SOME CONSIDERATIONS IN A MODEL BUILDING SYSTEM FOR SCENE ANALYSIS

By

Daniel H. Fishman*
Allen R. Hanson†
Edward M. Riseman*

COINS Technical Report 75C-2
March 1975

*Computer and Information Science Department
University of Massachusetts
Amherst, MA 01002

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

This research was partially supported by the Office of Naval Research under grant No. N00014-67-A-0230-0007

ABSTRACT

This paper outlines the design of a system, called VISIONS, whose goal is to build a semantic 3-dimensional model from a 2-dimensional digitized scene. There are many kinds of information that must be employed in model construction, ranging from processed visual data to highly structured semantic information emobdied in context frames. The modular subsystems that process this information interact through an executive which is responsible for the model construction.

We discuss a variety of considerations in making such a system both flexible and feasible. Brief arguments are offered for dealing with confidences, expectations, and importance of objects, attributes, and partial models in a rough manner. This allows a search of the space of models to be directed by the quality with which information in the model fits together. A deductive system under the control of the model builder and embedded in an AI language will allow the proper partition between programmer and system control. A high level search would be under the control of the programmer and very efficient low-level proofs of consistency of models would be under the automatic control of the deductive system. The ability to capture simple heuristic relationships of complex processes, e.g., perspective, as simple declarative assertions, allows the use of both procedural or declarative information to be employed in various subsystems. A simplified scenario of model construction demonstrates how the system might work.

TABLE OF CONTENTS

FIGURES

Introduction

This paper is a rough outline of a variety of considerations in designing a system to build a conceptual model of information in a two-dimensional scene. It represents a first pass of a number of ideas that are currently being examined for integration into a full visual perception system called VISIONS (Visual Integration by Semantic Interpretation of Natural Scenes) [1]. The ideas are not yet fully developed and a discussion in full detail will require far more space than is desirable at this point.

Before we discuss the processes of model building, an overview of the entire system will be presented. Figure 1 is a sketch of the major subsystems and information flow in VISIONS. The system is to be applied to 2D color images of natural outdoor scenes which include objects with complex shapes, texture, and color such as trees, roads, fields, fences, cars, etc. What is to be the desired output or function of the system? Firschein and Fischler [2] have used the term "encyclopedia concept" to represent a repackaging of the information contained in a visual scene so that general questions concerning this information may be asked. For example: "Is there a tree in the scene? Can I get from point A to point B? How?" We adopt this orientation at this early stage of development of the system; we are not embedding our vision system within any goal specific domain, but intend to devise a system general enough so that this adaptation is natural.

Our expected output is a gross description of the scene; details of the description will be filled in depending upon the application or use to which the system is put. The goal of the model builder in this research is the construction of a rough conceptual model of a 2D scene including identification of major objects, three dimensional information relating these

objects, relative distances between objects, gross terrain features and general information concerning the structure of the scene as a whole. A 2D model might include the labelling of all major regions ("objects") visible in the 2D image; the 3D model might take the form of a graph with nodes and arcs which show the relevant attribute-value pairs associated with each object and the relationships (spatial, functional, and temporal) between all objects in the scene. This would allow a robot to both answer questions about the scene and to achieve goals (e.g., navigation plans) using information from the scene.

A parallel computational structure, a "processing cone", transforms and reduces large amounts of visual data in a layered fashion [3]. Information flows up, down, and laterally within the cone by defining local parallel functions which are duplicated to operate across the entire array. Parallel line finders, region growers, texture analyzers, and color mappings, etc., will operate on a 256^2 grid and reduce it layer by layer to a 16^2 grid, and in some cases even to the 1 x 1 level which contains information extracted from the entire scene. Cueing specialists or demons can look for major features in the reduced image near the top of the cone; e.g., a large expanse of blue and white at the top of the image, a roughly textured green region, a set of straight lines, etc. The presence of prominent features can then be used by the model builder to invoke individual vision procedures which attempt to confirm the presence of a particular object (e.g., the sky, or a tree). These procedures examine the upper layers of the cone fairly completely; they examine the large arrays at the lower levels much more selectivly.. They return a value denoting the confidence that a particular object is present in a given area of the scene. Thus, the bulk of the visual data can be examined under the direction of procedures which use information in the upper levels of the cone as well as higher level semantic information.

