A PROGRESS REPORT ON VISIONS: REPRESENTATION AND CONTROL IN THE CONSTRUCTION OF VISUAL MODELS

Allen R. Hanson
Edward M. Riseman

COINS TECHNICAL REPORT 76-9

July 1976

COMPUTER AND INFORMATION SCIENCE

UNIVERSITY OF MASSACHUSETTS AT AMHERST

AMHERST, MASSACHUSETTS 01002

U.S.A.

A PROGRESS REPORT ON VISIONS: REPRESENTATION AND CONTROL IN THE CONSTRUCTION OF VISUAL MODELS

Allen R. Hanson*
Edward M. Riseman+

COINS TECHNICAL REPORT 76-9

July 1976

This research was supported by the Office of Naval Research under Grant N00014-75-C-0459, and the National Science Foundation under Grant DCR75-16098.

*School of Language and Communication
Hampshire College
Amherst, MA 01002

+Computer and Information Science
University of Massachusetts
Amherst, MA 01002

ABSTRACT

This report is an interim progress report on the evolving structure of VISIONS, a computer system for general visual perception. The goal of the system is the segmentation and interpretation of a digitized color image of natural outdoor scenes. We outline the multi-level data structures used for representing both a visual model of the scene and the semantic data base of stored knowledge about the world. A flexible modular strategy controls the operation of processes which embody diverse forms of knowledge, and allows both data-directed and knowledge-directed model building. A model search space is used to store a sketch of the processing history during model formation, so that limited, directed backtracking will be facilitated.

A symbolic data structure (RSE for Regions, line Segments and Endpoints) interfaces the results of low-level segmentation processes with the interpretation processes which form hypotheses about surfaces, objects, and frames of visually familiar situations. The RSE structure represents syntactic two-dimensional image information while the three higher levels of representation organize semantic concepts in three-dimensional space. Utilization of the RSE structure decomposes the development of the low-level and high-level systems; it provides a clear statement of the requirements imposed on the low-level segmentation processes, and delineates the form of the data which will be the input to the high-level processes. The summary contains a discussion of the major design goals of VISIONS.

Two forthcoming reports will supplement this paper. The low-level system is only briefly discussed here but will be treated in more detail in [1], while further details of the model builder will be provided in [2].

TABLE OF CONTENTS

I. OVERVIEW

This paper represents a brief overview of a general computer-based visual perception system, called VISIONS [1-7], currently under development by our group. The system is being designed to function in the real world environment of full-color natural outdoor scenes. The structure of VISIONS is quite complex and the ideas discussed here will only be an outline of the many representations and subsystems that are being developed in detail; the paper is meant to provide an introduction to and progress report on the evolving structure of VISIONS.

Figure 1 represents the global organization of VISIONS. The system divides roughly into two major subsystems:

a) low-level processes whose goal is the segmentation of the image into regions representing (major parts of) conceptual objects, and extraction of a set of visual features associated with each region;

and b) high-level processes whose goal is the construction of a conceptual model of the three-dimensional world represented in the scene; this involves the use of a semantic data base, expectations about the scene provided by context, deductive mechanisms, analyses of perspective, occlusion, shadows, a representation of shape and a spatial processor for manipulating volumes and surfaces in space, a flexible modular strategy which controls the utilization of the available processes, and a model search space where a sketch of the processing history during model formation is maintained for limited and directed backtracking.

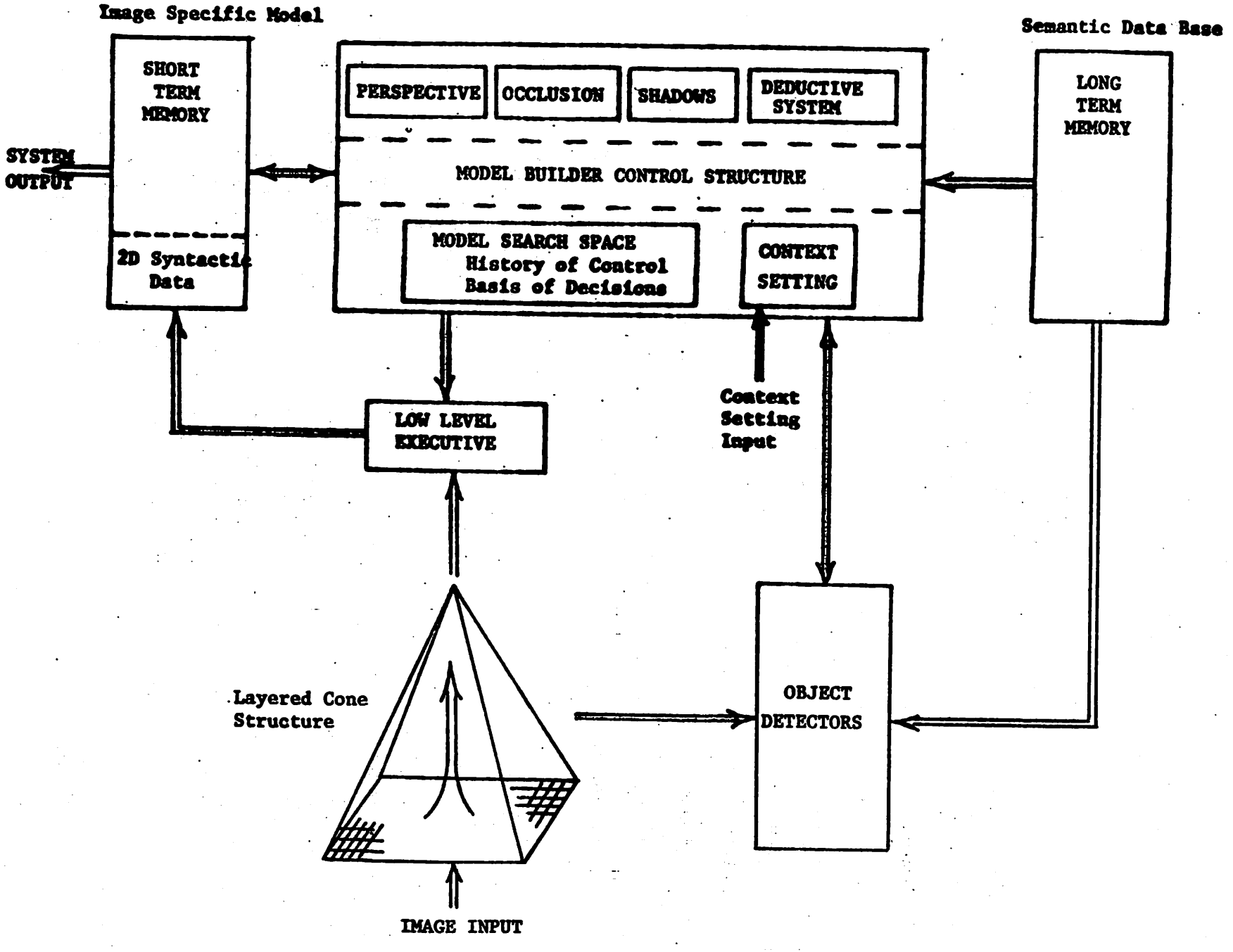


Figure 1 Overview of the VISIONS System

The goal of the high-level system is the construction of a model which describes the major conceptual entities and three-dimensional space of the scene under consideration. Both the image model and world knowledge contain information represented at different levels of symbolic abstraction (refer to Figure 2), similar to that described in the HEARSAY speech understanding system [8]. Again, this portion of the system may be divided into several structures:

a) Image Model (short term memory--STM): The image-specific model is formed as a multi-level network (directed graph). This consists of the information contained in the six planes on the left hand side of Figure 2. The bottom three planes (RSE) represent two-dimensional syntactic information derived from the segmentation process and stored symbolically. The upper three planes represent the semantic interpretation and definition of three-dimensional space.

b) World Knowledge (long term memory--LTM): This contains the general world knowledge in the same multi-level representation as the image model and is depicted on the right hand side of Figure 2. Pointers from the left side to the right side provide the linkage between the image-specific entities and the general classes to which they belong; pointers into the range of allowable values for attributes can be used to specify a particular attribute value of an image-specific entity.

c) Control Processes and Knowledge Sources: These are the functions which are responsible for the formation of hypotheses on the upper

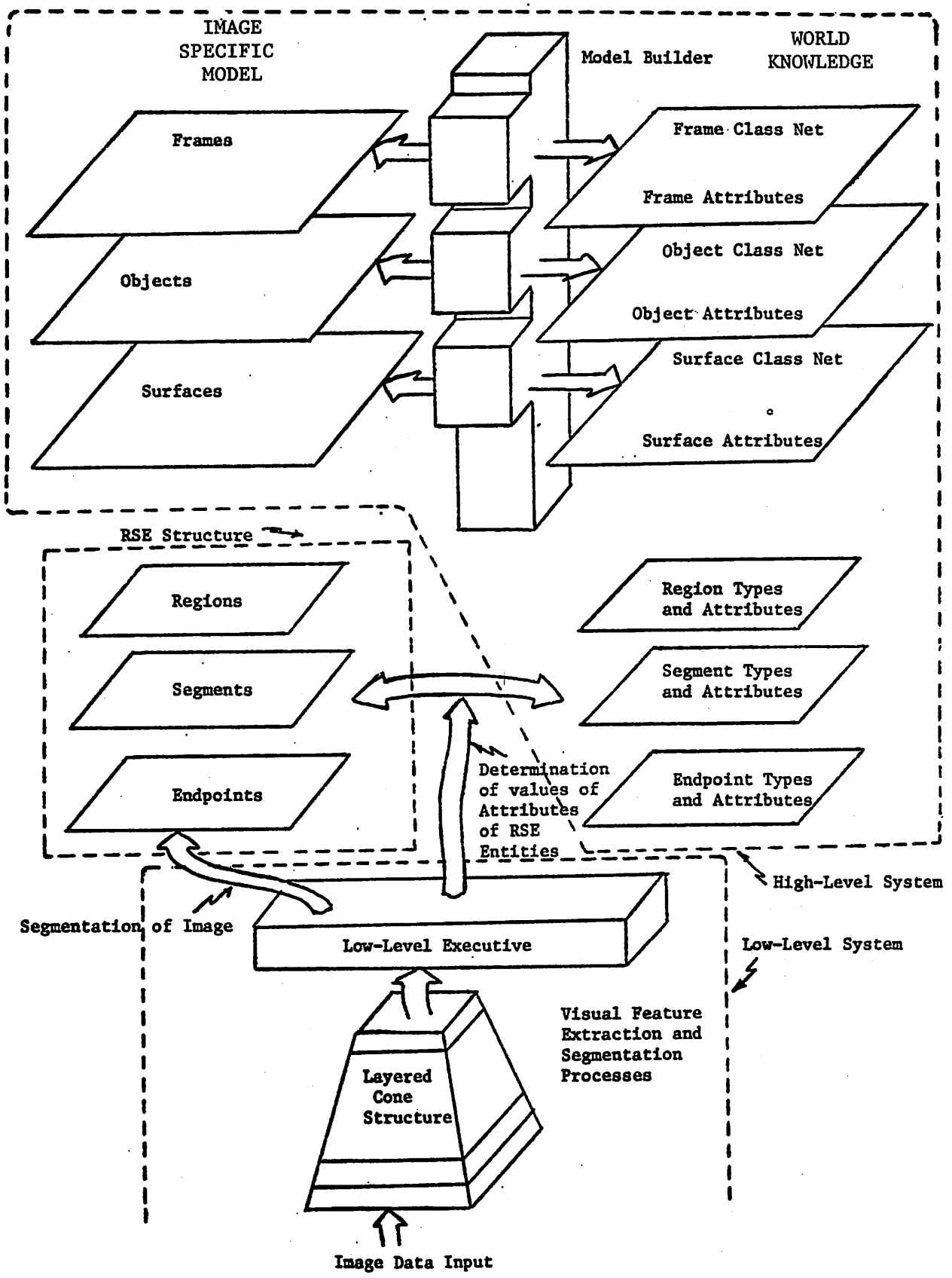


Figure 2 Levels of Representation

three planes of the model. They utilize the symbolic information in the RSE planes, contextual expectations concerning the image, and other hypotheses stored in the partially formed model. All entities in the model are linked to the relevant concepts in world knowledge as the model is formed.

The ideas presented in this paper address many of the issues in scene analysis and AI--the use of multiple knowledge sources, multiple levels of representation of knowledge, local and global processing, serial and parallel processing, model-directed analysis, backtracking, etc. (refer to Summary). One of the most important issues is effective control of the processes responsible for formation of hypotheses concerning the scene. In VISIONS, flexibility of control is afforded by both bottom-up and top-down analyses. For example, local data-directed analysis (bottom-up) might call for a hypothesis for the identity of a region on the basis of its visual features (color, texture, size, shape, etc.), while the instantiation of a frame (a representation of a familiar scenario) would allow knowledge-directed analysis (top-down) to provide global direction in setting up expectations about the context of the scene.

The high-level system is currently being developed utilizing a LISP base, the context trees of CONNIVER[22], and modifications of GRASPE [28] a graph processing language. These tools contain the necessary mechanism to support the model building process that we describe.

