

THE DESIGN OF A SEMANTICALLY DIRECTED
VISION PROCESSOR (Revised and Updated)

Allen R. Hanson*
Edward M. Riseman†

COINS TECHNICAL REPORT 75C-1

February 1975

This research was supported by the Office of Naval Research
under Grant N00014-67-A-0230-0007

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

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

This paper is a highly revised and updated version of "The Design of
a Semantically Directed Vision Processor," COINS Technical Report 74C-1.

ABSTRACT

This paper updates our design of a semantically directed vision processor. The system will carry out model-directed analysis of outdoor scenes by applying semantic knowledge at an early stage of processing. The goal is to quickly and flexibly interface low-level visual features (e.g., edge detectors, texture and color analyses) and high-level conceptual knowledge (e.g., trees stem from the ground, general knowledge associated with road scenes, and the like) in the perception of complex images.

The computational structure for rapidly extracting visual features is called a "processing cone." The cone consists of parallel spatial arrays of micro-computing elements, each of which operate on a local window to reduce the data layer by layer. Information flows up, down, and laterally in the cone via a sequence of local parallel operations. Routines for detecting objects will examine the data at the top of the cone and will selectively analyze the lower level mass of data. Rough confidences of the presence of objects in various regions will be passed to the model builder.

Model construction will employ many types of information in modular subsystems. Perspective and occlusion routines which utilize heuristic and mathematical analyses can be applied in both procedural and declarative form. The presence of partial models can be used to direct the system through a search space of possible models. Semantic information structured as sub-models will be used wherever possible to direct this complex process. A deductive system will be employed to check for model consistency at each stage.

TABLE OF CONTENTS

I.	INTRODUCTION	1
I.1.	Low-level Processes and Semantics	1
I.2.	A Cognitive Experiment	3
I.3.	An Overview of the System	4
I.4.	Review of Previous Research	8
II.	LOCAL FEATURES AND PARALLEL PROCESSING	13
II.1.	Hints from Pattern Recognition	13
II.2.	Data Reduction	13
II.3.	Other Local Functions	19
II.4.	Focus of Attention	20
II.5.	Comparison with Past Parallel Processing Machines	22
III.	THE VISION PROCEDURES	24
IV.	REPRESENTATION OF KNOWLEDGE	27
IV.1.	Deductive Semantic Processes	27
IV.2.	Application of Semantic Information to Scene Analysis	30
V.	MODEL BUILDING AND VISUAL PROCESSING	34
V.1.	Gross Organization of the System	34
V.2.	Further Considerations: Fuzzy (or Imprecise) Information in Model Building	37
V.2.1.	Quantification of Attributes and Symbolic Rep- resentation	37
V.2.2.	Expectation and Importance	40
V.3.	Confidence in Building Models	43
	ACKNOWLEDGEMENTS	45
	REFERENCES	46

I. INTRODUCTION

I.1. Low-level Processes and Semantics

Much of the previous research in computer vision or scene analysis has been concentrated in the constrained worlds of "blocks" or laboratories with objects of simple shape, straight lines, and little texture and color [1-8]. This concentration is understandable and valuable insights into complex processes have been provided from this domain. Some of the techniques developed will undoubtedly continue to form a basic library for some years to come; many others will be discarded almost immediately. However, it is generally agreed among researchers in this area of artificial intelligence that the world of polyhedra has served its purpose and that there appear to be two long-range directions for vision research to take.

First, there is a need to cope with the enormous complexity of real world scenes. This will force consideration of many global properties of scenes as opposed to the micro properties employed in the block's world and elsewhere. The second direction involves the interface of computer vision research with powerful semantic techniques with the ultimate goal that perception will take place as a knowledge-directed process. As Tenenbaum [9] points out:

We feel that the time is now ripe to confront a number of these crucial perceptual issues--information overload, segmentation of textured objects, representation of irregular objects, generality of strategies--that do not arise in the blocks world. Instead of simplifying the environment, we must learn to cope with the complexity of real-world scenes by capitalizing upon their natural redundancy of descriptive features and contextual constraints.

These two avenues co-exist in several pieces of recent research [9-16] including our own [18-21]. Consideration of both avenues simultaneously is

a reasonable approach. Lately, work in speech recognition [25-26] has been blending the syntactic and semantic approaches of natural language processing with the more classical approaches of pattern recognition applied to acoustical data. The redundancy of information present in the complexity of the real world, in both vision and speech, provides opportunity for applying high level information in a very natural way, thereby affecting computational savings which the complexity alone would belie.

The "semantics" or "meaning" associated with a visual image depends to a great extent upon the intent of the system perceiving the image. This is particularly true of biological systems, which seem to tune their perceptual system as a function of the goal of the organism, e.g., driving a car or searching for a lost set of keys. This "tuning" suggests a considerable amount of goal-directed processing, even to the lowest levels in the perceptual system.

It is very difficult to conceive of a perceptual system using local analysis isolated from the context, meaning, and goals existing at higher levels. On the other hand, it is fairly easy to see how the interplay of low-level analysis and higher-level processing can be very valuable. By allowing semantic feedback to direct complex processes such as region formation and edge detection in the presence of texture, color variation, and shadowing, problems which are unyielding to isolated analysis may succumb to an integrated approach. It very well may be true that no region formation algorithm can ever work properly without semantic feedback. We may have to let the low-level systems take guesses at parameter settings and tune in one direction or another as a result of difficulties in interpretation in distinct subareas.

I.2. A Cognitive Experiment

As a simple demonstration of our use of internal models and semantics, one need only look at his own surroundings through a tube of rolled up paper and examine what he "sees" as opposed to what he "knows" is there. The junction of low-level processes and model building can be examined in a human perception experiment of constrained vision. Suppose we supply a person with a large photograph 1 foot square, and constrain his view of it through a movable cardboard sheet with a 1/2" diameter circular window so that at any single moment he sees a very local region of the photograph. If this individual is given the problem of determining roughly what is in the picture, we feel that he will use very different procedures than those used by the current vision systems. Rather than attempting to construct an outline drawing, he will look for prominent features as clues to what the image represents. He will probably scan quickly in many different directions, but once he finds a prominent feature, his strategy might become highly context sensitive and model-directed. He might form hypotheses which encompass large numbers of assumptions and use these to direct the processing until they are verified or disproven.

Informal experiments of these conjectures were conducted using projections of 35 mm slides. A half dozen individuals in group discussion carried out an analysis of the images. We were able to observe the entire image as well as the position of the moving window. The window was small enough so that texture was discerned somewhat, but difficult to use. The result of the analysis was very similar to the feature search and model building process described above. The final models of the natural scenes were usually correct although quite rough. An interesting note is that 70-90% of the information

in the final model was obtained during the first 30 seconds when the students were allowed to view the scene for 5 minutes.

Our interest here is not to assert a model of human problem-solving, but rather to emphasize the need for higher-level approaches to computer vision. The difficulty of human perception in this kind of experiment gives a useful perspective on computer vision and its difficulty. We have a hard time when absorbing information locally and sequentially. The implications are that we use very different kinds of processes; otherwise we should lower our expectations of performance of computer vision systems. Before accepting the latter conclusion, it would be wise to explore the former possibility.

I.3. An Overview of the System

As we see it, one of the crucial problems is the interface between low level visual information associated with local features and high level conceptual knowledge. We propose that this can be done by quickly filtering upward processed information from local features, finding prominent features and possible objects in the scene, and then invoking world knowledge to direct further processing. Thus, the processing initially will be bottom-up until hypotheses are formed and then will switch to top-down. The design that we outline is an ambitious project. Consequently, we will discuss some of the distinct subproblems whose solution will contribute to an effective vision processor, but which can be independently investigated.

Figure 1 is a sketch of the major subsystems and information flow in the vision processor, called VISIONS (for Visual Integration by Semantic Interpretation of Natural Scenes). Although there have been refinements, this block diagram will serve as a reasonable overview. The goal of the system