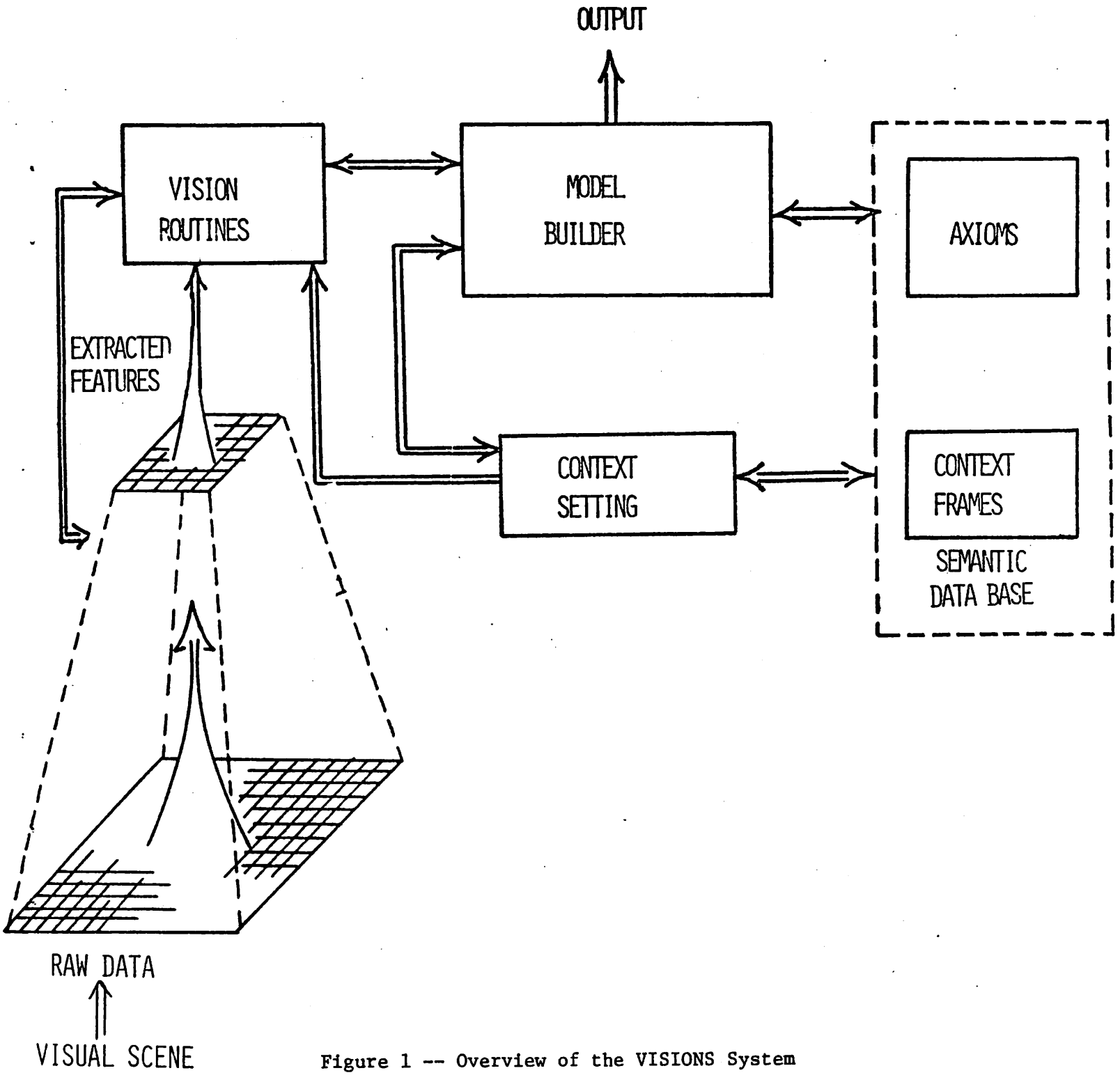

Figure 1 -- Overview of the VISIONS System

Construction of a plausible model will involve retrieval of conceptual information from a semantic data base. We are investigating the use of a formal deductive system to perform the retrieval and manipulation of low level semantic information. The model builder, however, must be a flexible system which accesses and manipulates many forms of information: the output of procedures operating upon visual data in the cone, semantic knowledge of the world (cars move on roads), the context of the scene (e.g., afternoon, summer, wooded area, far from ocean, etc.), and procedures which deal heuristically with perspective, occlusion, shadows, and horizons, etc. Some of this information could be embodied in frame-like submodels [4] or 'slide-box' submodels [5] so that expected or typical subscenes can direct the processing.

The remainder of this paper will discuss the kinds of information that must be employed and the major subsystems embodying this information and their interaction. We will outline some ideas concerning deductive processes under the control of the model builder, required control structures, and examples of formal axioms that capture simple but powerful heuristic information concerning perspective and occlusion. In the last section, a scenario of the model builder applied to a real scene is presented.

Gross Organization of the System

The complexity of the model building process becomes apparent if one examines the types of information that must be utilized. Figure 2 is an expanded view of the model builder. It is shown in a modular form because this seems to be the appropriate way to handle diverse forms of information and to keep the system manageable. The model building executive will be a control structure which invokes each of the subprocesses when necessary and which acts as a message switching center, allowing communications between the remaining subprocesses. It is responsible for integrating the responses, examining the implications, resolving conflicts, etc.

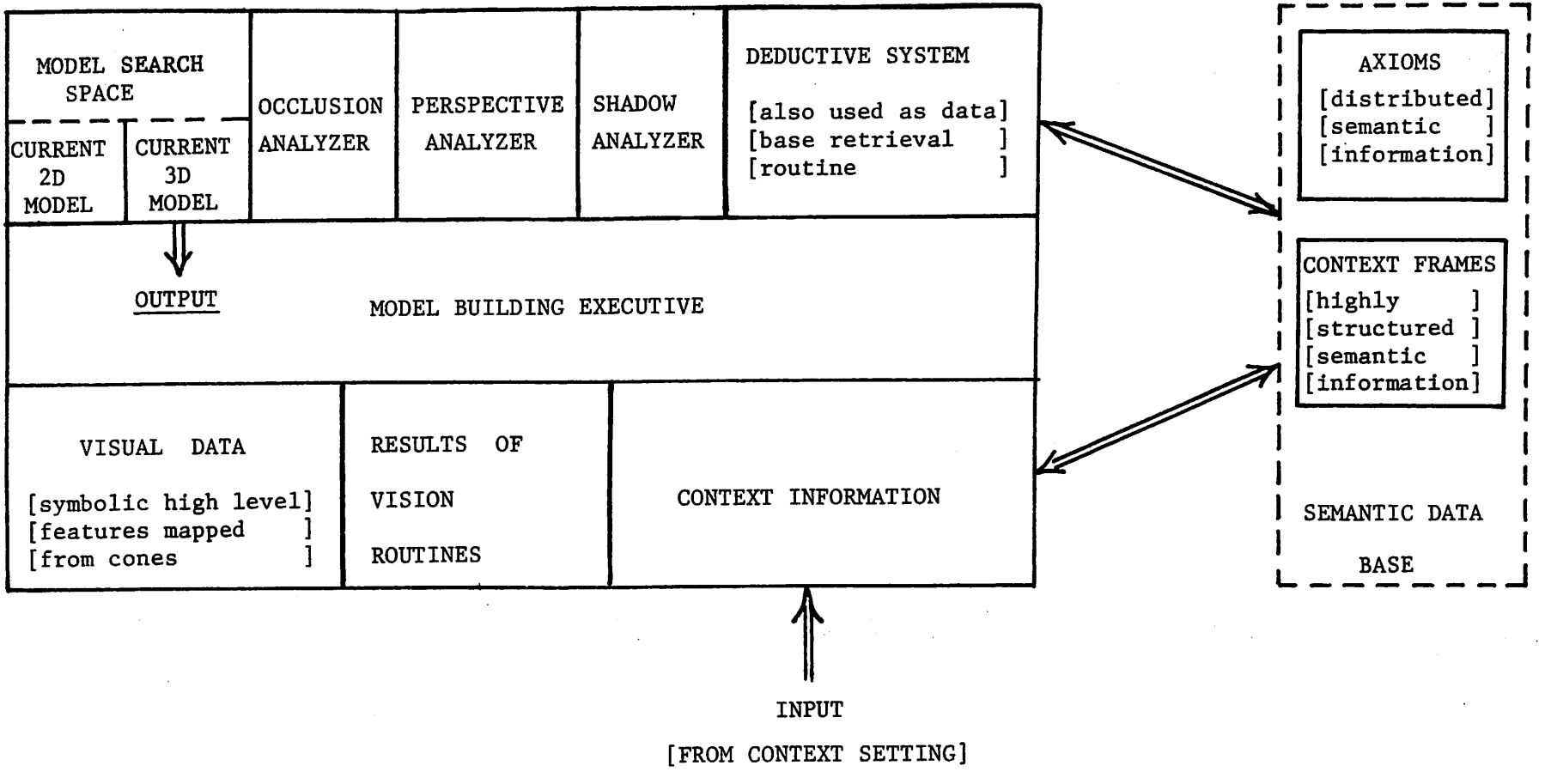

Figure 2 -- The Model Builder

Let us first outline the several sources of information that the model builder must refer to and manipulate:

1) Visual data -- this includes:

   a. visual features extracted in the processing cones; often they will be at the top of the cone, but sometimes more detailed data further down must be accessed. These features must also be mapped to symbolic terms consistent with information in the semantic data base.

   b. vision routines which respond with a rough confidence for various alternative identities of a region.

