addition to the spectral attribute matching process, we are implementing three major types of processing which lead to hypothesis formation at the region, surface, volume, object, and schema levels. These are hypotheses based on: (i) curve fitting of region boundaries and contours [HAN78 in this volume,YOR78b]; (ii) surface and volume constructions using 2D shape descriptions, results of perspective, shadow, and occlusion analyses, and additional semantic information; and (iii) manipulation of stored 3D descriptions of objects.[1] Several of these processes, or portions of them, are discussed in other sections of this paper or in the companion paper. It is our contention that although much of the interpretive power of a system is embodied in the active model construction processes, there is an important component captured in the redundancy of information obtainable from disparate sources. The highly idealized example which follows illustrates the kinds of hypotheses which are possible and exemplifies typical KS interactions that are expected.

Consider the 2D image of a scene shown in Figure 6. Even with all color, texture, and lighting information removed from the image, one still is able to identify several objects. However, what is of primary importance to the discussion at hand is the rich set of cues that allow a volume-surface plan of three-dimensional space to be constructed without access to the identity of objects. The key point is that three-dimensional space can be constrained prior to object recognition. Of course much additional information will be made available by accessing stored descriptions of objects and by further constraining 3D space via top-down processing and checks for consistency of the resulting hypotheses. However, for the moment we will just examine a few of the hypotheses which can be formed at the surface and volume levels by analyzing data at the RSV levels.

The vertex cues for polyhedral objects have been studied extensively [GUZ68,CLO71,HUF71,DUD73, WAL75,SHA77]. Of the five building vertices where the visible portions of planar surfaces meet in the 2D image, in Figure 6 we have distinguished only $E_1$, $E_2$, and $E_5$. There are four more vertices, where the right tree occludes the building, that provide further cues of occlusion. Boundary dominance seems to be a striking visual cue. At the points $E_3$ and $E_4$, for example, the heuristic associated with boundary dominance at polyhedral T-junctions can be generalized to deal with non-planar surfaces. Consider the segments $S_1$, $S_2$, and $S_3$ which meet at $E_3$. The similarity of the shape properties of segments $S_2$ and $S_3$ provide a hypothesis that $R_1$ occludes $R_2$.

A rather different cue is available from a perspective analysis of the segments bounding regions $R_3$ and $R_4$, which represent two of the surfaces of the building. The nearly parallel sides of the segment pairs $S_4$-$S_5$ and $S_6$-$S_7$ provide an interesting cue for focussing attention on this area. A supporting cue is that there are a pair of vertical parallel boundaries in this area. If the

[1]This research, currently in progress, is being performed by B. York, COINS Department, University of Massachusetts.

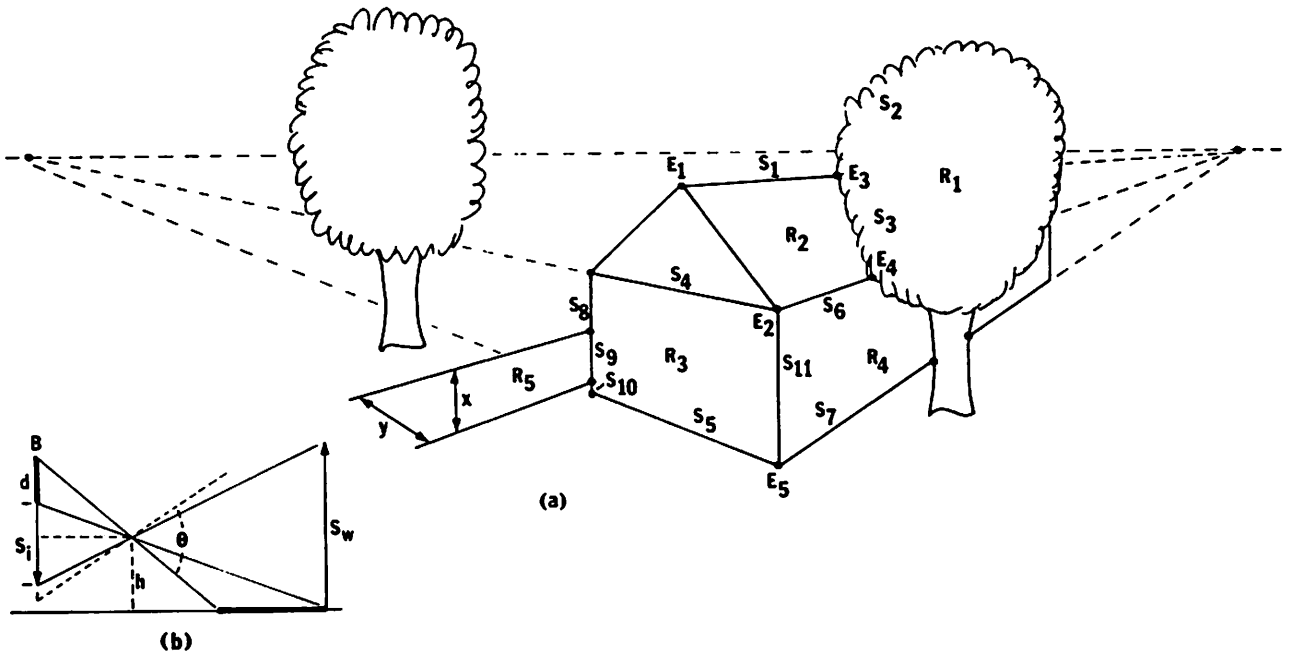



Figure 6. Bottom-Up Cues from Shape, Perspective, and Occlusion Interaction. (a) By occlusion analysis the vertices E3 and E4 provide information about the depth of the surface associated with R1 relative to the surfaces associated with R2 and R4. Shape analysis allows R3 and R4 to be hypothesized as rectangular planar surfaces, which then allow perspective analysis to determine vanishing points and the horizon line. Perspective and shape analysis employ projective geometry to compute the following. With assumptions concerning the height, angle (relative to the ground plane), and focal length of the camera, as well as levelness of the ground plane, then the distance of E5 from the bottom of the image (refer to 6b) allows its physical 3D distance to be computed. This allows the height of S11 in the physical world to be computed from its length in the image, under the assumption that S11 represents a line (surface boundary) perpendicular to the ground plane. The analysis proceeds without knowledge of the object identities of the regions. (b) Segment $S_w$ in the physical world projects as segment $S_i$ in the image. The distance from the camera of a point in the ground plane is determined by the distance d from the bottom of the image B.

system assumes that $R_3$ and $R_4$ are planar surfaces with parallel sides, the two vanishing points can fix the horizon line. Once the horizon is determined, additional cues are sometimes available (although not in this case) to provide hypotheses about whether the ground plane is level.

The near parallel lines to the left side of the image lead to two alternative hypotheses, although several assumptions are necessary to generate them. The first assumption is that the pair of line segments are parallel in the 3D world and that they bound a planar surface. Two further alternative assumptions are that this surface lies either: a) perpendicular to the ground plane (actually parallel to the gravitational vertical), or b) in the ground plane. In the first case region $R_5$ is "wall" and distance X represents the projection of its actual height onto the image; in the latter case $R_5$ is "road" and distance Y is the projection of its actual width onto the image. The objects do not have to be identified in order to determine the placement of this planar surface in the system's surface-volume model of 3D space.

Finally, an interesting interaction between the shape KS and the perspective KS is available at region $R_3$. The 2D shape analysis determines that $R_3$ is trapezoidal. Information is available in LTM that rectangular planar surfaces often project as trapezoids. Thus, it can make available to the perspective KS the assumption that $S_{12}$ (a new segment formed from $S_8$, $S_9$, and $S_{10}$) and $S_{11}$ are parallel, have equal length, and lie in a planar surface. We utilize knowledge of the focal length, height, and orientation of the camera,[1] and the additional assumption that the ground is level; then the orientation of the surface relative to the camera can be computed using relatively simple principles of projective geometry [DUD73]. If the vertex $E_5$ is assumed to be the projection of a point in the ground plane (i.e., $R_3$ represents a surface rooted to the ground plane), then the height of vertex $E_5$ from the bottom of the image allows the distance of the house from the camera to be computed; the trigonometry is sketched in Figure 6(b). If a house is hypothesized, a verification of this analysis is available by checking that the

[1] As default values we assume that the height of the camera is 5'6" and the orientation is horizontal. These can be modified once a known physical size is detected in the image.

height of this surface (via $S_{11}$) is roughly within the range of the height of one story of a house. This semantic information about houses is available in LTM. This is just one example of KS interaction. Thus, occlusion, shape, perspective, and projective geometry can all be used to determine the position of the house in a surface-volume plan of 3D space.

## IV. Search and the Model Search Space

The organization of our system leads quite naturally to the notion of model formation as the sequential incremental expansion of a partial model. Given a partial model and a strategy for expanding it (see Section V), there will be many places where the model could be expanded. At each of these points a subset of the available KSs will undoubtedly be applicable and each KS might produce several competing hypotheses which are promising and worth following. Thus, there is a large implicit search space of partial models which can be generated, only a small subset of which will be consistent with each other and adequately "explain" the image. From this point of view, the process of model construction decomposes into two highly related subproblems -- control strategy and representation of a search space of competing partial models.

Since the size of the model search space is enormous, one cannot expect to explore even a small fraction of possible interpretations. On one hand biological systems seem to plunge effortlessly and quickly down correct paths, probably due to a combination of factors which include the use of constraints available from the many sources of cues which somehow operate in parallel.[1] On the other hand we are quite sure (unfortunately) that alternative hypotheses have to be considered because KSs will be unreliable, and at other times the best local hypothesis does not fit the global set of hypotheses.

In response to this problem, several speech recognition systems [ERM75,LOW76,WOO77] tried to explore as many paths as computational costs permitted using various strategies to explore promising paths, while leaving unpromising paths unexplored. The number of paths that must be examined is certainly a function of the quality and reliability of the KSs that are employed, just as the amount of search necessary in game playing programs is a function of the quality of the heuristic evaluation function [NIL71]. However, we do not have a single evaluation function, but rather a number of KSs, and the manner in which the hypotheses from various sources is brought together is a crucial factor in selecting and constraining the search paths.

Rather than seek to explore many paths, our initial research methodology has been to develop a tool whose use will allow insights into the ways of bringing the diverse sources of knowledge together. While some form of parallel interaction of various constraints ultimately appears to be desirable, we have committed ourselves to the sequential development of partial models. The goal will be the examination and an understanding of a few paths in the search space in an attempt to plunge down a correct interpretation path. Examination of the history of search and the history of applied processes should provide valuable insights into where and why incorrect decisions were made. Consequently, we are interested in developing a general representation for storing alternative search paths in a tree of partial models even though our strategies will seek to move down few paths.