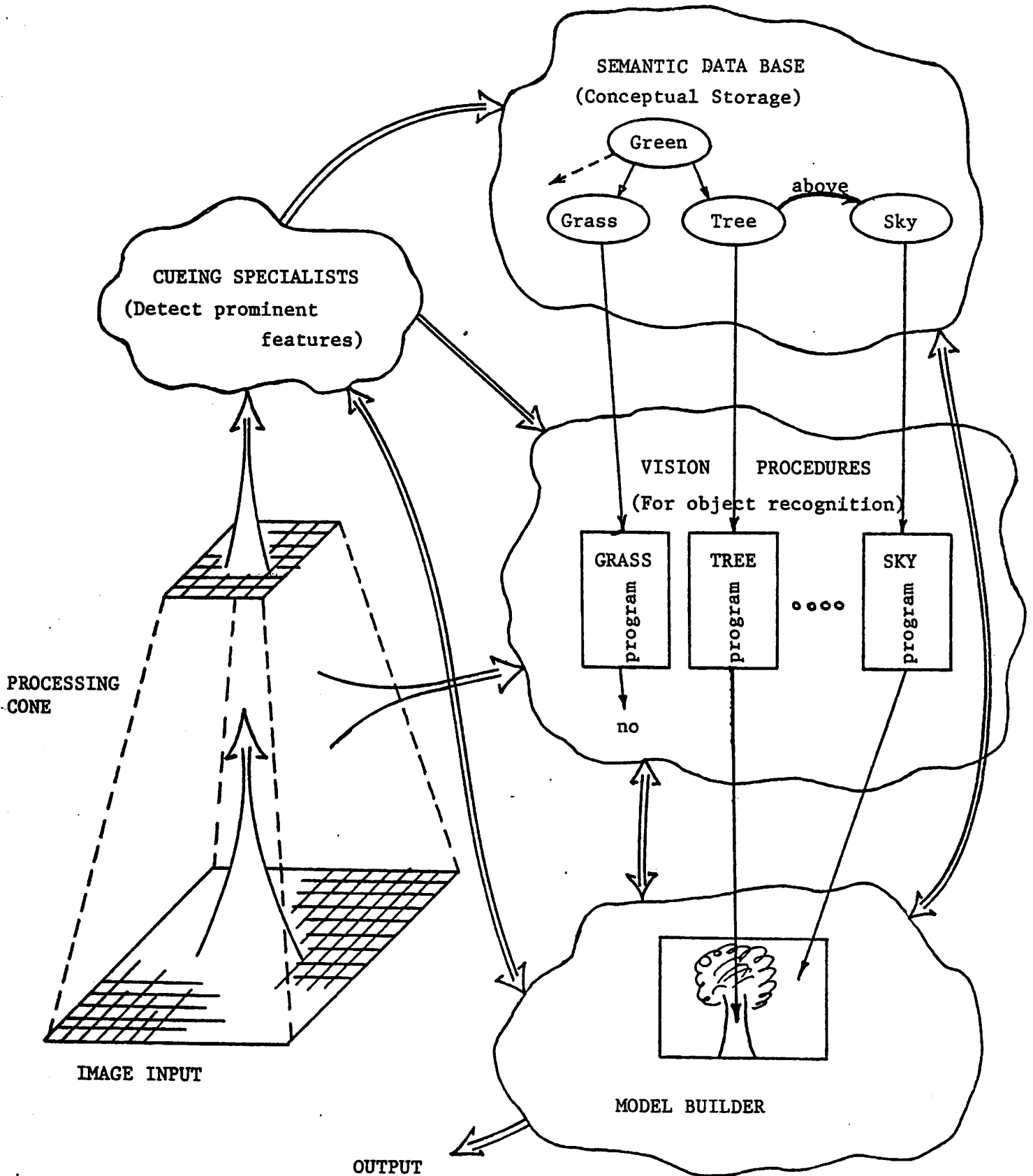


Fig. 1. The Vision Processing System

is to develop a conceptual three-dimensional (3D) model of the important information perceived in a two-dimensional (2D) image. This 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 structure whose nodes and arcs show the relevant attribute-value pairs associated with each object and the relationships (spatial, functional, 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., generate navigation plans) using information from the scene. In a system expanded past the point discussed here, the "plans" generated by an organism might be used to modify the visual processing itself, thereby facilitating the execution of those plans.

The system functions by applying parallel local operators designed to reduce the considerable amount of information present in the scene. Parallel line finders, region growers, texture analyses, color mappings, etc., will operate on a 256^2 grid of the original image and reduce it layer by layer to an image on a 16^2 grid or even less; in fact the cone reduces to the 1×1 level so that parameters such as global average gray level and variance can be extracted.

The vision routines examine fairly completely the upper layers of the cone, and much more selectively, the large arrays at the lower levels. They return a value denoting the confidence that a particular object is present. Thus, the bulk of the volume of data can be examined under the direction of the model builder which uses information in the upper layers as well as semantic knowledge, a partially completed model, and expectations concerning the general setting of the image.

Construction of a plausible model will involve retrieval of conceptual information from a base of world knowledge. There is a wide degree of latitude in the form of the semantic base; it could range from a semantic

net to an extended theorem prover or deductive system. 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), procedures which deal heuristically with perspective, occlusion, horizons, the context of the scene (e.g., afternoon, summer, wooded area far from ocean), etc. Some of the latter information could be embodied in frame-like submodels [27], called context frames, so that expected or typical subscenes and contexts could be dealt with. This allows the horizon dividing the ground plane to be an expected scenario; or a road scene with a road, cars, trees, and telephone poles to be treated as a unit.

Minsky [27] has argued that at some high level of cognitive processing, sequential analysis becomes necessary--that parallel processing mechanisms which have been proposed fail to deal with the figure-ground problem, perspective, occlusion, and the need to deal with complex symbolic structures as units. We fully agree that for many of these problems, the more intuitive sequential processes may be appropriate. However, we have been motivated by the idea that the sheer masses of visual data that are input to the human brain cannot be sequentially processed. Sequential analysis of two adjacent textured regions might be a computational quagmire; however, once the homogeneity of texture and therefore the distinctness of each region is established, then very possibly sequential processes might be appropriate. Now, this in no way implies the sufficiency of parallel processes for perception, but we believe many of the early stages of machine processing (defining major regions, boundaries, textures, colors, etc.) are more appropriate for parallel processes of the type we will describe. In fact, the interface between these parallel processes and the higher level sequential processes

occurs quite naturally at the top of the cone in the system we propose.

I.4. Review of Previous Research

We review briefly those areas of research which are, we believe, in keeping with this philosophy and which are most similar to our approach. We also note in passing that most of the more successful recent projects in AI share the general philosophy of utilization of semantic information coupled with model-directed processing [e.g., 11,25,28,29].

Tenenbaum [9] has described the preliminary structure of a knowledge-based perceptual system. Though this system operates in a constrained world of walls, doors, desks, chairs and telephones, it begins to utilize higher level semantic information. Tenenbaum's approach is to define a two-stage procedure for distinguishing the object sought from other objects appearing in a scene. The system relies heavily on such data as color and range (relative size) to quickly eliminate most objects from the set of possible objects. After this reduced set is found, features which pairwise disambiguate the objects are employed. Contextual information is utilized to form strategies directing search for a particular object or hypothesizing objects which might be nearby any objects found. Currently, the SRI work is organized to interactively develop both the features sufficient to distinguish individual objects and the strategies that can efficiently use them. A general model of each object in the scene is stored internally; when an object is found, its size and orientation is correlated with the internal model which is then displayed via a graphics display.

Possibly the most successful effort to date in the analysis of outdoor scenes is that of Yakimovsky and Feldman [11]. They utilized semantic information in a decision-theoretic approach to the analysis of several road scenes. The information includes properties of the boundaries between re-

gions (e.g., how likely is the adjacency of two regions) and properties of the regions themselves (color, shape, etc.). After initial clustering of picture points to form regions, a decision-tree analysis is used to further join and then identify regions according to a maximum likelihood analysis based on these properties. For more complex environments, we feel that the a-priori conditional probability of a feature given a region cannot be reliably estimated (usually the number of samples is very small) and changes drastically with respect to a different context and over time. Thus, it is becoming apparent that the inclusion of more complex semantic information is necessary; furthermore, the nature of this information must be such that it can be utilized in a highly flexible manner.

Our layered parallel processing structure (described in Section II) is very similar to the "recognition cones" of Uhr [30,31] and we have borrowed a portion of that name for our computational structure. Uhr describes a cone that transforms and reduces information layer by layer to a single cell. Although he discusses a number of possible preprocessing techniques for machine vision, real scenes were not examined; most of his effort has been directed towards classifying, describing, and in general applying parallel conceptual processes to simplified symbolic problem domains. Our focus is the development of techniques that work on real visual data from complex scenes.

In the conical structure, we seek the computational reductions that seem intuitively possible. In the spirit of Kelly [22], we wish to use small amounts of processed data to direct the examination of the vast amount of raw data. Kelly examined the problem of face recognition by averaging 8 x 8 arrays of raw data to get an averaged reduced picture. Lines found by sequential techniques in the reduced image are used as plans to sequentially

find lines in the 64-fold larger image. Kelly reports a 40 to 1 reduction in computation using only this one stage of data reduction. He has effectively demonstrated the power of this hierarchical direction. Our work that is in progress makes available a whole range of parallel operations, generalizes the degree and type of transformation, and allows parallel projection back into the expanded image.

Bajcsy has been systematically investigating texture measures [12], based on Fourier techniques applied to both aerial photographs and outdoor scenes, as features for higher level analysis. Coupled with this analysis has been a constrained semantic network characterizing features of the scenes and relations among regions and objects with good results reported. Bajcsy and Lieberman [13] have attempted to extend this work to outdoor scenes where the descriptions include color and texture at the micro-level (e.g., color and shape of a grass blade) which are then structurally joined, producing a description of the entire image. This effort is still underway and results are inconclusive but promising.

A group at Carnegie-Mellon [23] is preparing to extrapolate to a vision system the hypothesis and test paradigm successfully used in the HEARSAY System [25]. This paradigm involves hypothesizing a partial model with one type of information, testing its validity with other types, reformulating the model, etc.; its application to vision is still in an early stage of development and looks like an interesting effort.

Preparata and Ray [15] have described an approach to the recognition of events in two dimensional natural outdoor scenes. While their approach has some interesting formalizations at the conceptual model building level, it has some basic weaknesses in terms of practical applications. They util-

ize an input which is essentially the scene segmented into regions according to color and avoid dealing with the full complexity of visual data. A graph structure is created based on length of common boundaries and the vertical relationships between all regions in the image, along with the size of each region. The goal is to match this graph with a roughly similar subgraph extracted from a semantic network. However, problems of perspective, occlusion, and variability of spatial location of three dimensional objects in a two-dimensional image make this measure quite weak. We feel that it would be exceedingly difficult to extend this approach to handle the full range of complexity inherent in most outdoor scenes.

Bullock [17] has an excellent review of edge detection operators applied to real world scenes. The problems of finding boundaries of texture elements (micro-structure) and boundaries of objects with surface texture (macro-structure) are fully discussed. Extensive experimental results compare the performance of several algorithms. The problems of edge-detection on the two levels bear resemblance to some of our current work on region growing across both micro- and macro-texture [20] by extracting properties of local regions and structurally relating them. Bullock's work is also leading towards the inclusion of semantic information for higher level scene analysis [16].

Color is an area that is not very well understood. Many of the groups that are developing complex systems for scene analysis are employing color attributes of some form. However, the difficulties of mapping visual tristimulus color data into symbolic descriptors are formidable. A straightforward but effective use of color and color differences in region growing was reported by Yachida and Tsuji [24]. Again, the application was con-

strained to homogeneous solid colors of known type; this approach provides only limited insight to the problems we face.

We should also mention that a number of researchers are currently examining the effectiveness and economy of multi-sensory data from touch-sensors, mechanical position sensors on wheels or camera, radar, etc. [9,32,33]. In the system to be described below, only visual data is considered since we are trying to develop techniques to deal with this rich source of information. However, we are in no way precluding the use of additional data; in fact, it can be incorporated in a natural way into the system we envision.