2) Semantic data base of knowledge as axioms: general knowledge that is not scene-specific is brought in wherever applicable; similar to general data bases of world knowledge for natural language processing, but with obvious emphasis on information useful to visual perception. This information is in a distributed form in comparison to similar information in the highly structured context frames.

3) Context frames: expected stereotypes or submodels affect model construction; this involves information that is in the general semantic data base but organized in a modular fashion and in a flexible (possibly procedural) form. Frames should be a higher level structure to the semantic data base; a road scene with roads, cars, trees and telephone poles alongside the road, etc., is a unit of knowledge to be applied as an entire submodel.

4) Context setting: expectations about the setting of the model (time of day, season, etc.); they modify both information in the semantic data base and some of the processes which operate during model building. The set of active context frames employed in the development of the partial model will also be considered as part of the context setting.

5) Partial models constructed:

   2D Model -- as a particular partial model of the image is constructed, it obviously affects further processing; the 2D model includes information such as labelled regions and the adjacency of visible portions of the tentatively identified objects in the image;

   3D Model -- brings in the conceptual implications of the objects tentatively identified; this includes the spatial, functional, and temporal relationships in the 3D world and any relevant semantic information.

6) Model Search Space: a tree of partial models under consideration where each branch is a hypothesis about the identity of a region or assumption about the context setting of the image; thus, a history of the search is maintained and information concerning the dependency of any decision upon earlier decisions can be included to aid in intelligent and efficient backtracking.

7) Deductive system: this semantically guided deductive process is a powerful theorem prover which is subordinate to the model builder; it is used to check consistency of additional hypotheses with the partial model as the search tree of models is expanded; it should efficiently solve distinct subproblems provided by the model builder, and act as a straightforward information retrieval process from the semantic data base.

8) Perspective Analyzer: this module contains both procedural and declarative information (sometimes in axiomatic form) used to relate the 2D and 3D models, aid in object identification, and generally verify the partial model. The regions associated with tentative identities of objects in the model must satisfy perspective constraints between focal length, the size and placement of the region in the image, the physical size (implied by semantic data) and physical distance. Implications between all separate objects in the model must remain consistent and thereby provide powerful clues for model verification or refutation a declarative form, simple basic shapes such as

also provide information concerning the likelihood of being ground planar or off the ground plane.

9) Occlusion Analyzer: axioms can also be used here to represent heuristic relationships between 2D adjacent regions and whether they represent objects in or out of the ground plane; procedural analysis of the dominance of boundaries also provides powerful cues:

10) Shadow Analyzer: checks consistency of the light source and the shadows produced by objects off the ground plane; examines gradients of intensity on objects with approximately uniform hue; compensates for the variation in the strength of boundaries of objects running in and out of shadows.

The overall control of information processing in the vision system directly affects the strategy required to achieve the desired objective of the system. Specific paths of information flow between modules is not shown in Fig. 2 because many of the modules are only partially designed. It is clear, however, that the way in which these processes are invoked is critical to the success of the system. Consider the analysis of an image of a person. The occlusion routine will be looking for two regions with similar characteristics but separated by a third region. It will consider the hypothesis that the two regions are visible portions of a single object. If this routine is called in too early it might conclude that the two arms hanging alongside the body are occluded by the chest and that they are one object. However if the occlusion routine is invoked by a "person" frame, then it would be examining the scene at the proper level: Is the body region occluding the background? Of course, the system must consider many hypotheses. However, the complexity of this analysis might be reduced by the proper communication and control of these modules.

In summary, the executive in the model builder will facilitate communication between each of the modules defined, and consequently will be the major tion between each of the modules defined, and consequently be the major control center for the system. In addition, it must contain a strategy for correlating the diverse forms of information. It must have the ability to explore alternative models, form additional hypotheses, invoke a variety of validation processes, and resolve conflicts. The executive allows these sources to be highly interactive, structured predominantly in a hierarchical fashion, although heterarchical control is allowed where useful. Each of the modules will be able to request further information through the executive of the system. This should keep the required complex interactions reasonably under control in the design process.

Fuzzy (or Imprecise) Information in Model Building

A characteristic of our world is that descriptions of it and objects contained in it are imprecise; descriptions are not provided as truth or falsity of conditions, concepts, or actions. Rather, many conflicting conditions may be simultaneously feasible with varying degrees of confidence or many concepts might be applicable with varying degrees of "truth" or applicability. Theoretical work in fuzzy set theory and fuzzy logic has not provided any practical insight into analysis of natural systems in which information is often vague and/or confusing. Our use of the term "fuzzy" is not directly comparable to the theoretical usage; perhaps "imprecise" or "vague" information would be more accurate.

Although we do not, as yet, know how to handle this form of information, we do know a number of the processes in which it would appear to be highly relevant (perhaps even necessary). A major problem that must be dealt with is the range of information in the real world associated with a single symbolic descriptor. The description of real objects covers an incredible variety of shapes, sizes, colors, textures, etc., many of which apparently defy precise specification. A single semantic descriptor of color or texture might represent a wide range of values of colors or textures. In addition, fuzziness enters into the expectation and importance of the values of attributes for objects. [Note the difference in these two concepts.] These weights must be handled in some way as the model builder determines a confidence for the identity of each object. To avoid these problems seems to deny the complexity of the real world.

Although each of the following areas deserves careful discussion, we choose here only to list the areas where fuzzy information might affect building models of the real world:

1) the quantification of attributes and their symbolic mapping;

2) the expectation (in the semantic data base) that attributes of objects take on particular values;

3) the importance of attribute-values in recognizing objects;

4) the confidence of various identities for each major region;

5) the importance of subparts of an object in recognizing objects;

6) the ease with which the parts of a model fit together in determining an overall confidence of a model.