II. THE LOW-LEVEL SYSTEM

The low-level system further subdivides into two components: the processing cone [1,4] and the low-level executive. The processing cone depicted in Figure 3 is a simulation of a parallel array computer that is hierarchically organized into a layered system. It is meant to provide a general computational structure for the analysis of visual data in both numeric and symbolic form. Its major function is the transformation and reduction of the large amounts of data normally found in digitized images.

Information flows up, down, and laterally within the cone by defining a function to be applied at time t to a local window at a given level of the cone. This function is applied simultaneously in parallel to local windows across the entire array.

There are three major types of processing operations in the cone. During a reduction process upward through the layers in the cone, the data is reduced because portions of each window are nonoverlapping. An iteration process allows the data to be analyzed and/or transformed at a fixed level of the cone; the size of the array remains constant due to overlapping of windows. A projection process allows information in upper layers to influence computation in lower layers.

Algorithms for forming edges and grouping them into lines, region growing, texture analysis, and color mappings, among others, have been defined as sequences of parallel operations applied up and down the levels in the conical structure. In the cone, the results of these parallel operations are stored in pseudo-image arrays which are also available for further processing by local operators. Many of these algorithms produce segmented or transformed images at various levels of

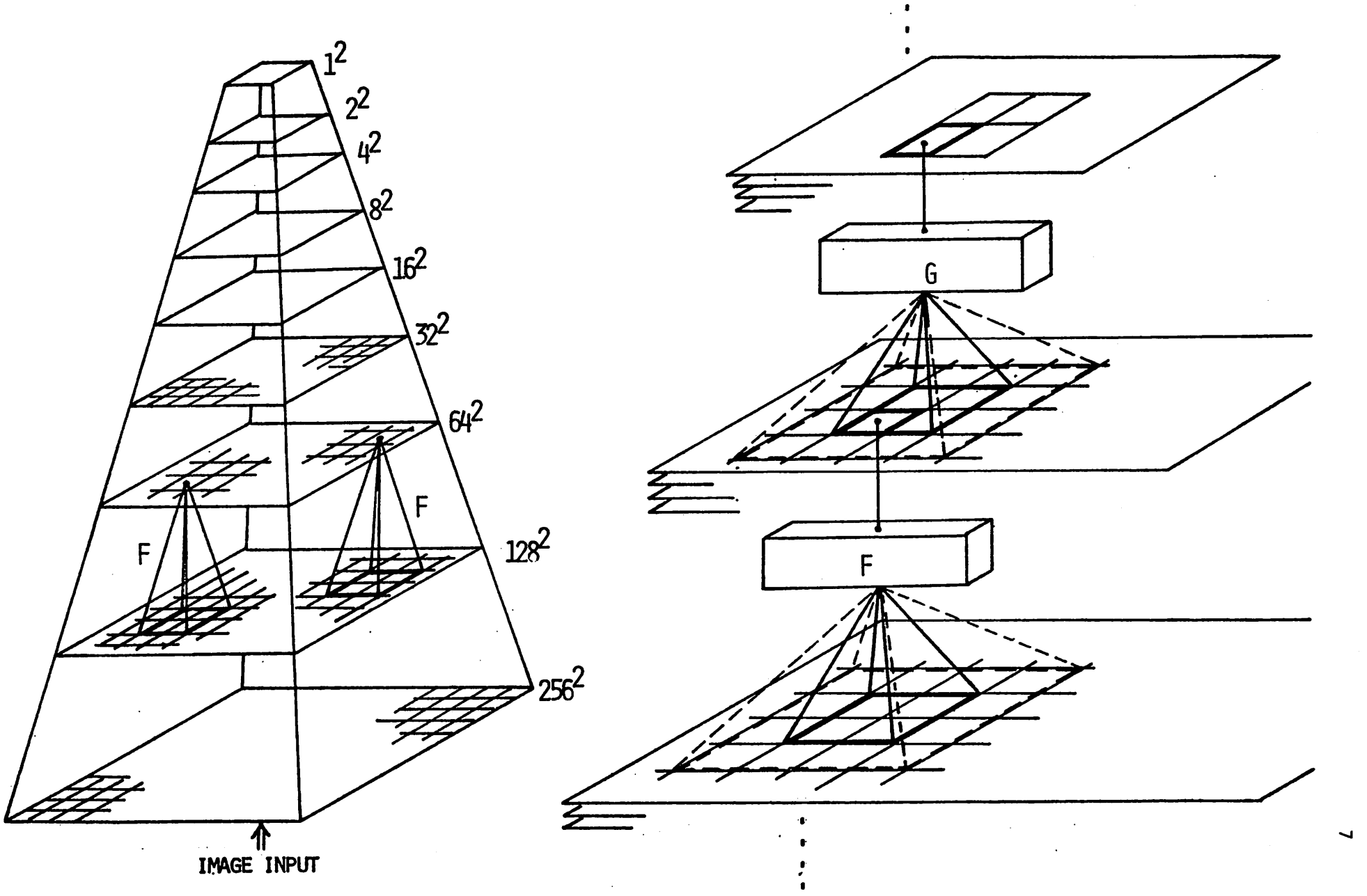


Figure 3 Processing Cones

resolution--they operate on the 256^2 grid of image data and can reduce it, layer by layer, up to a single cell which contains information extracted from the entire scene. For related work on hierarchical structures, see Kelly [9], Rosenfeld and Thurston [10], Uhr [11], Klinger and Dyer [12], and Tanimoto and Pavlidis [13].

The idea of a general-purpose low-level system, which understands the environment at different levels of description, has recently been discussed by Zucker, Rosenfeld and Davis [14]. It has become apparent to us that no single low-level process is robust enough to provide the information required for reliable segmentation. This implies that a system of well-coordinated algorithms is needed which is effective in extracting different types of information. The nucleus of the low-level system, the executive, will be responsible for control of the algorithms, including the dynamic selection of features upon which boundary and region analyses are based, the order in which they are invoked, cooperation between them, an understanding of the strengths and weaknesses of each for resolution of conflicts, etc. Since the processes are somewhat redundant and produce alternative forms of the same data, they provide the executive with the ability to determine consistency of analysis. This stage of the system is still in the midst of development and will be reported at a later time.

In the following discussion, we assume that the organization described has allowed textured regions to be processed reliably, i.e., that a textured area may be represented as a single region with global texture descriptors.

III. INTERFACING THE HIGH- AND LOW-LEVEL SYSTEMS

III.1 Symbolic Representation of Segmentation Results

What should be the output of a low-level system? How should the low-level system communicate with the high-level system? If one considers the process of image understanding as a transformation (or series of transformations) from the numeric data representing the sensory stimulation to a symbolic structure representing a model of the world as found in the image, then at some point numeric entities must be mapped to symbolic entities and structures. The question to be decided, then, is where in the transformation this conversion takes place and how it is accomplished.

Marr [1975] has recently suggested that low-level vision in humans moves onto the symbolic level at a very early stage, prior to the initiation of semantic processes. He believes that high-level analysis only has access to a "primal sketch"--symbolic descriptions of lines, edges, shadows, blobs, and abstract groupings of these elements (including the "places" defined by these primitive features).

The advantage of symbolic communication for computer vision is that the vast amounts of data in the intensity arrays do not have to be continually referenced. They can be compacted and easily examined by semantic processing structures. Of course the symbolic storage of every faint edge, gradient, and shadow can itself be overwhelming so there is still a need for suppression of detail, whose loss can be compensated for by allowing feedback to attention mechanisms to extract further foveal detail when desired.

Grouping operations and other transformations should be applied to symbolic entities spatially close to each other in the

2D image. Thus, it seems clear that although one may operate on symbols, maintenance of this information in a data structure which preserves spatial relationships (such as the processing cones) is highly desirable.

From the point of view of building effective vision systems, there is some question as to whether it is desirable to move immediately to the symbolic level. A value for a parameterized feature on some scale might serve better than some symbol. One might operate on numeric values and move to a symbolic level after some degree of processing, so that the amount of symbolic data is of manageable proportions. A textured region would have one label and a set of descriptors, rather than the symbolic representation of each texture element of the textured region. If the texture element is of high interest, it can be focussed upon as a subimage itself.

We have chosen a symbolic representation for macro-segmentation results for two main reasons. The first is that it formally separates the segmentation process from the interpretive processes. A flexible symbolic data structure will provide a formalism in which the results of segmentation algorithms can be collected. At the same time it defines the input to the semantic processes. Thus, as a methodology for system development, one gains a decomposition of two major systems which allows independent implementation of each, yet insures that they can be compatibly interfaced.

The second reason for the symbolic representation is that very different representations and techniques are called for in the low- and high-level systems. Semantic processes generally operate on symbols--not numbers--and

the symbolic naming of the primitive visual entities facilitates communication between them. This means that the processed visual information can be stored in a form similar to the semantic data base. This does not preclude feedback from internal models to the feature extraction stage--it just structures this communication.

III.2 The RSE Structure

It appears that the minimal information that must be contained in a symbolic representation of a segmented image is the labelling of distinct regions and boundaries that we have been taking great pains to extract. If it allows regions and lines to be consistently represented, then we have a single data structure which accepts the results of many different algorithms and provides effective communication between them. An algorithm that finds boundaries can check whether they fit with regions that may have been extracted by utilizing a (possibly) different set of features. This redundancy may detect errors and direct the invocation of more powerful, but computationally costly, processes in a selective manner.

Our syntactic representation of 2D information is stored in three planes--region, line segments, and endpoints. It is very important to make the relationships between these image elements easily accessible. There is no reason to expect that regions and objects are in a one-to-one relationship; objects are usually composed of several regions. Thus, it is necessary to know which segments bound a region and which regions are adjacent to each other.

A convenient representation is a directed graph broken into layers of nodes; the nodes represent names of the entities on each plane, and the unlabelled bidirectional arcs between planes represent key two-dimensional, spatial relationships between them. This representation is interesting in that relationships between nodes at one level appear as a node at the next level of representation with its own descriptors and relationships. The fundamental relationship of adjacency of regions is implicitly available as a line segment node; it is represented by an arc from each region node to their common line segment node. Thus, a region is defined by the set of line segments which form its boundary, while a line segment is defined by a pair of adjacent regions, unless it is a non-bounding line segment contained in a single region. Line segments are anchored in two space by the position of their endpoints. Thus, the concatenation of line segments can be represented by arcs to their common endpoint node. If regions must later be split or joined, this representation affords the flexibility for redirecting a few pointers to update the low-level visual data.

Figure 4 is an illustrative example. The R plane projects down upon the S plane which in turn projects down upon the E plane. In particular, regions are enclosed by segments (and can, in turn, enclose non-bounding line segments, such as S_9). Note that R* is a special region which includes everything outside the picture and therefore points to the line segments on the boundary of the image.

If a segment is not enclosed by a region, then it separates two (and only two) regions; this follows by definition because if a boundary separates more than two regions, it will be partitioned into segments such that each

segment separates exactly two regions. In this example we have further subdivided a line segment into segments which are straight when this property is applicable (except in the case of the picture boundary where they are artificial). Obviously other properties could be used to subdivide line segments also. Also note that segments such as S_5 and S_9 which are entirely contained in a region will be called isolated segments and are immediately apparent because only a single region node points to them.

Segments are specified by their two endpoints. Segments which meet will have a common endpoint, allowing any connected sequence of segments to be extracted. Closed line segments have no endpoints; therefore, it is useful to add an arbitrary starting point so that the line can be fixed in two space.

It is crucially important to realize that only a limited subset of the possible two-dimensional spatial relationships between regions and lines are being used to form the logical structure of this layered network, namely the adjacency of regions mapping onto segments and the connectedness of segments mapping onto endpoints. There are many other relationships that can be represented and extracted as explicitly labelled directed arcs between the primitive visual entities. For example, containment of region R_2 by region R_1 can be represented by a "C" arc from R_1 to R_2. In this fashion the syntactic graph representation can be enriched with any further relationships that the user might define, such as arcs between parallel line segments or endpoints near each other.

A set of properties and values can be hung off each region or line segment node and made accessible to the semantic system which is examining the information. In particular the cone algorithms are to extract visual

features of regions and boundaries and dump them into the symbolic structure. These descriptive properties might include:

regions -- hue, saturation, intensity, texture, location, size, shape, orientation, etc.;

line segments[1] -- location, quality (straight, curvilinear, or various characteristics of irregularity), width of gradient, orientation, etc.; and

endpoints -- location, type of vertex such as the polyhedral fork, arrow, T, etc.

The model builder begins the model inference process with the RSE structure, but the RSE structure need not be static. The complex of nodes and arcs can change over time as the low-level system extracts further information from the image data. Given the manner in which the segmentation algorithms will operate in the processing cones, there could be an RSE structure of each level of resolution in the cone. This would allow the dynamic development of RSE under hierarchical direction. Segmentation and RSE results at higher levels of the cone would direct and refine segmentation and RSE at lower levels.