We make no claims regarding the completeness of this overview. There are important areas ignored such as parallel machines for image processing [34-41], mathematical formalisms for systems [42-43], biologically oriented systems [44-46], and various sub-areas including a multitude of methods for dealing with texture, color, shape, structural description, segmentation, etc., which should be included in a comprehensive review. It does, however, provide a flavor for some of the recent research in this area and provides a context for a discussion of the system proposed here.

II. LOCAL FEATURES AND PARALLEL PROCESSING

II.1. Hints from Pattern Recognition

Research in pattern recognition has made it unmistakably clear that identification of the category of a pattern of information cannot be separated from either selection of the features to be employed or the pragmatic consideration of the dimensionality of this data. Suppose we utilize an array of input points that ensures fairly good resolution, say 256 x 256. Then each snapshot of a scene is comprised of 64K points; each point consists of 6 bits (64 distinct levels) of intensity for each of 3 colors. In terms of the classical approaches to pattern recognition, this is a staggering computational overload. From this point of view, an immediate necessity is the reduction of this data to a manageable level while retaining a subset of the most relevant data or by carrying out a transformation which emphasizes that data useful for classification. However, the problem we have here is far more complex than the typical pattern recognition problem, irrespective of the dimensionality of the patterns. Nevertheless, we ask the same questions: How can the data be usefully reduced in size and which features should be employed?

II.2. Data Reduction

Many of the approaches in scene analysis to date have operated on the raw digitized data exclusively. Visual systems in the animal world, however, carry out extensive parallel preprocessing. We think this is highly desirable in our problem domain. By automatically reducing the data and preserving coarse, though significant, information, detailed analysis of critical

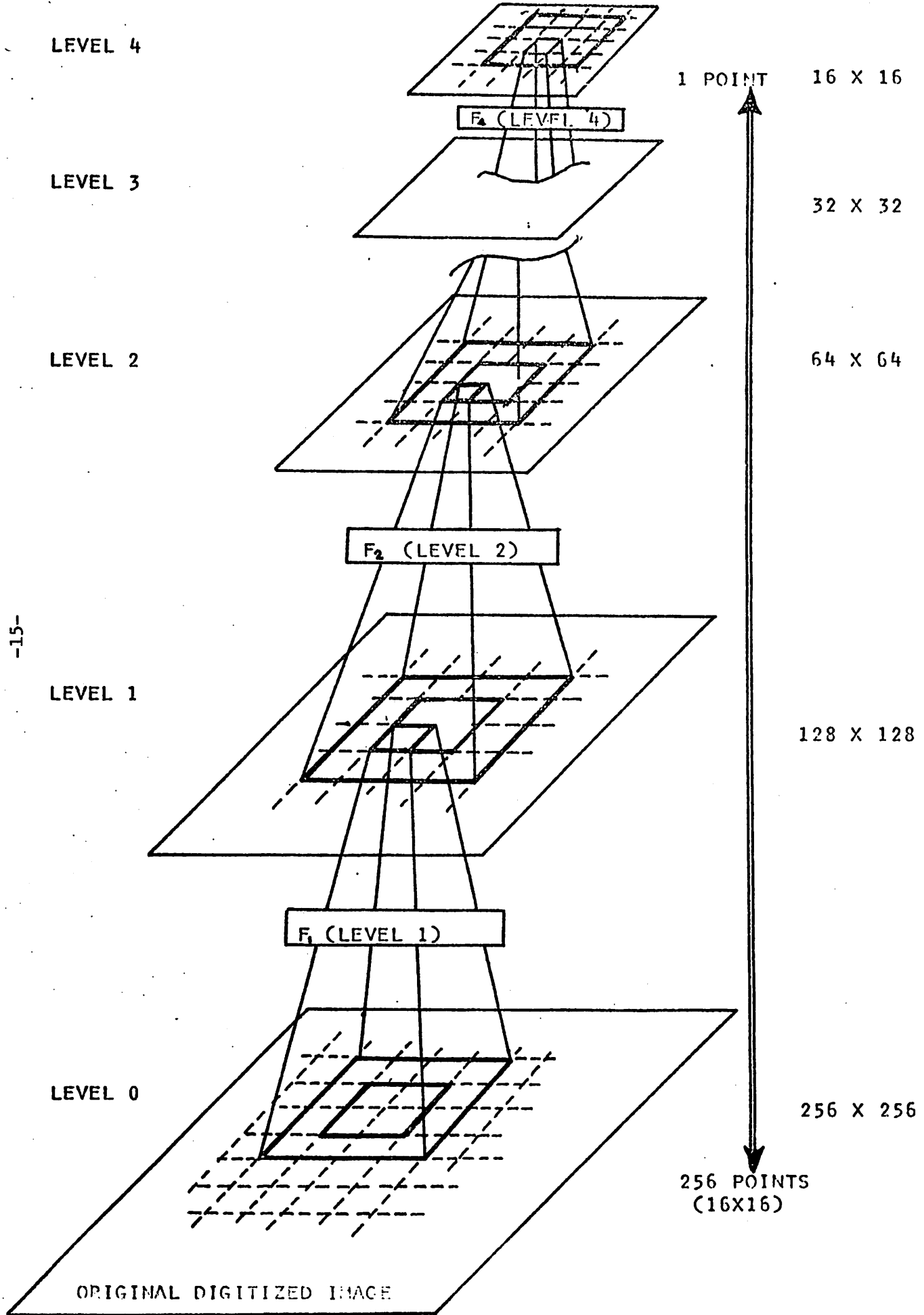
regions of the image can be selectively carried out where it would not be practical across the whole scene. Our goal here is to specify the type of information that should be retained and a simple structure for this data reduction.

Figure 2 is a sketch of a computational "processing cone" [19]. It represents a parallel structure which transforms and reduces large amounts of visual data in a layered fashion. The raw data is depicted at level 0 on a 256^2 grid. The remaining layers are square arrays of size: 128^2 , 64^2 , . . . , 2^2 , 1 (although the top of the cone is not shown in Figure 1), with each point at a particular level storing the information extracted from a subcone below it. In this way, spatial relationships of extracted features are preserved. This system generalizes the computational advantages described by Kelly in his face recognition system [22].

The goal of the structure is to bring in uniform, automatic, processing functions which reduce the dimensionality* of huge arrays of data in such a way as to enhance various features useful to perception of 2D images. Obviously, this requires exploration of the utility of the various characteristics (boundaries, regions, texture, color, shape, etc.) in the image, the techniques for quantifying each, and finally their mapping into symbolic information at the upper levels of the cone (say the 16^2 or 8^2 levels).

Information flows up, down, and laterally within the cone via a sequence of local parallel functions. Each function will be a procedure specified by a computer program (subroutine) operating on its window of input values. Thus, we simulate an array of general purpose micro-computers. These three major types of transformations have been labelled reduction, pro-

* Note: By dimensionality, we refer to the number of points in an array, not the two dimensions of the planar array.



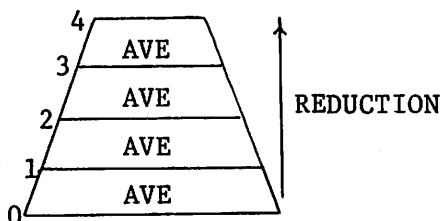
-15-

Figure 2. Processing Cone

jection, and iteration, respectively. At any moment in time, a particular array is operated upon by a set of identical local functions that is uniformly duplicated across the entire array, each with its own window. The windows are currently of size 4 x 4 or 5 x 5, and are somewhat overlapping to avoid problems such as line boundaries falling on window borders. Let us describe each of the forms of computation within the cone.

Reduction (Upward Processing)

As we have already described, a function maps a 4 x 4 array (which is associated with its 2 x 2 center) of values on level n into a single value on level $n + 1$.^{*} Adjacent windows shown in Figure 3 are placed on non-overlapping 2 x 2 centers of 4 x 4 neighborhoods. As an example application, a local function which outputs the average of the 2 x 2 center, if repeated between all levels, will produce an "average" picture at the 16 x 16 level. Each point on level 4 will be the average of the 16 x 16 subcone below it at level 0.



One should note that 4 levels of reduction have been performed but that only a single function definition is required.

* For simplicity we refer here to a single value associated with each cell at each level. Actually, each cell has a k -tuple of values for some constant k . This allows the system to function with a memory or carry out multiple parallel processes.