We do not believe that statistical decision theory is a viable approach to solving these problems. Rather, heuristic approaches such as the mapping of physical measurements into semantic descriptors representing ranges of values might interface visual data to conceptual data with sufficient flexibility. Confidences probably can only be quantized meaningfully to several levels, say 8, but would still roughly provide the desired direction for searching the model space. Rough weighting of importance of attribute-values follows in a similar manner.

Our View of the Search Problem

The model builder portion of the scene analysis system may be viewed as a search strategy which must explore a search space of partial models, each describing the scene. The space may be represented as a tree, each of whose nodes may be considered to be a partial model. The root node will contain no scene specific information. Rather, it will contain facts and general rules about scenes in general, and other information which must be valid in all models. The root may be regarded as a semantic data base of assertions and of general rules which describe how classes of assertions may interact to define new assertions. Subsequent nodes of the search tree could contain such scene specific items as: (1) assertions about a context setting, e.g., that the scene is outdoors, and it is summer; (2) assertions about objects in the scene, e.g., that a certain region is the sky;

(3) assertions about the relationships which hold between objects, e.g., that the house is behind the trees; (4) deletions of assertions entered in an earlier node, e.g., delete the assertion that a particular region is a (single) tree; (5) general rules which describe relationships which are only applicable in a given context, e.g., the trunk of a tree is often occluded by the crown in summer.

It should be noted that while each node of the search tree represents a partial model, it does not actually contain all of the information describing the model. Rather, it contains only information describing an incremental change of the previous model. Thus, given a node N in the model search tree, the model it represents is understood to be the union of all the information in the nodes in the path from N back to the root node.

As with any other search strategy, at each step the model builder must select from among all nodes which have already been generated which node to "expand" next. While we have not yet considered this decision process in detail, it seems clear that an evaluation function which assesses the likelihood that a given model is correct would be of value. The model builder could then pursue the policy of always extending the best fitting model. Or, this choice could be weighted somewhat by the depth of the model in the tree, to avoid going too deeply with a model at the expense of ignoring another which has only a slightly lower likelihood value. Other considerations might enter in as well; for example, several node expansions might be carried out under control of a single frame.

Once the node to extend has been selected, the model builder will have several tools at its disposal with which to carry out this extension. For example, it might use a context frame as a guideline for selecting particular regions in the scene for further analysis in light of objects one would expect to find in the assumed context. Vision routines might then be activated to attempt to confirm the identity of particular regions. Spatial relationships

between objects and/or regions might be hypothesized and entered into a new node N'. The deductive system could then be invoked in order to confirm or refute the hypotheses in light of the current model. Information concerning this model is found in the path from N' to the root.

An additional approach of the model builder might be the exploration of local semantic implications of the current model. That is, in trying to assign a name to a green "thing", or in trying to determine what might be found near a green thing, the model builder, or one of its resource submodules would locate matching items in the semantic base, using the deductive system as the retrieval mechanism. Items so located could then be used to locate still other items in the attempt to find something that fits. Note that this last method of attempting to identify scene components is less model-directed and under less control than when a particular frame is directing this process. However, we believe that it is unlikely that frames can cover all situations, so that at times we expect to find it necessary to operate in this more unrestrained search.

Features of the Deductive System

Since the idea of applying theorem-proving methods to larger, or more "practical" problems has been the subject of some skepticism [4,6, 7], we wish to briefly indicate here how we plan to use our deductive system, its capabilities, and justification for its use in our system.

The deductive system will be made available as a programming language resource for the implementation of the model builder. That is, we plan to extend a procedural programming language with statements which invoke, limit, and control the deductive system and which manipulate and structure its data base. The data base structure we are planning to support is similar to CONNIVER's [8] context tree. This data structure is quite natural for problem solving and it matches very well the model search space discussed above.

Two principal ways in which the deductive system will be used are (1) as an information retrieval mechanism which is capable of both direct (table look-up) and indirect (deductive) retrieval; and (2) as a consistency checker. In either case, a path in the tree, that is, a partial model, would be used as its data base. Some methods for user control of the deductive system which we are planning are a command to limit a particular invocation to consider only assertions in order to effect table look-up and avoid deductive search; an incremental capability so that partial searches may be suspended and resumed at milestone points (other searches may be conducted using different models while a given search is suspended); and the strong use of the partial model to make available general rules which the model builder deems to be appropriate for a given problem.

Some of the capabilities of the language package will be quite similar to those of CONNIVER -- the context tree and the return of sets of solutions to a given problem. However, the capabilities and philosophy diverge considerably as regards the use of the deductive system. In CONNIVER, the programmer has complete control over and must direct every deductive search, which in essence is why the IF-NEEDED functions were devised. If the interaction of IF-NEEDED's is at all complex, and if the data base is large, then the programmer has a formidable task to manage the search well. In many cases, he may resort to a simple depth-first search. On the other hand, we take the view that there are two search problems going on simultaneously as in STRIPS [9]. One is a high-level search, e.g., the search represented in the model search-space, while the other is a low-level search, e.g., the deductive search associated with proving a new assumption is consistent with a given model. Our goal is to free the programmer to as great an extent as possible from getting involved in the detailed decision making associated with the low-level search. To this end, the deductive system under development is very powerful. The search procedure to be used extends the Q*-search algorithm [10] by guaranteeing not to solve the same subproblem twice for a given data base. It does

this by monitoring the search along all paths of the (low level) search tree. Anytime a subproblem occurs along one branch which is already under attack, or has been solved along another, then the results of the first attack will be applied directly to any other occurrences of the subproblem. This can be implemented with surprisingly little overhead. Other features of the search procedures include the use of problem-specific information to filter out the use of semantically inappropriate assertions and general rules, and to do pruning operations on the low-level search tree.

The deductive mechanism to be used is based upon Π-resolution [11, 12] in which deductive operations correspond to sets of operations performed in the simpler syntax of first-order logic. While Π-resolution was originally devised for large data base applications, its use will also impact on (1) deductive efficiency, since multiple resolution paths are explored simultaneously in Π-resolution; (2) semantic restrictions in the inference process since the notation of Π-resolution permits the typing of variables and the Π-operations involve type checking and intersecting; (3) deriving a set of solutions to a given problem, since the set of all solutions is carried along simultaneously, and is refined with each deductive step (note that there is an avoidance of backtracking implicit in this mechanism); and (4) representing and dealing with objects described by intervals of values rather than by discrete values alone, e.g., intervals of time, or intervals of the frequency domain, etc. This last feature, which may be achieved by interpreting the restriction of variables to be intervals rather than sets, should prove to be useful for dealing with texture and hue, for example.