For the reader who believes that this symbolic representation must surely produce an unwieldy mass of data, Figure 5 gives an example of a fairly complex road scene. It was produced by hand from an actual 35 mm slide of the image by blurring the slide and tracing the boundaries of the

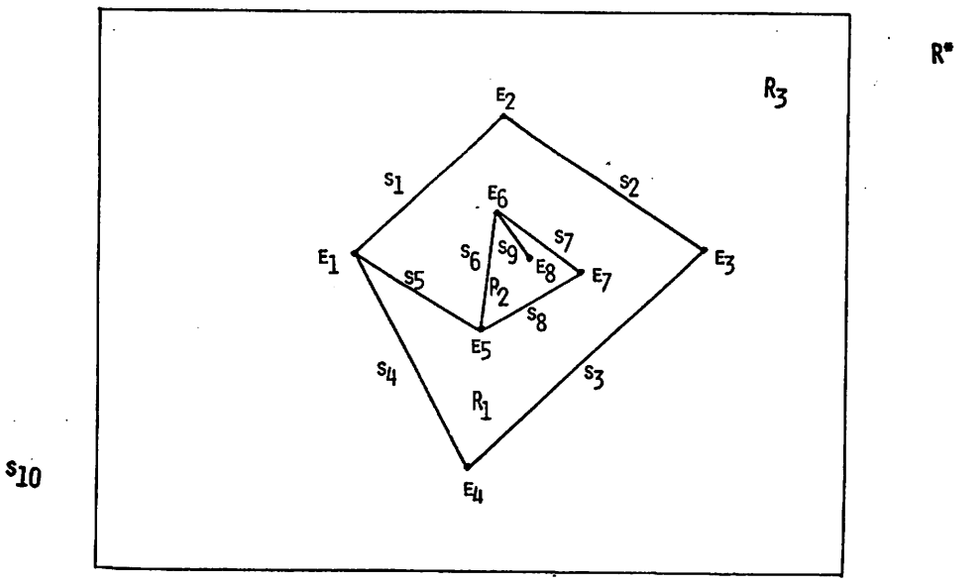
---

[1] The chain encoding [15] of a line segment on a rectangular grid might be stored so that further extraction of properties can be carried out later. This can be done at different resolution levels so that a jagged line which globally is straight would have the local and global properties consistent and accessible. Of course the chain encoding at a coarser resolution can be obtained directly from the chain encoding at a fine resolution.

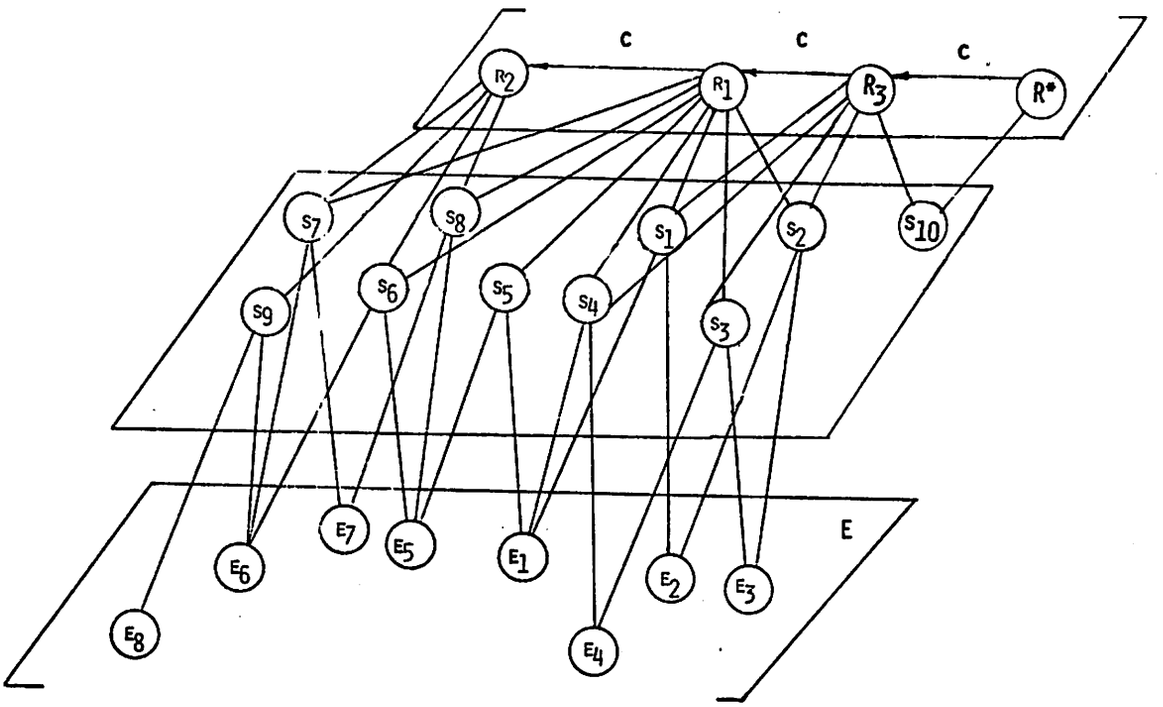
major regions. Each region has been labelled (R_1 - R_{19}) as has each line segment (S_1 - S_{44}), including the picture boundaries. A collapsed representation of R-S and S-E planes is shown in Figure 6(a) and (b), respectively. Note that the road scene has had much of the finer textural characteristics processed and grouped in the tree regions of R_1 and R_4, and the grassy region of R_6. Otherwise this representation might produce a bewildering mass of information.

III.3 Advantages of RSE for Segmentation and Description

It is worth pointing out the advantage of our region-segment definition, both for the low-level segmentation processes which must extract the line segment, and for the facilitation of determining adequate descriptors of the line segments. If local edges which are part of a common boundary are to be grouped into distinct line segments, then some criterion of similarity is needed. In addition to spatial proximity and orientation, the similarity of edge strength is a strong cue for edge grouping. However, the regions surrounding any given region are bound to have different properties. Therefore, no matter upon what feature the strength of the gradient is based, one must expect widely varying values as the boundary of a single region is tracked. Hence, the argument for forming line segments each of which lies between no more than one pair of regions. Local edges can be expected to exhibit characteristics which have less variance. In fact comparison of



(a)



(b)

Figure 4 Simple Example of RSE Structure

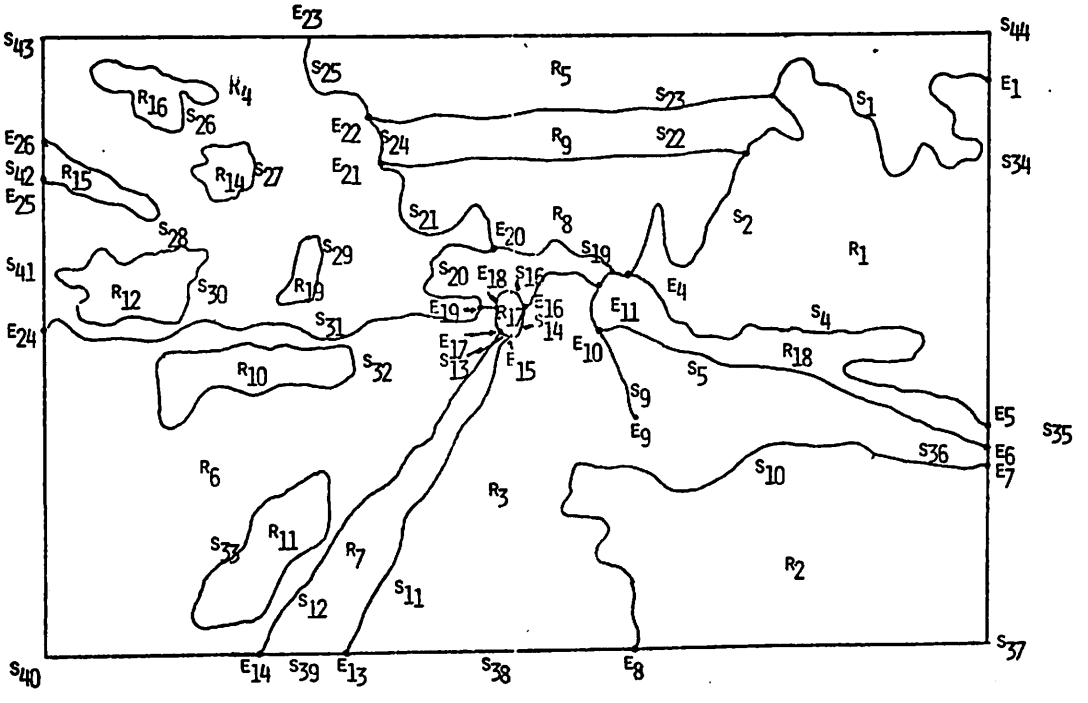
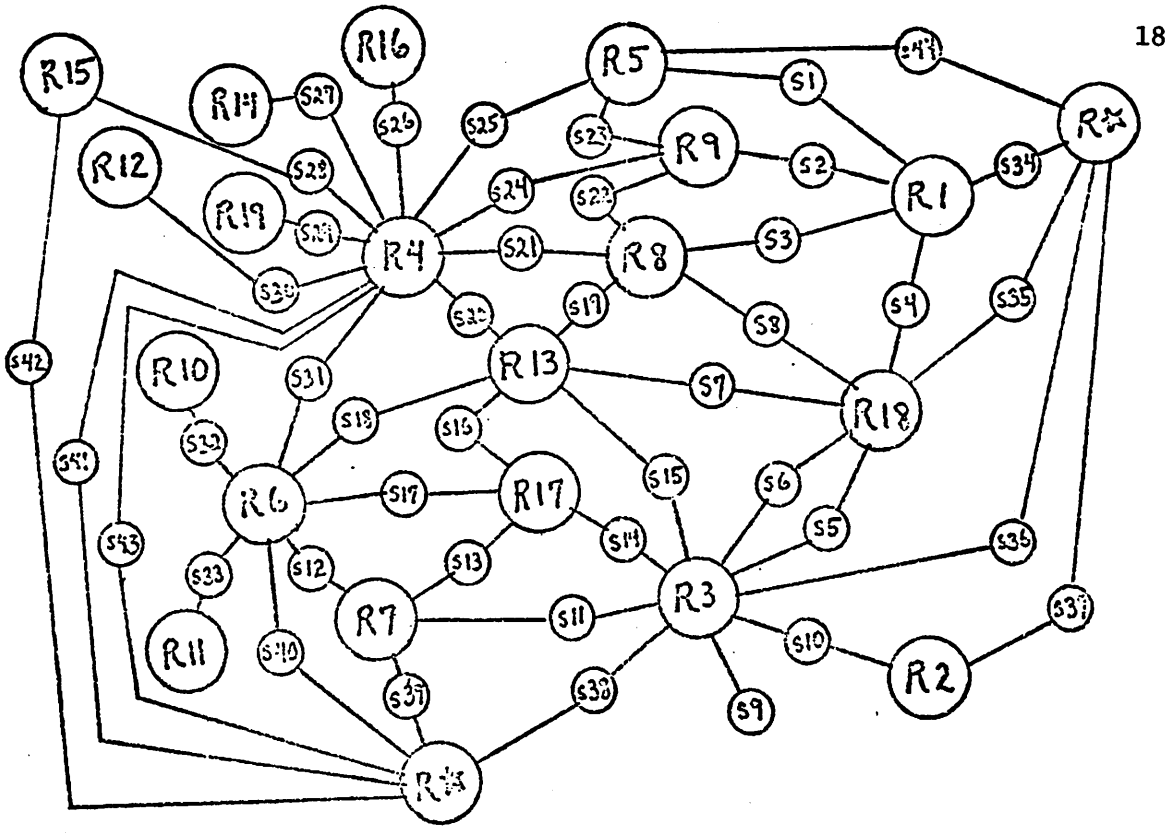
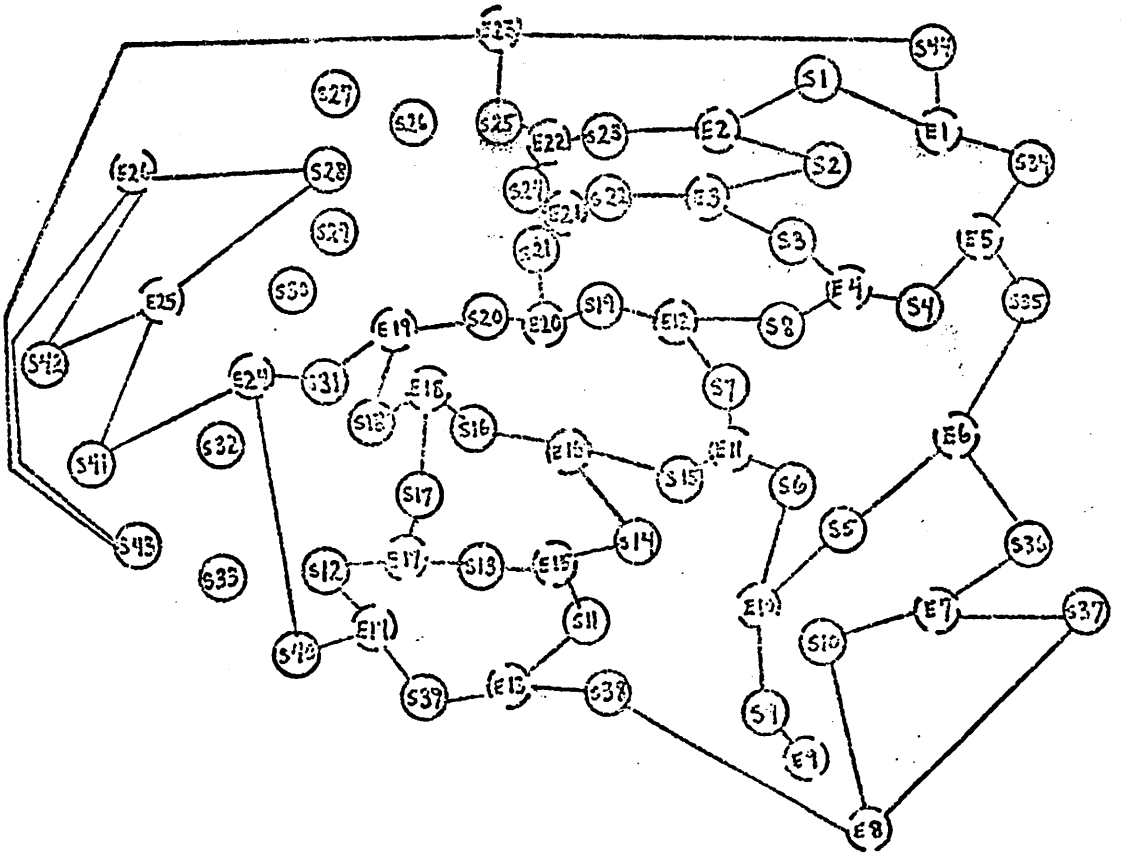