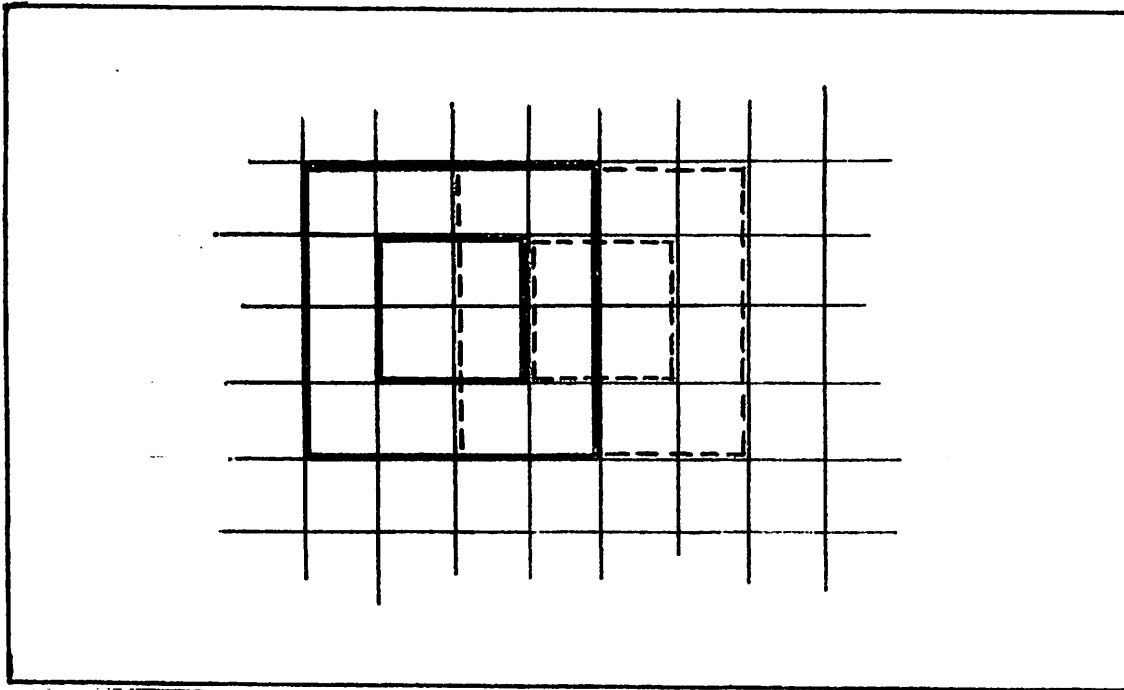


Figure 3. Non-Overlapping 2 x 2 Centers of 4 x 4 Neighborhoods

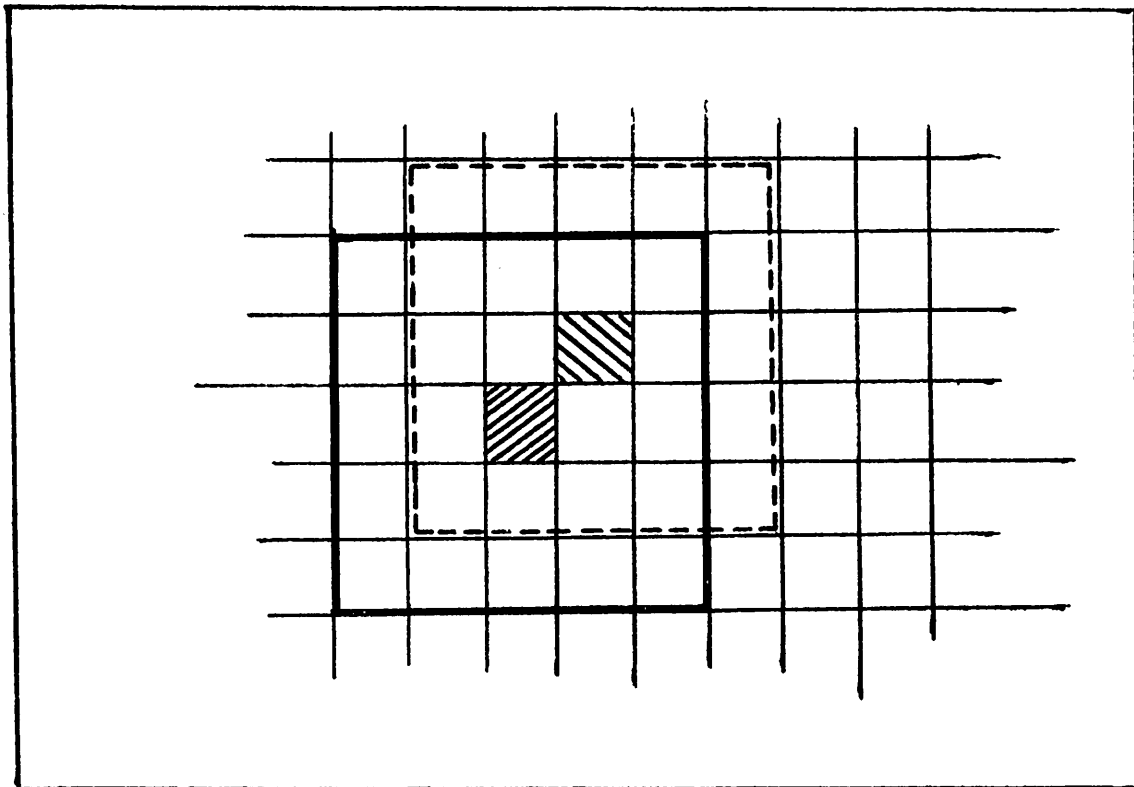


Figure 4. Iterative Neighborhoods on a Given Level.

Iteration (Fixed Level or Lateral Processing)

Here a local function is defined to operate repeatedly on a 5 x 5 neighborhood at a given level and map the resultant value into the central cell of that neighborhood. The size of this neighborhood has been chosen for symmetry about the central cell and so that each window covers a reasonable area, but of course this could be varied. Every cell in the array looks at a neighborhood so that all possible 5 x 5 windows are used in the computation (with suitable conventions for edge effects); two of these neighborhoods are depicted in Figure 4.

The iterative mapping is not significantly different from a number of parallel processing machines. However, we do not limit ourselves to logical functions of the points in the window. We see uses such as parallel region-growing, low-pass filtering or blurring, color normalization of raw three-primary color data, local examination for certain conditions or flags from above or below, etc.

Projection (Downward Processing)

Each point, considered a member of a 2 x 2 center of a neighborhood, has a parent cell which this neighborhood would map into by reduction. Successive reductions denote a set of ancestral cells, exactly one cell on each level above the given cell. If a cone has already been computed, then information can be passed down in parallel by making all the ancestral information of a given cell available during projection.

An example of projection is the downward mapping of a mask which has been set at the 16^2 level. This mask could be set in many ways. It could be the result of growing a region about, let us say, a blue point in the

upper portion of the image. Consider the following sequence in a simple, well-constrained domain: reduction by averaging, region growing by iteration at the top level, and then projection downward; this could proceed in a fraction of the time required for region growing directly at the 256^2 level.* Of course, this description ignores the practical difficulties of automatically setting the proper thresholds in various areas of the image.

Our particular implementation of the projection process allows a mixture of iteration and projection. In addition to the 5 x 5 neighborhood during the iteration process, the ancestors of the center cell are also provided. Thus, a local iteration function operating on level 1 will have available simultaneously all the storage elements of the 25 cells of level 1 and the single ancestor cell from each of levels 2,3,4,...,8 (level 8 is (1 x 1)). Pure projection takes place by operating only upon the ancestral information while combinations of iteration and projection are also possible.**

II.3. Other Local Functions

Parallel preprocessing techniques for extracting features are currently under investigation [19,20]. These include procedures within the cone to detect lines and boundaries, to quantify texture, coarseness and orientation, grow regions, name colors, characterize the straightness or irregularities

* Region growing in the parallel structure proceeds by expanding the boundary of the partially grown region. The rate at which the region is grown depends upon the location of the "seed" point in the region. Region growing in a single direction proceeds linearly and therefore the parallel process of iteration on level 0 could take as many as 255 iterations before termination.

** If combinations of reduction with iteration and projection prove valuable, this flexibility can be provided. In the current implementation, reduction is a distinct process. However, sequences of reduction and iteration projection can achieve many of these algorithms without any changes.

of boundaries, extract shape, features, etc. Each of these sub-problems is quite complex and requires individual investigation.

There are many interesting ways to allow these features to interact and produce results more useful than could be extracted individually. Two investigations of cooperating processes are currently underway. A conservative line-finder which returns prominent line segments can be interfaced to a conservative region grower which returns the centers and rough size of prominent regions. The two cooperating processes may be less sensitive to noise, texture, and the general clutter often found in unstructured scenes.

A particularly interesting investigation involves a region grower that is guided by both average color and texture concurrently. This takes place near the top of the cone and may provide a flexible means of dealing with both micro-texture and macro-texture simultaneously [20].

II.4. Focus of Attention

Now we will describe a modification in the layered data structure that was considered. An interesting biological mechanism suggests itself as a way to reduce the data to be processed the foveal view of the human visual system. Only the central field of vision (several degrees) is in focus and carries detailed information of the scene; the remaining field of view seems to transmit information on a relatively gross level. Motion can be detected on the extreme periphery, but not detailed color, form, etc. As a simple experiment, one need only look at a particular object and examine what he "sees" on the periphery. However, it seems we "know" most of what is in a particular scene by using internal models to add a large amount of informa-

tion to the gross view.*

The suggestion for our model of a mechanical visual system is not to process all areas of the scene with equal effort. Presumably, if the region of interest were known, the major portion of computation could be carried out on the points comprising this region and points falling outside this region would receive only crude processing. One way in which this type of "selective focus" may be approximated in the proposed system is by constructing the first layer in the following manner. The central field samples dense information and areas progressively further from the center sample information with an increasingly coarser grid. The region of "focus" or central interest is resolved at the finest level and surrounding regions become less and less detailed, as shown in Figure 5.

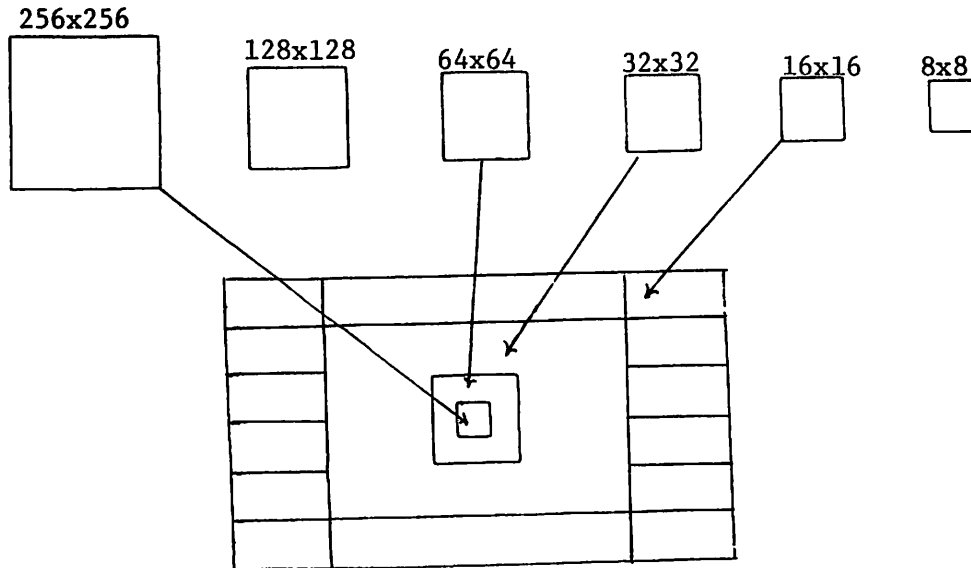


Figure 5. First layer of a system employing a "foveal view"

* For those interested in more detailed and biologically motivated models of vision processing that bear resemblance to the ideas in this paper, we refer you to [44-46].