Thus, while many of the language features contained in CONNIVER will also be present in the language we are developing, we believe that the programmer will have available a more powerful set of tools with which to represent and conduct his high-level search. His need to get involved in the low-level search will be minimal. We are confident that this will be the case since the deductive system being developed is considerably more powerful than MRPPS [13] which has been successfully applied to search problems in the presence of up to 500 assertions and general rules in its data base [11,14].

Two Axioms for Perspective and Occlusion

There are many forms in which knowledge can be expressed. Certainly functional processes must be embedded in the perspective and occlusion modules. However, it is useful to have some of this information available to the deductive system in a logical form. Some of the visual heuristics can be captured in this axiomatization.

We will present here a few examples of the kinds of general axioms which may be employed to support the model building process. The following axiom might be used to heuristically determine which of two identified objects in a scene is closer to the camera:

$$EQ(SIZE_p(X),\ SIZE_p(Y)) \wedge GT(SIZE_I(X),\ SIZE_I(Y)) \longrightarrow CLOSER_p(X,Y)$$

This axiom states that if the physical size of object X is roughly equal to the physical size of object Y and the image size of X is greater than of Y, then X is closer to the camera than Y. Notice that the size functions could be evaluated when the axiom is used. The physical size function would do a table look-up to determine the size or size ranges of the given objects, and the image size function would make measurements on

the image. Predicate evaluation routines could then be used to determine the truth values of size comparisons. It should be noted that the conclusion of the axiom is valid regardless of the topological terrain features.

The following axiom might also be used to determine which of two items (assumed to be resting on the ground) is closer to the camera:

$$\text{LT}(\text{DIST-FROM-BOT}(X), \text{DIST-FROM-BOT}(Y)) \longrightarrow \text{CLOSER}_p(X,Y)$$

This axiom uses the heuristic of comparing the distance from the bottom of the image to the base of each object to determine relative distances from the camera. The conclusion of this axiom is <u>not</u> invariant with respect to terrain features. The conclusion follows if the objects are in the same horizontal plane. However, as a heuristic, the axiom has the virtue with respect to the first axiom of not requiring the identification of the objects.

A final example deals with the relationship between two adjacent regions, X and Y, in the 2D image. Use of this axiom requires the assumption that X is in the ground plane of the 3D world:

$$\text{GROUND-PLANAR}(X) \wedge \text{2D-ADJACENT}(X,Y) \longrightarrow \text{OCCLUDES}(Y,X) \vee [\text{GROUND-PLANAR}(Y) \wedge \text{3D-CONNECTED}(X,Y)]$$

One can conclude that either object Y occludes object X <u>or</u> that Y is in the ground plane and 3D-connected or adjacent to the ground planar region X.

The three axioms we have presented demonstrate the kinds of heuristics that can be described axiomatically. These particular axioms also demonstrate a way in which we plan to blend procedures with the declarative syntax. Many similar axioms can be devised which embody heuristics for perspective and occlusion.

A Simplified Example

This section is devoted to a simplified scenario of model-directed image analysis, assuming the system is configured as described previously. There are clearly many strategies available for identifying regions in a scene and constructing a conceptual model to represent the scene. The strategy described here is a simplified one and where convenient we will ignore many crucial questions and problems. Furthermore, the analysis presented here represents the "correct" path, thereby eliminating any discussion of the control and back-up mechanisms clearly required in any actual application of the system.

Assume that the scene in question is as shown in Figure 3; clearly this represents a highly processed version of the original image, with boundaries marked and regions indicated. Figure 4 contains a partial set of attributes associated with each region. The values of these attributes are quantified very roughly in verbal terms and the descriptors given are not indicative of the actual representation of these values within the system.

In the analysis to follow, we utilize the concept of "context frame" introduced earlier; we note that the context frames and submodels produced during model building are loosely equivalent to, yet highly simplified versions of, the frames and slots proposed by Minsky [4]. They are also similar to the 'slides' and 'slide-boxes' suggested by Arbib [5]. Groups of objects may be conceptually joined as submodels or local context settings; e.g., roads, telephone poles, and cars can produce a road-scene context while buildings with doors, walls, windows, roofs, driveways, and lawns produce a human-habitation context. The examination of the scene once one or two of the objects associated with a frame (or sub-frame) are found can be further directed by a particular sub-frame, instead of the general model