Figure 5 Example Road Scene



(a) The Collapsed R-S Plane of the Road Scene



(b) The Collapsed S-E Plane of the Road Scene

Figure 6

features of the regions to either side of a pair of adjacent edges can be very useful in directing the edge binding process [16].

The extraction of line segment descriptors is also facilitated. Occlusion of one object by another (see Figure 7) causes the boundary of an object to vary depending on which of the regions represents the occluding surface. One cannot easily provide a single descriptor for the entire boundary of region R_1 (irregular for one portion and two straight line portions at different orientations) unless it is subdivided. By defining line segments as we have, it is more likely that simple features will be adequate descriptors for each segment.

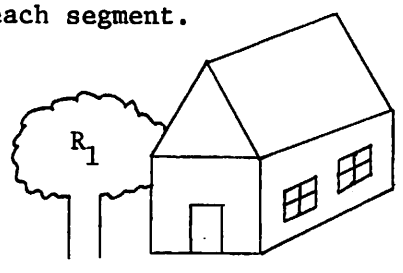


Figure 7 Characteristics of the boundary of the tree region change due to the occlusion by the house; a single descriptor of the boundary of R_1 is not meaningful.

III.4 Operations on the Symbolic RSE Structure

The purpose of such a symbolic representation of the visual syntax is to allow other procedures flexible access and manipulation of this processed information. Referring to our simple example in Figure 5, let us describe a few operations which illustrate the manner in which the RSE data structure will be utilized for model building.

It is assumed that the segments bounding a region, the regions about a segment, and the corresponding relationships for the S-E plane are primitive relationships, i.e. the results are easily accessible by following the proper arcs in the graph. Some operations are:

(1) determination of the common boundary of two regions - for regions R_1 and R_2, intersect the two sets of segments which they point to on the S plane; thus,

$$\{S_1,S_2,S_3,S_4,S_5,S_6,S_7,S_8\} \wedge \{S_6,S_7,S_8,S_9\}$$

yields the common boundary $\{S_6,S_7,S_8\}$;

(2) determination of the set of regions adjacent to a given region - for region R_2, obtain its non-isolated segments on the S plane $\{S_6,S_7,S_8\}$; trace back to the R plane all regions other than R_2 which point to these segments and form the set $\{R_1\}$ as the adjacent region set; note that each non-isolated line segment is recognized because it has exactly one other region node pointing to it; and

(3) determination of the connected boundary around a region - for region R_2, select one of its non-isolated line segments on the S plane, say S_6, and then select one of its endpoints, say E_6; trace back from E_6 to another of its line segments which is not isolated, in this case S_7 (note that selection of S_9 will require backtracking when it is found to be an isolated segment); follow its pointer down to its next endpoint E_1 and repeat the process until the boundary closes on itself back at S_6.

There are many other simple operations that are straightforward. The comparison of properties of adjacent regions is available from the descriptors of the region nodes obtained in (2) above. Line dominance cues are available by comparing the properties of the line segments entering an endpoint node (refer to Fig. 7).

Sometimes a line has a break in it allowing two regions to be incorrectly merged. Since endpoints have a location descriptor, arcs on the E plane can be formed between pairs of endpoints on the basis of a distance threshold. Then hypotheses for extending or inserting line segments can be examined for implications at a semantic level. Certainly feedback to segmentation algorithms can provide them with a focus of attention and increased sensitivity for examining the visual evidence, possibly quite weak, to confirm these hypotheses.

The operation of determining region containment is quite messy in the RSE structure, because it is locally unclear whether R_2 contains R_1 which contains R_3, or R_3 contains R_1 which contains R_2. One must recursively trace out to the picture boundary and R* to determine the correct case. Consequently we assume that containment relationships are computed once and C arcs placed in the R plane. Once C arcs are available, it is easy to extend operation (3) above to determine inner and outer boundaries for regions which contain other regions.

One must also face the problem that segmentation errors undoubtedly will occur. This will require that regions be merged to form larger regions, and the messier problem of later splitting one region into several regions. However, this should only require locally redirecting the pointer structure to reflect those changes.

IV. SEMANTIC REPRESENTATION OF THE MODEL AND WORLD KNOWLEDGE

The upper three planes of the image-specific data structure represent the meaning of the visual data. They define space in terms of surfaces (such as the planar surface of the ground), objects (tree), and frames as an embodiment of familiar scenarios or submodels (road scene, house scene). Here, all the entities serve to define the relationship of semantic components and map down upon the regions which demarcate the visible portions of these entities. These "conceptual grouping" mappings go from frames to objects to surfaces to regions, etc.; however, the inverse mappings are also available since the arcs are bidirectional.

IV.1 Frames

Our concept of a frame is similar to that of Minsky [23], although it is simplified and tailored particularly to vision. A frame defines a conceptual grouping of objects and their relative positions in three space. The frame of a road scene will describe the relationships[2] between the road, cars, guard rails, telephone poles, etc. The frame will specify objects or surfaces, their importance to the instantiation of the frames, and any other information useful to guide the model building process. The frame must also point to related frames, just as objects in a semantic net have access to related objects. Objects appearing in a frame may also have their own frame specification. A tree may be

[2] The relationships of interest for our immediate purposes are primarily spatial, but generally we expect spatial, functional, and temporal relationships to be stored in a frame.

part of a road scene frame and be treated as an object with attributes. On the other hand, a more detailed description of the parts of a tree require spatial definition and relationships. Thus, "tree" can be treated as an object or a frame during model building. Long-term memory may then be treated as a network of frames and objects in which many of the entities may be dealt with in either way by the remainder of the system.

The relative importance of particular objects and relationships to the frame can be captured as weights. Weights from objects to frames (upward weights) rate the likelihood that the presence of objects implies the presence of the frame, while weights from frames to objects (downward weights) store the likelihood that the presence of the frame implies the presence of the object. Guard rails strongly imply the road scene frame, while the road scene frame must have guard rails as optional since their absence is to be expected in some cases. Of course this is a crude approximation to the dependencies between objects in a frame. However, it is clear that accurate estimation of the joint distribution of all the subsets of the objects in a road scene is not feasible. Thus, we approach this problem heuristically with intuitively selected weights, and we will deal with the problem of tuning, dependencies, and adaptation later.

In the model an instantiated frame of a road scene provides access to other frames which might also be instantiated as part of the model. It must also project down upon the object plane, specifying the objects of this frame which are <u>actually</u> in the scene. Thus, the model builder uses the frame to guide the analysis of the picture in terms of sets of related objects with dependent relationships, rather than process each individual object relatively independently.

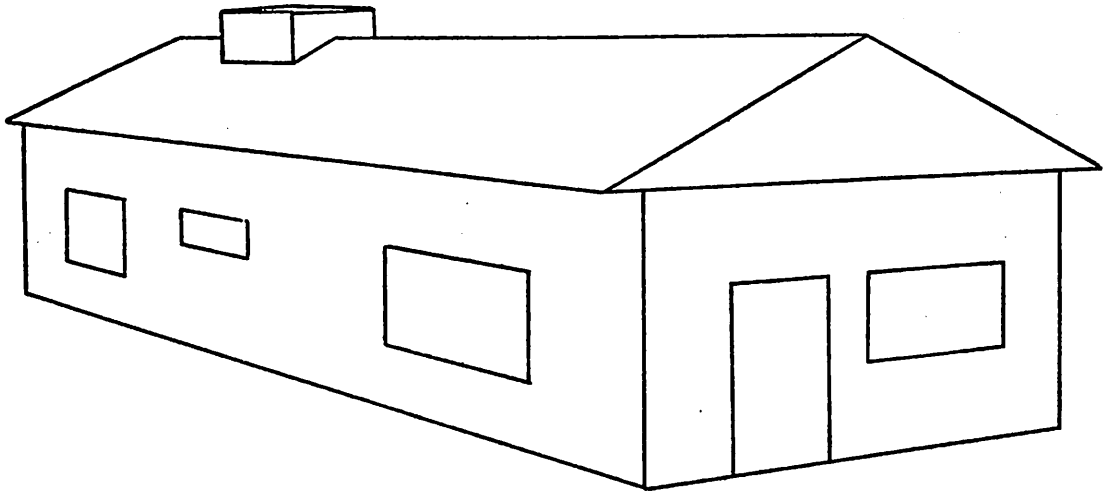
IV.2 Shape, Surfaces, Spatial Processing, and Perspective

Representation of three-dimensional space is required for both frames and objects. As we stated earlier, the spatial relationships of the frame entities are crucial to the application of the frame information. Similarly, we must provide a description of the shape of an object. If the system is expected to understand viewable space, it must have access to the volume that the object is expected to fill. Since surfaces bound volumes, the object level will be mapped down upon the surface level.

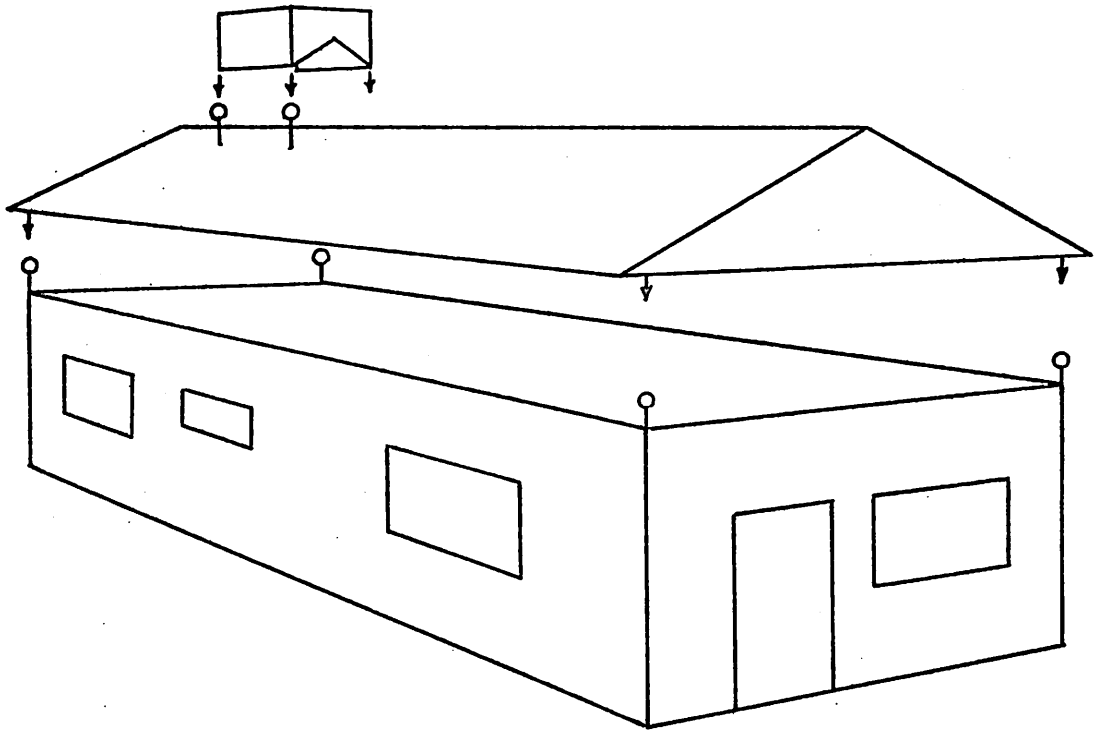
The complexity of shape at the semantic level rivals or exceeds the difficulties caused by texture at the segmentation level. Our initial approach to this problem is an attempt to define a coarse description of space in terms of primitive surfaces and volumes.