If necessary, a layered processing system similar to the original layered system can now be constructed by mapping central information to the level of coarser surrounding bands and repeating this process with the new larger central area. Of course, the region of central interest may be shifted across the entire scene by altering the camera position. Although this is an interesting structure and might be useful, it will not directly affect the success or failure of the visual perception system proposed here. Therefore, we will not employ this mechanism in our initial research.

II.5. Comparison with Past Parallel Processing Machines

The various layers of processed information for each of the functions employed will be assumed to be simultaneously available to the vision routines and model builder. One could think of computational modules at each cell in each layer that can compute any of the necessary functions desired. Then we would allow the higher level processes (Sections I-V) to control the switching of the functions on all cells. In the limit one can imagine a very simple general purpose micro-computer at each point in the array at each layer. We feel that this can be intelligently discussed only when we understand the amount, the complexity, and the manner of use of the processed data.

There has been extensive research on the design of general image processing machines and parallel transformations from the late 1950's to the present [34-40]. Most of that effort was not concerned with any particular application. The design that is outlined here should not be hastily compared with such general purpose systems. Any of these structures that can effectively perform our computations could be utilized in our scene analysis. For the time being, however, we refer the interested reader to recent papers

which will serve as an effective introduction to this area. Additionally, it should be pointed out that the limitations associated with Perceptron-like parallel processing [41] are not a factor here because of the fundamentally different structure and the nature of the processing involved (sequential and multi-layered parallel with feedback).

III. THE VISION PROCEDURES

Once a sufficient set of features have been extracted at the top of the cone, the vision procedures provide one form of interface between the processed data and the model builder. The purpose of these procedures is to determine roughly the likelihood of the presence of various objects in the different regions whose characteristics have been tentatively identified. One vision routine will be developed for each major object and/or region which may appear in the scenes being analyzed. Thus, the vision routines may be considered to be procedural representations of objects as embodied in a semantic net or axiomatized in a formal deductive system.

The purpose of these vision-routines is to examine the visual information resident in the cones and, by using advanced pattern recognition techniques (both heuristic and mathematical), to provide a likelihood that the object which the routine is designed to detect is actually present. Each routine contains explicit information on how to utilize the features necessary for the recognition of a particular object and their importance.

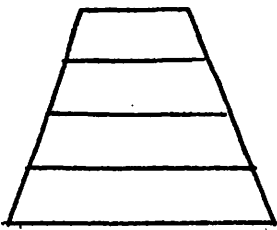
Pattern classification is generally viewed as at least a two-part procedure: feature extraction and classification. The selection of a set of features upon which the decision will be based is most crucial. Reliability of decisions is directly limited by the quality of information in the feature measurements. We have already discussed the extraction of some features through extensive parallel preprocessing. Now the task is to determine the subset of information from any of the layers that signal the presence of, say, a tree. We must decide which of the concepts associated with an object have measurable and relatively invariant visual components (sometimes dependent upon such things as the season of the year, etc.). Some features

describing a tree are color, shape, size, and texture of the trunk and foliage.

Features from upper levels which are simple functions of the processed data may be quite useful. There is a great deal of flexibility in writing a procedure to examine specific portions of the information. The uppermost layers contain the coarsest information, but they are of low dimensionality so they can be examined quickly in a sequential manner. The rough shape, color, and texture might be sufficient to determine the likely absence or presence of a tree. However, since any of the more detailed layers can be accessed, as we pointed out earlier, specific areas may be selectively examined to provide whatever additional information is required. Features which are effective in separating some pair of categories (objects) may be particularly useful in reducing critical ambiguity [9,47,48].

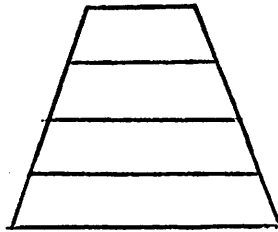
The second phase is the classification process. Here we view this as determining the likelihood of the presence of the goal object given the values of the features measured. The goal here is to get a coarse evaluation such as "improbable," "low," "medium," "high," "almost certain." This crude decision information will be manipulated by the model builder so that the object identification can be integrated into a global model of the scene, with associated global confidence.

Two different routines, say the water routine and the sky routine, looking at the same region of the image, may report the presence of water and sky, respectively, with differing confidences. The features extracted in the cones might be represented as follows:



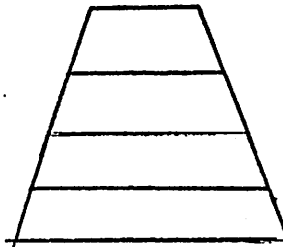
REGIONS

(Location: upper third)



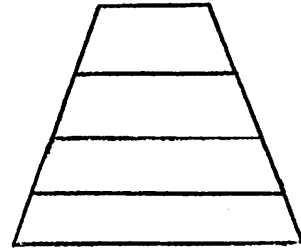
LINES

(Irregular lower boundary for region)



TEXTURE

(Smooth and homogeneous)



COLOR

(Blue and often unsaturated)

The routine associated with sky might hypothesize that the region contains sky with confidence .92, while the water routine may look at this same region and hypothesize that the region is water with confidence .53. The model builder is then responsible for determining which of the two objects is actually present using higher level information and perhaps requesting that additional measurements in the cone be made in the region under question. A more complete discussion of the use of "fuzzy" or "imprecise" information may be found in Section V.

IV. REPRESENTATION OF KNOWLEDGE

One of the prime goals of this research is to determine how to bring knowledge of the world (both general and specific) to bear upon the visual perception of images. The image will be processed for the purposes of identifying all large or "important" objects, and construction of a rough 3-D model of the scene. By "object" we mean each physical region whose boundaries should be identified (such as sky), although they are not literally objects (in the sense of manipulatable objects).

Useful knowledge might be of a general form--that trees are green and basically immobile; that people have two feet, are potentially mobile, and often appear on sidewalks, which are elongated planar objects; and that if the sky appears in an image, it usually appears above all other objects. Thus, we are concerned with spatial, temporal and functional relationships between the visual events of interest as well as 2D and 3D types of information. The extensive base of such general information allows one to view the world in the highly structured way in which it exists. In addition, there might be available specific information about the environment under consideration. This might vary from a list of the objects that are likely to appear in the image to a complete topographical map of all objects in the environment. This information must be organized in a form which allows easy transformation into visual processes; that is, in such a way that it interfaces naturally with the visual analysis to be performed.

IV.1. Deductive Semantic Processes

The semantic information can be embodied in many forms. The specific structure does not seem to be critical at this point of the research, although

two forms of representation are immediately evident, semantic networks, or an axiomatized data base coupled with a deductive system. In an earlier version of the system design, we intended to employ a semantic network. However, since then, we have decided to employ a powerful deductive system that is being developed by D. Fishman [57].

In the first alternative, semantic information can be embedded in a directed graph structure in which the nodes are used to represent conceptual objects or their modifiers while arcs represent the relationships between them. All information which bears directly or indirectly upon the visual image or its processing should be embodied within such a network. Thus, all physical visual attributes of an object will be associated with a node; e.g., a tree has a certain color, shape, texture, and size, and these can be stored as attribute-value pairs associated with the node for tree. In fact, the internal model of "tree" must be rich enough to embody both the general concept of a tree as well as all variations of trees that might appear in the scene being processed.

In addition to physical attributes of objects, spatial, temporal and functional relationships between objects bear useful information. One "knows" that trees are rooted in the ground and often appear beside sidewalks. In addition, sidewalks are used by people to walk upon; therefore, one often will find people with their feet upon them. All of this information can be embedded in the network as a directed arc between the objects labelled with the name of the relationship. Quillian [49] has examined various ways of storing and retrieving information from semantic nets; e.g., the length of the path between related objects may be used as a rough estimate of the "relatedness" between objects. One problem with a semantic net is that addi-

tional processes must be employed to draw deductions not explicitly stored in the net. Thus, a set of deductive rules and the mechanisms to apply them are necessary. As a second alternative, semantic information would be embedded in a modified theorem-proving environment [50-54]. These theorem-provers axiomatize the semantic information at the same time that they utilize heuristic information to efficiently direct large-scale searches.

The system we are intending to employ is under development by D. Fishman [55-59]. As in a straightforward theorem prover, a set of rules (or clauses) is provided describing the permissible deductions, and a set of predicates and functions describe the relationships between objects. Instead of a "blind" theorem proving system which can get lost in huge searches, techniques are being developed to utilize semantic information from the model; clues from prior proof procedures can be blended with heuristic search procedures in order to guide the current proof procedure.

We see a theorem prover under the direction of a model building executive which ships it individual subproblems to solve. The theorem prover can be used as a simple information retrieval system or to solve larger problems such as consistency of a hypothesis with another set of axioms which form a partial model. Thus, there is automatic low-level search under the control of the theorem prover. On the other hand, high level search among alternative models being constructed probably should be somewhat under the control of the model builder (and therefore the programmer). Thus, the structure of this system allows a middle ground between automatic backtracking and CONNIVER, which forces all control on the user. Fishman is proposing

the development of a semantic deductive system of this type embedded in an AI language that allows the saving of multiple contexts.

IV. 2. Application of Semantic Information to Scene Analysis

There has been little attempt in vision research to apply semantic information. One possible reason is that this information is not in a functional form. If the information is stored as symbolic labels (e.g., the actual words associated with the concepts), then it must be translated into a form in which the knowledge may be applied to the image. This is a distinct limitation of typical representations. We may retrieve the information that the "trunks" of trees are generally "vertical." However, in terms of processing the image, this means that the lower portion will have a narrow boundary running up and down. Winograd [28] successfully represented knowledge in a procedural form in his natural language processing system. Procedural representations can utilize declarative symbolic information in a far more flexible manner. For example, a piece of descriptive information about some object can be associated with a subprocedure represented by a set of programming statements. This subprocedure can describe how the image should be analyzed in order to detect that characteristic of the object. This representation of information allows it to be functional, a mechanism by which it can actively operate on the image rather than remain as a passive notation.