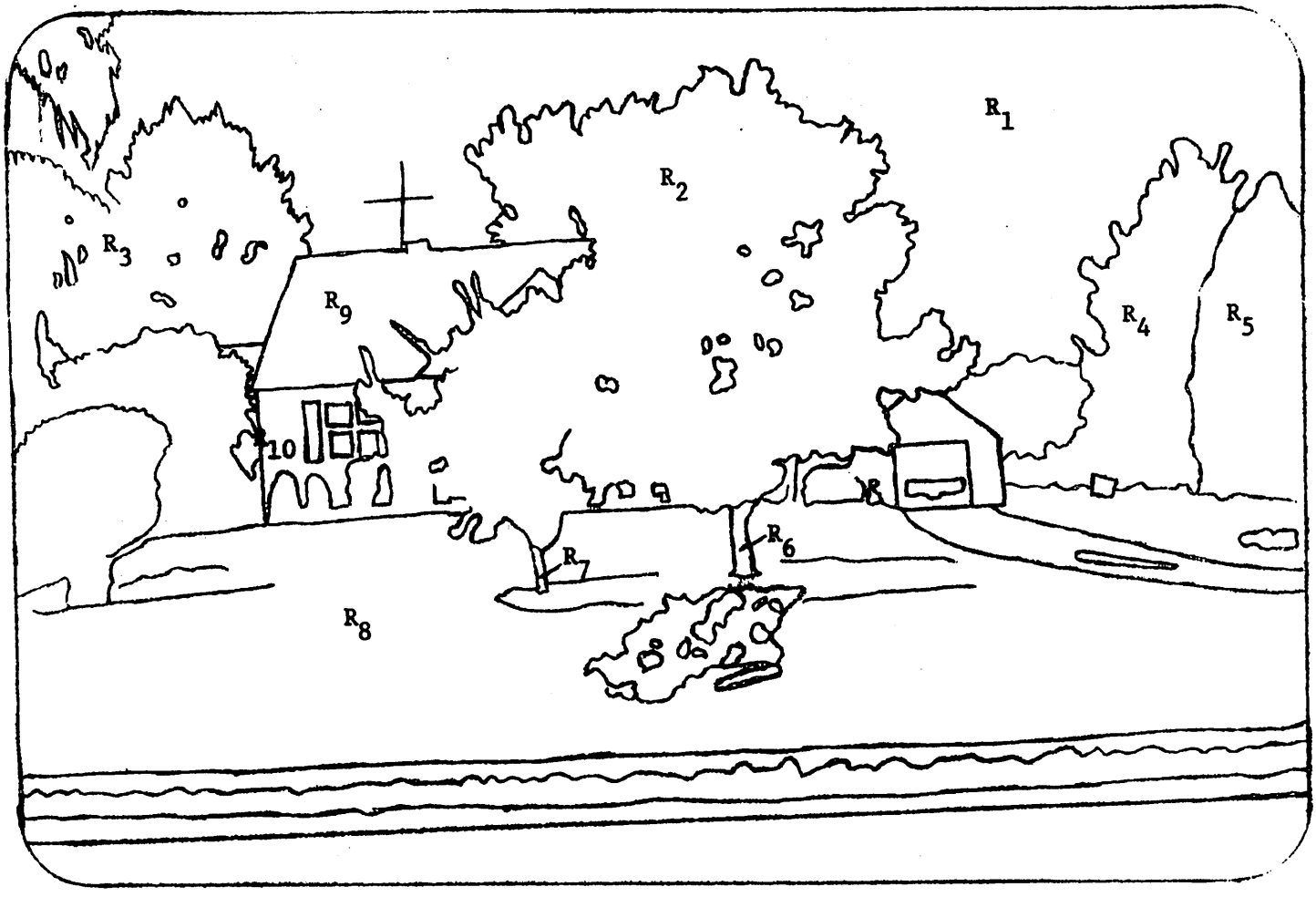

Figure 3 -- Image Used In Text

Region	Hue	Texture	Image Location	Shape	Orientation	Boundary
R_1	Blue	Smooth	Top	Irregular	horizontal	Irregular:lower
R_2	Green	Micro:rough Macro:green & blue	Middle	Blob-like	--------	Irregular
R_3	Green	"	"	"	"	"
R_4	Green	Micro:rough Macro:-----	"	Elongated	vertical	"
R_5	"	"	"	"	"	Semi-smooth
R_6	Brown	----------	"	"	"	Smoothly-varying
R_7	"	"	"	"	"	"
R_8	Green	Micro:semi-smooth Macro:green & green	Bottom	Irregular	horizontal	"
R_9	Red	Micro:smooth Macro:------	Middle	Geometric	--------	Mostly straight, some irregular
R_{10}	White	Micro:smooth Macro:------	Middle	Georetric Irregular	-------- --------	Half straight, Half irregular

-------- Indicates data not available or attribute not applicable.

Figure 4 -- Partial List of Attributes for the Regions Marked in Figure 3

builder and semantic data base. Of course, all the information contained in the data base will still be available.

A frame must contain information concerning the necessary objects for the verification of that particular frame in addition to information on how to put all this information together to obtain a confidence value for the submodel produced from the frame. This could involve the specification of objects absolutely necessary for verification and what to do if some objects are missing. Thus, the contextual frame contains information not found in the general data base which relates to entire groups of objects and their relationships. The main objects of the frame may be jointly sought and their relationships explored as a unit submodel. We are <u>not</u> suggesting a frame system with the full generality of Minsky, rather a highly simplified (probably hierarchically organized) system specifically tailored for outdoor scenes.

We will assume that the most general context for the scene is given; e.g., outdoors. However, strategies exist for selecting this context automatically and entering this selection into the set of decisions made during model building. If further analysis were to indicate a change of context it could be implemented in the same way as erroneous identification of an object.

As an example, many outdoor scenes consist of a sky-ground region, as depicted in Figure 5. Given this setting, heuristics for relating general perspective, occlusion and distance functions to the image are made available. For example, the actual boundary of the sky, as it appears in the scene, is formed by objects from the ground plane protruding into the sky. If an object overlaps the sky-ground boundary (as shown in Figure 5) the object must be non-planar (that is, not entirely in the ground or sky plane) and, depending on illumination, may cast a shadow.

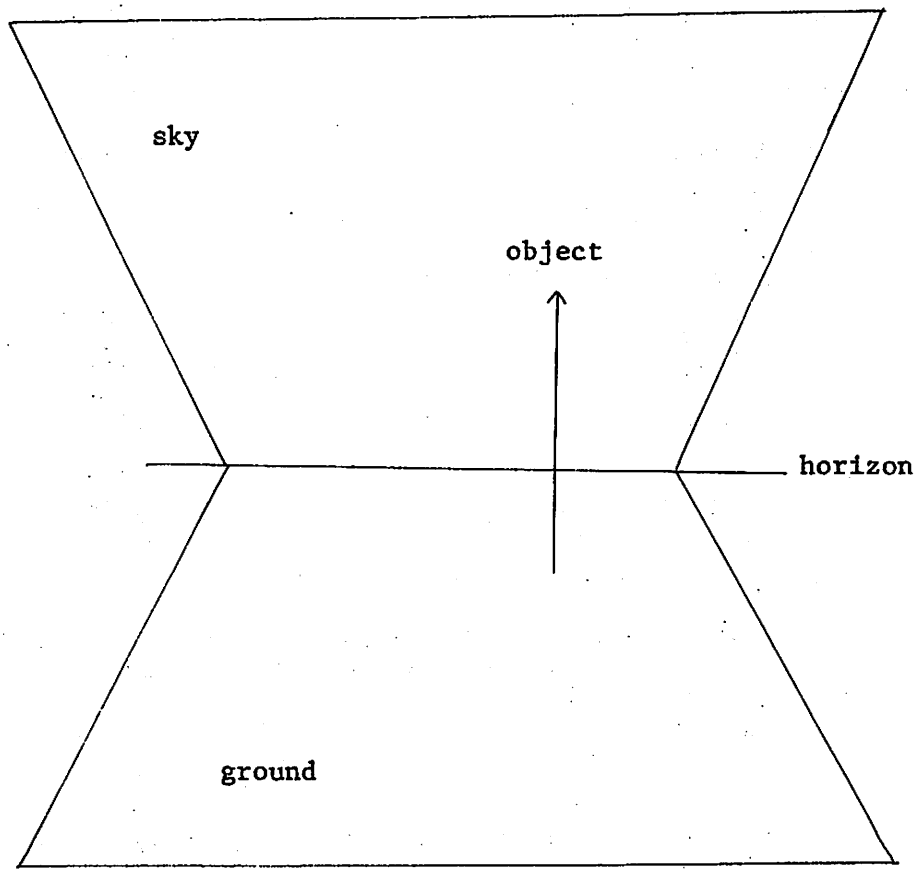

Figure 5 -- One Context Setting for Outdoor Scenes

One strategy for beginning the analysis of an image is to start with the largest region closest to the top of the image (R_1 in Figure 3). Using the information obtained from the cone structure, the attributes ascribed to this region, and the outdoors context setting, a tentative identification of "sky" is made. Note that other identifications are possible (and are considered) but that sky has the highest confidence as determined by the vision routines.