### VI.1 Approaches to Search

Let us discuss some alternative ways in which related partial models can be represented for some type of search process. At one extreme a complete, separate structure could be built for each partial model generated during the search. This leads to gross space inefficiency when one considers that each partial model varies from its parent only by some small amount, and to gross time inefficiency when attempting to determine similarities or differences among alternative models. This approach is totally unacceptable.

At the other extreme only the current partial model is saved, and updates are made to this partial model. Note that this could involve destroying old data, e.g., if later in the search one hypothesis is replaced by an alternative hypothesis. This can be implemented as a single directed graph which contains all alternative partial models; Figure 7(a) portrays a simple example. Essentially this is the "blackboard" approach of the HEARSAY-II speech understanding system [ERM75]. In some respects this is a very useful representation because it allows all competing or consistent submodels to be directly available. If a better alternative to a hypothesis is developed later, it can just be added. This is equivalent to having, at any moment in time, a network containing the relationships between the entities in the tip nodes of a partial model search tree. Its primary disadvantage is that the history of the model development is not available.

An alternative technique for managing the history of search is provided in the "contexts" of the CONNIVER programming language [McD74]. Here, a tree of contexts is formed during the search. This design was in response to the problems encountered using automatic backtracking as the control mechanism in PLANNER [HEW68]. At any moment during processing, a context can be "pushed" so that any further changes in the stored environment could be undone at the user's discretion by a "pop" context command. In addition, other contexts can be examined while the system is bound to a given context. This allows earlier contexts to be examined while deciding what to do in the current context. This approach provides the flexibility necessary to reconsider previous decisions, to determine why they were made, as well as examining the implications resulting from a change to an earlier hypothesis or set of hypotheses. In particular, hypotheses subsequent to the change(s) which do not depend on them may be retained without contradiction; this point is elaborated on in the next section. Figure 7(b) sketches this approach

[1] Note that even biological systems admit ambiguous and sometimes incorrect interpretations in cases where competition between alternative explanations of data cannot be resolved [ESC71,GRE66].

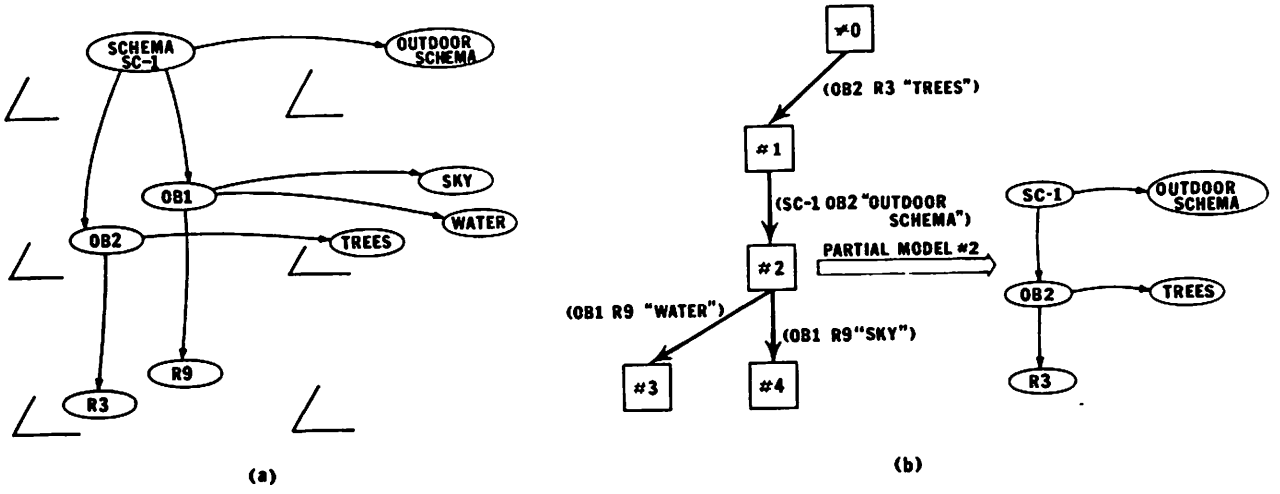



Figure 7. Alternative Representations of Models during Search. For simplicity note that Schema-Parts have not been specified. (a) In the blackboard approach of HEARSAY-II [ERM75], all partial models are stored in a single network, e.g., two partial models involving different hypotheses for OB1 are simultaneously represented. This provides the advantage of easy access to competing partial models, but has the disadvantage of not maintaining the history of the development of each partial model. (b) In the VISIONS approach, contexts [McD74] are used to store each partial model in the sequential model-building process. This tree of partial models has the advantage of maintaining the search space history as both an interactive debugging tool and for control decisions, but it has the disadvantage of making access to competing models more difficult.

to search. Any node in the search tree (other than the root node which contains the initial state -- RSV and all of LTM) explicitly contains only those items which differ between it and its parent. Each node implicitly contains the initial state as well as all changes along the path leading to that node from the root.

These two representations have different advantages. While history is maintained by using context packets, linkages between "brother" paths emanating from some higher level node are not provided without additional modifications. Thus, the system does not have easy access between competing models, nor does it have easy access to compatible submodels in different parts of the search tree. In addition, user control of backtracking has turned out to be a non-trivial task [FAH74].

For the initial design of VISIONS we have chosen the context representation of the search space because of the availability of the history of decisions. This allows interactive examination of search paths by the human user during system development. However, as the problems of model-building are explored further, we may provide some form of scratch space for collapsing the tree (as in HEARSAY) into a network of parallel competing models, or incorporate pointers between relevant contexts that are not in the same partial model path.

### IV.2 History of Search and Error Recovery

Error recovery and backtracking are important aspects of search, particularly when the hypothesis generators are potentially unreliable. They have serious implications relating to the feasibility of systems which have large computational overhead [WIL77].

The first problem has to do with error detection: how does the system know an error (or errors) has occurred and how does the system begin to isolate candidates for those hypotheses which might be in error? A partial solution to this problem is to maintain a model confidence measure based on confidences of individual hypotheses and their semantic relationships to each other. Presumably, as a model is developed, the confidence of the model should increase (if it is a correct model) as more and more hypotheses are developed and expected semantic relationships between them are satisfied. A decrease in the confidence of the model signals the possibility of an erroneous hypothesis. However, it is important to realize that there is no guarantee that it was the last hypothesis which was in error. It could just as easily have been an earlier hypothesis which was erroneous and which only now is causing the contradiction. The problem of reliable error detection has not been adequately explored in the context of image understanding systems and deserves much more attention.

Assuming errors are detectable and their cause determinable, then the second problem relates to the correction of the error (or errors). Is the error correctable without backtracking? If provision is made to store in the search space the interdependencies of sequential decisions, then it would be possible to tell by inspection if a previous erroneous instantiation could have propagated other instantiations. If it did not, then such an error could be corrected by "un-doing" the error rather than backtracking and regenerating the search tree from the point of error down. Even if it did propagate errors, those instantiations which were not directly or indirectly dependent upon the

error are acceptable without redoing the computation.

The final problem has to do with the efficiency of error correction if backtracking is necessary. Is the process of backtracking easy and efficient? When returning to the state where an error occurred, how much computation is going to be redundant? During the generation of alternative hypotheses, the KSs applied may have required a large amount of computation. If this computation is not to be repeated, then the alternatives must be saved for later use. This is not too difficult -- the system must know which processes produced the results so that they will not be reapplied, while at the same time others which have not yet contributed to the analysis can be invoked. More generally, is directed backtracking possible where the analysis of the error provides information about what to work on next? This is a difficult question to answer and will be the subject of future research.

In order to deal with the problems raised, the information which must be collected and stored during sequential model building includes:

a) the order in which the hypotheses were generated;
b) the KS(s) responsible for generating each hypothesis;
c) the alternative hypotheses that were sufficiently interesting to be saved, but were not followed; and
d) the dependencies of each hypothesis; i.e., at the point of generating a hypothesis, the set of earlier hypotheses which contributed to the generation of the new one.

It may be worth a brief aside to compare the differences in our form of search with that of general problem solving. In the construction of a plan for a robot, for example, the order in which operations are carried out may be an integral part of the solution because of the problem of interdependent subgoals [SAC75]; certain operations necessary to achieve one subgoal may interfere with achieving another subgoal if the operators are applied in the wrong order. In our work, order is important only insofar as it is useful in arriving at a "solution." The solution is a set of hypotheses, with constraints between them, but there is no order inherent in the representation of a semantic interpretation of a scene.

In summary, the search space mechanisms we have described will serve two purposes. The first is that it will allow a user to interactively explore the model-building process so that we can understand the contexts in which the system makes errors. The second is that it opens the possibility that the model-builder could, ultimately, employ strategies to automatically recover from errors by examining these contexts.

## V. Hierarchical Modular Control

As we have shown, the complexity of the virtually unconstrained image interpretation task necessitates the integrated application of many different processes. While the preceding sections have been concerned mainly with problem space representation and decomposition, this section focusses upon issues arising from the need for controlled application of the system resources (e.g., knowledge sources, utilization of semantic constraints, etc.) during the model-building process. It is often very difficult to anticipate some of the underlying issues prior to a system implementation and the familiarity with the problem space that the implementation provides. The penalty can be a complete redesign of the system in order to take into account unforseen problems which arise. Although major changes to a system may be unavoidable, it is possible to structure the system so that these changes are localized with minimal disturbance to the remainder of the system. This has been our approach to problems of representation and search and is also evident in the design and implementation of control strategy mechanisms. Again, we emphasize that our approach is designed to provide a tool which can be used to explore issues of control in the model construction aspect of the image understanding task.

### V.1 Decomposition of Control

"Strategy" is our term for a set of mechanisms (or instructions) which examine the prevailing situation and then decide what to do next, eventually leading to some extension of the current partial model. Rather than attempt to define a single very large strategy, we have chosen to decompose it into a hierarchical set of smaller strategies. The purpose is to allow the researcher flexibility of expression and ease in making local changes, while at the same time maintaining a clear functional structure. We hope it will provide us with a better understanding of the tradeoffs between parallel/sequential, top-down/bottom-up, local/global, and distributed/centralized control mechanisms.

Our decomposition of control corresponds to the structure of the rest of the system. A strategy must determine which partial model in the model search space to expand, which level of representation (the submodel) within the model is to be selected, and which instantiated node (hypothesis) at the selected level will be expanded.

This allows the strategy at the model search space level to be decoupled from the strategy for node selection in that model. This seems to be quite reasonable since the two decisions are based on very different types of reasoning. One model search space strategy could be biased towards a depth-first strategy while another might lead to a more breadth-first search which attempts to explore a larger number of parallel paths. The strategy for focussing on a region node, on the other hand, might be based upon the size and color of the set of regions. This strategy ought to be independent of the strategy for focussing upon an object node based on an instantiated schema, etc. From the point of view of the KS processes, we are also suggesting that the strategy for applying the object-region attribute matching KS, let us say, be separated from the strategy for applying the more expensive spatial processor and occlusion analysis in the verification of the shape of an object volume.