Some of the relationships between the real 3D world and the 2D image upon which it is projected are embodied in rules of perspective, occlusion, and shadows. As outlined a little later, this information can be organized in functional modules to be applied to the 2D image. Thus, throughout the system, various forms of information will be employed.

3D and 2D Information

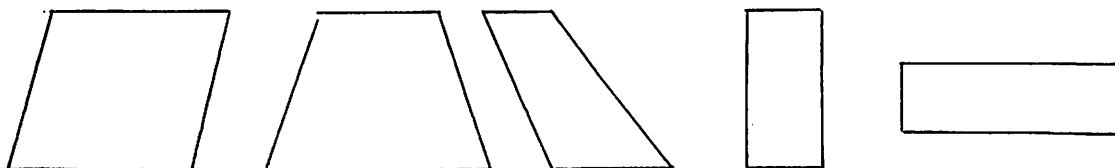
Another distinction in the type of data other than declarative-procedural involves the differences in knowledge in the 2D and 3D domains. Representation of shape, size, and orientation of objects is complicated because three-dimensional (3D) information has a two-dimensional (2D) projection dependent upon the particular perspective. For example, in 3D the sky can be thought of as planar,* above the horizon (or high), and for our purposes, it covers an infinite area. Once we determine that a particular region is possibly sky, then semantic physical information allows one to interpret its place within the model of the world: the sky is planar and above all other objects in the image (as opposed to being a vertical plane behind the most distant visible object on the ground plane). Thus, the conceptual 3D knowledge tells us the sky can't be walked upon or bumped into.

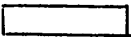
However, the 2D projection (for a horizontal camera) of the sky often results in its appearance at the top of the picture with any shape. Although we usually don't talk about the 2D shape of sky, it is clear that in viewing images we can accept any shape if the rest of the scene makes sense. The particular shape depends upon its occlusion by objects resting on the ground plane. If a nearby object occludes the sky at the top, it is possible for portions to appear lower in the picture. Since trees are often the uppermost objects occluding the sky, the bottom boundary is often irregular.

This incomplete example points out the difference in utilizing 3D and 2D information. Consequently, we intend to store conceptual information from the "physical" 3D world--size, shape, orientation--as well as "picture" in-

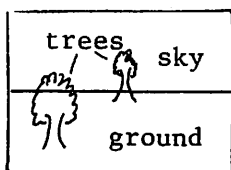
*Note: for simplicity, the sky is being represented as a "ceiling" plane rather than occupying a volume.

formation from expected 2D mappings--size, shape, orientation, and boundary characteristics. This last attribute could be a simple measure of straightness or irregularity of the lines bounding a region. The model builder must use the perspective analyzer to check the consistency of an actual 2D region with the 3D value. Although some 2D shapes will be stored (such as: a tree trunk is composed of roughly parallel vertical lines spaced 3" to 3' wide) they cannot be stored from all possible perspectives. More generally, the perspective module will be able to check the consistency of the 3D value of the shape of an object with the 2D shape of a unknown region. However, one can still make great use of several simple types of shapes that have wide occurrence:



Some of these usually represent objects in the ground plane, while others usually represent objects perpendicular to the ground plane, and some represent both (e.g.,  can be a road running left to right or a fence railing).

There are other simple heuristics that can be embedded in these modules which sometimes give useful hints: e.g., the "picture area" of a region affects its likelihood of being sky. In many pictures, but obviously not all, the sky covers 1/4 to 1/2 of the image when the camera is held horizontal. Also, rough distance estimates between objects may be made using perspective information; consider the following simple scene:



Assuming the ground is flat (perhaps not a viable assumption), one can hypothesize that the smaller tree is farther away. This hypothesis is not on the basis of size (which may be misleading) but by the simple heuristic that the smaller tree is rooted to the ground plane higher in the image.

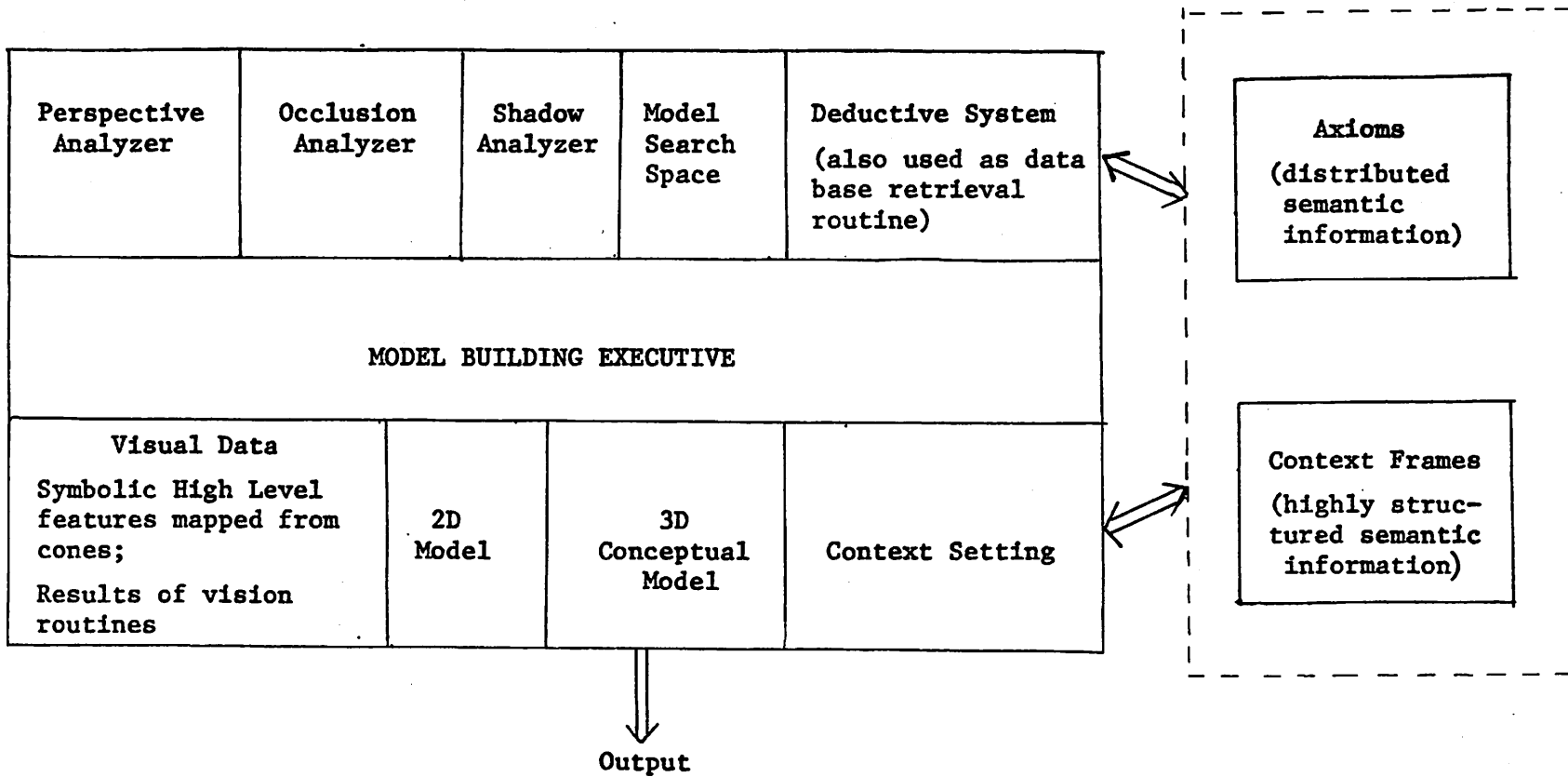


Figure 6 -- The Model Builder

V. MODEL BUILDING AND VISUAL PROCESSING

V.1. Gross Organization of the System

The desired objective of the model builder in this research is the construction of a rough conceptual model of the 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.

The complexity of the model building process becomes apparent if one examines the types of information that must be utilized. Figure 6 is an expanded view of the model builder. It is shown in a modular form because this seems to be the only way to make the system manageable. The 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. Then it is responsible for integrating the responses, examining the implications, resolving conflicts, etc. However, the control structure is not as straightforward as this discussion implies; in many cases a more heterarchical organization is implicit.

Let us first outline the several sources of information which the model builder must utilize 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; a road scene with roads, cars, trees, and telephone poles alongside the road, etc., is a unit of knowledge (some pieces of which may be missing) to be applied as an entire subframe.
- 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 is a semantically guided deductive process 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 is used to efficiently solve distinct subproblems provided by the model builder; it is also used as a straightforward information retrieval process from the semantic data base.
- 8) Perspective Analyzer: this module contains both procedural and declarative information 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 and 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; in 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. The executive in the model builder will facilitate communication between each of the modules defined. In addition, it must contain a strategy for correlating the diverse forms of information. Therefore, the model builder, and in particular the executive, will be the major control center for the system. It must have the ability to explore alternative models, form additional hypotheses, invoke a variety of validation processes, and resolve conflicts. The executive will allow 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. Of course, multiple strategies exist for the construction and verification of models based on the considerations outlined above ; some of these are discussed in [21, 60-62].

V.2. Further Considerations: Fuzzy (or Imprecise) Information in Model Building

A characteristic of our world is that descriptions of it and the 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 [63-65] 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 some of the processes in which it would appear to be highly relevant (perhaps even necessary) and some of the properties it must exhibit. These questions and many of the ideas which follow overlap and interact; individual discussions do not imply independence of the problems.

V.2.1. Quantification of Attributes and Symbolic Representation

In order to construct a model of a scene, one must understand the

world being modelled. However, if one examines human descriptions of the various features in the world, it becomes evident that we use imprecise descriptors. 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. The measurement of the attributes of these primitives and the subsequent assignment of symbolic descriptors remains an important area of research.