Given a tentative identification of R_1, information associated with "sky" may be obtained from the semantic data base; this base contains both general relationships and specific information such as

- trees are very often adjacent to and occluding the sky, usually below or to the side; the boundary between sky and trees is usually very irregular;
- water (lake or ocean) may be adjacent to and below the sky; the boundary is usually smooth if the other shore cannot be seen;
- mountains may be adjacent to and below the sky; the boundary is usually smoothly varying (depending upon distance);

and so on.

It is evident that region adjacency is an important factor in the tentative identification of regions; Figure 6 represents a region adjacency table for the image of Figure 3. If region i is adjacent to region j, and $i \leq j$, then the i,j^{th} entry of the table is a 1; otherwise it is zero. Only the upper triangular portion of the table need be maintained. From Figure 6 we note that regions adjacent to R_1 include R_2, R_3, R_4, R_5 and R_9. Furthermore, we note that $R_2 - R_5$ are predominantly green with irregular or semi-smooth boundaries and rough micro texture. A hypothesis may be made that $R_2 - R_5$ are in fact trees and activate the tree frame as a subframe of the outdoor frame. This frame contains both general and specific information related to trees:

Region	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	1	0	0	0	1	0
2		0	0	0	0	1	1	1	1	1
3			0	0	0	0	0	1	1	1
4				0	1	0	0	0	0	0
5					0	0	0	0	0	0
6						0	0	1	0	0
7							0	1	0	0
8								0	0	1
9									0	1
10										0

Figure 6 -- Region Adjacency Table

- trees very often have parts: a crown and a trunk, which are usually adjacent.
- shape varies widely although it usually is blob-like or vertically elongated.
- trees are rooted to the ground and may extend into the sky.
- the macro texture of tree crowns may be green and "sky"-color (i.e., the sky may show through tree branches) or green and dark-green (or black) due to shadowing.

etc. Therefore, regions $R_2 - R_5$ are tentatively identified as tree crowns. This identification may be verified using other attributes such as shape. Furthermore, the confidence associated with this (and, in fact, with every) identification is obtained from the vision routines and modified using available information from other sources.

The fact that trees usually have parts (crowns and trunks) may now be used to guide the model building process. Of the regions $R_2 - R_5$, only R_2 has adjacent regions which appear to satisfy the attributes of "trunk". However, we now note that trees usually have only one trunk, while R_2 has two regions (R_6 and R_7) adjacent to it, both of which are potential trunks. Perhaps the most reasonable hypothesis to make is that R_2 is in fact two regions representing the crowns of two trees, with one adjacent to or partially occluding the other, such that the boundary separating them is undetected. We can now return to the cone structure at a very low level (high resolution) and attempt to find this boundary, searching only in R_2. In this case, only a portion of the boundary is found, but it is enough to suggest that the initial hypothesis might be correct. Even if the boundary was not found, the hypothesis is sufficiently reasonable that it may be worthwhile to do further processing in an effort to increase the confidence of the hypothesis.

Consider the problem of determining the size of the objects in question. Simple determination of distances is derived from knowledge of

the parameters of the camera used to produce the image (e.g., lens focal length, distance above ground plane, degree of inclination) in conjunction with size ranges of objects in the world. The image size of an object is a function of its distance from the camera, the camera parameters, and its actual size. In this case, the known distance from the bottom of the image, the length of each trunk, and the probable height of each crown (using a vertical distance from trunk to the point on the boundary of the crown directly above the trunk) results in sizes and distances consistent with the expected size range of trees. This would tend to strengthen the hypothesis that these three regions constitute a group of two trees.

Of course, distance and size estimates of this kind contain a fairly large margin of error. However, lack of reasonable consistency between estimates adds further information about the assumptions required to produce the estimates. We will therefore assume the hypothesis is correct and continue with the analysis of the image.

The tree frame provides information relating trees to other objects often found nearby. Also noted is the fact that trees are most often rooted to the ground. This information further relates the tree frame to the outdoor frame and allows the identification of upright objects in the outdoor frame with the tree. The very general information concerning upright objects may now be applied to the tree(s); e.g., that they can cast shadows. In addition, we find that trees are often surrounded by grass (which is part of the ground plane), or are found in a field or part of a forest, etc. Once this information is obtained the attributes of region R_8 (which is adjacent to both R_6 and R_7) may be correlated with the attributes of grass. In addition, selected vision routines may be activated to examine R_8, returning confidences of its identity; in this case assume that "grass" has the highest confidence.

Relating the proposed identity of R_8 with the ground plane in the general outdoor context frame invokes further evidence concerning the

validity of the model constructed thus far. For example, the ground plane is often occluded by objects; here the tree trunks occlude portions of R_8 and the crowns of the trees extend into the sky, thereby increasing the confidence of the identity of R_6, R_7, and R_2. It is this richness of the problem domain that provides the redundancy of information which allows the model building process to construct a consistent model of a complex image.

We have declined to discuss in detail the heuristic mechanisms which guide the model builder in the selection of regions for investigation. In some cases, the frame structure or semantic data base contains sufficient information to provide this direction. However, in many cases some other external selection mechanism must be provided. The form of this mechanism might be a stack which contains unexplored regions with a heuristic ordering imposed; e.g.,

- regions adjacent to regions whose identity has been hypothesized, in order of the confidence of the hypotheses;
- regions in order of decreasing size;
- regions displaying high color saturation (possibly indicating man-made objects).

The region on top of the stack could be selected whenever the current line of investigation is deemed unprofitable, when an impasse is reached, or when some other avenues appear more promising.

For the current image, several choices of regions are now available; we could continue the analysis using further information from the semantic data base, probably looking next at one of the several regions (not labelled in Figure 3) adjacent to R_8. There are several arguments against continuing this line of investigation. It seems advisable to group dependent

subsets of hypotheses to avoid massive unraveling of hypotheses and identifications if subsequent processing reveals an erroneous decision at an earlier stage.