In the simplest case, say a parallelepiped, each face may be defined in terms of a surface as a primitive area (on a plane in two space) and then the relative orientation of these surfaces are used to "hook" them together to bound a volume in three space [17]. Representations of simple surface areas and volumes can be stored for use as primitive descriptions. However, we can extend primitive volumes to include a collection of related surfaces which has a macro-definition with a label name. This representation should be sufficient to build a simple stereotype of an object or a frame; descriptive structural examples of a house, car, and road scene are shown in Figure 8-10.

In the house description (Figure 8) three primitive volumes are hooked together at relative orientations and positions in order to roughly



(a)



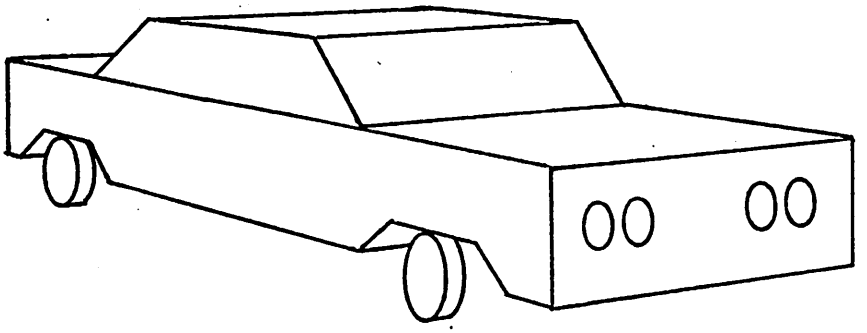
(b)

Figure 8 Representation of Shape of House

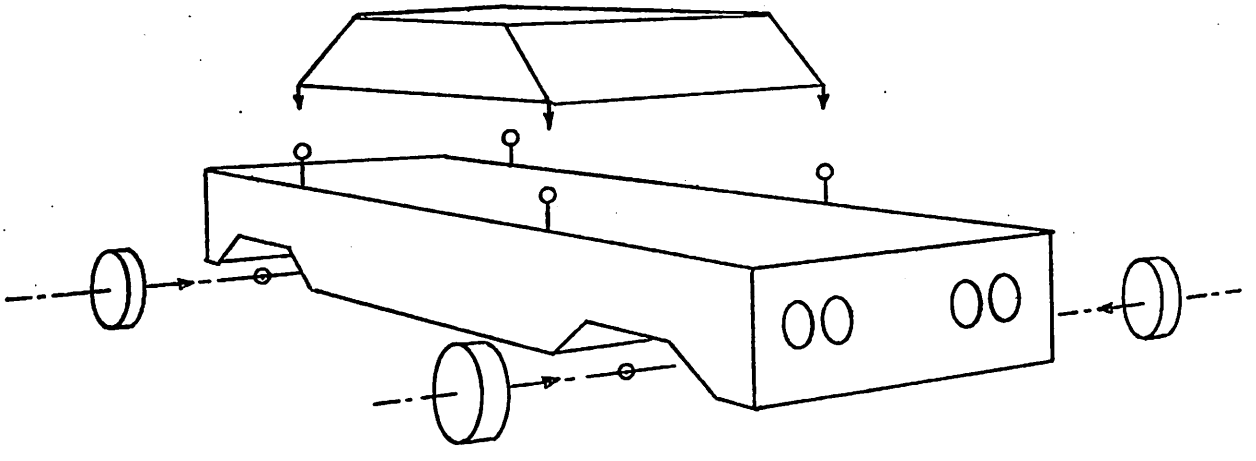
define the volume occupied by the prototype house. Windows and doors are shown as primitive areas lying on the surface planes of the volumes. It is, of course, necessary to specify a set of constraints on the way in which volumes and surfaces are attached. The location of windows and doors are variable within certain placement limits and structural considerations. While the chimney may be placed anywhere along the roof line, it must be attached to the volume representing the roof.

Depending upon the representation, the chimney itself may or may not be a primitive volume. It would be considered as primitive if its volume is defined in terms of surfaces; on the other hand, the chimney can be made up of three primitive volumes: two prisms and a rectangular parallelepiped. This collection could then be labelled "chimney" and treated as a non-primitive volume because it is built out of other volumes. In the case of the chimney, the object shown is the prototype for chimneys which straddle the roof line; for chimneys wholly within one plane of the roof, either a different primitive representation is required or possible modifications to a single representation must be provided.

Similar considerations hold for the prototype car, as depicted in Figure 9. For the car body volume one can build it as a primitive in terms of surfaces or as a non-primitive in terms of other volumes (some of which may be primitive). In the latter case, the car body could be described either by adding volumes together or subtracting volumes from another. In general, there are many representations of an object or part of an object.



(a)



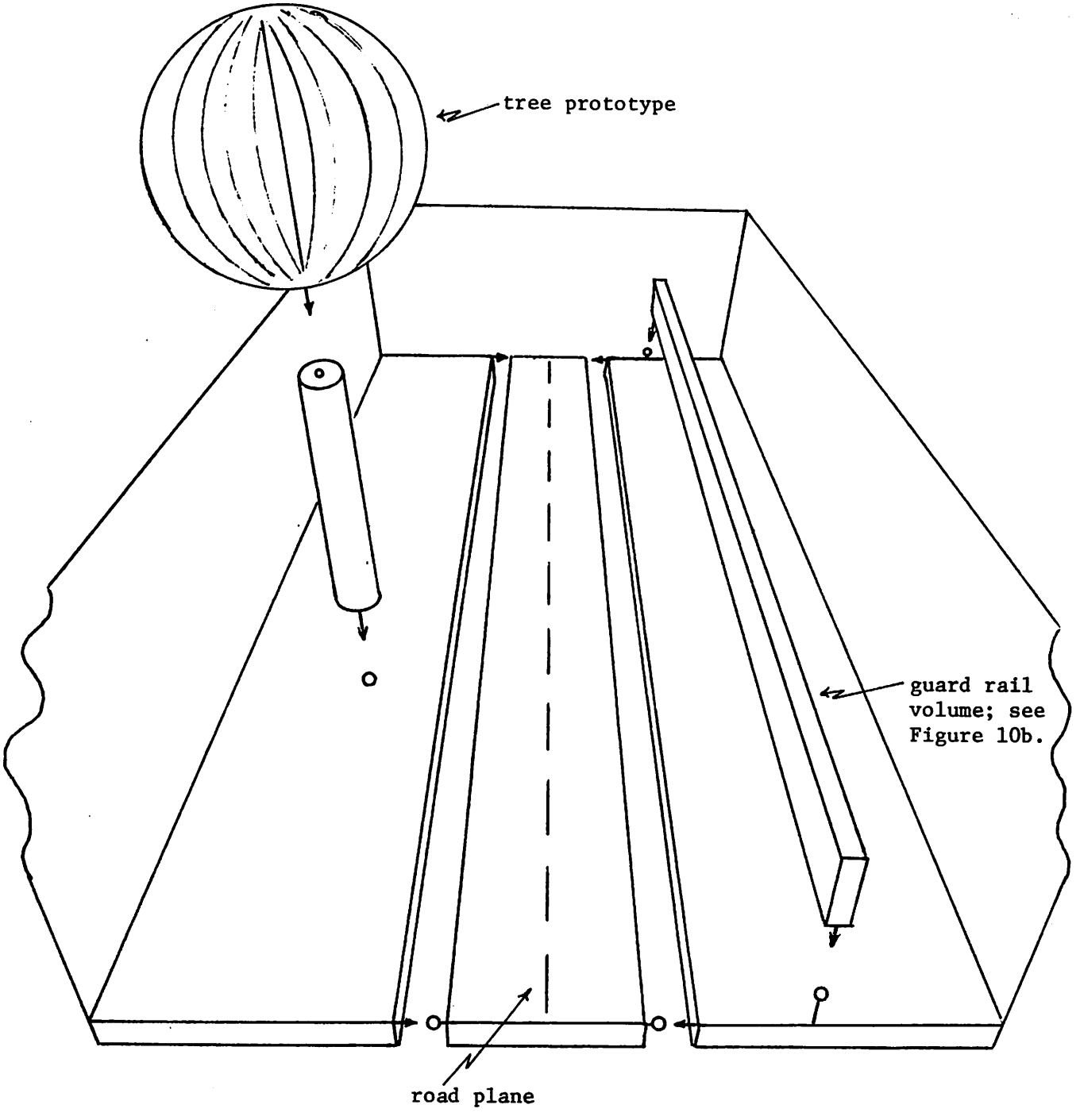
(b)

Figure 9 Representation of Shape of Car

There may be constraints upon the ways that surfaces and volumes attach to each other and these constraints could be critical to the description. The attachment points between the roof and body of the car are clearly variable, but only along the z-axis; the volumes must be connected (Figure 9b) so that the planar surfaces representing the sides line up. However, the points at which the upper volume attaches to the body volume can slide backward and forward within a specified range. If it is moved outside this range, the perception of the hood of the car will begin to disappear so that it will no longer look like a standard car.

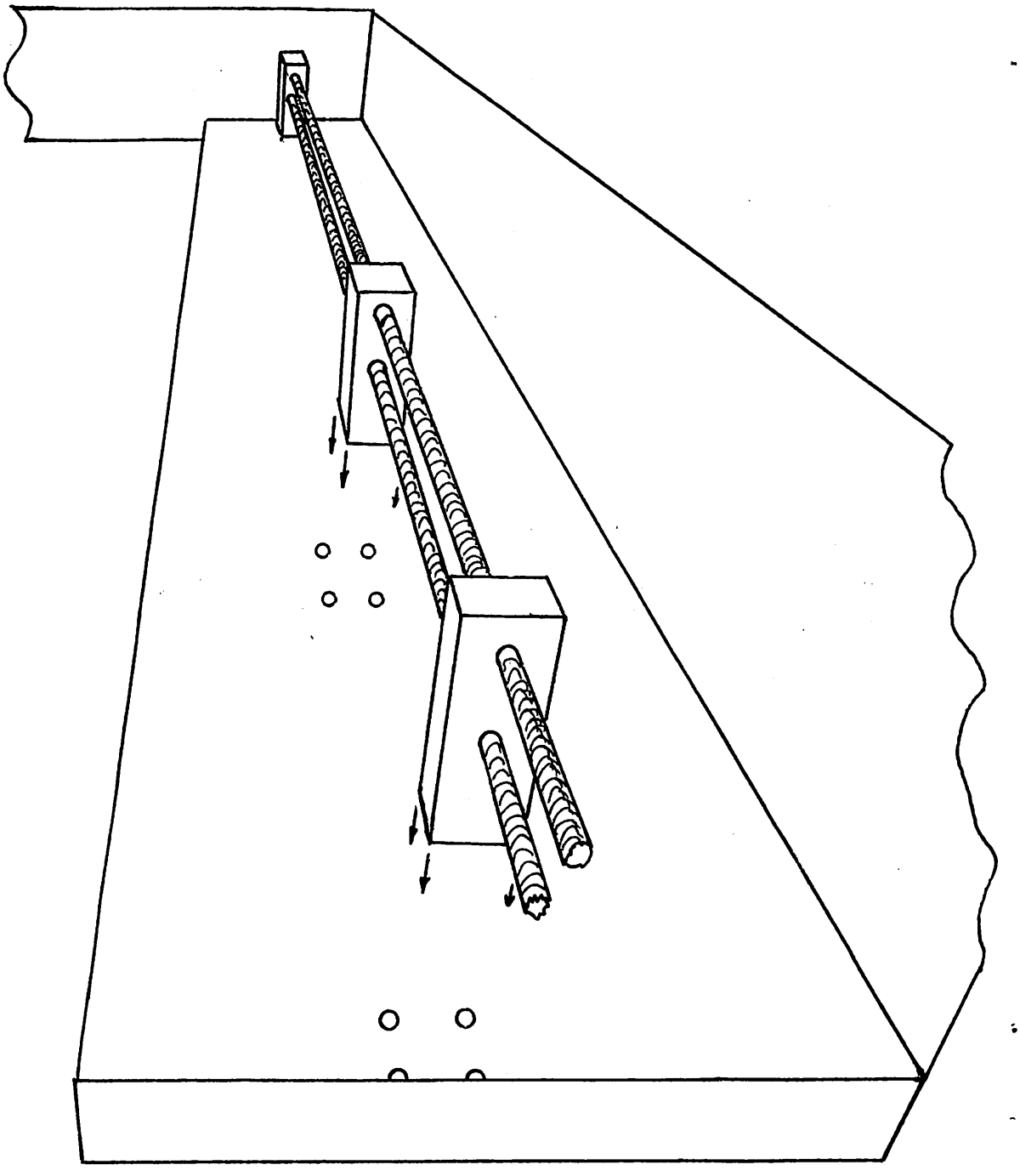
Figure 10 is a description of space in the road scene frame. Here, some of the conceptual entities are surfaces such as the road and roadside. The same descriptive language, though, can be used to attach the surfaces and volumes; the orientation and placement of the hooks (including constrained variation) describe the relationships of the parts in building the whole. In Figure 10a, the volume in which the guard rails can be located is represented; then the description of a prototype guard rail (Figure 10b) and their relative placements completes this representation.