## V.2 Structure of the Control Hierarchy

Now let us describe our hierarchy of modular control (Figure 8) more carefully.[1] We have already provided arguments for the vertical decomposition according to the data acted upon: model search space, model, level, and node. In addition, we decompose control horizontally in the classic hypothesize-test paradigm into types of control modules:

1) FOCUSing on an element of the task,
2) EXPANDing that element by generating new hypotheses, and
3) VERIFYing the new hypotheses.

Each control module makes strategic decisions at its level by determining which lower level control modules and/or KSs to apply. For example, the model search space expander calls: 1) a model focusser (strategy to select a partially developed model from the search space), 2) a model expander (strategy to hypothesize an incremental change to the partial model), and then 3) a model verifier (strategy to determine the correctness of the hypothesis based upon its relationship to the rest of the partial model). Figure 9 provides a dynamic view of the activity of four control units which would be executed during a typical model expansion task. The region-object attribute matcher is the KS invoked in node expansion in this example.

Notice that both Focussers and Verifiers at any level are atomic units of control because they do not call other units of control; rather they cause programs to be executed which operate on data (the model search space or the partial model) to make a decision. The program(s) which they execute, however, can be arbitrarily complex. Initially our Focus strategies are relatively simple, while Verification will vary from being absent (i.e., accept the highest ranked hypothesis) to calls to complex programs before accepting a hypothesis.

All Expanders are non-atomic units of control except at the bottom level. In order to expand the model search space (by generating a hypothesis), a model must be expanded, a level must be expanded, and finally a node must be expanded. Each action is sandwiched around the execution of a Focus program and a Verification program, but centrally involves a call to another Expander which must return some value from the bottom before it

[1] These ideas were discussed in [WIL77] in a somewhat different form, but with the same general purpose in mind. Further research on these topics is currently in progress by J. Lowrance, COINS Department, University of Massachusetts.

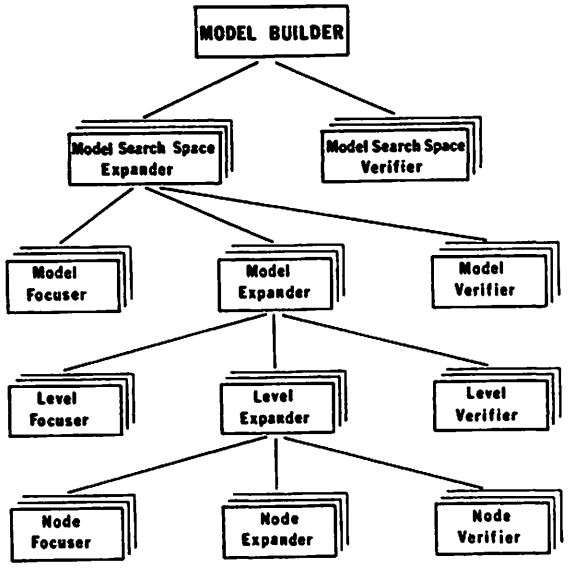



Figure 8. The Hierarchical Modular Structure of Control. The control modules are organized according to type (FOCUS, EXPAND, VERIFY) and the data acted upon (Model Search Space, Model, Level, Node). All Expanders, except the Node Expander, are non-atomic in that they call other control modules. The remainder are atomic because they call KSs to be applied in the model-building process. The functions for focussing, hypothesizing, and verifying are distinct so that commitments of resources can be specified in different model-building strategies.

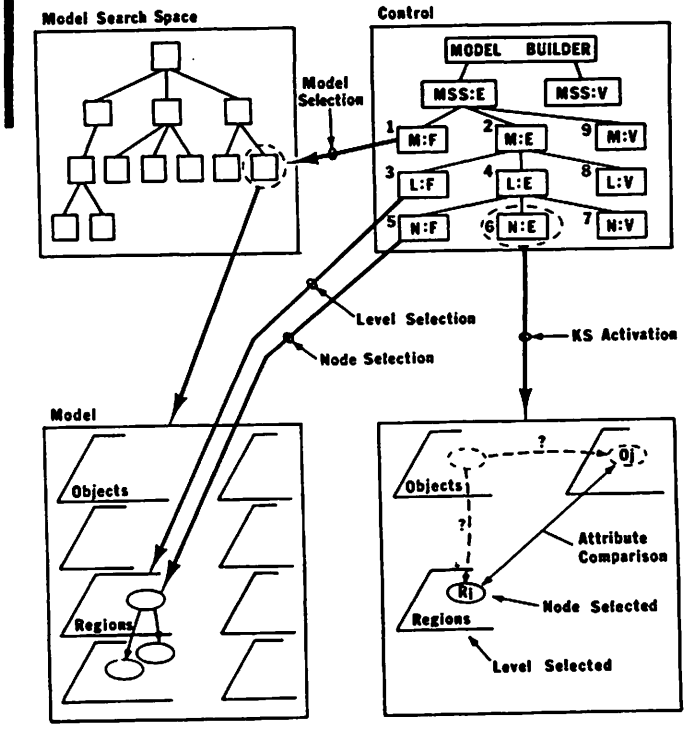



Figure 9. A Dynamic View of Control. A model Focusser is a sub-strategy to select a partial model in the search space. The level and node Focussers are distinct sub-strategies to pick a level and a node to work on. The node Expander calls in the KS which actually forms the hypotheses; the KS attribute matcher is shown in the example. The remaining control units are not shown in their execution. The selection of the level, node, and particular KS to be applied specify whether model development proceeds in a bottom-up or top-down fashion.

has completed execution. The key calls are made by the node expander and node verifier since these will involve the execution of the most reliable KS(s) that we have been able to construct.

This control structure decouples each local control module from the others except to the extent that Expanders will call a Focusser, Expander, and Verifier. We see the possibility of having libraries of control strategies and easily replacing individual modules during experimentation. This approach does impose constraints on the strategies that can be attempted and makes hybrid strategies more difficult. However, it also imposes a discipline concerning the function of control, and is an attempt to avoid entirely ad-hoc mixtures of a wide range of programs. Although we will step outside this paradigm as necessary, it appears to suit our needs in many ways. Default strategies at all control levels have been defined and are just beginning to be used, but we have not yet had sufficient experience to make substantive comments.

## VI. Schemas and the Organization of Visual Information

There is a great deal of structure in our visual environment and it seems evident that such expectations are useful in processing visual information. Although our layered semantic network directly incorporates simple relationships between elements, there are larger complexes of related elements whose organization will be quite important to efficient analysis of scenes. Many of the higher order dependencies that we are interested in are a function of the three-dimensional spatial relationships between surfaces, volumes, and objects. It will be quite useful to have this information collected as a single packet of information in a standard data structure. This organized collection of information, built around a stereotyped scene, event, or object, is stored in our version of a visual schema.

Let us assume that we have stored a fairly extensive body of information about surburban house scenarios. In particular, information about a driveway to a house could include facts such as its functional purpose (to provide a path for autos), its shape and spatial extent (planar, approximately 12' wide, arbitrary length but usually less than 50' long), its spatial relationship to other entities (often leading to a garage, in the ground plane, often perpendicular to the road, etc.). The driveway information as part of the house context has a particular meaning and an expected visual appearance. It could be used to direct analysis (at the region, surface, and/or object level) of the area around a 2D region in order to verify that the region represents a roughly planar surface leading to a house. This analysis can be data-driven (bottom-up) because the hypothesis could have been formed on the basis of the visual attributes of the driveway region. On the other hand, the recognition (or hypothesis) of a house would allow a driveway to be predicted and searched for in a goal-oriented manner (top-down).

Local hypotheses emanating from attributes of individual regions or boundary segments are usually not sufficient in themselves for interpretation. The reader can easily verify the importance of more global contextual information and semantic relationships by restricting the field of view of an image to a small local area, so that the surrounding context is not available. Only in restricted circumstances are the semantics of a scene available from local spectral and shape attributes. This implies that in many cases a representation which captures more global information is required to adequately disambiguate between locally competing hypotheses.

The full description of Minsky's frames [MIN75] involves many characteristics which appear useful, but which would be very difficult to implement. Rather we seek to define the minimal requirements of a visual frame (our schema) and gain some experience with simple structures before attempting to capture relatively esoteric information processing capabilities. It has been our recurring experience that many of the problems which appear when operating on the actual data of real scenes cannot be anticipated. When design considerations become rather elegant and yet fail to face the critical issues which appear, the effort is a pointless exercise. Our initial evaluation of the utility of simple types of schemas must necessarily involve complicated processes operating across many representations and data structures. Until we understand the reliability of the knowledge sources which are producing hypotheses, the redundancy of this information, and a range of top-down and bottom-up strategies for scene interpretation, we believe it prudent to be rather conservative in our design of schemas.

There are three types of information stored in our schemas[1] corresponding to the roles that schemas must play:

1) the parts named in a schema provide a hierarchical partitioning of the knowledge base;
2) the 3D spatial relationships of these parts provide an understanding of shape and space for object recognition and the construction of a surface/volume description; for each part there must be information about the size, the shape, number of parts contributing, etc.; and
3) rough estimates of the conditional probabilities between the presence of the schema and the presence of each part (i.e., part-whole relationships) provide control information for data-driven instantiation of a schema and top-down direction of hypothesis formation.

### VI.1 An Object in a Schema -- An Object as a Schema

Schemas will be used to represent both physical objects and scenes when they are composed of a set of related parts. Our schema defines a stereotypic grouping of elements and focusses upon the relationships between elements. An object with a set of parts (e.g., a house composed of roof, wall,

[1]The ideas involved in the structure and control information in schemas is the work of J. Lowrance, while the use of shape in schemas is the research of B. York, both in the COINS Department at the University of Massachusetts.

window, door, etc.) requires a mode of representation similar to that of a scene with a set of objects (e.g., a road scene composed of road, car, guard rail, centerline, tree, etc.).

The naming of the schema-parts defines the semantic entities which are to be grouped under a label and also defines the predictive capabilities of the schema in top-down processing. All elements which either have a relatively high likelihood of being in the schema or are of strong semantic importance should be named. In effect, this serves to partition the major semantic level in the knowledge base -- the network of representations of objects in the world. Schemas provide the ability to hierarchically group subsets of items by defining their parts in a recursive fashion until each tip node is defined by an object class node which is not decomposed further. Thus, objects appearing in a schema may also have their own schema specifications.[1] For the tip nodes we are assuming that the primitive attributes of color, texture, and shape are a sufficient description of this object and more detailed information on its parts is unnecessary.

Let us examine more carefully the ways in which a concept can be a part of a context, and also be providing a context. Many concepts can be described at three different levels:

a) as a symbolic entity with properties that are independent of both larger contexts and potential decompositions;
b) as a part of a schema where the associated information is relevant to the role that it plays within the schema context; and
c) as a schema itself with specification of its parts and their relationships.