Let us, therefore, return to region R_1; we are now under the guidance of the general outdoors frame once again. Associated with this frame might be a heuristic which states that straight lines and highly saturated colors are often important clues to the existence of man-made objects. Region R_9 contains a number of straight lines, several regular angles, and is of saturated color; we choose that region next. Note that these considerations could have been used to order the regions on the stack so that R_9 was automatically moved to the top of the stack.

The outdoors frame contains a partial list of man-made objects listed in order of their importance (using some criteria) such as buildings, roads, cars, sidewalks, fences, people, etc. Some of these may themselves reference a frame structure. Let us assume that, on the basis of the attributes of R_9 and the information supplied by the vision routines, the building frame is activated. Using information associated with this frame in conjunction with process and validation procedures similar to those described above, the most likely hypothesis is that R_9 represents a portion of a house (building).

Instantiation and validation of regions continues in this way until the global characteristics of the scene are obtained and the conceptual representation is constructed. At each step, maximum use is made of the massive redundancy of information available in the visual image. For example, integration of the house (R_9) with the trees (R_2), coupled with the dominance of boundary characteristics as analyzed by the occlusion operator, implies

another missing boundary. Surprisingly, there is information implying a boundary separating the front trees from a tree behind the house. This implies, of course, that region R_2 is probably three separate regions!

The subtle interplay of perspective, occlusion, and image data provides very powerful model validation or refutation techniques. Furthermore, this interplay is essential for the construction of a conceptual model of the image. Unfortunately, these processes are still only partially understood as they relate to the problem.

Conclusion

In this paper, we have provided an overview of the approach we are pursuing in the development of a computer vision system. We have pointed out many of the problems which must be dealt with effectively and have discussed some of the processes and semantic structures which must be applied to these problems. In some cases, we have sketched particular mechanisms which have already been implemented or are currently undergoing preliminary design. Ultimately, the successful implementation of the vision system will require the development of mechanisms for dealing with such phenomena as perspective, occlusion, and shadowing; the extraction of visual features; the representation and use of structured knowledge; the application of deductive processes; and last, but not least, the model building process of coordinating and controlling the application of these mechanisms in a systematic, and hopefully, efficient search of the space of scene-describing models.

Our primary design goals are to (1) modularize the processes dealing with distinctly different kinds of information; (2) interface these processes in such a way that control of this very complex system is manageable; (3) determine ways to fit the information together into plausible models with meaningful confidences; and (4) quickly bring to bear as much semantic knowledge as possible and to maintain a model-directed analysis throughout.

REFERENCES

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

2. O. Firschein and M. A. Fischler, "Describing and Abstracting Pictorial Structures," Pattern Recognition Journal, Vol. 3, November 1971, pp. 421-443.

3. A. R. Hanson and E. M. Riseman, "Preprocessing Cones: A Computational Structure for Scene Analysis," Tech. Report 74C-7, Computer and Information Science, University of Massachusetts, September 1974.

4. M. Minsky, "A Framework for Representing Knowledge," AI Memo No. 306, Artificial Intelligence Center, M.I.T., June 1974.

5. M. A. Arbib, The Metaphorical Brain, Wiley-Interscience, N.Y., 1972.

6. T. Winograd, Understanding Natural Language, Academic Press, N.Y., 1972.

7. D. Anderson and P. J. Hayes, "An Arraignment of Theorem-Proving or the Logician's Folly," Memo No. 54, Dept. of Computational Logic, School of AI, Univ. of Edinburgh, 1972.

8. G. J. Sussman, and D. V. McDermott, "From PLANNER to CONNIVER--A Genetic Approach," Proc. FJCC, AFIPS Press, Montvale, N.J., 1972, pp. 1171-1179.

9. R. E. Fikes and N. J. Nilsson, "STRIPS: A new approach to the application of Theorem Proving to Problem Solving," Artificial Intelligence 2, pp. 189-208, 1971.

10. J. Minker, D. H. Fishman, and J. R. McSkimin, "The Q* Algorithm--A Search Strategy for a Deductive Question-Answering System," Artificial Intelligence 4, 3/4 (Winter 1973), 225-243.

11. D. H. Fishman, "Experiments with a Resolution-Based Deductive Question-Answering System and a Proposed Clause Representation for Parallel Search," Ph.D. Dissertation, Dept. of Computer Science, University of Maryland, 1973. (Published without appendices as TR-280, Computer Science Center, University of Maryland, November 1973.)

12. D. H. Fishman and J. Minker, "Π-Representation: A Clause Representation for Parallel Search," TR-74A-4, Dept. of Computer and Information Science, University of Massachusetts, November 1974. (To appear in Artificial Intelligence.)

13. J. Minker, J. R. McSkimin, and D. H. Fishman, "MRPPS--An Interactive Refutation Proof Procedure System for Question-Answering," Int. Jour. of Computer and Information Sciences 3, 2 (June 1974), 105-122.

14. D. H. Fishman, "Experiments with a Deductive Question-Answering System," TR-74C-10, Department of Computer and Information Science, Univ. of Massachusetts, December 1974. (Submitted for publication.)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER COINS Tech. Report 75C-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* SOME CONSIDERATIONS IN A MODEL BUILDING SYSTEM FOR SCENE ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED INTERIM
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* Daniel H. Fishman Allen R. Hanson Edward M. Riseman		8. CONTRACT OR GRANT NUMBER*(s)* N00014-75-C-0459
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer and 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 3/75
		13. NUMBER OF PAGES 33
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)*

Model-Building, Scene Analysis, Computer vision, Semantic Knowledge, Deductive Processes, Perspective, Occlusion, Shadows

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

This paper outlines the design of a system, called VISIONS, whose goal is to build a semantic 3-dimensional model from a 2-dimensional digitized scene. There are many kinds of information that must be employed in model construction ranging from processed visual data to highly structured semantic information embodied in context frames. The modular subsystems that process this information interact through an executive which is responsible for the model construction.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

We discuss a variety of considerations in making such a system both flexible and feasible. Brief arguments are offered for dealing with confidences, expectations, and importance of objects, attributes, and partial models in a rough manner. This allows a search of the space of models to be directed by the quality with which information in the model fits together. A deductive system under the control of the model builder and embedded in an AI language will allow the proper partition between programmer and system control. A high level search would be under the control of the programmer and very efficient low-level proofs of consistency of models would be under the automatic control of the deductive system. The ability to capture simple heuristic relationships of complex processes, e.g., perspective, as simple declarative assertions, allows the use of both procedural or declarative information to be employed in various subsystems. A simplified scenario of model construction demonstrates how the system might work.