Depending upon the shape of the entities involved, different types of primitives seem to be more natural. Most of the house and car seem to be naturally represented by planes and polyhedral volumes. However, the wheels of the car or the telephone pole are better represented as cylinders. The generalized cone [18,29] takes a cylindrical representation and specifies the modifications down its axis to fit specific shapes (e.g., a screwdriver). It allows parts to be hinged and this seems to be a reasonable way to approximate the shape of a



(a)

Figure 10 Representation of Spatial Relationships in Road Scene Frame



(b)

Figure 10 Representation of Spatial Relationships in Road Scene Frame

person as well as the degrees of freedom in their movement. If the language of primitive elements is rich, there should be sufficient flexibility in the description of a wide range of objects and situations.

One should note that these descriptions can be symbolically represented independent of any particular point of view. This would allow a single representation to be picked up by a perspective process which could rotate the symbolic description (not manipulation of the equations representing the vertices) so that the appearance from a particular direction can be deduced. Now the problem of fitting the shape description of an object or frame to the shape and size of regions is well defined, although the actual process by which a collection of regions is made to match a prototype shape descriptor is still very difficult. A spatial processor must rotate these descriptions and scale them in order to take into account the effect of distance upon the size of the regions appearing in the image. The scaling might be facilitated by normalizing each description into a unit cube whose orientation and size can be manipulated to fit the particular image example. This process can be driven bottom-up from the shape of the regions produced by the segmentation processes, or top-down from the shape representation by predicting the appearance of regions representing objects or surfaces.

Finally, it may be useful to provide a hierarchical description of the shape of an object from coarse to fine. This allows a simple shape representation as an index into complex shape representations. Also the hierarchy can be used to roughly provide information about the appearance of regions at different levels of the cone. Thus, the coarsest description of tree links a sphere (or hemisphere) above a cylinder. More detailed

decriptions should allow distortions on the hemisphere, but even more importantly is the surface description capturing the "texture" of protrusions and cavities in the shape of the crown. Fine descriptions must move down to the level of branches and leaves which partially fill the sphere. Thus, the global crown shape can delimit the volume containing the cylindrical representation of branch volumes and the planar representation of leaf areas (a two-dimension area carries most of the information for specifying the volume of the leaf).

IV.3 Example of a Model

Let us summarize the discussion of model representation using as an example the simple hand-drawn tree scene shown in Figure 11. A partial sketch of the model and its relationship to the permanent knowledge base of the system is depicted in Figure 12. For clarity only some of the links in the model have been drawn. In addition there are many details that are still being developed and have been left purposely vague.

A model in this approach is defined by the specifications on the entire left set of planes shown in Figure 12. Note that this includes the three planes of the RSE interface data structure, since they represent the results of processing two-dimensional sensory data and define the entities upon which concepts map, i.e., the syntax of visual information. The upper planes define the organization of the conceptual three-dimensional information representing the system's understanding of the world depicted in the image. Let us consider each of these in a bit more detail.

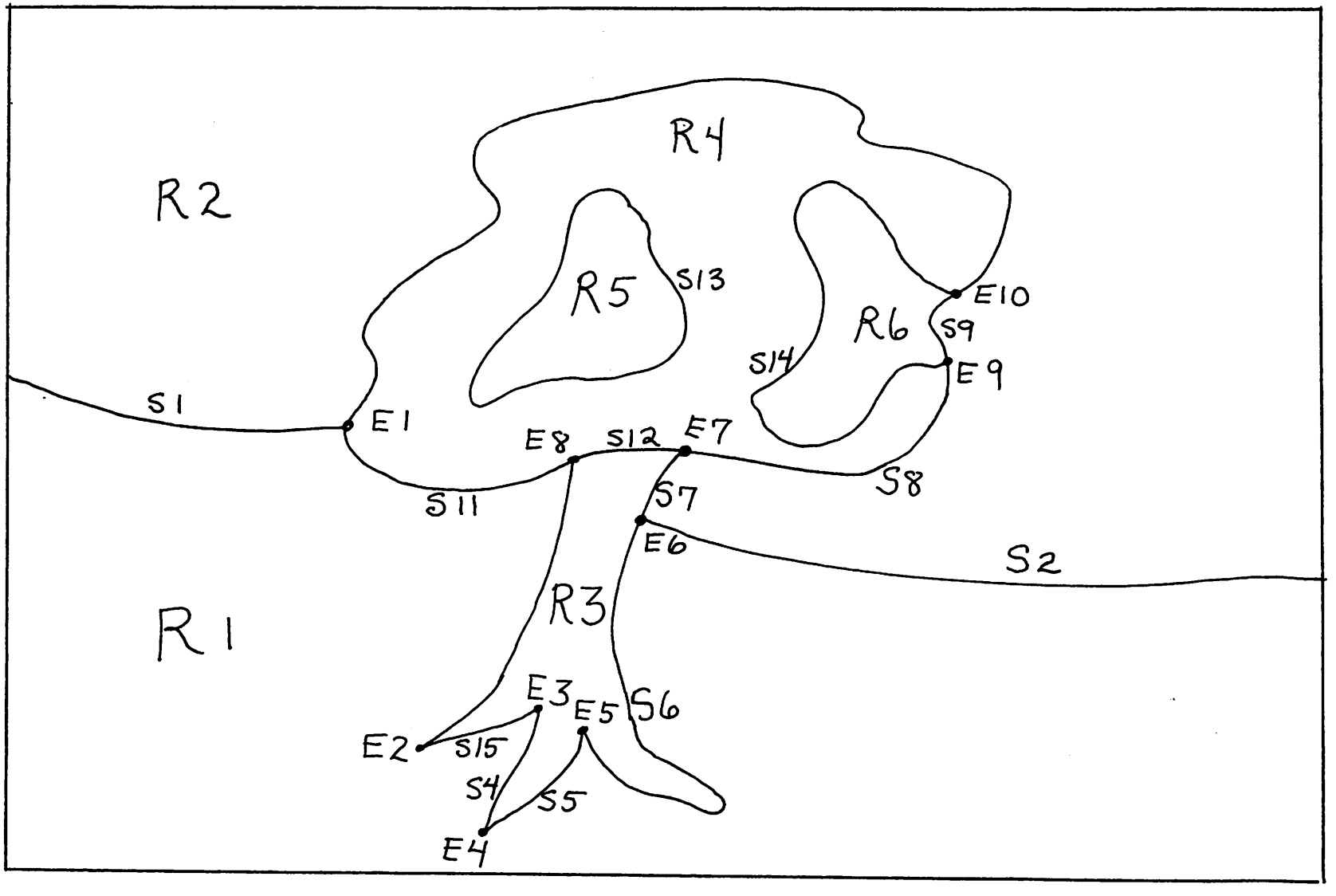


Figure 11 Example Tree Scene

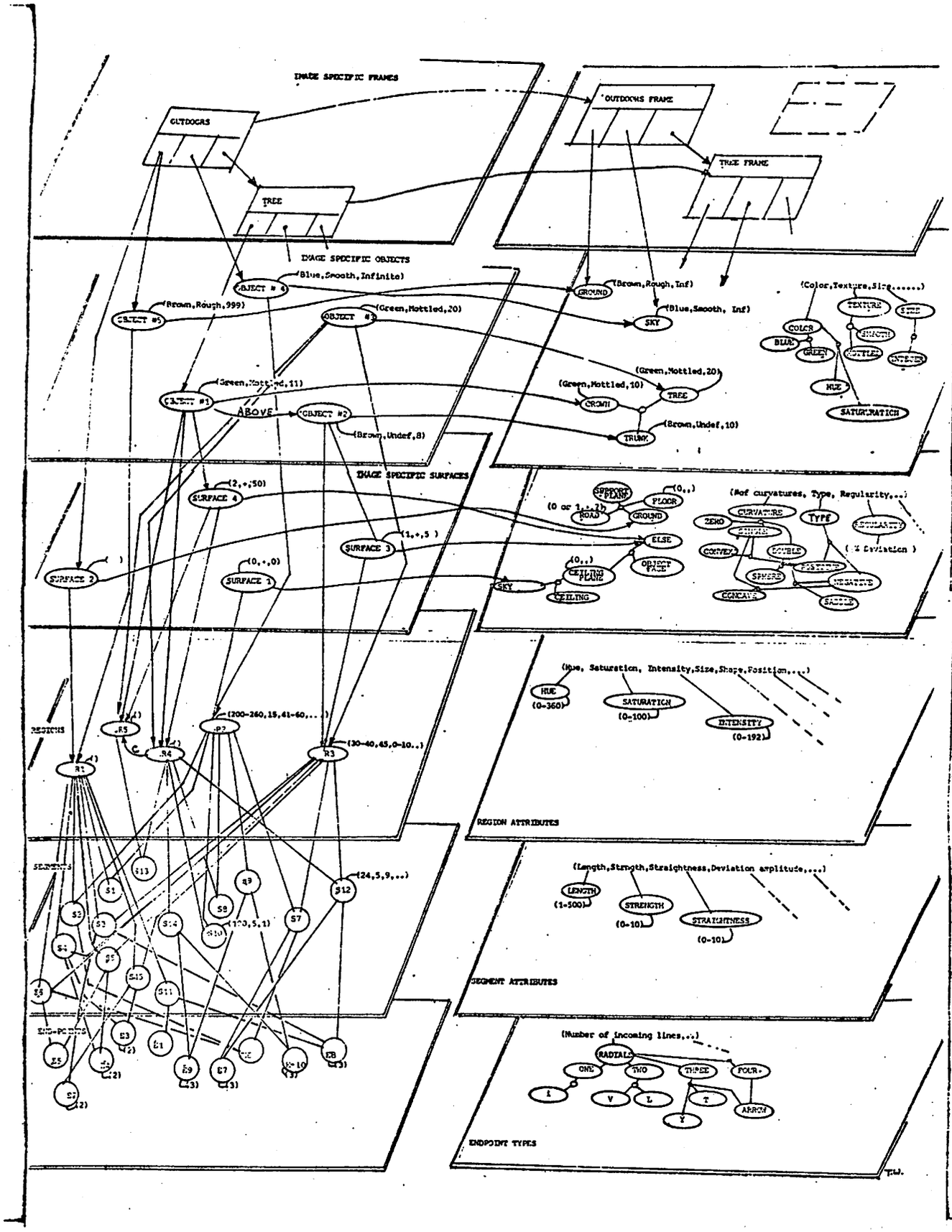


Figure 12 Partial Model of Tree Scene

Associated with entities at all levels (except possibly frames) is an attribute-value specification of the form that we have already described, i.e., a list of the features and values for each object, surface, region, line, and endpoint. In Figure 12 the entities in STM have vectors of values associated with corresponding vectors of attributes in LTM, without the need for pointers to the actual attributes in LTM. There are both advantages and disadvantages to this structure, and other alternatives exist for representing attribute value pointer structures [19].

One further point to be clarified involves the difference in the definition of the attributes of objects and regions. By following the appropriate pointers down into RSE and its long term equivalent, the "sensory" values of these features may be found. Objects and surfaces (which are semantic entities) have the commonly interpreted "colors" represented as symbolic entities, while regions (which are syntactic entities) have numeric specifications of hue and saturation, which represent a distribution of the wavelengths of light or some type of mapping closer to the sensory data.

If the pointer structure is carefully examined, one will notice the rich level of redundancy that is represented. For example as a model is constructed, the object O_4 might point to color-blue and texture-smooth, just as the concept "sky" in LTM does. This is part of the basis for O_4 being hypothesized as sky, and this hypothesis is represented by a pointer to sky completing the loop.

V. CONTROL OF PROCESSES IN BUILDING A MODEL

V.1 The Model Search Space

In this section, we discuss the process of constructing a model of the scene. The model builder begins with the RSE data representing the segmented image, as well as the world knowledge at all levels of representation (the right hand side of Figure 1). At any given point in time, a partial model consists of the entire set of nodes and arcs of image-specific data in short-term memory, including those arcs into the permanent data structure. The model is constructed incrementally, in a sequential mode, based on a model building strategy and a collection of modular sources of knowledge. These changes might be the instantiation of an object or frame, the addition of a relation arc on the object plane, a decision about objects occluding each other, an assumption about the context (such as the season or the time of day), etc. Thus a partial model which represents the current set of hypotheses is continually expanded. At each decision point there are usually a very large number of alternatives to choose from; thus, one can envision a collection of partial models, each representing a different interpretation of the image data.