As an illustration consider Figure 10, where the concept of a car is represented in three ways: as a schema, as a schema-part, and as an object. At the object level the CAR class node has properties that are independent of the contexts in which it may appear and its physical decomposition into parts. For CAR these attributes could include its overall physical size, color, price, etc. However, there will be contexts such as the Road-Scene Schema in which the concept of a car plays a role. Here, a node for CAR is created as a schema-part which can have information about cars in road scenes added to it; for example, the expected spatial relationships of cars to the scene, possible directions of movement, their relationship to stop signs, how likely their presence is in the scene, etc. Note that it is not necessary to repeat invariant properties of the CAR class node in the Road-Scene Schema class and that the same car node can be used for all the schemas in which it appears. Only additions and modifications in description that are relevant to the car in each different schema in which CAR appears will have its own schema-part node for CAR [MIN75], but all of these will point to the same CAR node at the object level. The third representation of the concept for

[1]The decomposition of scene-schemas into parts which are themselves scene-schemas (not objects) is currently under investigation.

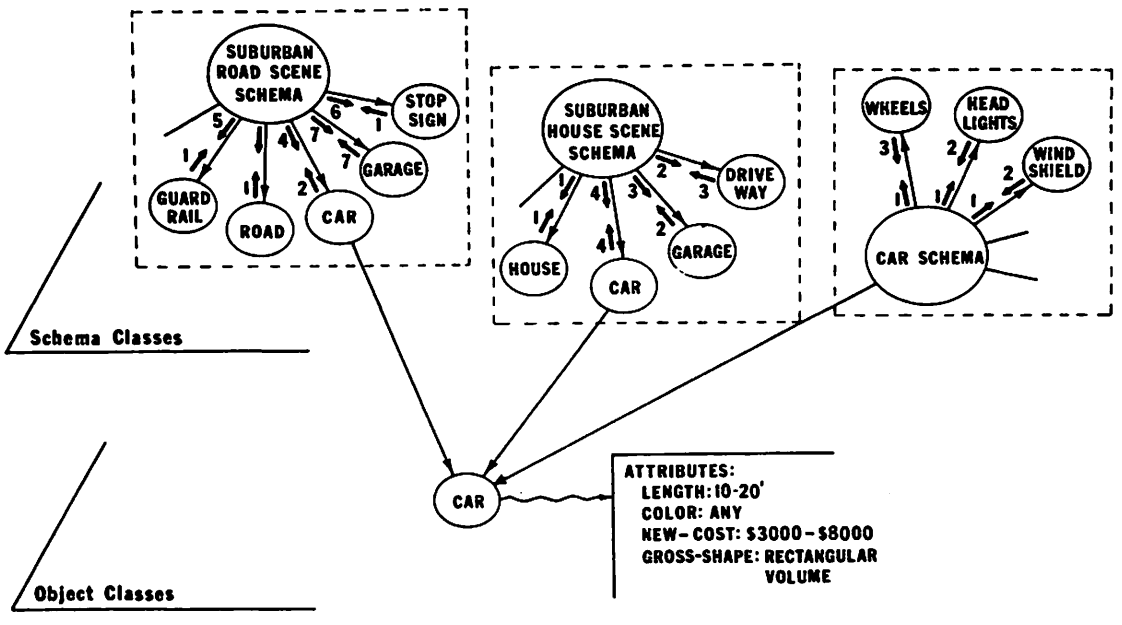



Figure 10. The Structure of Schemas. A physical object can appear as an Object, as a Schema-Part, and as a Schema. At the Object level CAR has attributes which are independent of its parts or the contexts in which cars may appear. As a Schema-Part CAR can be described as part of the Road-Scene Schema, e.g., where it will typically appear in the scene. At the Schema level the CAR-Schema describes the relationships of the parts of cars. Schemas provide control information in terms of rough conditional probabilities (as at all other levels of LTM) where ↑ ≡ P(Schema|Schema-Part) and ↓ ≡ P(Schema-Part|Schema). The probabilities are denoted on a scale of 1 to 7 where 1 denotes P(X|Y) = 1. and 7 denotes P(X|Y) = 0.

car is as a Car-Schema class, where the parts of CAR and relationships between parts of CAR are specified.

### VI.2 Schemas Organize Space and Shape

The second role for schemas is to provide information for building a volume/surface description of the world depicted in the image. In order to do this it is essential to have a volume/surface representation of typical objects and scenes stored in long-term memory. For example, the features of a tree in the distance (even if the trunk and crown are visible) are not sufficient to infer the three-dimensional structure of the object. In a goal-oriented situation where this level of detail is of interest to a human, much of the spatial information must be made available from memory (stored experience) of closer views and prior interactions with trees.

In vision we face a world with dependencies between a large number of variables [FIS78], and often we attempt to approximate these dependencies with binary relationships between elements. To some extent this is compensated for in our system because n-ary relationships become available by attaching 3D representations of objects and scenes at the schema level. If a particular view of a prototypical house is assumed, then a stored 3D representation allows potentially complex relationships between a set of regions representing window, door, roof, etc. in the 2D image to be inferred. In general, a 3D shape representation of an object or scene can provide n-ary relationships between entities at all the lower levels of objects, volumes, surfaces, regions, segments, and vertices. In our first implementation of VISIONS the stored relationships at the lower levels of LTM will be restricted to either binary or AND/OR relationships. These stored static dependencies can be enriched dynamically during the processing of a schema where more complex relationships can be extracted as they are needed.

Minsky suggests that a frame system consisting of a set of 2D projections of a scene (representing typical viewpoints) might be sufficient for recognition of the scene from most viewpoints. We believe that standard 2D views will be quite useful because they provide patterns for direct matches at the region level with little computational overhead; rotations of stored 3D descriptions are avoided. Different views of a house would show a window as a rectangle or as a trapezoid. We have already described ways for accessing a 3D view via pointers in LTM given such 2D region shapes (Section II).

If a set of 2D views are to be sufficient, however, many views must be stored or else there must be means by which the system can interpolate between views. If a human has a slightly unusual perspective of a scene, it does not cause a great deal of difficulty. Looking down upon a road at a 30° angle, the volumes and surfaces of the road, cars, and telephone poles can be interpreted without ambiguity. However, the 2D regions of the projected image change shape. At a minimum these changes of shape would be necessary in order to interpolate between pairs of views. Although such mechanisms can be quite useful, we believe this approach does not provide the desirable flexibility since the system will not really understand crucial aspects of our three-dimensional world. A human has a rough sense of the 3D form of an object; one can sketch the volume filled by an object if he has some artistic abilities and can imagine the 3D spatial extent of an object. These capabilities are not easily derived from a set of 2D views.

In addition to familiar 2D views of an object or scene, we will store 3D information about objects and scenes using the 3D shape representation described in Section III.4. This will open a range of matching strategies between the 3D representation and their expected 2D projections as regions in the image.

### VI.3 Schemas Provide Control Information

The third role of a schema is to guide the instantiation of hypotheses during construction of a model. Here, the minimum information necessary involves implications between each part and the schema, i.e. the rough specification of the importance of the presence of each object to the presence of the schema, and vice versa. In addition a priori likelihoods of the schema and its parts will be useful. Some type of information of this sort appears necessary for ordering expectations based on partial models.

In Figure 10 upward and downward arrows between schemas and schema-parts have associated conditional probabilities, or weights, representing P(Schema|Schema-Part) and P(Schema-Part|Schema), respectively. Since we do not have reasonable means of statistically estimating these probabilities, we choose to form intuitive estimates on a coarse scale of values between 0 and 1. They capture to a first approximation the relative importance between particular objects and a schema. For example, the presence of a guard rail strongly implies the road scene schema ($w = 1.$), while the inverse implication is much weaker ($w = .25$), since the absence of a guard rail in a road scene is to be expected in many cases.

Of course this is a crude approximation to the dependencies between objects in a schema. However, the accurate estimations of the joint distribution of all the subsets of the objects in a road scene is not feasible. Even with this simplication, the tuning of the weights might cause difficulty, but since we only intend to use these values as rough estimates, we expect the problems to remain tractable.

### VI.4 The Size and Overlap of Schemas

In our domain a driveway will not be of sufficient importance around which to organize its own context. It might be possible to store a model of the visual context around every object which is in the pool of identifiable objects. However, this could involve a great degree of redundant storage of information. Certainly a schema for the curb of a suburban street should not be necessary. Rather our schemas will initially be constructed around those entities which have significant semantic importance and will vary with

the goals of the vision system. For the moment, let us assume that the house scene and the road scene are situations which have schemas organized around them. In this case the driveway will play a secondary role in both the house-scene schema and the road-scene schema. At the object level, the node for driveway will have properties of a stereotypical driveway attached to it which are independent of the context of the driveway. This node can be used in the description of both the house-scene schema, the road-scene schema, and all larger contexts in which such a driveway may appear.

One should note the implications of developing a knowledge base in this fashion, particularly if each schema organizes a relatively small number of concept nodes. Essentially it divides our layered semantic network into overlapping partitions at the schema and object levels [HEN75]; each partition has a particular focus and with each there will be a packet of information useful for top-down and bottom-up processing, including 3D shape representations useful for matching.

One could put the road-scene and house-scene schemas together so that any investigation of a driveway would have a more complete context available in a single place. This decision appears to be poor in light of the purpose which schemas play in focussing attention upon objects which are likely to be present. The size and content of the schemas will directly affect the power and efficiency of knowledge-driven strategies. The presence of a subset of schema-parts (quite possibly a subset of one element) will predict as alternative hypotheses the remaining elements of the schema. If a schema is composed of a small number of very likely elements, then each of the few alternatives is a promising choice for further processing and this set is computationally manageable. If, on the other hand, the house scene and road scene schema are joined, gross inefficiency could result when only one of these contexts is actually present. By keeping them separate the instantiation of additional object nodes proceeds in two stages. First, the alternative schemas that are implied by the current partial model can be weighed. Then an ordering on the objects potentially present in each schema can be computed.

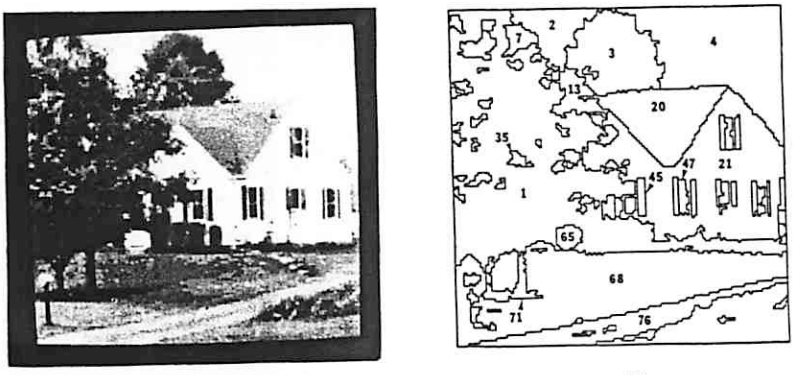
## VII. Results

The experiments that are reported here are the very first with the system in its current form. At this point we have not yet attempted to build a model nor have we examined interesting control strategies or grappled with the difficulties of a large search space. Rather, the analysis has focussed on the quality of the information produced by individual KSs, the content and structure of long-term memory, and the combined effect of KSs in forming hypotheses at the object and schema levels. The results reported were obtained just prior to the writing of the final version of this paper. They are included to provide a sense of the status of the system and the quality of information available from preliminary versions of several of our KSs. We expect that as the result of a more complete analysis of the data, we will begin to refine the knowledge sources and the contents of LTM.