Consider the measurement versus description of the attribute/concept "color." The term "red" conjures up a spectrum of colors ranging from pink to deep red and from orange to purplish-red. The measurement of a particular area in a scene, on the other hand, results in a known quantity of the red, green and blue primaries being ascribed to that area (or alternatively, values for hue, saturation, and intensity). The consistent naming of colors, given the measured values, is difficult, yet the interface to the semantic information is through exactly these symbolic descriptors. When we say "A tree is green," we are willing to accept a wide range in the actual measured values of green. Complicating the assignment of consistent symbolic descriptors to color information is the fact that human perception of color is modified by the context in which the color is found (e.g., surrounding colors), lighting, mood of the observer, surface conditions, etc. While color is such an important attribute for recognition and perception, it is not surprising that its use has been limited; color information in digital images has been employed only in reasonably simple ways [10,11,13,18,23,24].

Given that the measurement of the value of an attribute can be fuzzy, we feel that color should be quantified as a range so that subtle shadings

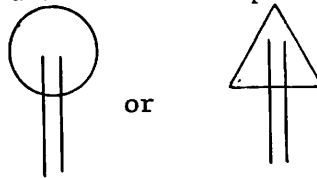
of color are not of critical importance. These ranges can then be aggregated in different ways and associated with symbolic descriptors. We now have the requisite flexibility for specifying the green of grass to range over a large angle of the chromaticity diagram from yellowish-green to pure green, while the green of pine trees can range from pure green to blue-green. Increased flexibility can be gained by allowing modifiers of color (or hue) such as saturation and intensity. One wants to be able to describe a pale (unsaturated) blue of the sky or a dark (low intensity) green for shadows within the foliage of a tree; these descriptions may then be mapped into the "every-day" terms for colors.

The utility of ranges is not confined to color. It allows us to deal with intensity, saturation, texture, and in fact almost all attributes, in a rough way. However, this does not alleviate the problem that these measurements are imprecise, and that any process using the value of an attribute must take that into account.

Consider the problem of quantifying texture; what are the proper symbolic descriptors of texture? It seems clear that humans utilize texture in their visual perception of objects or regions. However, when one attempts to verbally describe texture, a long groping narrative with many modifiers is likely to ensue--unless one uses a term such as "leafy" or "grass-like." This clearly implies that texture descriptors are concepts related to visual imagery and lacking in precise linguistic representations. Thus, a common texture might invoke the concept of objects that exhibit that texture. It is unclear what features of texture are being used in human perception as well as how the measurements take place. Thus, we seek simple features of texture which capture some aspect of the concept. If coarseness, homogeneity,

and directionality of texture are roughly extracted, we might use a confidence or fuzzy set membership to represent the center of a range of values. We might have simple properties such as a linear scale from smooth to rough, as well as homogeneous or non-homogeneous. Thus, the color and texture of a region might be represented as .7 green and .9 rough, recognizing that such a description is a many to one mapping and thus not unique.

There are attributes of objects such as shape which seem to defy simple quantification. The shape of a tree is incredibly complex if it is to be described in any level of detail. However, there are relatively simpler features of shape on a grosser level of detail that must be extracted and that might be described in a hierarchical process of subparts:



It is this latter level of detail that is appropriate for our purposes. We seek to find many simple features which can be used both for discrimination between objects and perception of objects. This is leading us to search for ways to heuristically quantify complex attributes. P^2/A , a measure of compactness [1], is one such simple feature. Boundary characteristics of a region seem to carry much information about the shape of an object; techniques such as chain encoding or finding points of maximum curvature might be used to determine qualities of line boundaries. Orientation and skeletonization are other features of shape that can be examined. We need a set of values for each attribute that can be computed efficiently and map into reasonable symbolic representations.

V.2.2. Expectation and Importance

We have discussed previously the impreciseness in the measurement of

physical values of real-world attributes as well as the mapping into symbolic descriptors. However, the measurement of the color of a region and its mapping into a value of blue, for example, includes neither the expectation that the sky is blue nor the importance of the attribute color in the recognition of a region as sky.

What is the degree of expectation that the sky is blue? The values of some attributes vary or take on a few values: the sky is blue, white, and often blue and white; sometimes it is red and orange. However, these values take on different expectations, often dependent upon the context. If nothing is known about the context, the expectation is high the sky is blue or white, and low that the sky is red. If we know that we are looking west at 7:30 p.m. in the summer, there is a much higher expectation that the sky is red. Ignoring the problem of varying contexts for the moment, the point we wish to make is that it is useful to associate some degree of expectation with an attribute-value pair. In summer, the color of trees in general is green with expectation, let us say for purposes of discussion, of .95 and maroon with expectation .1.* This latter expectation increases sharply in the autumn in New England. We expect grass to be green but can deal with it being brown or yellow. We expect the sides of roads to be parallel. This type of information clearly affects the confidence of any hypothesized model.

However, the problem we are faced with is somewhat more complex. The strong expectation that an attribute-value is associated with a particular object may be of little aid in recognition processes if many objects have

* Particular trees such as Japanese Maple are a deep red or maroon all during the leafy season and turn crimson in the fall. The relative expectation probably should be .001 or lower.

that attribute value. Importance of a particular attribute-value in the recognition process is related to a feature's ability to discriminate between objects in the environment [9,47]. The question now arises as to how to effectively utilize expectation and importance and where the information is to reside. Furthermore, similar questions appear at various levels in the organization of the entire system.

The importance of an attribute, i.e., the weighting of attributes, is another form of vagueness in the detection of the presence of an object. The location of the sky on the top of the picture might be more important than an irregular boundary caused by trees at the bottom of the sky since the trees could very well be absent or located at a great distance away. The importance of a "feature" is also dependent upon the context. For trees in summer, the green of the crown may be more important than the shape. This information must be incorporated procedurally in the vision routines associated with tree and sky. Is it also necessary to place the importance of attributes in the semantic data base for use by the model builder? The answer is unclear at this time, but development of the model building system should determine the need for this redundancy.

Similar questions can be directed at a still higher level. There is a great deal of ambiguity in determining the relative importance of the parts of an object for purposes of recognition. If a house is composed of various planar faces (roofs, walls, doors, windows, etc.), how much weight should be given each part in deciding a house is present? In summer, the crown of a tree is more useful than the trunk for signalling the presence of a tree because the green foliage often occludes much of the trunk. In winter, the trunk is probably more helpful than the crown of branches because it is much

more invariant than the web of branches emanating in all directions. This weighting information can also be embodied procedurally in the pattern recognition processes of the vision routines.

In many cases, recognition of an object will be based on the recognition of some or most of the constituent parts of the object. In some cases, recognition of any of the parts ("OR") is sufficient to determine the object's presence while in other cases, all of the parts ("AND") are required. This can be generalized to cover all situations in between by using a fuzzy AND/OR which varies from 1("AND") to 0("OR"). The interrelationships are very complex and this approach would require considerable development. Briefly, a value of .9 implies that "most" of the parts must be found; for example, all parts with a medium confidence, or 3 or 4 parts with sufficient confidence, or 2 parts very strongly to balance one being very weak. An AND/OR value of .1 implies a set of conditions slightly stronger than one part of the set of parts. The manipulation of confidences in conjunction with the AND/OR condition yields the overall confidence of the object.

V.3. Confidence in Building Models

We have already discussed the uncertainty or fuzziness of the information that is passed on to the model builder from the vision routines and perhaps the data base of knowledge. The vision routines for various objects pass a confidence on the basis of the relative importance of the attribute-values and parts within the given context. However, the particular partial model into which it is being fit also affects the degree of confidence that the region is that object. This means that the confidence of the identity of a particular region must be adjusted with respect to the confi-

dences and identities of surrounding regions. In some cases, the confidence of an object may be sufficiently high and the possibility of fitting it into a particular model may be sufficiently low that refutation of the model is the only solution.

More generally, the model builder must use both the context of the partial model under construction and the region (object) being added to vary the confidence of the model. How well does the additional hypothesis fit the model? The overall confidence of the model can be greatly increased or decreased. All the relationships between objects in the model, the context, and the object under consideration should have a strong influence on the acceptability of the new object and continued acceptance of the current model. Here is where importance and/or expectation of relationships stored in the semantic base might aid in determining this confidence. Typical or common sets of these relationships might be called into play as contexts or submodels constructed as simple frames.

In summary, the confidence of any given model is based upon:

- a) the importance of individual attributes and parts of an object;
- b) the fuzzy measurement of the value of each relevant attribute;
- c) the output of each vision (object) routine executed;
- d) the context (e.g., season, general locality) in which the model is being built;
- e) the ease with which the parts of the model fit together; and
- f) whether expected submodels are found and how well they fit together.

We have raised many complex issues and only provided partial insight towards their solution. Our understanding of these problems at this point

is incomplete and we have chosen to avoid detailed speculation. For example, in dealing with confidences, simple linear weighting schemes might be useful, but we are cognizant of their limitations in a long history of pattern recognition applications. Thus, we expect the necessity of second-order relationships between some of the interacting attributes/parts/objects in determining various confidences. However, we have pointed out in numerous places the desirability of using more flexible heuristic mechanisms for allowing information to come into play. At this point, we feel it would be naive to attempt explicit descriptions which must evolve carefully and modularly.

Finally, there are a number of topics that have not been discussed in any depth, for example shadows, reflections, and motion (in a dynamic world); these problems might be handled by subsystems similar to perspective and occlusion. Their absence does not imply a lack of importance. We expect to tackle the problems associated with shadows at a relatively early point in the research, although initially the simplest set of heuristics that suffice will be employed. Reflections and motion will be topics of future study.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the continuing support of their colleagues, as manifested in many long and fruitful discussions, particularly Dan Fishman, Michael Arbib, William Kilmer and John Roy. The unceasing effort of many graduate students is also very much appreciated, particularly Tom Williams, Paul Nagin, Ed Fisher, Dan Purjes, Jonathan Post, and Elliot Soloway.