Our decision to build models sequentially instead of employing paradigms utilizing distributed computation [8] is worth a brief aside. It seems that once one understands many of the local interactions in a very complex system such as VISIONS, the system could be redesigned to run many of the processes simultaneously, and take advantage of parallel processing at the semantic level just as we are doing in the cones at the low level. However, there

may be difficulty in the development and debugging of a system with locally distributed computation [20]. The user may not be able to easily grasp the complexities of local interaction. We have chosen a conservative strategy during the development stages. The system is being configured so that a user can easily step in and assume the role of the control strategy [21]. Although this is a methodological consideration, we believe that advantages of parallel and locally distributed computation make it highly appealing for the effective development of real-time vision systems.

The model search space can be structured efficiently as a tree of models, where each node in the tree represents a different model. A new model is generated whenever a change (or collection of changes) is made in the current model. Each node stores only the incremental change to the current model and the reasons for this change. This representation is naturally embedded in the context tree structure of CONNIVER [22]. In this environment, the model, at any node in the tree, is the union of incremental changes in the (unique) path from the currently active model (decision point) back to the root node. Also, each node in the tree is a partial model, and is stored as a context giving the system the ability to return to any previous environment for further processing.

It is important to avoid confusion between the whole model search space and each model itself. Figure 13 is a simple example of a search space for our example scene. The union of the information contained in nodes 1, 2, and 3 constitutes the current partial model. Each node of the model search space contains an incremental change to the model, usually an addition of nodes and arcs to the model itself (STM) and between the model and prior knowledge (STM + LTM).

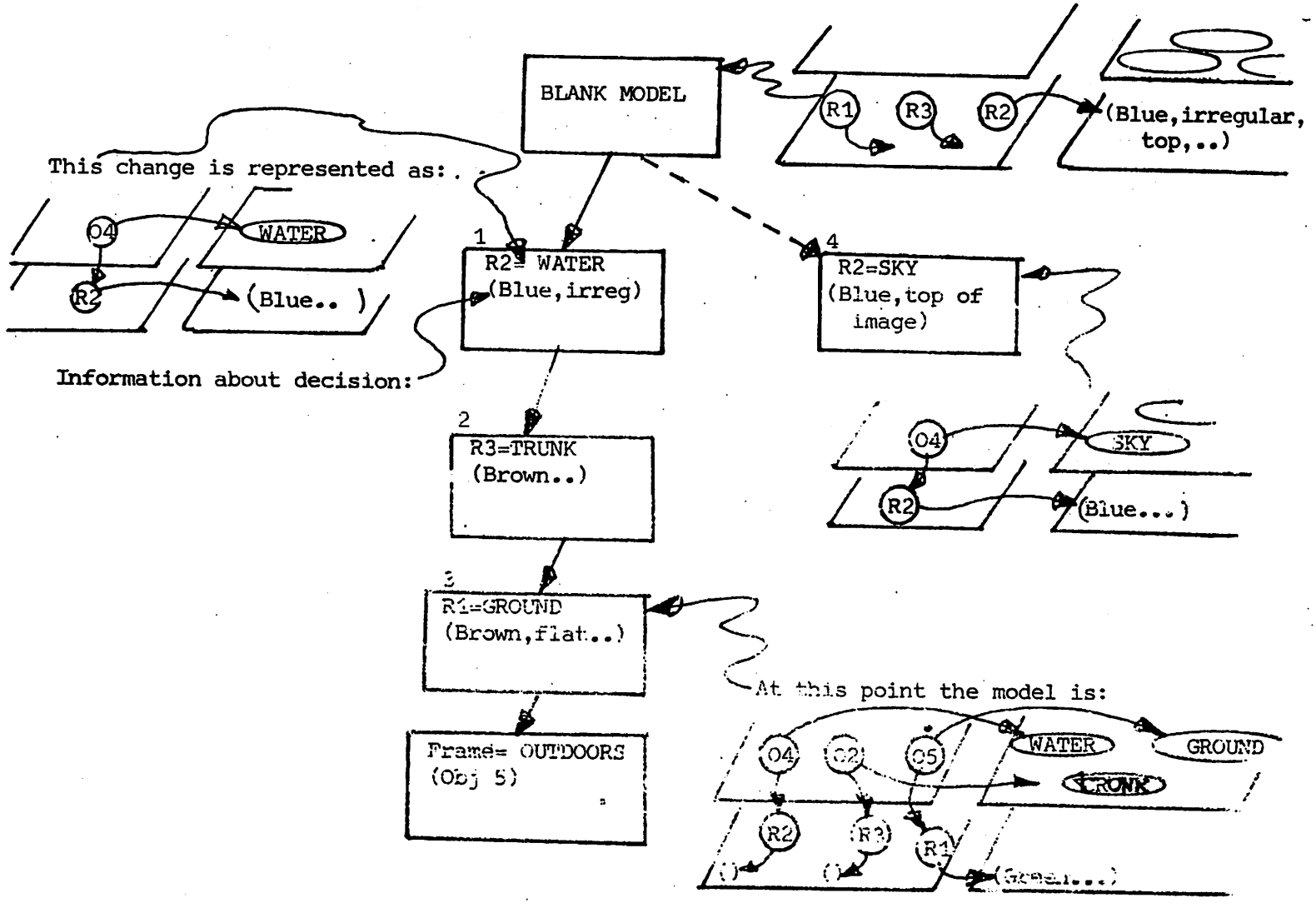


Figure 13 Model Search Space

One must decide upon the type and extent of information which should be saved at the model node. The purpose for storing this information is to recover from errors and conflicts without the necessity of undoing the entire model. Consider the following situation. During the course of model building, it may become apparent (perhaps as a result of further processing) that the identity of R_2 is in error. In this case, the current model must be suspended and a new one formed, assigning to R_2 the best hypothesized identity (e.g. sky). Now, we do not want to discard all the hypotheses that have been made since the incorrect decision -- the erroneous assumption might have been near the root of the search space! In fact any decisions in the suspended model which were independent of the assumed identity of R_2 should be retained in the new model without requiring much additional computation.

Intelligent and directed backtracking can be based on a trace of the decisions made during the model construction process. It might be possible to store the knowledge sources which were responsible for the decision, their confidences, the depth of their analyses (since there may be different investments of computation based upon the importance of the decision), the dependencies of the decision upon the partial model at the time, additional modules which have not been called but may be important, and the best set of alternatives to the decision being made

In general, one can view the model construction process as a search through the space of all possible models. If this whole approach to scene interpretation is to be successful, only a very small portion of the huge

search space should be explored. However, it is unlikely that all decisions in the path of the final model were made correctly on the first attempt--clearly, the efficiency of the search depends upon the sophistication of the strategy used to guide the process and the quality of the knowledge sources employed.

V.2 Knowledge Sources and Processes

The complexity of the model building process becomes apparent if one examines the types of information which must be utilized. The model builder must be a flexible system which accesses and manipulates diverse forms of information. In VISIONS there will be a system of processes which interact somewhat heterarchically; however, final decisions will be under the control of an executive which can be modularized into substrategies. The executive strategy is responsible for integrating the responses of the subprocesses, examining the implications of these responses in the context of the partially constructed model, resolving conflicts, etc.

Let us list the various semantic processes or knowledge sources which provide information to the modular control strategy and upon which decisions must ultimately be based:

(1) object detectors (formerly referred to as vision routines): under the control of the low level executive; these routines respond with a rough confidence for various alternative identities of a region; they will examine RSE and when necessary examine data in the cones;

(2) semantic data retrieval: semantic world knowledge that is not scene specific;

(3) perspective analyzer: the module contains information used to relate the 2D and 3D world, and aid in object and frame recognition by rotating the perspective of 3D symbolic descriptions;

(4) occlusion analyzer: this module utilizes heuristic information to determine the likelihood that one region represents a surface/object which occludes another; it examines the relationship of regions in the neighborhood of a given region, dominance of boundaries, etc.;

(5) shadow analyzer: checks consistency of the light source and the shadow produced by objects off the ground plane; examines intensity gradients on regions (objects) with approximate uniform intensity, and compensates for the variation in the strength of boundaries of regions running in and out of shadows;

(6) deductive system: a module which checks the logical consistency of the model within the data base; this function may be realized as a deductive system which utilizes the semantic data base as its axioms, and the model or portions of it as the theorem to be proved; others have proposed that this process be implemented as a constraint satisfaction [23] or relaxation procedure [24,25].

V.3 Modular Control Strategy

The construction of a consistent model clearly depends upon the strategy embodied in the model builder and, subsequently, the invocation order of the models. Generally each of the processes described will not be allowed to directly call another process. Instead, the executive will monitor the interaction and invocation of the set of knowledge sources as requests.

The importance of a correct invocation order is demonstrated by a simple example; Figure 14 represents a line drawing of a person. The occlusion module is responsible for determining the liklihood that an object (or region) partially occludes another object (or region). This is accomplished by various heuristic cues which take into account the similarity of two regions separated by a third, considerations of line dominance of adjacent regions,

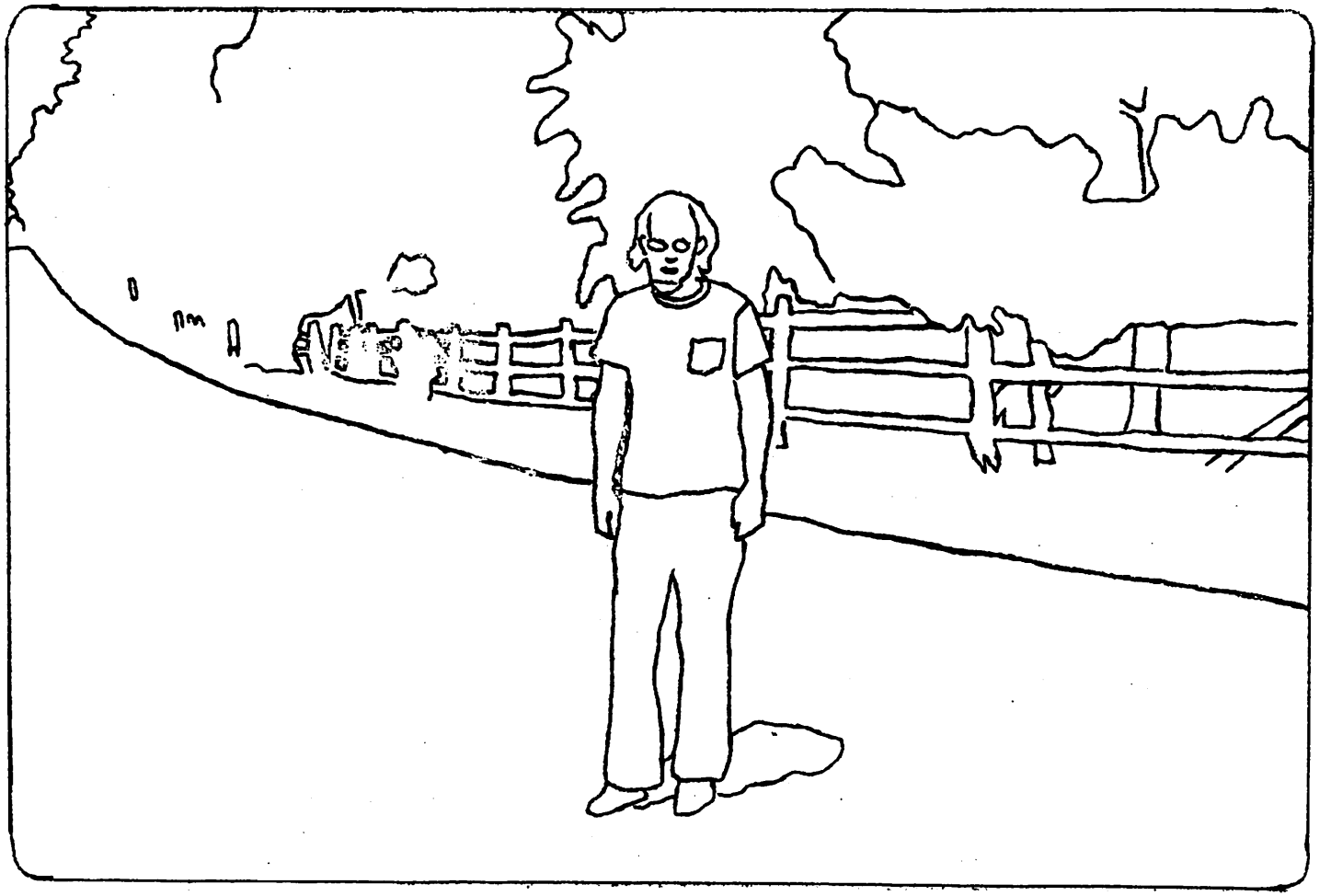


Figure 14 Example for Invocation of Occlusion Module

etc. If invoked blindly, the occlusion module will consider the hypothesis that since regions R_1 and R_2 have similar features they are part of a single object or surface which is occluded by R_3. This implies a conclusion that the arms (regions R_1 and R_2) are in fact one object, occluded by and separate from R_3 (the shirt or body). At a certain level, this might be a reasonable conclusion to reach; however, it is not the usual interpretation of the relationships between arms, bodies, and shirts. In this case, a faulty conclusion has been reached because of the prematurity of the invocation. However, if the module were invoked under the direction of a 'person' frame, then it would be examining the scene at the proper level of abstraction: Is the person occluding the background? The fence behind the person can be hypothesized (correctly) to be a single object. Thus, it is most important to have the processes communicating at the correct level of analysis. Similar examples may be advanced for the other modules and in general this is an important consideration in the development of good control strategies.