Although we attempted to avoid estimates of a priori and conditional probabilities in stages of earlier development of our system, it has proven very useful for integrating the output of the various KSs. Three KSs are reported here:

a) region-object attribute matches,
b) size estimates from perspective analysis,
and c) 2D shape fitting.

Each is being structured to output confidences of results in terms of a scale from -100 to +100, where a value of 0 implies no information. This allows normalization of the KSs by correlating the zero information point of each KS hypothesis (whether it be a hypothesis of a shape label or an object identity) with the a priori probability of that concept stored in LTM. Currently, our knowledge base includes 40 objects, 57 schema-parts, and 17 schemas. It is almost impossible to maintain consistency (according to Bayes rule) when the a priori and conditional probabilities in the network are subjectively estimated by a human. A method which reduces the severity of this problem and which includes some of the rigor of Bayesian decision theory is provided by the inference net approach of [DUD76]; this problem was also examined in the MYCIN system [SHO75].




(a) (b)

Figure 11. House Scene. (a) Black and white intensity image of suburban house scene. (b) Segmentation produced by early histogram-based region analysis. Numbered regions are those used in the experiments described in the text.

Figure 11a is a black and white image of a suburban house, while Figure 11b is a segmentation produced by a histogram-guided region analysis [NAG77]. It should be pointed out that these segmentation results do not utilize our most recent algorithms that are described in [HAN78] in this volume.

Table I lists sample results obtained by matching attributes of objects and regions. Due to the limited statistics of the training set, and the lack of spectral invariance for many objects, the attribute matching experiments were aimed at classifying only five "target" objects: bush, grass, tree (crown), road and sky. We have listed examples of regions whose actual object identity is in the target set and several in the non-target set. Again remember that the scale is between +100 and -100 using the weighted linear sum of features as described in Section III.3.

Note that if an object, such as the house roof of region 20, does not have expected spectral attributes, it could match some other object in the target set reasonably well, in this case grass. In many cases the errors made are not unreasonable given the visual appearance of the scene; for example, the white house wall (region 21) is matched as sky due to the fact that it has virtually no discernable texture and is almost the same color as the sky. These types of errors ought to be correctable from additional sources of information, such as shape, position in the image, and relationships with surrounding regions.

Table II summarizes the results of applying the attribute matching KS on two images, the example image of Figure 11 and the image of Figure 21a in the companion paper. The results are presented in various ways. Based upon size, we have examined a) all regions, b) those regions large enough to entirely contain at least one 5×4 window (i.e., those regions for which textural features are computable), and c) large regions which contain at least 65 pixels. In all three groups, the accuracy of recognition of regions whose identities were in the target set was fairly high -- greater than 75%. The majority of errors occurred between tree and bush which is not unexpected, since these objects have similar spectral characteristics and differ primarily in size. When these two object classes were collapsed into one, the recognition rate among the four target classes was greater than 90%.

In order to provide a sense of the confidence in erroneous hypotheses for non-target regions, Table II also summarizes the average values of the heuristic confidence measure for the best hypothesis for correctly identified targets, incorrectly identified targets, and non-targets. The associated distributions are also provided in Figure 12. These results are very promising in their overall accuracy and in the difference in confidences between target and non-target distributions. However, the distributions do overlap, and this single knowledge source cannot be expected to separate targets from non-targets without error. As part of the development of this knowledge source, we will be extending the set of features used and improving the prototype representations in LTM as a larger data base of images becomes available. Of course the single biggest problem with this KS will still remain -- many object classes in LTM are not characterized by features that tend to be invariant in color and texture.

Table III provides samples of the ranges of sizes (mainly heights, but all dimensions that can be simply stated will be specified) for object classes in LTM. Using this information, a computed

| Region Identification (Figure 11b) | Actual Region Identity | Hypothesized Region Identity | Hypothesis Correct? | Confidence Region is: Bush | Grass | Road | Sky | Tree | Area of Region (pixels) | Number of 5×4 Windows in Region |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | TREE | TREE | YES | 31 | -72 | -92 | -62 | 61 | 4300 | 2888 |
| 2 | SKY | SKY | YES | 16 | 1 | -55 | 35 | 16 | 407 | 299 |
| 3 | TREE | TREE | YES | 33 | -21 | -89 | -38 | 95 | 846 | 611 |
| 4 | SKY | SKY | YES | -58 | -44 | 0 | 68 | -78 | 1763 | 1577 |
| 7 | TREE | TREE | YES | -27 | -18 | -71 | -65 | 80 | 97 | 33 |
| 13 | TREE | TREE | YES | 12 | -17 | -89 | -74 | 84 | 195 | 74 |
| 20 | ROOF | GRASS | NO | -43 | 35 | 32 | -20 | -54 | 854 | 609 |
| 21 | WALL | SKY | NO | 2 | 3 | - 8 | 47 | -16 | 2207 | 1347 |
| 35 | TREE | TREE | YES | 37 | - 7 | -85 | -56 | 42 | 31 | 2 |
| 45 | SHUTTER | TREE | NO | -48 | -29 | -67 | -47 | - 4 | 42 | 0 |
| 65 | WALL | SKY | NO | -12 | -43 | -26 | 14 | 4 | 79 | 30 |
| 68 | GRASS | GRASS | YES | 0 | 69 | -56 | -50 | 31 | 2588 | 1904 |

Table I. Attribute Matching. Sample results obtained from the region-object attribute matching KS on the segmentation shown in Figure 11b. The last two columns show the area of the region in terms of the number of pixels it contains, and the number of 5×4 windows (used to compute the texture features) totally contained in the region.

| | Summary of Identification Accuracy | | | | | |
|---|---|---|---|---|---|---|
| | All Regions | | Regions in Which at Least One 5×4 Window Will Fit | | Large Regions (area ≥ 65 pixels) | |
| | All 5 Objects | 4 Objects After Collapsing Bush-Tree | All 5 Objects | 4 Objects | 5 Objects | 4 Objects |
| Total Number of Regions | 209 | | 83 | | 45 | |
| Number of Target Regions | 99 | | 50 | | 25 | |
| Number of Non-Target Regions | 110 | | 33 | | 20 | |
| Number of Target Regions Correctly Identified | 76 | 91 | 40 | 45 | 19 | 23 |
| Number of Target Regions Incorrectly Identified | 23 | 8 | 10 | 5 | 6 | 2 |
| % Target Regions Correct | 76.7 | 91.9 | 80. | 90. | 76. | 92. |
| | Summary of Averages of Confidence Measures | | | | | |
| Correctly Identified Target Regions | 31.8 | 29.9 | 49.5 | 49.1 | 63.2 | 59.8 |
| Incorrectly Identified Target Regions (highest value) | 21.4 | 23.8 | 36.9 | 28.2 | 37.6 | 25. |
| Non-Target Regions (highest value) | 8.4 | | 24.9 | | 27.3 | |

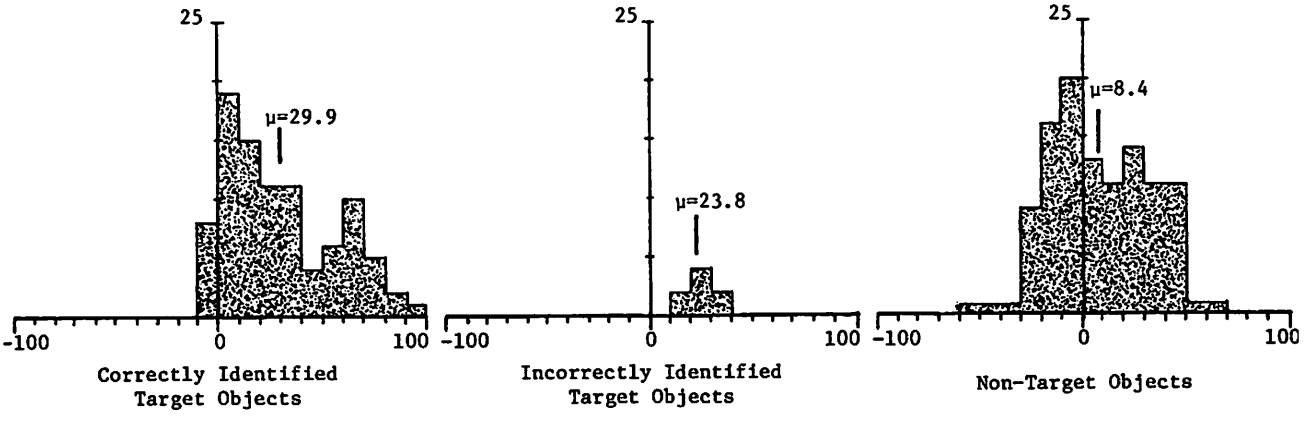
Table II. Summary of Region Identification Results from Attribute Matcher Applied to Two Images.

physical size will be assumed to provide no information (0) if it is exactly at the limits of the range, linearly increasing positive support for a hypothesis as it approaches the limits of the expected range, maximum support (100) inside the expected range, and negative support (-100) if it is outside the range. This value is then scaled by a coefficient which represents an importance measure based on the discriminability of a particular size value in a manner similar to that described in Section III.3. The a posteriori probability estimate of an object is a function of the a priori probability, and the match of the attribute information (here size) provided by the KS with the stored object attribute in LTM, modified by the discriminability of the attribute computed on the basis of the overlap of the attribute values with other objects. Thus, a particular physical size value, even if it falls inside the expected size range for an object, will not significantly raise the a priori probability estimate, if that size falls inside the expected range of many other objects.

We have already described (in Section III.5) the range of assumptions that is necessary before the physical size of an object can be computed on the basis of image size. Determining this size requires computing the distance of the object from the camera lens. The computation is very sensitive to the height and inclination of the camera and to the elevation of the object above the ground plane (perhaps available from perspective cues). This sensitivity is most pronounced for distant objects.