REFERENCES

1. R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, 1973.
2. A. Guzman, "Decomposition of a Visual Scene into Three-Dimensional Bodies," Proceedings of Fall Joint Computer Conference, Vol. 33, 1968, pp. 291-304.
3. C. R. Brice and C. L. Fenema, "Scene Analysis Using Regions," Artificial Intelligence, Vol. 1, Fall 1970, pp. 205-226.
4. M. B. Clowes, "On Seeing Things," Artificial Intelligence, Vol. 2, Spring 1971, pp. 79-116.
5. D. A. Huffman, "Impossible Objects as Nonsense Sentences," Machine Intelligence 6, B. Meltzer and D. Michie (eds.), American Elsevier, 1971, pp. 295-323.
6. P. H. Winston, "The MIT Robot," Machine Intelligence 7, B. Meltzer and D. Michie (eds.), American Elsevier, 1972, pp. 431-463.
7. D. L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," AI TR-271, MIT, Cambridge, Mass., 1972.
8. Y. Shirai, "A Heterarchical Program for Recognition of Polyhedra," Memo No. 263, Artificial Intelligence Laboratory, MIT, Cambridge, Mass., 1972.
9. J. Tenenbaum, "On Locating Objects by their Distinguishing Features in Multisensory Images," SRI Technical Note 84, AI Center, Stanford Research Institute, September 1973.
10. J. M. Tenenbaum, T. D. Garvey, S. Weyl and H. C. Wolf, "An Interactive Facility for Scene Analysis Research," SRI Tech. Note 87, Artificial Intelligence Center, Stanford Research Institute, January 1974.
11. Y. Yakimovsky and J. A. Feldman, "A Semantics-Based Decision Theory Region Analyzer," Proceedings of the Third Joint Conference on Artificial Intelligence, August 1973, pp. 580-588.
12. R. Bajcsy, "Computer Description of Textured Surfaces," Proc. of 34d IJCAI, August 1973, pp. 572-579.
13. R. Bajcsy and L. Lieberman, "Computer Description of Real Outdoor Scenes," Tech Report, Moore School of Electrical Engineering, University of Pennsylvania, 1974.
14. M. L. Baird and M. D. Kelly, "A Paradigm for Semantic Picture Recognition," Pattern Recognition Journal, Vol. 6, June 1974, pp. 61-74.

15. F. P. Preparata and S. R. Ray, "An Approach to Artificial Nonsymbolic Cognition," Information Sciences 4, 1972, pp. 65-86.
16. B. Bullock, "Formation of Scene Analysis Plans Using Pattern Recognition Replacement for Region Structure Analysis," Tech. Report, Hughes Research Laboratories, Malibu, California, May, 1973.
17. B. Bullock, "The Performance of Edge Operators on Images with Texture," Tech. Report, Hughes Research Laboratories, Malibu, California, October, 1974.
18. E. M. Riseman and A. R. Hanson, "Design of a Semantically Directed Vision Processor," Tech. Report 74C-1, Computer and Information Science, University of Massachusetts, January 1974. (Superseded by current report)
19. 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.
20. A. R. Hanson, E. M. Riseman, P. Nagin, "Region Growing Using Color and Texture," in preparation.
21. D. Fishman, A. R. Hanson, and E. M. Riseman, "A Model Building System for Scene Analysis," Tech. Report in preparation.
22. M. D. Kelly, "Edge Detection in Pictures by Computer Using Planning," Machine Intelligence 6, pp. 379-409, 1971.
23. R. B. Ohlander, "Analysis of Natural Scenes: A Model and Methodology for Research," Thesis Proposal, Carnegie-Mellon University, September 2, 1974.
24. M. Yachida and S. Tsuji, "Application of Color Information to Visual Perception," Pattern Recognition, Vol. 3, 1971, pp. 307-323.
25. D. R. Reddy, L. D. Erman, R. D. Fennell, and R. B. Neely, "The Hearsay Speech Understanding System," Proc. of 3rd Joint Conf. on AI, August 1973, pp. 185-193.
26. D. E. Walker, "Speech Understanding Through Syntactic and Semantic Analysis," Proc. of 3rd Joint Conference on AI, August 1973, pp. 208-215.
27. M. Minsky, "A Framework for Representing Knowledge," AI Memo No. 306, Artificial Intelligence Center, M.I.T., June 1974.
28. T. Winograd, Understanding Natural Language, Academic Press, N.Y., 1972.
29. R. Schank, "Identifications of Conceptualizations Underlying Natural Language," Computer Models of Thought and Language, R. C. Schank and K. M. Colby, Eds. W. H. Freeman & Co., 1973, pp. 187-247.

30. L. Uhr, "Layered 'Recognition Cone' Networks that Preprocess, Classify, and Describe," IEEETC, pp. 758-768, July 1972.
31. L. Uhr, "Describing, Using Recognition Cones," Tech. Report #176, Computer Science, Univ. of Wisconsin, February 1973.
32. B. M. Dobrotin and V. D. Scheinman, "Design of a Computer Controlled Manipulator for Robot Research," Proc. of 3rd Joint Conf. on AI, August 1973, pp. 291-297.
33. M. H. Smith and L. S. Coles, "Design of a Low Cost General Purpose Robot," Proc. of 3rd Joint Conf. on AI, August 1973, pp. 324-335.
34. S. H. Unger, "A Computer Oriented Toward Spatial Problems," Proc. IRE, Oct. 1950, p. 1744.
35. B. H. McCormick, "The Illinois Pattern Recognition Computer-ILLIAC III," IEEE Trans. on Electronic Computers, 1963, pp. 791-813.
36. M. J. E. Golay, "Hexagonal Parallel Transformations," IEEETC, Vol. C-18, August 1969, pp. 733-740.
37. E. G. Johnston, "The PAX II Picture Processing System," Picture Processing and Psychopictorics, Eds., B. S. Lipkin and A. Rosenfeld, Academic Press, 1970, pp. 426-512.
38. B. S. Gray, "Local Properties of Binary Images in Two Dimensions," IEEETC, Vol. C-20, May 1971, pp. 5551-5561.
39. M. J. B. Duff, D. M. Watson, R. J. Fountain, and G. K. Shaw, "A Cellular Logic Array for Image Processing," Pattern Recognition, Vol. 5, September 1973, pp. 229-247.
40. B. Kruse, "A Parallel Picture Processing Machine," IEEETC, Vol. c-22, December 1973, pp. 1075-1087.
41. M. Minsky and S. Papert, Perceptrons: An Introduction to Computational Geometry, MIT Press, Cambridge, 1969.
42. D. J. Parker and D. J. H. Moore, "End Points, Complexity and Visual Illusions," IEEE Trans. on Systems, Man, and Cybernetics, SMC-2 July 1972, pp. 421-429.
43. D. J. H. Moore, "A Theory of Form," Int. J. Man-Mach. Stud., Vol. 3, January 1971, pp. 31-59.
44. M. A. Arbib, The Metaphorical Brain, John Wiley and Sons, 1972.
45. R. L. Didday and M. A. Arbib, "Eye Movements and Visual Perception: A 'Two Visual System' Model," Technical Report 73C-9, Computer and Information Science, University of Massachusetts, Amherst, December 1973.

46. P. Dev, "Segmentation Processes in Visual Perception: A Model," Tech. Report, Computer and Information Science, University of Massachusetts, 1973.
47. E. Fisher, A. Hanson, and E. M. Riseman, "Feature Selection Using Thresholded Measures," COINS Tech. Report 74C-9, December 1974--an earlier version of this appeared in Proceedings of 1973 International Conference on Cybernetics and Society, November 1973, pp. 94-95.
48. A. Hanson and E. Riseman, "Context in Word Recognition," COINS Tech. Report 74C- , University of Massachusetts, To appear in Pattern Recognition.
49. M. R. Quillian, "The Teachable Language Comprehender: A Simulation Program and Theory of Language," Comm. of ACM, vol. 12, August 1969, pp. 459-476.
50. 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.
51. D. Hewitt, "PLANNER: A Language for Manipulating Models and Proving Theorems in a Robot," AI Memo 168, AI Project MAC, Cambridge, Mass: MIT, 1968.
52. G. J. Sussman and D. V. McDermott, "From PLANNER to CONNIVER - A Genetic Approach," Proc. of 1972 FJCC, pp. 1171-1179, 1972.
53. D. Bobrow and B. Raphael, "New Programming Languages for AI Research," Tutorial presented at 3rd Joint Conf. on AI, August 1973.
54. B. Meltzer, "A New Look at Mathematics and its Mechanization," Machine Intelligence 6, (B. Meltzer and D. Michie, eds.), Edinburgh: Edinburgh University Press, pp. 63-70, 1968.
55. 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.
56. 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).
57. D. H. Fishman and J. Minker, "[I]-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.)

58. 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.
59. 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.)
60. M. A. Fischler, "Machine Perception and Description of Pictorial Data," Proceedings IJCAI, Washington, D.C., 1971.
61. O. Firschein and M. A. Fischler, "Describing and Abstracting Pictorial Structures," Pattern Recognition Journal, Vol. 3, November 1971, pp. 421-443.
62. O. Firschein and M. A. Fischler, "A Study in Descriptive Representation of Pictorial Data," Pattern Recognition Journal, Vol. 4, December 1972, pp. 361-377.
63. L. A. Zadeh, "Fuzzy Sets," Information and Control, vol. 8, pp. 338-353, 1965.
64. R. Kling, "Fuzzy Planner," University of Wisconsin, Technical Report.
65. R. C. T. Lee, "Fuzzy Logic and the Resolution Principle," 2IJCAI, Advance Papers of the Conference, The British Computer Society, September 1971, 560-567.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

The computational structure for rapidly extracting visual features is called a "processing cone." The cone consists of parallel spatial arrays of micro-computing elements, each of which operate on a local window to reduce the data layer by layer. Information flows up, down, and laterally in the cone via a sequence of local parallel operations. Routines for detecting objects will examine the data at the top of the cone and will selectively analyze the lower level mass of data. Rough confidences of the presence of objects in various regions will be passed to the model builder.

Model construction will employ many types of information in modular subsystems. Perspective and occlusion routines which utilize heuristic and mathematical analyses can be applied in both procedural and declarative form. The presence of partial models can be used to direct the system through a search space of possible models. Semantic information structured as submodels will be used wherever possible to direct this complex process. A deductive system will be employed to check for model consistency at each stage.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)