The development of a model involves the instantiation of surfaces, objects, frames, and the relations between them. Thus, the control strategy defines how the links between these entities form. For example, an object can be instantiated bottom-up on the basis of regions (referred to as R-0), or top-down on the basis of hypothesized frames (referred to as F-0). Depending on how one wishes to drive model development, more or less weight can be given to each of these local model building strategies. In general this viewpoint (similar to HEARSAY knowledge sources) leads to a set of local

control strategies for linking entities between pairs of levels. One can define a particular strategy to be utilized in each case. For example, in hypothesizing an object for a region, the R-O strategy can call in an attribute matcher to determine the degree of match between region and object attribute values. But more reliable information might be obtained by examining object shapes and having the perspective analyzer attempt to fit the shape to the region by rotating and scaling the description. However, this approach is conputationally exorbitant and should be used only in cases where there is other strong evidence for the particular object. These types of trade-offs would be represented in the local R-O strategy. Other modular control strategies can be defined for F-O, O-F, S-O, etc. Now the executive can be flexibly programmed to prefer certain choices and the user of the system can more easily explore the space of control strategies.

VI. SUMMARY

This paper represents an outline of work in progress. We have focussed primarily upon the issues of representation and control. There are many different stages in the development of visual models and many diverse sources of knowledge which contribute to this development. Due to the obvious complexities involved in the design and implementation of such a system, we will be following a methodology based on incremental simulation [21]. Here the user can interactively play the role of unimplemented processes and methodically replace distinct functions by software over a period of time while the whole system is in operation.

The low-level system is being developed on a PDP-15 computer in FORTRAN and assembly language with a color monitor and disk. The high-level system is being implemented on a CDC Cyber-74 in LISP which has been extended by GRASPE [26] and portions of CONNIVER [22].

The overall operation of VISIONS can be summarized in terms of some of the major design goals that have guided the development or have evolved with increased understanding of the problem domain:

1) A flexible interface of visual sensory data (numbers) to semantic knowledge (symbols) - The image is segmented in the hierarchical parallel processing cones and mapped into symbolic entities and descriptors (the RSE structure). This representation of the processed 2D information is the interface to prior stored knowledge about the world.
2) A flexibility of processing control - The development of the model can be data-directed, knowledge-directed, or model-directed. The

system must be able to segment an image even if it is a semantically unrecognizible or nonsense scene. Here the sensory data may entirely drive the analysis in a bottom-up fashion. In other cases control of model building proceeds top-down in a predictive fashion by accessing world knowledge in the long term data base. Finally, the development of an image-specific model can and should be directed by the partial model already formed (which of course is another but more focussed entry into the semantic data base). As the current model nears completion, more and more of the processing can be expected to be model-directed.

3) A sketch of the control history for directed but limited backtracking - The model will be developed incrementally. Each addition to the model will be stored in a CONNIVER context tree so that previous model environments may be accessed. The basis for each addition to the model will be stored, including the knowledge source(s) responsible, their confidence, any dependencies of this decision upon other hypotheses in the partial model, and finally the subset of best alternatives for extending the model. If conflicts in the model arise later, the system can examine the basis of a previous decision, then possibly: a) call in a more complete analysis by knowledge sources which were not utilized or only carried out a limited analysis; b) decide to delete or modify a previous hypothesis, requiring an examination of the hypotheses that followed for dependency upon this decision; c) examine other alternatives for the decisions in conflict; etc. It may be useful to provide a global view of competing models by collapsing the context tree into a graph structure similar to the HEARSAY

system. This representation provides pointers between model hypotheses which would not be directly available between distinct branches in the tree of contexts. However, this global view makes it difficult to store directly the history of sequences of incremental model additions. More experience with these structures is called for.

4) Both local and global processing - In the processing cones an operation on a window at the lower levels provides a local view of the sensory data, while an operation on a window at a high level in the cone has a more global view due to the larger receptive field from which it draws information. The cone provides a simple linkage between local and global processing. At the semantic level there are local semantic relationships between, for example, objects via a semantic network. One can trace specific relations independent of a larger context. On the other hand the scenarios or frames at the highest level of representation, provide a global view of the knowledge base. The set of conceptual entities and the definition of three-dimensional space in a common situation such as a road scene acts as the global semantic structure. Similar statements are valid for the 6 plane image specific model.

5) Both serial and parallel processing - The cone functions as temporally ordered sequences of parallel operations at the different levels of transformed image representations. The interpretive processes, however, have been organized in terms of sequential control. This is not based upon a committment to sequential semantic processing, but rather a committment to a methodology that allows ease of human understanding of complex interactions between complex modular processes.

Given the state of the art of distributed computation, we believe that it would be difficult to debug improper interaction of simultaneous processes. By structuring sequential control, the user can play the role of a process (including the control process) at any point and only need to deal with a limited amount of information. Once the communication requirements between knowledge sources and the limitations and capabilities of each knowledge source are better understood, it should be far easier to redevelop the system in a parallel process of locally distributed computation.

6) Many levels of representation - The system has the ability to view the world in many ways. The lower three levels of representation involve syntactic two-dimensional information. One might be interested in the boundary between two regions or a property of a region. This will allow the system to talk about the image itself, as opposed to the world represented and implied by the image. The upper three levels of representation describe important aspects of a three-dimensional world. Surfaces bound space and more highly evolved biological systems have a well-developed sense of them. "Objects" or conceptual entities provide the common labels by which we communicate about the world that we share. Frames encompass highly structured sets of objects and surfaces with compact labels and descriptions. These representations are intimately woven together in a hierarchy since frames map onto objects and surfaces, objects are defined by surfaces and regions, surfaces are defined by regions and line boundaries, regions are bounded by line segments, etc. The deletion of any of the six levels leaves the system blind to certain aspects of visual perception that one expects of a general vision system.

7) Diverse knowledge sources - The system will employ modular processes based on independent areas of knowledge. These include processors for perspective, occlusion, and shadows. For example the perspective module requires an analysis relating physical real world dimension to image dimensions; it entails a process which must be able to rotate descriptions of the shape of the surfaces of objects and frames so that the point of view can be inferred. The occlusion module will make inferences on spatial relationships and missing portions of objects. The shadow module might hypothesize the location of the light source and the grouping of regions with different intensities into a single object. A deductive module must examine the consistency of the model, etc.

8) Redundancy of information - At the segmentation level there are many algorithms which can be based upon many possible features. Rather than hope to find the single choice for each which will provide a useful partition and description of all parts of an image, the low-level system expects that multiple representations can be brought together, much as the four representations of the frog's view of the world [27]. Here, partial redundancy is expected and utilized to increase the confidence in the results of the processing.

At the semantic level the arguments are similar. One might hypothesize a mountain in the distance on the basis of the shape of the upper boundary, the bluish cast of the region, the context of the scene, etc. One does not throw away secondary evidence and base a decision on the single strongest piece of evidence. The totality of the information should be integrated into globally strong decisions.

ACKNOWLEDGMENTS

The authors wish to acknowledge the major contributions to this phase of the research by Kurt Konolige, John Lowrence, Tom Williams, and Bryant York.

REFERENCES

[1] A. Hanson and E. Riseman, forthcoming paper on Processing Cones.

[2] T. Williams and J. Lowrance, "Building Models in the VISIONS System," forthcoming technical report, Computer and Information Science, University of Massachusetts.

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

[4] A. Hanson and E. Riseman, "Preprocessing Cones: A Computational Structure for Scene Analysis," Computer and Information Science Department Technical Report 74C-7, University of Massachusetts, September 1974.

[5] A. Hanson, E. Riseman and P. Nagin, "Region Growing in Textured Outdoor Scenes," Computer and Information Science Department Technical Report 75C-3, University of Massachusetts, February 1975.

[6] D. Fishman, A. Hanson and E. Riseman, "Some Considerations in a Model Building System for Scene Analysis," Computer and Information Science Department Technical Report 75C-2, University of Massachusetts, February 1975.

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

[8] L. Erman and V. Lesser, "A Multi-Level Organization for Problem-Solving Using Many, Diverse, Cooperating Sources of Knowledges," Proc. 4th Int. Joint Conf. on Artificial Intelligence, September 1975, p. 483-490.

[9] M. D. Kelly, "Edge Detection in Pictures by Computer Using Planning," Machine Intelligence 6, 1971, p. 379-409.

[10] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEETC, 1971, p. 562-569.

[11] L. Uhr, "Layered 'Recognition Cone' Networks that Preprocess, Classify, and Describe," IEEETC, 1972, p. 758-768.

[12] A. Klinger and C. Dyer, "Experiments on Picture Representation Using Regular Decomposition," Technical Report UCLA-ENG 7497, 1974.

[13] S. Tanimoto and T. Pavlidis, "A Hierarchical Data Structure for Picture Processing," Computer Graphics & Image Processing, June 1975.

[14] S. Zucker, A. Rosenfeld and L. Davis, "General Purpose Models: Expectations about the Unexpected," Proc. of 4th IJCAI, September 1975, p. 716-721.

[15] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," IRE Trans. on Electronic Computers, EC-10, June 1961, p. 260-268.

[16] E. Riseman and M. Arbib, "Computational Techniques in Visual Systems, Part II. Segmenting Static Scenes," Computer and Information Science Department Technical Report 76-11, University of Massachusetts, July 1976, to appear in Proceedings of IEEE.

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

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

[19] S. E. Fahlman, "A System for Representing and Using Real-World Knowledge," Massachusetts Institute of Technology, AI-Memo 331.

[20] R. Davis and J. King, "An Overview of Production Systems," Stanford AI Laboratory Memo AIM-271, October 1975.

[21] W. Woods and J. Makhoul, "Mechanical Inference Problems in Continuous Speech Understanding," Third IJCAI, August 1973, p. 200-202.

[22] D. McDermott and C. Sussman, "The CONNIVER Reference Manual," Massachusetts Institute of Technology Memo 259a, January 1974.

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

[24] A. Rosenfeld, R. A. Hummel and S. W. Zucker, "Scene Labelling by Relaxation Operations," IEEE Trans. on Systems, Man and Cybernetics, June 1976, p. 420-433.

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

[26] J. Lowrance and K. Konolige, "Portable LISP Subsystems: GRASPE and CONNIVER," forthcoming technical report, Computer and Information Science, University of Massachusetts.

[27] J. Y. Lettvin, H. Maturana, W. S. McCulloch and W. H. Pitts, "What the Frog's Eye Tells the Frog's Brain," Proc. IRE, 1959, p. 1940-1951.

[28] T. Pratt and D. Friedman, "A Language Extension for Graph Processing and Its Formal Semantics," CACM, 4, 1971.

[29] J. M. Hollerbach, "Hierarchical Shape Description of Objects by Selection and Modification of Prototypes," Massachusetts Institute of Technology AI-Memo 348, November 1975.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: COINS TR 76-9	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A PROGRESS REPORT ON VISIONS: REPRESENTATION AND CONTROL IN THE CONSTRUCTION OF VISUAL MODELS	5. TYPE OF REPORT & PERIOD COVERED: INTERIM
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Allen R. Hanson, Edward M. Riseman	8. CONTRACT OR GRANT NUMBER(s): ONR N00014-75-C-0459, NSF DCR75-16098
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Computer & Information Science, University of Massachusetts, Amherst, MA 01002	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Arlington, VA 22217	12. REPORT DATE: 7/76
	13. NUMBER OF PAGES: 57
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

computer vision
semantic data base
model building
segmentation
control structures

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is an interim progress report on the evolving structure of VISIONS, a computer system for general visual perception. The goal of the system is the segmentation and interpretation of a digitized color image of natural outdoor scenes. We outline the multi-level data structures used for representing both a visual model of the scene and the semantic data base of stored knowledge about the world. A flexible modular strategy controls the operation of processes which embody diverse

forms of knowledge, and allows both data-directed and knowledge-directed model building. A model search space is used to store a sketch of the processing history during model formation, so that limited, directed backtracking will be facilitated.

A symbolic data structure (RSE for Regions, line Segments and Endpoints) interfaces the results of low-level segmentation processes with the interpretation processes which form hypotheses about surfaces, objects, and frames of visually familiar situations. The RSE structure represents syntactic two-dimensional image information while the three higher levels of representation organize semantic concepts in three-dimensional space. Utilization of the RSE structure decomposes the development of the low-level and high-level systems; it provides a clear statement of the requirements imposed on the low-level segmentation processes, and delineates the form of the data which will be the input to the high-level processes. The summary contains a discussion of the major design goals of VISIONS.

Two forthcoming reports will supplement this paper. The low-level system is only briefly discussed here but will be treated in more detail in [1], while further details of the model builder will be provided in [2].