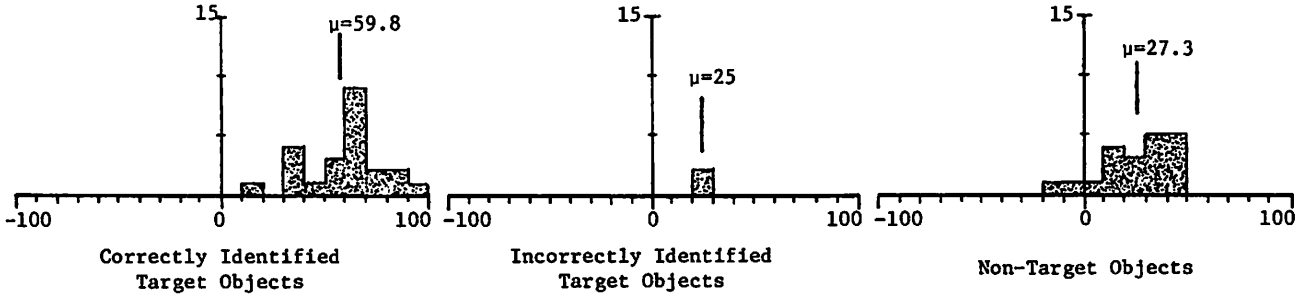
Table IV shows the computed physical sizes of particular regions in the image (Figure 11), and the assumptions that were necessary in order to carry out their computation. We have not yet automated the strategies by which these assumptions can be generated, and consequently the results shown were based on interactive guidance by a user.[1] The reader should note, however, that there are various sources for suggesting and constraining these assumptions. For example, the attributes of region 68 imply grass (the ground plane) with high probability, and it is at the lower part of the image where the ground plane is expected. This allows

[1]This work is currently in progress. It is based on initial investigations by Kurt Konolige (now at SRI, International) and is being continued by Daniel Corkill of the COINS Department at UMass. The results of this work will be reported in the future.




(a)

Distributions of confidence measure for all regions with 4 target objects (grass, road, sky, bush/tree).




(b)

Distribution of confidence measure for large regions (area ≥ 65 pixels) with 4 target objects (grass, road, sky, bush/tree).

Figure 12. Distributions of values produced by heuristic confidence measure for attribute-matching. The horizontal axis is the value of the measure and the vertical axis is the number of regions with that value.

distances of surfaces to be computed relative to their image location in region 68. The attribute matching KS implies the strong, likelihood of tree for regions 1 and 3 and they can be located in relation to the grass surface of region 68. The shape of region 45 suggests that there is a reasonable possibility that it is a surface perpendicular to the ground plane. There is also top-down guidance from stored sizes of objects. Thus, occlusion of the tree, region 3, by region 20, and the knowledge that regions 20 and 21 are part of a house (by other analyses) would allow the lower bound on the size of the tree to be raised by increasing, by the stored prototypical width of the house class, the distance at which the tree is rooted to the ground.

Admittedly, there is a non-trivial sequence of assumptions that must be made in order to arrive at the results that are shown. However, if the assumptions can be generated, then in many cases there is a check on the validity of each hypothesis because size must be consistent with stored expectations (as with the trees, window, and house). In addition, the assumptions must be consistent with any contexts which were employed (e.g., the relative locations of tree rooted to grass, house, windows, shrubs, and sky). There appears to be a rich and interesting line of research that is available here. These results only outline our first attempts in this area.

In order to make use of the 2D shape analysis described in the companion paper, the confidence of primitive shape labels associated with a region must be determined from the RMS errors of the curve fits and the constraints on the way in which the individual segments fit together. For example, quadrilateral, trapezoid, rectangle, and square are

| Object Class | Prototype Values (in meters) | | |
|---|---|---|---|
| | Minimum Expected Size | Expected Range of Computed Sizes | Maximum Expected Size |
| Car (length) | 2.2 | 3.7 - 5.0 | 6.7 |
| Door | 1.5 | 2.0 - 2.5 | 3.7 |
| Garage | 3.0 | 4.6 - 5.0 | 5.5 |
| Garage-Body | 2.2 | 3.0 - 4.4 | 4.9 |
| Garage-Door | 2.0 | 2.2 - 2.5 | 3.7 |
| House | 3.0 | 4.6 - 7.7 | 10.8 |
| House-Body | 2.2 | 3.0 - 5.5 | 7.7 |
| Human | 1.2 | 1.6 - 1.9 | 2.2 |
| Roof-Projection | 0.0 | 0.0 - 3.0 | 3.7 |
| Telephone-Pole | 3.0 | 5.0 - 8.0 | 11.0 |
| Tree | 2.5 | 4.3 - 9.2 | 15.4 |
| Tree-Crown | 1.3 | 2.5 - 5.5 | 9.7 |
| Tree-Trunk | 1.3 | 1.9 - 3.7 | 6.2 |
| Window | .6 | .9 - 1.5 | 2.5 |

Table III. Ranges of sizes (height unless otherwise specified) associated with sample object classes as stored in LTM.

shapes which share a common property (all are composed of four straight lines which intersect at four corners) but with increasing constraints on the manner in which the lines are combined; a quadrilateral is a superclass of a trapezoid, which is a superclass of a rectangle, etc. The computation of the confidences, then, should reflect not only the RMS errors of the fits of the individual lines but also the effects of these constraints. In the case of geometrically regular regions for which these labels might apply, all might be present with varying degrees of confidence. A heuristic function, which produces confidence values between 0 and 1, has been developed as a function of the RMS error fits and the expected constraints. Note that we have not yet examined the discriminability of shape labels, nor the points of zero information as used in the other two KSs. Because of the very preliminary nature of this research, we will present limited results. Table V summarizes the average RMS errors for straight line fits of sets of segments bounding selected regions in Figure 11b, and confidences associated with these regions (see also Section V in the companion paper).

The next experiment involved exercising the long-term knowledge base by combining the results of various KSs in order to obtain an initial test of the propagation of information. That is, we wished to see the way in which LTM allows KSs, which make hypotheses at different levels, to bring their influence together at a common node. The importance of these results resides not as much in the final a posteriori values to be shown, but rather in the manner in which the evidence propagates and the direction of the change in the a priori probabilities as more evidence is added. Again, we emphasize that the results are very preliminary, and do not adequately represent what we hope to achieve.

In order to combine the 2D shape KS and the size KS, the probability of the hypothesis of a 2D region shape label (e.g., region 47 as a rectangle) must be propagated upward to the object level via conditional probabilities on the arcs. This information may then be combined with the result of a size analysis and the results used as the new evidence of the object ("window"). The results of this experiment are tabulated in Table VI. Note that the size KS drives the a posteriori probability of the hypothesis that region 47 is a tree crown to 0, since the size of this region lies totally outside the minimum/maximum values of the expected size range for tree crown. This produces the proper results in this case, but for the case of region 3, it produces incorrect results. The maximum size of tree crowns was set at 9.7 meters (Table III) while the computed size of region 3 was 11.7 (Table IV). There are two ways to solve this problem: either an increase in the stored maximum value of tree crowns, or a weaker effect of sizes outside the range (proportional to the distance outside the extremes of the range). Several erroneous hypotheses have also been included in Table VI for comparative purposes.

The final experiment is tabulated in Table VII. Here several pieces of evidence from the various KSs are introduced and the cumulative effect of this evidence upon several schema is shown. Both the house-scene and lawn-scene schemas have significant increases in their a posteriori probabilities. It is worth noting that the influence of tree probably should not increase the probability of lawn-scene so sharply, nor should window increase drive-scene so dramatically (even though garages in drive-scenes have windows).

## VIII. Conclusion

### VIII.1 Flexibility for System Evolution

Our design involved modularization of data into levels, modularization of processes which transform data from one representation to another, and modularization of the strategies which employ them. It should allow us the flexibility to explore local interactions between executing processes, and the control issues at a more global level.

With this approach to representation, processes, control, and search, we believe that the system can be reconfigured incrementally with modest (although sometimes not trivial) effort. For example, suppose one of the paths to surface hypotheses turns out to be ineffective without the addition of another process, say motion cues (provided by optic flows of motion currently under investigation by our colleagues); this might well require some intermediate level of representation in order to combine those results with our own surface cues based on shape. It would not be surprising for such a change to have so major an impact upon a system that it would need to be entirely recoded. Although it would still take some effort in our system, we expect that most of

| Region(s) | Elevation | Distance | Height | Width | Actual Identity |
|---|---|---|---|---|---|
| 71 | 0.0 | 18.3 | .79 | .24 | Tree Trunk |
| 1 | .794 | 18.3 | ≥ 4.06 | ≥ 3.11 | Tree Crown |

Assumptions:
RG-71 is perpendicular and attached to ground plane
RG-71 is split into two sub-regions with only right part considered, and width measured on upper portion
RG-1 is perpendicular to ground plane and elevated above volume for RG-71
RG-1 is occluded by image boundary at top and left
RG-1 is subdivided by partitioning the shrub area in front of house; it was not sub-divided into two separate regions corresponding to the two actual trees

| | | | | | |
|---|---|---|---|---|---|
| 3(a) | 0.0 | ≥ 34.7 | ≥ 7.71 | ≥ 3.95 | Tree (Crown) |
| (b)<br>(c) | 0.0 | ≥ 57.9 | ≥ 11.7 | ≥ 6.57 | Tree (Crown) |

Assumptions:
RG-3 is occluded by RG-20 (from occlusion analysis)
(a) RG-3 is perpendicular and attached to ground plane at a distance further than the top of RG-68, whose spectral attributes imply grass
(b) house is identified; distance and size will be computed by adding the prototypical width of house stored in LTM to the physical distance of the top of RG-68 (grass)
(c) RG-3 is perpendicular and attached to ground plane behind RG-1, RG-20, and RG-21.

| | | | | | |
|---|---|---|---|---|---|
| 21 | 0.0 | 34.7 - 43.4 | 5.19 - 6.08 | 4.44 - 5.55 (a) (b) | House Wall |

Assumptions:
RG-21 is perpendicular and attached to ground plane at a distance between the physical distance to the top of RG-68, and the distance to bottom of RG-21 based upon a slight occlusion cue from shrub area (RG-1).
(a) RG-21 is perpendicular to the line-of-sight
(b) RG-21 is subdivided into two house walls; refer to Figure 28b in the companion paper

| | | | | | |
|---|---|---|---|---|---|
| 20 | 2.77 - 3.13 | 34.7 - 43.4 | 2.35 - 3.13 | 5.52 - 6.90 | House Roof |

Assumptions:
RG-20 is elevated above the volume for RG-21
RG-20 is perpendicular to ground plane (an incorrect assumption)

| | | | | | |
|---|---|---|---|---|---|
| 47 | 1.05 - 1.20 | 34.7 - 43.4 | 1.04 - 1.36 | .394 - .493 | House Window |

Assumptions:
RG-47 lies in plane of RG-21, and therefore its distance is defined by the range of RG-21
RG-47 is perpendicular to the line-of-sight

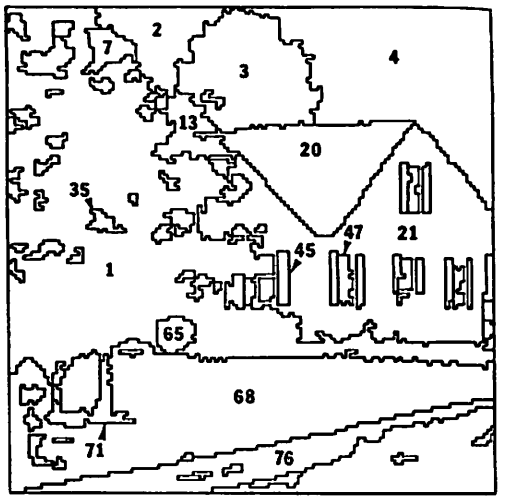
| | | | | | |
|---|---|---|---|---|---|
| 76 | 0.0 | Not Applicable | Not Applicable | 1.76 | Road (Narrow Driveway) |

Assumptions:
RG-76 is in ground plane
Two long segments forming RG-76 are actually parallel in the physical world

Table IV. Computed physical (real world) sizes of selected regions of Figure 11. The assumptions necessary to make the computation are summarized below the computed values.

| Region | Average RMS Error of Straight Segment Fits | Shape Label | Confidence of Label | Actual Identity of Region |
|---|---|---|---|---|
| 20 | .98 | Triangle | .971 | House roof |
| 45 | 0.0 | Quadrilateral<br>Trapezoid<br>Rectangle<br>Square | 1.0<br>1.0<br>1.0<br>.33 | Window shutter |
| 47 | 2.01 | Quadrilateral<br>Trapezoid<br>Rectangle | .92<br>.88<br>.45 | Window |
| 76 | .41 | Quadrilateral<br>Trapezoid<br>Rectangle<br>Triangle | .98<br>.912<br>0.<br>.96 | Road |

Table V. Preliminary results from 2D shape KS.




Region Identification
(same as Figure 11b)

| Region | Hypothesized Identity | a priori probability of hypothesized identity (stored in LTM) | a posteriori probability given evidence from attribute KS | size KS | shape KS | attributes and size | size and shape | Actual Identity of Region |
|---|---|---|---|---|---|---|---|---|
| 1 | Tree-Crown | .1211 | .657 | .237 | —— | .812 | —— | Tree-Crown |
| 3 | Tree-Crown | .1211 | .956 | 0. | —— | 0. | —— | Tree-Crown |
| 21 | Tree-Crown | .1211 | .101 | .268 | —— | .229 | —— | House-Wall |
| 47 | Tree-Crown | .1211 | .490 | 0. | —— | 0. | —— | Window |
| 47 | Window | .0049 | —— | .326 | .018 | —— | .639 | Window |

Table VI. Combination of evidence from various KSs at the object level.

| Hypotheses at Object-Class Level | KS(s) Used to Form Hypotheses at Object-Class Level | Effect of Combined Hypotheses on a priori Probability of Schema Classes | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | House-SC | | Drive-SC | | Lawn-SC | | Road-SC | | Walk-SC | |
| | | a priori | a posteriori | a priori | a posteriori | a priori | a posteriori | a priori | a posteriori | a priori | a posteriori |
| R47-Building-Window | Size | .2344 | .907 | .11 | .643 | .24 | .2354 | .13 | .1311 | .03 | .0304 |
| R1-Tree-Crown | Spectral Attribute and Size | .2344 | .738 | .11 | .1065 | .24 | .997 | .13 | .1311 | .03 | .0304 |
| R47-Building-Window and R1-Tree-Crown | Size<br>Spectral Attribute and Size | .2344 | .988 | .11 | .643 | .24 | .997 | .13 | .1311 | .03 | .0304 |
| R21-House-Body | Size | .2344 | .363 | .11 | .1065 | .24 | .2354 | .13 | .1311 | .03 | .0304 |
| R47-Building-Window and R1-House-Body | Size<br>Size | .2344 | .948 | .11 | .643 | .24 | .2354 | .13 | .1311 | .03 | .0304 |
| R47, R21, and R1 (hypotheses as above) | Size<br>Size<br>Spectral Attribute and Size | .2344 | .994 | .11 | .643 | .24 | .997 | .13 | .1311 | .03 | .0304 |
| R47-Human (incorrect hypothesis) | Size | .2344 | .236 | .11 | .110 | .24 | .240 | .13 | .136 | .03 | .038 |
| R1-Tree C R21-House-Body R47-Human (incorrect) | —— | .2344 | .845 | .11 | .1068 | .24 | .997 | .13 | .136 | .03 | .038 |

Table VII. Results from combining hypotheses at the object-class level and the effect on the schema-class level.

the energy expended in order to add these capabilities would require specification of the local relationships of these additional processes with various aspects of the system. The required intermediate representation would be added as a distinct level of representation without directly affecting the other levels. Next, processes would have to be defined to transform patterns from lower levels to this level and from the new level to higher levels. These should not affect the functioning of other modular processes except to the extent that the internal steps of the analysis are dependent upon internal steps of existing processes -- in such a case reorganization of the dependent existing processes would be inevitable. After adding the necessary knowledge sources for transformation of the patterns, strategies for deciding how to employ these new processes locally can be added without causing changes throughout the system. Finally, during the development of these changes, the careful examination of the history of selected local paths in the search space provides a dynamic trace of the processing and should aid in analyzing the effectiveness of these changes.

### VIII.2 Feedback to the Low-Level System

Once semantic hypotheses are formed, there is a rich source of information to direct segmentation processes which initially have no semantic guidance [HAN78, this volume]. The hypotheses of object identities allows selection of more effective features for segmentation, as well as extraction of more descriptive features for hypothesis verification. The fitting of shape descriptors to regions and segments also provides more global information for the refinement of boundaries which were initially based upon very local views of the sensory data. Investigation of these interesting problems is planned for the future.

### VIII.3 Results

The results shown in Section VII are our first attempts at exercising initial versions of several of the knowledge sources and long-term memory on actual data. Although they were obtained just prior to the writing of the final draft of this paper and should not be taken as conclusive, they are demonstrative of the kinds of experiments we are now beginning to run. Each of the KSs employed must be improved, and there are several others being implemented, such as 3D shape matching, whose influences have not yet been incorporated.

As we have repeatedly stressed, one of the motivations behind VISIONS has been the construction of a tool with which we can begin to explore the complexity of the problem domain of unconstrained complex images. Even this limited set of experiments has begun to show areas where modifications are necessary. Once the reliability of the KSs is understood, we can begin to explore interesting control strategies in the development of interpretations, including top-down schema-driven control.

### VIII.4 System Evaluation

It is impossible to predict the power and effectiveness of our approach to computer vision. Ultimately the power of the system is limited by the quality of the knowledge sources which are forming and evaluating hypotheses. We will consider the high level design of VISIONS successful if it permits the system to evolve with relative ease while exploring those limits. The success of a general vision system is another matter and awaits further stages of experimentation, redesign, and improvement of knowledge sources. This long range task, although rather formidable, appears feasible.

As of this writing, the core of the VISIONS system has been implemented; default strategies are available, the knowledge base has been filled out, initial versions of several of the knowledge sources are complete and being refined, 2D shape processing is being completed, and segmentation data is being passed from the low-level system. We are currently designing and carrying out a series of experiments to begin the assessment of the system. These results, as they become available, will be reported in the literature.

### Acknowledgements

This research was supported in part by the National Science Foundation under Grant DCR75-16098. We would like to express our gratitude to the Computer and Information Science Department at the University of Massachusetts for providing a congenial atmosphere in which to conduct this research. The support, constructive criticism, and advice offered by many of our colleagues, particularly Michael Arbib and Victor Lesser, is gratefully acknowledged. The research reported here is based primarily upon the untiring efforts of John Lowrance, Thomas Williams, and Bryant York, with contributions from Daniel Corkill and Kurt Konolige. These individuals and those who have contributed to the segmentation system of VISIONS have formed a rare cooperative research group whose dedication, and sometimes personal sacrifice, will always be appreciated. Finally, we wish to thank Ms. Janet Turnbull for her help and patience in producing the various drafts of this manuscript and many others during the last three years.

### References

[AGI72] G.J. Agin, "Representation and Description of Curved Objects," Stanford AI Memo 73, 1972.

[ARB72] M.A. Arbib, The Metaphorical Brain: An Introduction to Cybernetics as Artificial Intelligence and Brain Theory, Wiley-Interscience, 1972.

[ARB75] M.A. Arbib, "Artificial Intelligence and Brain Theory: Unities and Diversities," Annals of Biomedical Engineering, 3, 238-274, 1975.

[ARB77] M.A. Arbib, "Parallelism, Slides, Schemas, and Frames," in Systems: Approaches, Theories, Applications (W.E. Hartnett, Ed.), D. Reidel Publishing Co., 27-43, 1977.

[ARB77] M.A. Arbib, Personal communication.

[BAI75] M.L. Baird, J.J. Olsztyn, W.A. Perkins and L. Rossol, "The GM Research Laboratories' Machine Perception Project," Technical

Report, General Motors Research Laboratories, Warren, Michigan, October 1975.

[BAJ78] R. Bajcsy and A. Joshi, "A Partially Ordered World Model and Natural Outdoor Scenes," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[BAL78] D.H. Ballard, C.M. Brown, and J.A. Feldman, "An Approach to Knowledge Directed Image Analysis," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[BAR72] H.G. Barrow, A.P. Ambler, and R.M. Burstall, "Some Techniques for Recognizing Structure in Pictures," Frontiers in Pattern Recognition (S. Watanabe, Ed.), Academic Press, New York, 1972.

[BAR76] H.G. Barrow and J.M. Tenenbaum, "MYSYS: A System for Reasoning About Scenes," Technical Note 121, AI Center, Stanford Research Institute, April 1976.

[BAR78] H.G. Barrow and J.M. Tenenbaum, "Recovering Intrinsic Scene Characteristics from Images," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[BIN71] T.O. Binford, "Visual Perception by Computer," presented to the IEEE Conference on Systems and Control, Miami, December 1971.

[BRA78] J.M. Brady and B.J. Wielinga, "Reading the Writing on the Wall," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[BRE74] J. Brenner, E. Gelsema, T. Necheles, P. Neurath, W. Selles, and E. Vastola, "Automated Leukocytes," The Journal of Histochemistry and Cytochemistry, 22, 697-706, 1974.

[BUL78] B.L. Bullock, "The Necessity for a Theory of Specialized Vision," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[CAR77] S.G. Carlton and O.R. Mitchell, "Image Segmentation Using Texture and Gray Level," Proc. Pattern Recognition and Image Processing, Troy, NY, June 6-8, 1977, 387-391.

[CLO71] M.B. Clowes, "On Seeing Things," Artificial Intelligence, 2, 79-116, 1971.

[COO67] S.A. Coons, "Surfaces for Computer-Aided Design of Space Forms," MIT Project MAC TR-41, June 1967.

[COO74] S.A. Coons, "Surface Patches and B-Spline Curves," in Computer Aided Geometric Design (R.E. Barnhill and R.F. Riesenfeld, Eds.), Academic Press, New York, 1974.

[DUD73] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, 1973.

[DUD76] R.O. Duda, P.E. Hart, and N.J. Nilsson, "Subjective Bayesian Methods for Rule-Based Inference Systems," Proc. of the National Computer Conference, 1976.

[ERM75] L. Erman and V. Lesser, "A Multi-Level Organization for Problem-Solving Using Many, Diverse, Cooperating Sources of Knowledges," Proc. IJCAI-4, Tbilisi, USSR, September 1975, 483-490.

[ESC71] The World of M.C. Escher, (J.L. Locher, Ed.), Harry N. Abrams, Inc., New York, 1971.

[FAH74] S.E. Fahlman, "A Planning System for Robot Construction Tasks," Artificial Intelligence, 5, 1-49, 1974.

[FIS73] M.A. Fischler and R.A. Eschlager, "The Representation and Matching of Pictorial Structures," IEEE Trans. on Computers, January 1973.

[FRI69] D.P. Friedman, D.C. Dickson, J.J. Fraser, and T.W. Pratt, "GRASPE 1.5 - A Graph Processor and Its Application," Tech. Report, University of Houston, 1969.

[GOR74] W.J. Gordon and R.F. Riesenfeld, "B-Spline Curves and Surfaces," in Computer Aided Geometric Design (R.E. Barnhill and R.F. Riesenfeld, Eds.), Academic Press, New York, 1974.

[GRE66] R.L. Gregory, Eye and Brain: The Psychology of Seeing, McGraw-Hill, New York, 1966.

[GUZ68] A. Guzman, "Decomposition of a Visual Scene into Three-Dimensional Bodies," Proc. of Fall Joint Computer Conference, 33, 291-304, 1968.

[HAN75] A. Hanson and E. Riseman, "The Design of a Semantically Directed Vision Processor (Revised and Updated)," COINS Technical Report 75C-1, University of Massachusetts, February 1975.

[HAN76a] A. Hanson, E. Riseman, and T. Williams, "Constructing Semantic Models in the Visual Analysis of Scenes," Proc. of IEEE Milwaukee Symposium on Automatic Computation and Control, April 1976, 97-102.

[HAN76b] A. Hanson and E. Riseman, "A Progress Report on VISIONS: Representation and Control in the Construction of Visual Models," COINS Technical Report 76-9, University of Massachusetts, July 1976.

[HAN76c] A. Hanson, E. Riseman, and E. Fisher, "Context in Word Recognition," Pattern Recognition, 8, 35-45, 1976.

[HAN78] A.R. Hanson and E.M. Riseman, "Segmentation of Natural Scenes," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[HAR73] R. Haralick, K. Shanmugan, and I. Dinstein, "Textured Features for Image Classification," IEEE Trans. on Systems, Man and Cybernetics, SMC-3, 610-621, Sept. 1974.

[HEN75] G.G. Hendrix, "Expanding the Utility of Semantic Networks Through Partitioning," Proc. IJCAI-4, 115-121, August 1975.

[HEW68] D. Hewitt, "PLANNER: A Language for Manipulating Models and Proving Theorems in a Robot," MIT Project MAC, AI Memo 168, 1968.

[HUF71] D.A. Huffman, "Impossible Objects as Nonsense Sentences," in Machine Intelligence 6 (B. Meltzer and D. Michie, Eds.), Elsevier, 295-323, 1971.

[KAN77] T. Kanade, "Model Representation and Control Structures in Image Understanding," Proc. IJCAI-5, Cambridge, MA, August 1977.

[KEN77] J. Kender, "Instabilities in Color Transformations," Proc. of Conf. on Pattern Recognition and Image Processing, Troy, NY, 266-274, June 1977.

[KON75] K. Konolige, "The ALISP Manual," Univ. Computing Center, University of Mass., August 1975.

[KON77] "The ALISP Relational Database, COINS Technical Report 77-9, University of Mass., November 1977.

[LES77] V.R. Lesser and L.D. Erman, "A Retrospective View of Hearsay-II Architecture," Proc. IJCAI-5, Cambridge, MA, August 1977.

[LEV78] M.D. Levine, "A Knowledge-Based Computer Vision System," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[LIE74] L. Liebermann, "Computer Recognition and Description of Natural Scenes," Ph.D. Thesis, University of Pennsylvania, Philadelphia, June 1974.

[LOW76] B.T. Lowerre, "The Harpy Speech Recognition System," Tech. Report (Ph.D. Thesis), Carnegie-Mellon University, 1976.

[LOW78] J. Lowrance, "GRASPER Reference Manual," COINS Tech. Report, University of Mass., in preparation.

[MAC76] A.K. Mackworth, "Model-Driven Interpretation in Intelligent Vision Systems," Tech. Report 76-2, Dept. of Computer Science, University of British Columbia, June 1976.

[MAC78] A.K. Mackworth, "Vision Research Strategy: Black Magic, Metaphors, Mechanisms, Miniworlds, and Maps," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[MAR76a] D. Marr and H.K. Nishihara, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," MIT AI Memo 377, August 1976.

[MAR76b] D. Marr, "Analysis of Occluding Contour," MIT AI Memo 372, October 1976.

[MAR78] D. Marr, "Representing Visual Information," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[McC69] J. McCarthy and P.J. Hayes, "Some Philosophical Problems From the Standpoint of Artificial Intelligence, in Machine Intelligence 4 (B. Meltzer and D. Michie, Eds.), University of Edinburgh Press, 463-503, 1969.

[McD74] D. McDermott and C. Sussman, "The CONNIVER Reference Manual," MIT Memo 259a, January 1974.

[MIN75] M. Minsky, "A Framework for Representing Knowledge," The Psychology of Computer Vision (P. Winston, Ed.), McGraw-Hill, 211-277, 1975.

[NEI76] U. Neisser, Cognition and Reality: Principles and Implications of Cognitive Psychology, W.H. Freeman and Company, 1976.

[NEV74] R. Nevatia, "Structured Descriptions of Complex Curved Objects for Recognition and Visual Memory," Stanford AI Lab Memo AIM-250, October 1974.

[NEV78] R. Nevatia, "Characterization and Requirements of Computer Vision Systems," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[NIL71] N.J. Nilsson, Problem Solving Methods in Artificial Intelligence, McGraw-Hill, New York, 1971.

[PIA71] J. Piaget, Biology and Knowledge: An Essay on the Relations Between Organic Regulations and Cognitive Processes, Edinburgh University Press, 1971.

[PRA71] T. Pratt and D. Friedman, "A Language Extension for Graph Processing and Its Formal Semantics," Communications of the ACM, 4, 1971.

[QUI68] R. Quillian, "Semantic Memory," Semantic Information Processing (M. Minsky, Ed.), MIT Press, 1968.

[RED78] D.R. Reddy, "Pragmatic Aspects of Machine Vision," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[RIS74] E. Riseman and A. Hanson, "The Design of a Semantically Directed Vision Processor," COINS Technical Report 74C-1, University of Massachusetts, January 1974.

[ROS71] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEE Trans. Computers, 562-569, 1971.

[RUB77] S.M. Rubin and D.R. Reddy, "The LOCUS Model of Search and Its Use in Image Interpretation," Proc. IJCAI-5, Cambridge, MA, August 1977.

[SAC75] E.D. Sacerdoti, "The Non-Linear Nature of Plans," IJCAI-4, Tbilisi, USSR, September 1975, 206-214.

[SCH77] R.C. Schank and R.P. Abelson, Goals, Plans, Scripts and Understanding: An Enquiry into Human Knowledge Structures, Erlbaum Press, NJ, 1977.

[SHA77] R. Shapira and H. Freeman, "Reconstruction of Curved-Surface Bodies From a Set of Imperfect Projections," Proc. IJCAI-5, Cambridge, MA, August 1977, 628-634.

[SHI78] Y. Shirai, "Recognition of Real-World Objects Using Edge Cues," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[SHO75] E.H. Shortliffe and B.G. Buchanan, "A Model of Inexact Reasoning in Medicine,"

Mathematical Biosciences, 23, 351-379, 1975.

[SIM73] R.F. Simmons, "Semantic Networks: Their Computation and Use for Understanding English Sentences," in Computer Models of Thought and Language (R.C. Schank and K.M. Colby, Eds.), H. Freeman and Co., 1973.

[TEN73] J. Tenenbaum, "On Locating Objects by Their Distinguishing Features in Multisensory Images," SRI Technical Note 84, AI Center, Stanford Research Institute, September 1973.

[TEN76] J.M. Tenenbaum and H.G. Barrow, "Experiments in Interpretation-Guided Segmentation," Technical Note 123, AI Center, Stanford Research Institute, 1976.

[TUR74] K.J. Turner, "Computer Perception of Curved Objects Using a Television Camera," Ph.D. Thesis, Dept. of Machine Intelligence, School of Artificial Intelligence, University of Edinburgh, 1974.

[UHR78] L. Uhr, "Recognition Cones and Some Test Results," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[WAL77] D.E. Walker, W.H. Paxton, et al., "Procedures for Integrating Knowledge in a Speech Understanding System," Proc. IJCAI-5, Cambridge, MA, August 1977.

[WAL75] D. Waltz, "Understanding Line Drawings of Scenes with Shadows," in The Psychology of Computer Vision (P.H. Winston, Ed.), McGraw-Hill, 19-91, 1975.

[WEC75] H. Wechsler and J. Sklansky, "Automatic Detection of Contours of Ribs in Chest Radiographs," Univ. of California at Irvine, TR-75-2, 1975.

[WIL77] T. Williams and J. Lowrance, "Model-Building in the VISIONS High Level System," COINS Technical Report 77-1, University of Mass.. January 1977.

[WIL78] T. Williams, forthcoming COINS technical report, University of Massachusetts, 1978.

[WIN75] P.H. Winston, The Psychology of Computer Vision, McGraw-Hill, 1975.

[WOO77] W.A. Woods, "Final Report on Speech Understanding Systems," Bolt, Beranek and Newman, Inc., Cambridge, MA, 1977.

[WOO78] W.A. Woods, "Theory Formation and Control in a Speech Understanding System with Extrapolation Towards Vision," in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, New York, 1978.

[YAK73] Y. Yakimovsky and J.A. Feldman, "A Semantics-Based Decision Theory Region Analyzer," Proc. IJCAI-3, 580-588, August 1973.

[YOR78a] B. York, "Shape Representation in the VISIONS System," COINS Technical Report, University of Massachusetts, in preparation.

[YOR78b] B. York, "Symbolic Classification of Primitive Two-Dimensional Shapes," COINS Technical Report, University of Massachusetts, in preparation.